Sequential Decision Making under Structured Partial Observability

by

Chinmaya Kausik

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mathematics)
in the University of Michigan
2026

Doctoral Committee:

Professor Ambuj Tewari, Co-Chair
Professor Martin Strauss, Co-Chair
Professor Victoria Booth
Dr. Adith Swaminathan, Netflix

Chinmaya Kausik
ckausik@umich.edu
ORCID iD: 0009-0009-7694-2153

DEDICATION

Dedicated first and foremost to my parents, who have loved me through all the stages of raising me. I am so grateful to them for teaching me to approach the world with optimism, curiosity, ambition and a healthy dose of skepticism. Second to my sisters, whose presence in my life has been so important in who I am today. Finally, also to every queer scientist, mathematician, engineer, lawyer, doctor and creative who came before me and all those who will continue to appear, showing queer students out there that queer people can accomplish great things.

ACKNOWLEDGMENTS

I am first and foremost grateful to my advisor Ambuj Tewari, who was so kind to take me in as his student when I was switching out of geometry and topology in the first year of grad school, and has so profoundly shaped how I think about research. His boundless enthusiasm for research and his infectious optimism are well-known among his students, and were a big part of the support system that kept me going during my PhD. He has taught me to carry a child-like excitement for my research while asking the skeptical questions that a mature researcher would at every step. I am also grateful to my co-advisor Martin Strauss, whose generous support as I navigated my PhD has made the experience so smooth and effortless.

I am also thankful for all my other collaborators and mentors during my PhD – Kevin Tan, Yangyi Lu, Maggie Makar, Yixin Wang, Mirco Mutti, Aldo Pacchiano, Marc Brooks, Adith Swaminathan, Harald Steck, Nathan Kallus, Yonathan Efroni, Nadav Merlis, Aadirupa Saha, Kashvi Srivastava, Rishi Sonthalia. In particular, I would like to thank my primary collaborator Kevin, whom I really appreciate having had around to bootstrap my confidence and knowledge with during the early years of my PhD. I still fondly remember our all-nighters rushing to meet conference deadlines in the math department's atrium. Later in my PhD, I was fortunate to be mentored by Adith, whose deep insights about the research process itself and ability to make wild ideas seem doable have helped refine my research abilities in an entirely new way. Finally, I am grateful to my committee member Prof. Booth, who has been very supportive as I wrapped up my PhD.

I cherish all the friends I made during my time at UMich. I have such cozy associations with the stats office and all the banter sitting near Sahana, Jaylin, Gabe, Yash, Jake, Marc, Paolo, Abhiti, Josh, Vinod, Unique. I have the fondest memories of our board game nights and trips with Sahana, Yash, Jake, and Saptarshi, from Austin to Spain. The PhD would have felt so different without my life outside of the office and the lab – gym with Yash, Jake and others, lunches with Joseph, driving around with Cole, coffees with Mia and Sahana, volleyball with the Volleybros, salsa and bachata in the city. I am very grateful to have met Tuhin and Aishani in the middle of my PhD – they have grown to be my closest friends in Ann Arbor. Hosting the SPAM+ mixers every month has become one of the highlights of my social life as a PhD student, and I am grateful to all the

guests and friends who made the parties such a success.

I don't know how I would have survived the PhD without my many, many housemates at the Ella Baker Graduate Cooperative House. Lindsey's one hug a day, tea sessions with Emilia, long kitchen conversations at night with Vicki, Jared, Max, Collin, Bailee, Ezekiel's hilarious takes on everything, dancing with Bailee, our weekly music jams, all the delicious dinners I got to eat and make at Baker, our Bollywood nights thanks to Gauri, Kashmiri food from Sidra, seeing Sami and Rose in the joint (our common space) every evening, TV shows in the pit, Gio, Stephanie, Alhan, Cat, Elis, and many others. Max and Jared in particular have watched me grow and change in my five years at Baker, and have been there for me through it all. I feel so warm remembering all the times I would enter the kitchen at night to see Max making his tofu salad and then we would talk about math and life.

I am also grateful to my therapist, who has helped me through so many stages of my PhD and through so much personal growth.

Finally, I am very grateful to the family and friends who have helped me get through my PhD from outside of Ann Arbor. I am grateful for the friends I made during internships (shoutout Srikar, Vibha and the Jane Street gang), for getting to share stories with my sister Yaashia who started her PhD the same year as me, and all the support from my other sisters Divyangna and Malvika. I am grateful for my friend Abishek from college, who has always been available on text, heard me out non-judgmentally so often and been such a pillar of support through my PhD. I am also grateful for Rimika, Vidhi and Vrunda and our funny little group chat. Finally, I am deeply grateful to my parents for all the affection they showered me with when I needed support, for cheering me on through every milestone, for celebrating all my successes with so much enthusiasm. Thank you for being there for me.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGMENTS iii

LIST OF FIGURES viii

LIST OF APPENDICES xiii

ABSTRACT xiv

CHAPTER

1 Introduction 1

1.1 Sequential Decision Making 1
1.2 The Problem of Latent Information 2
1.2.1 The Central Compass 2
1.2.2 A Running Example 5
1.3 Why These Settings Are Still Hard 9
1.4 Tractability through Structured Latent Information 10
1.5 Algorithms, Guarantees, and Experiments 12
1.6 Outline 15

2 Preliminaries 17

2.1 Markov Decision Processes 17
2.1.1 Regret and Risk 17
2.1.2 Optimism in the Face of Uncertainty 18
2.1.3 Visitation Distributions and Mixing 20
2.2 Bandits 20
2.2.1 Linear Structure 21
2.3 Offline Reinforcement Learning and Policy Evaluation 22
2.3.1 Importance Sampling 22
2.3.2 Fitted Q-Evaluation and Fitted Q-Iteration 23
2.4 Reinforcement Learning from Human Feedback 24
2.4.1 Current Approaches 25
2.5 Structured Partial Observability 26
2.5.1 Partially Observable MDPs 26

2.5.2 PORMDPs . . . 27
2.5.3 Latent Time Series and Mixture Models . . . 28
2.5.4 Taxonomy of Structured Partial Observability . . . 29
2.6 Notation . . . 30

3 Learning Mixtures of Markov Chains and MDPs . . . 31

3.1 Introduction . . . 31
3.1.1 Summary of Contributions . . . 33
3.2 Background and Problem Setup . . . 34
3.3 Algorithm . . . 35
3.3.1 Setup and Notation . . . 35
3.3.2 Overview . . . 36
3.3.3 Subspace Estimation . . . 36
3.3.4 Clustering . . . 38
3.3.5 Model Estimation and Classification . . . 40
3.4 Analysis . . . 40
3.4.1 Techniques and Proofs . . . 42
3.4.2 Subspace Estimation . . . 43
3.4.3 Clustering . . . 44
3.4.4 Model Estimation and Classification . . . 44
3.5 Practical Considerations . . . 45
3.5.1 Subspace Estimation . . . 45
3.5.2 Clustering . . . 46
3.6 Experiments . . . 47
3.7 Discussion . . . 49
3.7.1 Future Work . . . 50

4 Offline Policy Evaluation and Optimization under Confounding . . . 51

4.1 Introduction . . . 51
4.2 Setup and Assumptions . . . 55
4.2.1 Background . . . 55
4.2.2 Assumptions on Sensitivity and Memory . . . 55
4.2.3 FQE and Confounded FQE . . . 56
4.2.4 Model-Based Method For Stationary Transition Kernels . . . 58
4.2.5 Hardness of OPE for Confounders with Memory . . . 61
4.2.6 Clustering-Based OPE for Global Confounders . . . 61
4.2.7 Policy Optimization under Confounding . . . 63
4.3 Numerical Experiments . . . 65
4.4 Conclusion and Future Work . . . 67

5 A Theoretical Framework for Partially-Observed Reward States in RLHF . . . 69

5.1 Introduction . . . 69
5.1.1 Related Work . . . 71
5.2 Defining RL with Partially-Observed Reward States (PORRL) . . . 72
5.2.1 PORMDPs . . . 73

5.2.2 Reinforcement Learning in PORMDPs (PORRL) with Cardinal and Dueling Feedback . . . 75
5.2.3 A General Yet Tractable Case . . . 77
5.3 Optimistic Algorithms for Cardinal PORRL . . . 79
5.3.1 Improving over Naive History-Summarization with Model-Based Methods 79
5.3.2 Leveraging Recursive Structures Using Model-Free Methods . . . 81
5.4 Dueling to Optimism Reduction . . . 83
5.4.1 The Naive Reduction Always Fails . . . 83
5.4.2 Reducing Dueling to Optimistic Cardinal PORRL . . . 84
5.5 Conclusions and Future Work . . . 85

6 Leveraging Offline Data in Linear Latent Contextual Bandits . . . 87

6.1 Introduction . . . 87
6.2 Linear Bandits With Latent Structure . . . 90
6.3 Estimating Latent Subspaces Offline . . . 92
6.4 Offline Data Sharpens Online Optimism . . . 95
6.5 Lower Bound . . . 96
6.6 Practical Optimism with ProBALL-UCB . . . 97
6.7 Experiments . . . 100
6.8 How General Are Latent Bandits? . . . 102
6.9 Discussion, Limitations and Further Work . . . 104

7 Conclusion . . . 106

7.1 Summary through the Compass . . . 106
7.2 Cross-Cutting Themes . . . 107
7.3 Practical Considerations . . . 108
7.4 Future Directions . . . 108
7.5 Closing Remarks . . . 110

APPENDICES . . . 111

BIBLIOGRAPHY . . . 285

LIST OF FIGURES

FIGURE

1.1 The central compass of this thesis as causal DAGs. Solid circles denote observed variables; dashed circles denote latent variables. Black arrows are standard MDP edges; solid red arrows are active latent influence channels; dotted gray arrows are channels removed by confinement. Blue rectangles denote the behavior policy π_b (present in offline settings). The annotation below each latent node indicates the complexity of the latent channel: K mixture components, Γ sensitivity parameter, d_E/d_{HABE} eluder/history-aware Bellman eluder dimension, d_K latent subspace dimension. **Top:** In a general POMDP, the latent state u_h influences transitions, rewards, and the behavior policy simultaneously, leading to statistical intractability. **Bottom:** Each chapter confines the latent variable to a subset of channels, making the problem tractable. Chapters 3 and 4 confine the latent state away from rewards; Chapters 5 and 6 confine it to rewards alone. 3

1.2 The autonomous driving running example across the four chapters. In each panel, red boxes denote latent variables and red arrows show their influence channels. **(a)** and **(b)** are offline settings with an unobserved road type affecting transitions and behavior policy; **(b)** additionally includes memoryless hazards (⚠). **(c)** and **(d)** are online settings where the latent variable affects only the reward. In **(c)** the passenger's internal state evolves over time; in **(d)** the preference is fixed, and the action is an entire driving policy over a full episode. 6

3.1 Breaking up a trajectory into 4 segments and G blocks per segment ($G = 4$) for the single-step estimator. Observations are only recorded at the orange points. 36

3.2 Histogram of the average ordered eigenvalue energy (the square of the eigenvalue) where the mean is taken over states and actions. There are two large eigenvalues, corresponding to $K = 2$. 45

3.3 Histogram (and KDE) of pairwise squared distance estimates in projected subspace above, and accuracy against thresholds below. Note how there is a spurious mode around the 0.00015 mark, and picking any threshold past it yields a significant drop in accuracy. 46

3.4 Clustering error v.s. trajectory length on 1000 trajectories, with a comparison between using $\mathbf{V}_{s,a}^T$, $I_{S\times S}$ or a random projector to a K-dimensional subspace in Algorithm 2. The same threshold was used for each trajectory length. Results averaged over 30 trials. The mixing time of this system is roughly $t_{mix} \approx 25$. 48

3.5 End-to-end error v.s. trajectory length on 1000 trajectories, comparing initializations of the soft EM algorithm using (1) random initializations, (2) models from $\mathcal{N}_{clust}$, and (3) classification and clustering labels from $\mathcal{N}_{clust}$ and $\mathcal{N}_{sub}$. Results averaged over 30 trials, with 30 random initializations for randomly-initialized EM within each trial. . . 49
3.6 Scatter-plot of likelihoods v.s. clustering accuracy achieved by the randomly-initialized soft EM algorithm over 30 trials on gridworld. Randomly-initialized soft EM does not achieve the global maximum all of the time. 49

4.1 OPE for Memoryless Confounders. Comparison of our model-based method, its non-stationary relaxation (Alg. 17), its projected gradient descent variant (Alg. 6), and CFQE on state 13 in a 16-state gridworld. Confidence intervals (CIs) are one standard deviation wide and computed over 30 trials. $H = 8$. 66
4.2 Policy Improvement for Memoryless Confounders. Top Left: Loss curve dynamics of max-min gradient descent. Top Right: Resulting policy $\hat{\pi}^*$ for $\Gamma = 10$ in 4x4 gridworld with actions indexed by WENS. Brighter colors indicate higher $\hat{\pi}^*(a \mid s)$. Bottom: Increase in the lower bound on $V_1^{\pi_\theta}$ as gradient ascent iterations progress. $H = 8$. 67
4.3 Top Left: Average performance of the clustering method from Kausik et al. Top Right: Average relative error of clustering-based OPE with different clustering algorithms. Bottom: Improvement in estimates of policy values under gradient ascent coupled with different clustering algorithms, see Appendix B.1 for details. We average over 30 trials, confidence intervals are 1 standard deviation wide. $H = 60$. 68

5.1 Illustrating how a human's internal states (represented by emojis) affect their feedback to an agent or LLM. Top: Cardinal or good/bad feedback. Bottom: Dueling or preferential feedback. In line with Definition 5.2.1, $u_h \in \mathcal{U}$ are represented by the emojis, $p = 2$ and $\mathcal{H}_p = \{2, 4\}$ in both cases. 72

6.1 **Left:** Geometric interpretations of LOCAL-UCB. Showing $\mathcal{C}^t_{\text{on}}(\boldsymbol{\beta}) \cap \mathcal{C}^t_{\text{off}}(\boldsymbol{\beta})$ in green for three timepoints $t = t_1, t_2, t_3$. The dotted lines delineate the subspace confidence set. **Right:** Geometric interpretation of ProBALL-UCB. $\mathcal{C}^{t_1}_{\text{on}}(\boldsymbol{\beta}) \not\subset \widetilde{\mathcal{C}}^{t_1}_{\text{off}}(\boldsymbol{\beta})$, so we continue to use projections; but by time t_2, $\mathcal{C}^{t_2}_{\text{on}}(\boldsymbol{\beta}) \subset \widetilde{\mathcal{C}}^{t_2}_{\text{off}}(\boldsymbol{\beta})$, so we stop using projections. 98
6.2 Left to Right. **First:** Simulation study comparison of ProBALL-UCB against LinUCB for $\tau = 5$. **Second/Third:** Comparison of ProBALL-UCB initialized with SOLD against {LinUCB, mUCB, and mmUCB, TS, mmTS, and MixTS}, for $\tau = 0.1$ and various confidence bound constructions. ProBALL-UCB outperforms all other algorithms, and approaches the performance of LinUCB when Hoeffding confidence sets are used. **Fourth**: ProBALL-UCB regret on MovieLens against offline samples used in SOLD, compared to LinUCB on ground-truth low-dimensional features. Here, $\tau = 0.1, T = 200$. As the number of offline samples increases, SOLD recovers a low-rank subspace almost as good as ground-truth. The shaded area in each subfigure depicts 1-s.e. confidence intervals over 30 trials with fresh $\boldsymbol{\theta}$, accounting for the variation in frequentist regret for changing $\boldsymbol{\theta}$. 101

A.1 Block structure of the matrix of squared pairwise distance estimates (after sorting). . . 111

A.2 End-to-end error v.s. trajectory length on (left) 1000 MDP trajectories from the grid-world dataset and (right) 750 Markov chain trajectories from the Last.fm dataset, comparing various initializations of the soft and the hard EM algorithm. Results averaged over 30 trials, with 30 random initializations for randomly-initialized EM within each trial. 112
A.3 Clustering error using random projections of varying dimension for a trajectory length of 100, benchmarked against the performance of the "with subspace" and "without subspace" versions. The gridworld MDP dataset is on the left, while the Last.fm Markov chain dataset is on the right. 113

D.1 Plot of eigenvalues of aforementioned matrix. Notice the drop after 18 eigenvalues. . 274
D.2 Log-plot of eigenvalues of aforementioned matrix. Notice the drop after 18 eigenvalues. 275
D.3 Comparison of ProBALL-UCB with LinUCB, for different choices of τ and confidence bound constructions. All variants perform no worse than LinUCB, with martingale Bernstein performing the best. The shaded area depicts 1-standard error confidence intervals over 30 trials. 276
D.4 Comparison of ProBALL-UCB with LinUCB and TS algorithms, for different choices of τ and confidence bound constructions. All variants perform no worse than LinUCB and outperform the TS algorithms, with martingale Bernstein performing the best. The shaded area depicts 1-standard error confidence intervals over 30 trials. 277
D.5 Comparison of ProBALL-UCB and ProBALL-TS initialized with SOLD against LinUCB, TS, MixTS, and mmTS, for different choices of τ and confidence bound constructions. ProBALL-UCB outperforms LinUCB, and ProBALL-TS outperforms MixTS and mmTS. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh $\boldsymbol{\theta}$. The confidence intervals on regret thus account for the variation in frequentist regret for changing $\boldsymbol{\theta}$. 278
D.6 Comparison of ProBALL-UCB initialized with SOLD against LinUCB, mUCB, and mmUCB, for different choices of τ and confidence bound constructions, in terms of regret. All variants of ProBALL-UCB perform no worse than LinUCB, and outperform mUCB and mmUCB. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ. 279
D.7 Comparison of ProBALL-UCB initialized with SOLD against LinUCB, mUCB, and mmUCB, for different choices of τ and confidence bound constructions, in terms of rolling average rating over 25 timesteps. ProBALL-UCB performs no worse than LinUCB, and outperforms mUCB and mmUCB. 279
D.8 Comparison of ProBALL-TS initialized with SOLD against TS, mmTS, and MixTS, for different choices of τ and confidence bound constructions, in terms of regret. All variants of ProBALL-TS outperform TS, mmTS, and MixTS. 280

D.9 Comparison of ProBALL-TS initialized with SOLD against TS, mmTS, and MixTS, for different choices of τ and confidence bound constructions, in terms of rolling average rating over 25 timesteps. All variants of ProBALL-TS outperform TS, mmTS, and MixTS. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ. 280
D.10 Comparison of ProBALL-UCB initialized with SOLD against LinUCB, mUCB, and mmUCB on low-dimensional ground-truth features, for different choices of τ and confidence bound constructions. When τ is small enough, all variants of ProBALL-UCB perform no worse than low-dimensional LinUCB, and outperform mUCB and mmUCB, on ground truth features. This showcases the efficacy of SOLD, and demonstrates that we recover subspaces that are just as good as ground-truth. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ. . 281
D.11 Comparison of ProBALL-UCB initialized with SOLD against LinUCB, mUCB, and mmUCB on low-dimensional ground-truth features, for different choices of τ and confidence bound constructions. When τ is small enough, all variants of ProBALL-UCB perform no worse than low-dimensional LinUCB, and outperform mUCB and mmUCB, on ground truth features. This showcases the efficacy of SOLD, and demonstrates that we recover subspaces that are just as good as ground-truth. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ. . 281
D.12 Comparison of ProBALL-UCB initialized with ground truth subspaces against LinUCB, mUCB, and mmUCB, for different choices of τ and confidence bound constructions. All variants of ProBALL-UCB perform no worse than LinUCB, and outperform mUCB and mmUCB. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ. 282
D.13 Comparison of ProBALL-UCB initialized with ground truth subspaces against LinUCB, mUCB, and mmUCB, for different choices of τ and confidence bound constructions, in terms of rolling average rating over 25 timesteps. All variants of ProBALL-UCB perform no worse than LinUCB, and outperform mUCB and mmUCB. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ. 282
D.14 Subspace estimation error of SOLD against the number of offline samples, in the Frobenius norm. This was performed on the MovieLens dataset. We compare the error of SOLD against the parametric rate of $1/\sqrt{N}$. This shows that the error of SOLD indeed decreases very quickly in practice. 283
D.15 End-to-end regret at $T = 200$ timesteps of ProBALL-UCB initialized with SOLD, against the number of offline samples used in fitting SOLD. With a low enough τ, the regret of ProBALL-UCB approaches the regret of LinUCB on ground-truth low-dimensional features, showing that we lose next to nothing from needing to estimate the subspace with SOLD. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. 284

D.16 End-to-end regret at $T = 200$ timesteps of ProBALL-TS initialized with SOLD, against the number of offline samples used in fitting SOLD. With a low enough τ, the regret of ProBALL-TS approaches the regret of TS on ground-truth low-dimensional features, showing that we lose next to nothing from needing to estimate the subspace with SOLD. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. 284

LIST OF APPENDICES

A Supplementary Material for Chapter 2 . . . . . **111**

B Supplementary Material for Chapter 3 . . . . . **157**

C Supplementary Material for Chapter 4 . . . . . **191**

D Supplementary Material for Chapter 5 . . . . . **243**

ABSTRACT

Sequential decision-making problems in domains like healthcare, recommendation systems, and language model alignment are frequently affected by latent variables such as unobserved confounders, hidden mixture structure, or partially-observed internal states that influence the data-generating process. This is compounded by the practical constraint of learning from pre-collected offline datasets rather than a lot of online interaction. While the partially observable MDP (POMDP) provides the most general framework for such settings, learning in POMDPs is statistically intractable in general. Understanding what structural assumptions make learning tractable, and how to design algorithms that exploit them, is therefore of great importance. This thesis presents four contributions to this area, unified by the following insight:

Sequential decision-making is tractable when the latent variable's influence is *confined* to one part of the data-generating process (such as the state/context or the reward), and within that part it acts through a *low-complexity channel*, such as a low-dimensional subspace or a function class of bounded dimension.

We develop spectral methods for recovering latent mixture and subspace structure, optimistic methods calibrated to the complexity of the latent channel, and conservative estimation methods for settings where the latent structure prevents consistent estimation. In key settings, we first establish impossibility results that delineate what is and is not achievable. For instance, we show that consistent policy evaluation is impossible under memoryless confounding even with sensitivity constraints, that naive history-summarization in RLHF leads to complexity exponential in the horizon, and that even full data coverage is not sufficient for latent subspace recovery when certain natural independence conditions are violated. We then identify natural assumptions that must be made to avoid these impossibility results. Finally, we provide algorithms whose statistical complexity scales with the dimension of the latent channel rather than the ambient problem. We frequently chase lower bounds and optimality guarantees in an effort to identify and achieve the statistical limits of what can be learnt in these settings. Our results thus include various novel theoretical guarantees and structural characterizations of these settings.

For each contribution, we present experiments with both synthetic and real-world data, often in medical and recommendation settings. We provide practical recommendations for choosing hyper-

parameters, perform ablations studying the impact of major algorithmic components, and compare our methods with existing approaches, demonstrating that they outperform existing methods in realistic settings.

CHAPTER 1

Introduction

1.1 Sequential Decision Making

Many problems in machine learning require not a single prediction, but a sequence of decisions whose consequences unfold over time. A recommendation system selecting which content to surface [Li et al., 2010], a clinical algorithm deciding on a treatment at each hospital visit [Lu et al., 2021b], and a language model choosing how to phrase a response to a user [Ouyang et al., 2022] all share this structure. At each step, an *agent* observes something about the world, takes an *action*, and receives a *reward* signal encoding how well it did. The goal is to learn a *policy* (a mapping from observations to actions) that maximizes cumulative reward over time.

Two axes organize the space of such problems. The first is whether decisions are *stateful*. When an action today changes what the agent sees tomorrow, the standard framework is the *Markov Decision Process* (MDP) [Puterman, 2014], where a state s_h evolves according to a Markovian transition kernel $\mathbb{P}(s' \mid s, a)$ and the agent optimizes over an episode of H steps. A large body of theory and algorithms applies when the Markov property holds [Sutton and Barto, 2018, Auer et al., 2002, Jin et al., 2018]. When each decision is instead a one-shot choice in context — selecting a treatment dose, choosing which ad to display — the problem reduces to a *contextual bandit* [Lattimore and Szepesvári, 2020]. The second axis is whether the agent learns by *interacting* with its environment or from a fixed dataset. The former is the *online* setting; the latter is the *offline* or *batch* setting [Levine et al., 2020]. In many of the settings we study, online interaction is not economical, safe, or even possible — a hospital evaluating a new treatment protocol has only historical patient records — and learning from such pre-collected data is fundamentally harder, since the agent cannot query the environment for missing information. This thesis studies problems across these axes: Chapters 3 and 4 work in the offline MDP setting, Chapter 5 in the online MDP setting, and Chapter 6 bridges the offline and online bandit settings. Formal definitions are given in Chapter 2.

1.2 The Problem of Latent Information

Real environments contain *latent variables*: quantities that influence the world but are never directly observed. Unfortunately, classical MDP and bandit frameworks assume that the observed state s_h (or context x) captures everything relevant about the world, so that two agents in the same state facing the same action experience the same distribution of outcomes. What happens when the state does not capture everything relevant?

In personalized medicine, patients with similar observed symptoms respond very differently to the same treatments because of underlying genetic or physiological differences that are rarely measured [Kallus and Zhou, 2020]. A treatment policy learned from historical records will be confounded by these unrecorded characteristics: clinicians who generated the historical data had intuitions and contextual knowledge that shaped their decisions, and that knowledge is absent from the dataset. In recommendation, each user has an unobserved set of preferences that determines which items they find rewarding [Hong et al., 2020]; a dataset of past interactions aggregates trajectories from users with very different latent states, none of which are directly revealed. In autonomous driving, the physical environment type (icy road, dry highway, urban intersection) governs how the vehicle responds to steering and braking, but this type may not be directly measurable from onboard sensors alone.

In each case, the latent variable is a genuine feature of the problem. Ignoring it causes real harm: confounded offline evaluation leads to dangerously biased policy estimates; recommendation algorithms that treat a heterogeneous user population as a single entity explore inefficiently; a control policy unaware of latent environment type may behave safely in one regime and catastrophically in another. The most general framework for handling latent information is the *Partially Observable MDP* (POMDP) [Kaelbling et al., 1998a], where a hidden state u_h can influence both the transition dynamics and the rewards in an arbitrary, history-dependent way. In the applications above, each interaction with the environment is costly, so sample complexity is a practical concern. However, learning in POMDPs is known to be statistically intractable in general [Krishnamurthy et al., 2016, Jin et al., 2020]. No algorithm can learn efficiently without further structural assumptions, regardless of how much data is collected.

1.2.1 The Central Compass

What are some realistic POMDP settings where learning might be tractable? The central message of this thesis is that many tractable and realistic islands can be found in the intractable ocean of POMDPs via the following compass:

The latent state's influence is confined to one part of the observables, and within that part it acts through a low-dimensional/low-complexity channel.

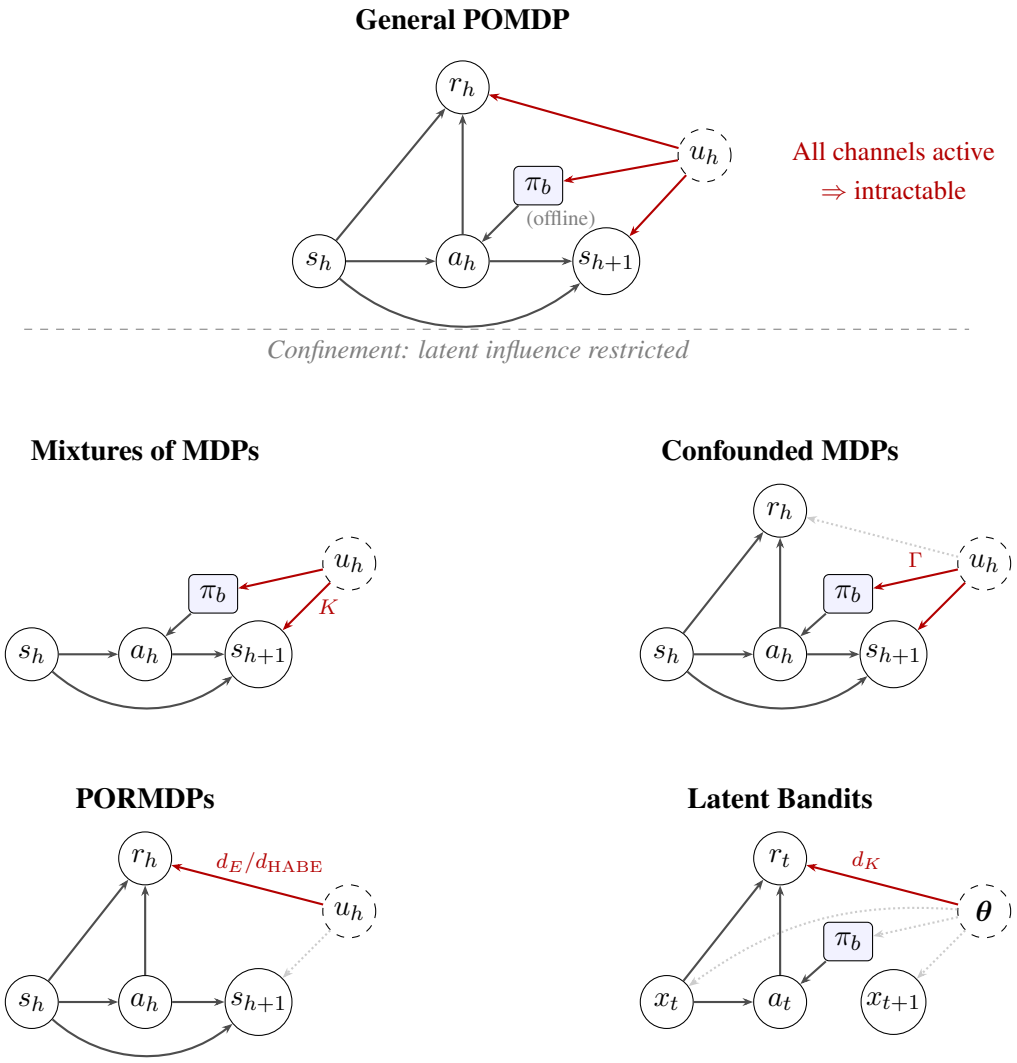

Figure 1.1: The central compass of this thesis as causal DAGs. Solid circles denote observed variables; dashed circles denote latent variables. Black arrows are standard MDP edges; solid red arrows are active latent influence channels; dotted gray arrows are channels removed by confinement. Blue rectangles denote the behavior policy π_b (present in offline settings). The annotation below each latent node indicates the complexity of the latent channel: K mixture components, Γ sensitivity parameter, d_E/d_{HABE} eluder/history-aware Bellman eluder dimension, d_K latent subspace dimension. **Top:** In a general POMDP, the latent state u_h influences transitions, rewards, and the behavior policy simultaneously, leading to statistical intractability. **Bottom:** Each chapter confines the latent variable to a subset of channels, making the problem tractable. Chapters 3 and 4 confine the latent state away from rewards; Chapters 5 and 6 confine it to rewards alone.

Confinement means the latent state does not permeate the full data-generating process. In

Chapters 3 and 4, the latent variable can affect both the transition dynamics and the data collection process, but the reward is defined from observables alone. In Chapters 5 and 6, the latent variable affects the reward, but transitions and contexts remain clean. In each case some part of the world remains unentangled with the latent state, and that clean part is what makes the problem tractable.

The *low-complexity channel* means that within its domain of influence, the latent state acts through something small. Sometimes this is geometric, such as a literal low-dimensional subspace of transitions or reward parameters. Sometimes it is learning-theoretic, such as a small eluder or coverability dimension of the reward function class. The form varies, but in every case the complexity of the latent channel is much smaller than the complexity of the ambient problem.

The standard POMDP literature sometimes assumes a form of confinement — for instance, Jin et al. [2020] assume rewards depend only on observations, noting this is "natural" in most applications. Such work benefits algorithmically from this assumption (e.g., it makes observable operator representations sufficient for planning), but the tractability conditions that emerge — such as invertible emission matrices or revealing conditions — can be somewhat abstract. Making confinement an explicit design principle, as in this thesis, allows us to choose interpretable conditions that directly describe the complexity of the channel through which the latent variable acts.

In particular, we will see that standard algorithms are not designed to take advantage of these low-complexity channels. This thesis can be considered as a series of independently useful examples in applying this philosophy when working with POMDPs. Namely, we suggest a two-step process:

- Step 1: Identify whether and how the latent variables' effects are confined. Crucially, this will require domain knowledge and making untestable assumptions (such as adequate coverage of the latent variables).
- Step 2: Use the three step template developed in the thesis (decoupling, leveraging the low-complexity channel and aggregating across trajectories, composing) to devise good learning algorithms. Namely, we decouple the parts affected and not affected by the latent variable. We handle the latter with standard techniques, and handle the former by aggregating across trajectories using spectral methods and other techniques developed in this thesis. Finally, we compose the results from both parts while accounting for sequential decision-making considerations.

Step 1 explicitly captures why we believe that an application is tractable to learn; this is a crucial missing step in prior work. And Step 2 provides a general recipe for devising good algorithms beyond ad hoc procedures for specific real-world POMDPs. Without these steps, applying state-of-the-art algorithms to POMDPs can fail silently, even when structure (low-complexity confined latents) is present.

The entire framework assumes the latent variable *cannot be directly accessed*. In some applications, the agent can pay a cost to make the latent observable — running a diagnostic test to determine a patient's genotype, requesting a user to fill out a detailed preference survey, or equipping a vehicle with additional sensors to identify road surface type. Such actions lead to problems in cost-aware or multi-fidelity decision making. Our framework addresses the complementary setting: the latent information is genuinely inaccessible, and the agent must learn to act well despite never observing it, relying on the structure of its influence.

1.2.2 A Running Example

The four chapters of this thesis each address a distinct form of latent information, distinct in where the latent variable enters, what part of the observable world it affects, and what kind of harm it causes when ignored. To see the differences clearly, consider an *autonomous driving system* operating across varied road environments. At each timestep h, the vehicle observes its driving situation s_h (position, speed, road geometry, surrounding traffic), takes a driving action a_h (acceleration, braking, steering adjustment, lane change), and receives a reward r_h measuring ride quality. The granularity of the action varies: in the first three chapters, a_h is a concrete driving decision at each timestep; in the fourth, it is an entire driving policy parameterized by a vector. Across all four chapters, the observable state stays the same, but the latent variable, its role, and the details of the reward shift from chapter to chapter.

Latent road types and mixture structure (Chapter 3). Road surfaces are not homogeneous. Dry asphalt, wet pavement, packed gravel, and black ice all produce very different vehicle responses to the same steering and braking inputs. On dry asphalt, a sharp turn at moderate speed is routine; on black ice, the same maneuver causes a skid. These are latent *road types*: distinct physical regimes in which the transition kernel $\mathbb{P}(\text{next vehicle state} \mid \text{current state, driving action})$ genuinely differs. The road type is determined at the start of each trip segment and persists throughout, but is not directly observed in the historical data — for privacy reasons, the vehicles that collected it did not record camera footage or fine-grained location data that could identify the road surface, only tire slip, lateral acceleration, and coarse GPS position. The *state* s_h is the vehicle's observable driving situation at time h, the *action* a_h is the driving decision, and the *reward* r_h measures safety and efficiency from the available sensor readings. We observe N unlabeled trip segments from varied road conditions without knowing which type each belongs to, and the goal is to recover the latent road types and their distinct dynamics purely from transition data. The human drivers who generated these trips perceived and adapted to the road conditions — they drove more cautiously on ice, more aggressively on dry asphalt — so the behavior policy π_b also depends on the latent road

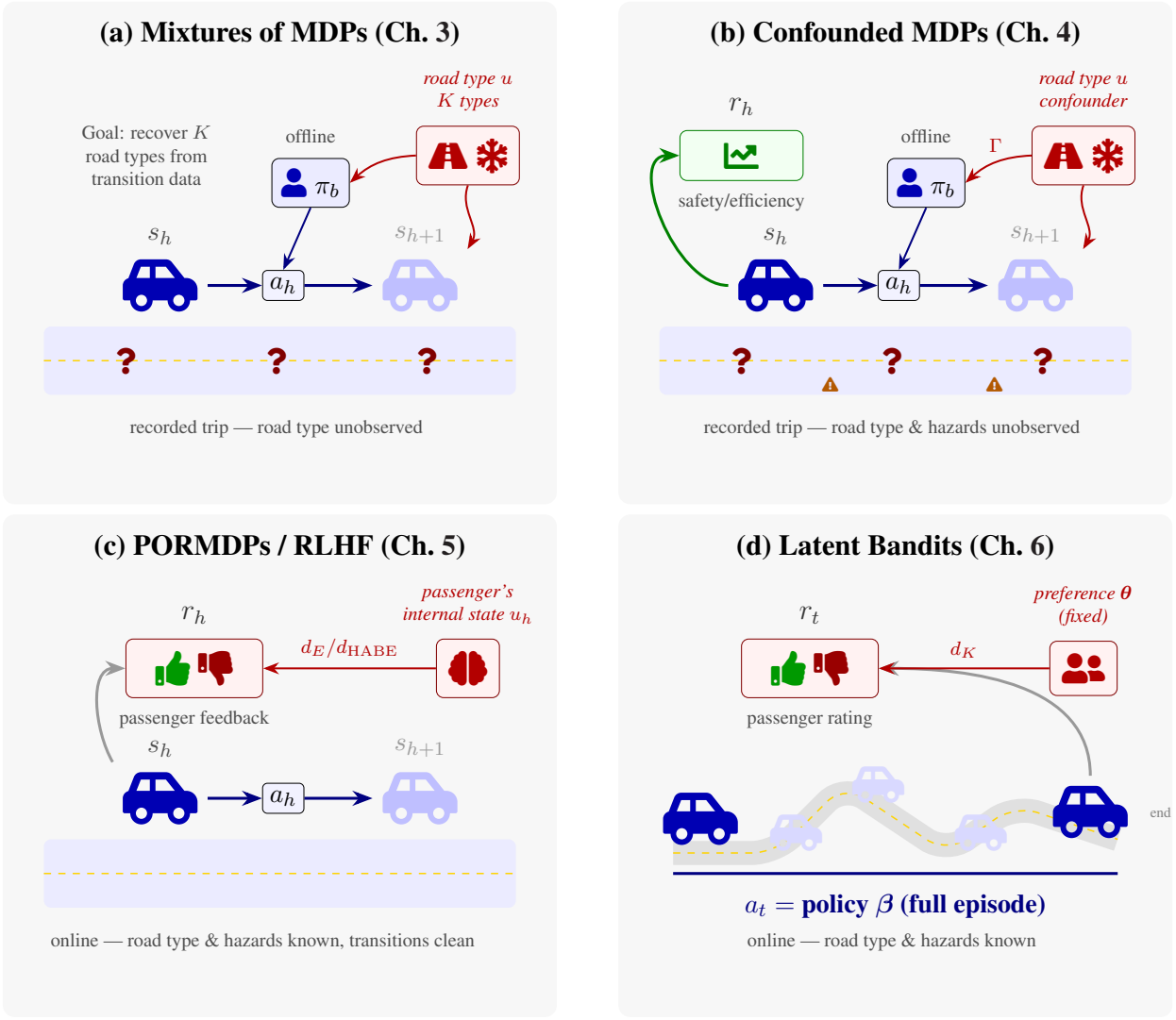

Figure 1.2: The autonomous driving running example across the four chapters. In each panel, red boxes denote latent variables and red arrows show their influence channels. **(a)** and **(b)** are offline settings with an unobserved road type affecting transitions and behavior policy; **(b)** additionally includes memoryless hazards (⚠). **(c)** and **(d)** are online settings where the latent variable affects only the reward. In **(c)** the passenger's internal state evolves over time; in **(d)** the preference is fixed, and the action is an entire driving policy over a full episode.

type. This is the *mixture of K MDPs* setting [Kwon et al., 2021]. The latent road type is confined to the transitions and the behavior policy: it determines how the vehicle responds to control inputs and how the driver acts, but the reward (defined from observable sensor data) plays no role in the clustering task. The low-complexity channel is geometric, in that the K transition kernels span at most a K-dimensional subspace of the probability simplex for each state-action pair, small enough to recover from population data without ever observing which road type generated any given trip.

Confounded offline evaluation (Chapter 4). Now suppose we want to use historical driving data — collected from human drivers — to evaluate how an autonomous policy would have performed, without deploying it on the road. Again, the state, action, and reward are as above. As before, the latent road type affects the transition dynamics, and the human drivers who collected this data perceived the road conditions and adapted accordingly. The difference is the task: the driver's dependence on the road type, which was merely a feature of the data in the clustering setting, now becomes the central obstacle. The latent road type acts as a *confounder* [Kallus and Zhou, 2020, Bruns-Smith, 2021]: it affects both the transition dynamics and the behavior policy, biasing the historical trajectories in ways invisible to standard offline analysis. The reward remains defined from observable safety and efficiency metrics. The confounder structure is particularly rich: the road type itself (dry, wet, icy) functions as a *global confounder*, stable across an entire trip segment and shaping both the vehicle dynamics and the driver's baseline behavior, while moment-to-moment hazards — a pothole, a road bump, a patch of gravel — constitute *memoryless* confounding, refreshed independently at each timestep: each affects both how the car responds and how the driver reacts, but is not recorded in the data. The confinement is that the reward remains defined from observables alone, exactly as in Chapter 3. For global confounders, there is additional low-complexity structure: the number of latent road types is small. Memoryless confounders lack this, and as we will see, admit fundamental unidentifiability — confinement alone is not enough. This is the *confounded MDP* setting.

Partially observed reward states and RLHF (Chapter 5). The first two settings optimize safety and efficiency — objectives defined entirely from sensor data. Once basic safety and efficiency are handled, a natural next step is to personalize the driving experience to passenger comfort, which sensors alone cannot capture. Passengers can provide real-time feedback during the ride (a thumbs up or down after a lane change or braking event) as well as an overall rating at the end [Christiano et al., 2017, Ouyang et al., 2022]. Human annotators can also compare pairs of recorded trajectories and judge which provided the better ride. At forking decisions — take the highway or surface streets, brake early and gently or late and firmly — feedback can be solicited from passengers or annotators choosing between the two options. The state s_h is still the vehicle's driving situation, the

action a_h is still the driving decision, but the reward is now the passenger's or annotator's feedback rather than a sensor metric. The difficulty is that this feedback depends not only on the observable trajectory but also on the passenger's *internal state* [Flavell et al., 2022]: their anxiety level during a highway merge, whether they were reading or watching the road, their accumulated trust (or distrust) in the system, their sensitivity to lateral motion. This internal state evolves throughout the trip but is never directly observed. Crucially, it affects the *reward* (how the passenger rates a maneuver) but not the vehicle's transition dynamics — the physics of driving does not depend on whether the passenger is anxious. Having learned the latent road types and their distinct dynamics from the historical data of the first two settings, the system can now go online — equipped with cameras and sensors that identify road conditions directly — and treat the transition model as known. The latent variable is thus confined to the reward channel; the transitions remain clean. The low-complexity channel, when it exists, is in the reward function class: its eluder or history-aware Bellman eluder dimension can be far smaller than the full history space. This is the *partially-observed reward-state MDP* (PORMDP) setting. The comparison-based feedback from annotators and passengers is a natural instance of the *dueling* feedback model, a special case of the same setting.

Latent subspace structure in bandits (Chapter 6). The previous setting handles a generic PORMDP where the passenger's hidden reward state u_h can evolve arbitrarily over an episode. In principle, the algorithms of Chapter 5 solve this problem, but the resulting history-dependent policies can be data-intensive to learn and difficult to deploy reliably when many state-action histories are rarely visited. Often, though, the hidden state is simpler than the general case: a passenger's comfort preferences may be a fixed "personality" vector $\boldsymbol{\theta}$ that stays constant throughout the ride, determining how every maneuver is rated. When the hidden reward state does not evolve — when it is simply a latent preference — the history dependence collapses, and the problem reduces to personalization: given a family of safe driving policies parameterized by a vector $\boldsymbol{\beta} \in \mathbb{R}^{d_A}$ encoding tradeoffs between smoothness, speed, fuel efficiency, and dozens of other factors, which setting of $\boldsymbol{\beta}$ suits each passenger? Across the passenger population, effective preference directions lie in a d_K-dimensional subspace with $d_K \ll d_A$ — the low-complexity channel: most passengers' preferences are combinations of a small number of archetypal comfort styles. We have offline data from many past rides (observed driving situations, policy parameters used, and passenger ratings) and want to learn the latent subspace of preference directions from this data, then adapt faster online to a new passenger. Each ride segment is a single decision conditioned on the current driving situation, with no sequential transition dynamics to model. The latent preference is confined to the reward; crucially, the context x_h (the current driving situation) is generated independently of the passenger's latent preference $\boldsymbol{\theta}$. This independence — the confinement — turns out to be *necessary* for the latent subspace to be identifiable from offline data at all, as we establish formally.

The lack of statefulness brings us to a bandit setting, where the context is x_h and the action is the d_A-dimensional policy parameter vector. This is the *linear latent contextual bandit* setting.

Across all four settings the same pattern recurs: the latent variable is not everywhere. In the first two settings it leaves the reward clean; in the last two it leaves the transitions clean. Within its domain of influence it acts through something small rather than something arbitrarily complex. Understanding exactly why naive approaches fail to exploit this, and what the right tools are, is what the rest of this introduction develops.

1.3 Why These Settings Are Still Hard

The four settings above carry more structure than a generic POMDP and have some level of confinement and low-complexity structures, but that does not immediately buy tractability. Two kinds of hardness persist. First, without appropriate coverage and occasionally additional confinement assumptions, the settings admit fundamental impossibility results. Second, standard algorithmic approaches fail in ways that demand new methods that utilize the confinement and low-complexity structures. The results below are novel contributions of this thesis in their own right - they define the exact statistical boundaries of these settings that our algorithms will then be designed to reach.

Fundamental unidentifiability. Latent information can make two very different worlds look identical from the outside. In the confounded MDP setting, one can construct two environments $\mathcal{M}_1, \mathcal{M}_2$ and behavior policies $\pi_{b,1}, \pi_{b,2}$ whose observed data distributions are identical, yet they have a value gap of $\Omega(H)$, the worst possible error (Theorem 4.2.1). Consistent estimation is impossible even with infinite data. In the latent bandit setting, the failure is even more severe: even with full coverage of every context-action pair and infinitely many infinitely long trajectories, if any of three natural conditions hold (contexts within a trajectory are mutually dependent, contexts depend on the latent state, or the behavior policy uses the latent state) then two bandits with *orthogonal* latent subspaces produce identical offline data (Lemma 5.2.1). An action that gives maximum reward on one bandit gives reward 0 on the other. Full coverage is not enough. Infinite data is not enough. A subtler form of the same problem is that the coverage condition that *would* make subspace recovery possible, namely that the offline population spans the full latent subspace, is invisible from the data alone. An offline dataset can have perfect action coverage and still be useless if all past users share the same latent type.

Algorithmic failures of natural approaches. What goes wrong if we use standard tools? In general, the geometry or learning-theoretic structure of the problem creates a trap that any algorithm

must navigate around explicitly. The EM algorithm [Dempster et al., 1977] is the standard approach for mixture model learning, but it is not convex, and on mixtures of MDPs it frequently converges to poor local optima with no reliable signal of its own failure. Empirically, low log-likelihood and low clustering accuracy coincide (see Chapter 3), so the objective function cannot distinguish failure from success. Since the mixture and latent bandit settings both involve recovering a low-dimensional subspace from trajectory data, dimensionality reduction is a natural next attempt. However, principal component analysis fails silently in both settings: per-trajectory estimation noise makes the empirical second-moment matrix full rank, causing PCA to conflate signal and noise even with infinite offline data. Probabilistic matrix factorization, another natural unsupervised approach, provides point estimates but no confidence bounds and no principled estimate of the latent dimension. In the RLHF setting, the most direct approach is to treat the full trajectory as the state, but this produces a policy class of size $(SA)^{\Omega(H)}$ before any learning has occurred. A more refined approach is to use the Markovian transition structure and decouple reward learning across timesteps, but even this fails exponentially on problems with recursive internal state structure (Proposition 5.3.3). For confounders with memory, standard fitted-Q evaluation — the workhorse of offline policy evaluation — incurs $\Omega(H)$ irreducible error regardless of dataset size (Theorem 4.2.6).

1.4 Tractability through Structured Latent Information

The previous section established that these settings are hard without the right assumptions and the right algorithms. How do confinement and low-complexity structure then translate into working methods? Three mechanisms recur across the four chapters. We first decouple the parts of the observables affected and not affected by the latent state, use standard methods to handle the latter and develop novel algorithmic approaches to handle the former by aggregation across trajectories, then compose the results for both kinds of parts while accounting for sequential decision-making considerations.

Leveraging confinement via decoupling. The impossibility results of the previous section confirm that proper confinement is *necessary*, not merely convenient. The direct algorithmic payoff of confinement is *decoupling*: because some part of the data-generating process is clean, it can be estimated by standard methods independently of the latent structure. In Chapters 3 and 4, rewards are defined from observables alone and can be estimated independently, while the latent structure in the transitions and data collection process is handled through spectral clustering or sensitivity bounds. In Chapters 5 and 6, Markovian transitions are handled by standard model-based RL while the reward channel is handled through function approximation calibrated to its own complexity. None of this would work if the latent state entangled the full data-generating process simultaneously,

as in a general POMDP.

Methods leveraging low-complexity/low-dimensional channels, like spectral recovery. Once decoupling separates out the part of the observables directly affected by the latent state, handling it efficiently needs algorithmic approaches attuned to the low complexity channel through which the latent state affects this part. In Chapters 3 and 6, where the channel is a literal subspace, the algorithmic payoff is *spectral recovery* via the double estimator [Vempala and Wang, 2004]: split each trajectory into two independent halves, estimate the relevant quantity from each half, form the cross-product, and average across trajectories. This targets the second-moment matrix $\mathbb{E}[\boldsymbol{\beta}_1\boldsymbol{\beta}_2^\top]$, whose top eigenvectors span the latent subspace. The splitting is essential. Squaring a single per-trajectory estimate and averaging gives a full-rank matrix due to per-trajectory estimation noise; this is exactly why PCA fails silently. The recovered subspace comes with explicit confidence bounds Δ_{off} that propagate into downstream algorithms in a principled way. In Chapter 5, where the channel has no finite parametric form, the analogous payoff comes through *reward-error or Bellman-error-based updates*: POR-UCRL, POR-UCBVI, and GOLF [Jin et al., 2021a] exploit the fact that transitions are Markovian in the observed state, reducing the full complexity of the problem to learning a reward function over the latent channel alone, with complexity governed by the eluder [Russo and Van Roy, 2013] and coverability dimensions rather than the size of the full history space. The central slogan is — identify the small object that captures the latent channel, and build an algorithm whose complexity scales with it.

Aggregation across trajectories and latent coverage. The latent state is never directly observed in a single trajectory; the structure only becomes visible through *aggregation across trajectories*. In Chapters 3, 4, and 6, this aggregation happens offline: by averaging cross-products across N trajectories, subspace estimation error decays at the parametric $N^{-1/2}$ rate once trajectory length exceeds a mixing threshold, shifting the statistical burden from trajectory length to trajectory count. In Chapter 3 this means sample complexity scaling linearly in S rather than exponentially in trajectory length. In Chapter 6 the offline-to-online benefit is captured exactly by the N-dependent term in the regret bound, making the offline phase a qualitative reduction in the effective dimension of the online problem, well beyond a warm start (which would reduce an additive initialization cost in the regret, but not change the dimension governing the per-round rate). In Chapter 5, aggregation happens online through episodic regret accumulation: the algorithm builds up evidence about the latent reward channel across rounds of interaction, and the regret bounds reflect how quickly that evidence accumulates relative to the channel's complexity.

Settings involving latent information also introduce a coverage requirement with no analogue in standard RL: the trajectory population must be diverse enough to span the latent structure, not just

cover state-action pairs. A dataset with perfect action coverage can be entirely useless for subspace recovery if all past trajectories share the same latent type. In Chapter 6 this is the requirement $\lambda_\theta > 0$; in Chapter 3 it is the model separation Δ and mixing time requirements; in Chapter 4 the sensitivity parameter Γ quantifies how far the behavior policy's knowledge of the confounder has distorted the observable distribution. In each case, the right coverage condition is a property of diversity in the latent space, not of the observable data alone. Like causal assumptions, these conditions are not testable from observed data and must be posited based on domain knowledge.

Testability and interpretability of assumptions. The assumptions underlying this thesis — confinement, model separation, mixing, latent coverage — are not fully testable from observed data. One cannot verify from offline sensor logs alone that the latent road type affects transitions but not rewards, or that the offline driving population spans all road types. In this sense, they share the status of causal assumptions like unconfoundedness. A natural concern is whether we are merely replacing one set of untestable conditions (e.g., the revealing or invertibility conditions of the POMDP literature) with another.

The difference is that our assumptions are considerably more *interpretable*. A domain expert can reason directly about whether "the latent road type affects the vehicle dynamics but not the definition of a safe outcome" — this is a statement about the physics of the application, not about the algebra of a statistical model. Because the assumptions describe the application rather than the statistical model, it is easier to understand what it means for them to *fail*, easier to judge from domain knowledge whether they are likely to hold, and easier to devise heuristics for partially checking them from data. Some quantities are directly estimable: the number of latent types K from the eigenvalue spectrum of the double estimator (Section 3.5.1, Figure 3.2), the latent dimension d_K from the rank of the SOLD estimator, and the model separation Δ from a scatter plot of pairwise trajectory distances. Others are partially checkable: the mixing time t_{mix} manifests as a threshold in trajectory length beyond which clustering accuracy improves sharply (Figure 3.4), and action coverage λ_A can be read from the minimum eigenvalue of the design matrix. Confinement itself, latent coverage λ_θ, and memorylessness of confounders remain genuinely untestable and must be justified by domain knowledge — but the impossibility results in this thesis (Theorem 4.2.1, Lemma 5.2.1) delineate precisely what goes wrong when they fail.

1.5 Algorithms, Guarantees, and Experiments

The latent dimension governs statistical complexity. The central quantitative payoff of the thesis is that in every setting, the right measure of statistical complexity is governed by the latent

channel, and the algorithms deliver guarantees that scale with it rather than with the ambient observable space. In Chapter 3, the end-to-end guarantee for learning mixtures of MDPs requires sample complexity linear in S and trajectory length linear in the mixing time t_{mix}. This replaces the exponential-in-trajectory-length dependence of naive methods entirely, and is the first such end-to-end guarantee. In Chapter 6, LOCAL-UCB's regret transitions smoothly from scaling with the ambient feature dimension d_A to scaling with the latent subspace dimension d_K as offline data grows, with a matching minimax lower bound (the first in any hybrid offline-online setting) confirming this is unimprovable. In Chapter 5, the algorithms' complexity is governed by the eluder and coverability dimensions of the reward function class rather than the size of the full history space; and when the channel has additional recursive internal state structure, model-free methods achieve an exponentially better dependence on horizon length than model-based methods provably can. The history-aware eluder dimension captures exactly when and why. Across all four chapters, the message is the same: find the dimension of the latent channel, and the algorithm's complexity reduces to scale with it.

Complete structural characterizations. Beyond individual algorithmic results, we prove several results that characterize entire problem classes, delineating what is and is not possible. Chapter 4 provides a complete picture of offline RL under confounding, distinguishing four regimes by confounder memory and sensitivity assumptions, with matching upper and lower bounds showing the landscape is tight throughout. In Chapter 5, the naive dueling-to-cardinal reduction fails for structural reasons: for any PORMDP and any sublinear cardinal algorithm, the feedback and regret objectives are fundamentally misaligned. The whitebox reduction developed in Theorem 5.4.2 resolves this by restricting both policies to those plausibly optimal under the current confidence set. This is the first explicit reduction from cardinal to dueling regret for MDPs, and it immediately converts all cardinal PORRL guarantees into dueling ones.

Chapter 6's de Finetti theorem (Theorem 6.8.1) serves a different purpose: it establishes the linear latent bandit as the canonical form of any stateless decision process whose latent preferences are stable within a trajectory and exchangeable across timesteps. The algorithms that follow are therefore solving the *right* model. Together, these results delineate what is possible in each setting and confirm that the models studied here are the right ones.

Experiments. The empirical results across the chapters make the theoretical phenomena visible and demonstrate that the practical stakes are real. In Chapter 3, the 96.6% clustering accuracy against 73.2% for randomly initialized EM is only part of the story; the scatter plot of log-likelihood against accuracy (Figure 3.6) reveals that EM cannot detect its own failure, since low likelihood and low accuracy coincide, while the spectral algorithm's improvement tracks the mixing time threshold

precisely as predicted. In Chapter 4, the naive FQE lower bound is far outside the useful range in the gridworld experiments; the model-based method's tighter bounds are a qualitative difference in usefulness. In the sepsis simulator [Oberst and Sontag, 2019], confounder-aware policy gradient significantly outperforms confounder-oblivious FQE and policy gradients, demonstrating the cost of ignoring latent structure in a safety-critical domain. In Chapter 6, SOLD's subspace error decreases at the $1/\sqrt{N}$ parametric rate on MovieLens-1M data [Harper and Konstan, 2015], and by roughly $N = 10^3$ offline samples ProBALL-UCB's end-to-end regret is indistinguishable from LinUCB given ground-truth low-dimensional features. The offline phase recovers the latent channel so completely that the online algorithm behaves as if the structure had been given for free.

Graceful degradation. Since our assumptions describe the application rather than the algebra of a model, we can ask concretely whether the guarantees degrade gracefully as assumptions weaken. There are two dimensions: degradation of the low-complexity channel, and degradation of confinement itself.

As the low-complexity channel degrades, the guarantees throughout the thesis widen continuously rather than breaking. In Chapter 3, as the model separation Δ or occupancy α shrinks, the mixture components become harder to distinguish in the offline distribution; if the downstream application also cannot distinguish them, this is benign, and if it can, the sample complexity grows polynomially in $1/\Delta$ and $1/\alpha$ rather than failing outright. In Chapter 4, the OPE error bounds scale as $O(\varepsilon H^2)$ where $\varepsilon = \Gamma - 1$. In Chapter 5, the guarantees scale with the eluder, HABE, and coverability dimensions of the reward function class. In Chapter 6, overestimating d_K is harmless — SOLD learns a subspace containing the true one, and the online algorithm pays for the extra dimensions but does not break — while underestimating d_K means missing part of the true subspace, causing the regret to default to the standard $d_A\sqrt{T}$ LinUCB rate. If the coverage constant λ_θ vanishes, the offline-to-online transfer weakens, but the regret never exceeds $d_A\sqrt{T}$: the algorithm falls back to ignoring offline data.

When confinement itself is violated, the picture depends on the setting. In Chapter 3, the clustering algorithm does not actually use the assumption that rewards are unaffected by the latent type; clustering remains possible as long as the latent type visibly affects *something*, and we can cluster on whichever observable quantity it influences. In Chapter 4, if the latent variable also affects rewards, the misspecification introduces additional bias; bounding this error and producing more conservative value estimates is a natural direction for future work. In Chapter 5, the model-free guarantees for GOLF depend on the HABE dimension of the Q-function class and do not use that the latent state leaves transitions unaffected; if the latent state also influences transitions, the HABE dimension may grow but the guarantees remain valid. The generic optimistic templates and

dueling-to-cardinal reductions depend on the learning complexity of the transition model through the confidence set or bonus construction — this complexity grows if transitions depend on the latent state, but the structure of the guarantees is preserved. In Chapter 6, if the latent state affects the context distribution or the behavior policy, the algorithms can continue to perform well unless these effects "cancel out" the latent state's effects on the reward, making two different latent bandits produce identical offline data. This is precisely the situation our impossibility result (Lemma 5.2.1) identifies, and in such adversarial cases the guarantee again degrades to the $d_A\sqrt{T}$ LinUCB rate.

1.6 Outline

Chapter 2: Preliminaries. We introduce the formal definitions and technical background used throughout the thesis, including Markov decision processes, Bellman equations, the regret and risk paradigms, the principle of optimism in the face of uncertainty, linear contextual bandits, offline policy evaluation, reinforcement learning from human feedback, and the taxonomy of structured partial observability studied in this thesis.

Chapter 3: Learning Mixtures of Markov Chains and MDPs. We develop an end-to-end algorithm for recovering the profile types and transition dynamics of a mixture of K MDPs from short unlabeled trajectories, combining spectral subspace estimation via the double estimator, pairwise distance-based trajectory clustering, and EM refinement. Formal guarantees show that the number of trajectories needed scales linearly in S and trajectory length linearly in the mixing time t_{mix}. To our knowledge, this is the first such end-to-end guarantee. We establish that the double estimator is the critical ingredient: naive PCA and random projections both fail. Experiments on gridworld mixtures attain 96.6% clustering accuracy against 73.2% for randomly initialized EM.

Chapter 4: Offline Policy Evaluation and Optimization under Confounding. We map the complete landscape of offline policy evaluation for confounded MDPs, distinguishing regimes by confounder memory structure (memoryless, global, general) and sensitivity assumption. For memoryless confounders under sensitivity constraints, we provide algorithms with tight lower bounds on policy value and matching information-theoretic hardness results. For global confounders, we reduce to Chapter 3 and obtain consistent estimates. For general confounders with memory, we prove an $\Omega(H)$ hardness result. We also present offline policy improvement algorithms with local convergence guarantees. Experiments on gridworld and a sepsis management simulator validate the theoretical landscape.

Chapter 5: A Theoretical Framework for Partially-Observed Reward States in RLHF. We introduce the PORMDP model for RLHF with partially observed internal states and intermediate feedback. We present POR-UCRL and POR-UCBVI, two model-based algorithms exploiting Markovian transitions and achieving regret $\tilde{O}((\mathrm{poly}(H, S, A) + p\sqrt{d_E d_C})\sqrt{T})$, subsuming and improving prior RLHF results. We study model-free methods and introduce the history-aware eluder dimension, showing model-free methods can be exponentially better than model-based methods on problems with recursive internal state structure. We prove that the naive reduction from dueling to cardinal feedback always fails, and provide the first explicit whitebox reduction converting any cardinal PORRL algorithm into a dueling one with the same asymptotic guarantee.

Chapter 6: Leveraging Offline Data in Linear Latent Contextual Bandits. We study offline-to-online transfer in linear latent contextual bandits, using N offline trajectories to accelerate online learning. We present SOLD, an offline spectral algorithm for learning the latent subspace with explicit confidence bounds. We present LOCAL-UCB, achieving minimax-optimal regret $\tilde{O}(\min(d_A\sqrt{T},\ d_K\sqrt{T}(1 + \sqrt{d_A T/d_K N})))$ with a matching lower bound (the first in any hybrid offline-online setting) and ProBALL-UCB, a computationally efficient variant with slightly looser guarantees. We establish the generality of the latent bandit model via a de Finetti theorem. Experiments on synthetic data and MovieLens-1M validate the end-to-end approach.

CHAPTER 2

Preliminaries

We establish common definitions, frameworks, and notation here; each subsequent chapter provides its own detailed setup tailored to its specific setting. Readers familiar with reinforcement learning may wish to skim this chapter and return to specific sections as needed.

2.1 Markov Decision Processes

We consider a (tabular, episodic) Markov decision process specified by a tuple $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r, d_0)$, where $\mathcal{S}$ is a finite state space with $|\mathcal{S}| = S$, $\mathcal{A}$ is a finite action space with $|\mathcal{A}| = A$, H is the horizon, $\mathbb{P}_h(s' \mid s, a)$ is the transition kernel at step h, $r_h(s, a)$ is the reward function, and d_0 is the initial state distribution. A policy $\pi = \{\pi_h\}_{h=1}^H$ maps states (and possibly histories) to distributions over actions. The value function and action-value (Q) function of a policy π are defined recursively via the Bellman equations:

$$Q_h^\pi(s, a) = r_h(s, a) + \sum_{s' \in \mathcal{S}} \mathbb{P}_h(s' \mid s, a) V_{h+1}^\pi(s'),$$
$$V_h^\pi(s) = \sum_{a \in \mathcal{A}} \pi_h(a \mid s)\, Q_h^\pi(s, a),$$

with the boundary condition $V_{H+1}^\pi(s) = 0$ for all s. The goal of reinforcement learning is to find a policy π^* that maximizes V_1^π over all policies. An optimal policy satisfies the Bellman optimality equation $V_h^*(s) = \max_a Q_h^*(s, a)$.

2.1.1 Regret and Risk

How should we measure a learning algorithm's performance? There are two dominant paradigms.

In the *online* setting, an algorithm interacts with the MDP over T episodes, producing policies

$\pi_1, \ldots, \pi_T$, and performance is measured by the *cumulative regret*

$$\mathrm{Regret}(T) = \sum_{t=1}^{T} V_1^*(s_1) - V_1^{\pi_t}(s_1),$$

which captures the total cost of exploration. Algorithms that achieve sublinear regret $\mathrm{Regret}(T) = o(T)$ are learning effectively: the average per-episode loss relative to the optimal policy vanishes.

In the *offline* setting, one is given a fixed dataset and must evaluate or optimize a policy without further interaction. The algorithm cannot explore to reduce uncertainty, so the primary concern shifts from regret to *risk*: for an evaluation policy π_e and behavior policy π_b, the goal is to estimate the value $V_1^{\pi_e}(s_1)$ from data collected under π_b, and the difficulty is governed by the distribution shift between π_b and π_e. When the behavior policy provides poor coverage of the evaluation policy, errors compound across the horizon H through the Bellman recursion, and even consistent estimation may be impossible without structural assumptions on the data collection process.

2.1.2 Optimism in the Face of Uncertainty

A widely used principle for online learning in MDPs is *optimism in the face of uncertainty* (OFU) [Auer et al., 2002, Jin et al., 2018]. Given a confidence set for the unknown model parameters constructed from past data, the learner acts according to the most favorable (highest-value) model in the confidence set. Optimism ensures systematic exploration, since the learner is driven to visit parts of the state space where uncertainty remains large, and hence optimistic value estimates are high. We describe three representative instantiations, differing in what the confidence set is built around.

UCRL: model-based optimism. The UCRL (Upper Confidence Reinforcement Learning) algorithm of Jaksch et al. [2010] takes a direct, model-based approach. At the start of each epoch, the algorithm constructs confidence sets $\mathcal{P}_t$ over the transition kernels using the empirical transition frequencies gathered so far. Concretely, for each state-action pair (s, a), the set $\mathcal{P}_t$ contains all transition distributions $p(\cdot \mid s, a)$ that lie within an ℓ_1-ball of radius roughly $\sqrt{S/n_t(s, a)}$ around the empirical transition $\hat{\mathbb{P}}(\cdot \mid s, a)$, where $n_t(s, a)$ is the number of visits. The learner then solves for the *optimistic MDP*: it finds the transition kernel $\tilde{\mathbb{P}} \in \mathcal{P}_t$ and corresponding policy $\tilde{\pi}$ that jointly maximize the long-run average reward. Because the true MDP lies in $\mathcal{P}_t$ with high probability, the optimistic value is an upper bound on the true optimal value, and any suboptimality of $\tilde{\pi}$ in the real environment is controlled by the diameter of the confidence set. UCRL operates in the infinite-horizon average-reward setting and achieves regret $\tilde{O}(S\sqrt{AT})$, where T is the total number of timesteps [Jaksch et al., 2010].

UCBVI: value-based optimism with exploration bonuses. The UCBVI (Upper Confidence Bound Value Iteration) algorithm of Azar et al. [2017] takes a value-based approach that avoids explicitly maintaining confidence sets over the model. Instead, UCBVI adds *exploration bonuses* directly to the estimated Q-function. At the start of each episode t, the algorithm performs backward induction: for each step $h = H, H-1, \ldots, 1$, it computes

$$\hat{Q}_h(s,a) = r_h(s,a) + \hat{\mathbb{P}}_h \hat{V}_{h+1}(s,a) + b_h(s,a),$$

where $\hat{\mathbb{P}}_h \hat{V}_{h+1}(s,a) = \sum_{s'} \hat{\mathbb{P}}_h(s' \mid s,a)\, \hat{V}_{h+1}(s')$ is the empirical Bellman backup and $b_h(s,a) \geq 0$ is a bonus term that shrinks as the triple (s,a,h) is visited more often. The greedy policy with respect to $\hat{Q}_h$ is then executed for the episode. The bonus inflates value estimates in under-explored regions, producing the same exploratory effect as explicit optimistic planning but with a simpler algorithmic structure. With Hoeffding-style bonuses of order $\sqrt{H^2/n_t(s,a,h)}$, UCBVI achieves regret $\tilde{O}(\sqrt{H^3SAT})$. Azar et al. [2017] showed that tighter Bernstein-style bonuses, which adapt to the variance of the value function, yield the minimax-optimal rate $\tilde{O}(\sqrt{H^2SAT})$ (up to logarithmic factors) for the episodic tabular setting. The exploration-bonus paradigm was further developed by Jin et al. [2018], who gave a streamlined analysis and extended it to the Q-learning setting.

GOLF: optimism with general function approximation. When the state-action space is large or continuous, tabular methods become infeasible, and one must work with function approximation. The GOLF (Generalized Optimistic Local Function approximation) algorithm of Jin et al. [2021a] provides an optimistic framework for MDPs with general function classes. Rather than maintaining confidence sets over transition models or adding bonuses to Q-function estimates, GOLF maintains a *version space* $\mathcal{Q}_t$ of value functions consistent with past observations. That is, $\mathcal{Q}_t$ consists of all Q-functions from a given function class $\mathcal{F}$ whose empirical Bellman errors on the collected data are small. At the start of each episode, the algorithm selects the most optimistic element of the version space:

$$Q_h^t \in \operatorname*{arg\,max}_{Q \in \mathcal{Q}_t} \; \mathbb{E}_{s \sim d_0}\big[\max_a Q_1(s,a)\big].$$

The learner then executes the greedy policy with respect to Q^t. As data accumulates, the version space shrinks, and the optimistic values converge to the true optimal values. The regret of GOLF is governed by the *Bellman eluder dimension* of the function class $\mathcal{F}$, which quantifies how quickly Bellman errors on visited state-action pairs constrain Bellman errors on unvisited ones. This complexity measure unifies and generalizes earlier notions like the eluder dimension for bandits [Russo and Van Roy, 2013], and recovers near-optimal rates for linear MDPs and other structured settings as special cases.

These optimistic algorithms and their variants recur throughout this thesis. In Chapter 5, we design optimistic algorithms for settings with latent, partially observed reward states, and in Chapter 6, we develop optimistic methods that exploit latent low-dimensional bandit structure.

In the settings studied in this thesis, the latent variable makes parts of the model unidentifiable: confidence sets over the latent-affected components may not shrink to a singleton even with infinite data. It is therefore crucial to construct confidence sets only for the parts not affected by the latent state, and to handle the latent-affected parts through separate mechanisms — sensitivity bounds, spectral methods, or function approximation calibrated to the latent channel's complexity. This distinction drives the algorithmic design throughout Chapters 3–6.

2.1.3 Visitation Distributions and Mixing

For a policy π interacting with transition kernel $\mathbb{P}$, the induced *state-action visitation distribution* at step h is $d_h^\pi(s,a) = \mathbb{P}^\pi(s_h = s, a_h = a)$. When the transition dynamics are stationary ($\mathbb{P}_h = \mathbb{P}$ for all h), the Markov chain induced by a policy π on $\mathcal{S} \times \mathcal{A}$ may converge to a stationary distribution d^π. Convergence to a unique stationary distribution is guaranteed when the chain is irreducible and aperiodic. More generally, any finite-state Markov chain either converges to a stationary distribution determined by its starting state, or is periodic, cycling through a finite set of distributions; in either case, the mixing analysis applies with respect to the appropriate underlying distribution. The *mixing time* t_{mix} measures how quickly this convergence occurs: it is the smallest t such that $\|d^{(t)} - d^\pi\|_{TV} \leq 1/4$ for all starting states (any constant below $1/2$ yields an equivalent definition up to constant factors). The mixing time does not require an infinite horizon, only that trajectories are long enough for the chain to approximately reach stationarity. This quantity governs the trajectory length needed for concentration arguments in Chapters 3 and 4, where the non-i.i.d. dependence structure within trajectories is handled using the blocking technique of Yu [1994].

2.2 Bandits

The multi-armed bandit problem is a special case of the MDP with a single state ($S = 1$) and horizon $H = 1$: at each round, the learner selects an action (arm) and observes a reward. In the *contextual* setting, a context $x_t \in \mathcal{X}$ is revealed before the learner acts, and rewards depend on both the context and the chosen arm. Regret, optimism, and confidence sets carry over directly from the MDP setting described above. In the non-contextual case, the UCB (Upper Confidence Bound) algorithm [Auer et al., 2002] maintains a confidence interval for the mean reward of each arm and

selects the arm with the highest upper confidence bound. Concretely, at round t, UCB selects

$$a_t = \arg\max_a \ \hat{\mu}_a(t) + \sqrt{\frac{2\log t}{N_a(t)}},$$

where $\hat{\mu}_a(t)$ is the empirical mean reward of arm a and $N_a(t)$ is the number of times arm a has been pulled up to round t. The bonus term $\sqrt{2\log(t)/N_a(t)}$ is an upper confidence bound on the deviation of $\hat{\mu}_a(t)$ from the true mean, derived from Hoeffding's inequality. This achieves regret $O(\sqrt{AT\log T})$ [Auer et al., 2002, Lattimore and Szepesvári, 2020], and the principle of optimism in the face of uncertainty extends naturally to contextual settings where the confidence sets are constructed over a shared parameter space rather than per-arm means.

2.2.1 Linear Structure

The key additional structure we impose in the bandit setting is *linearity*. In a linear contextual bandit, each context-action pair (x, a) is mapped to a feature vector $\phi(x, a) \in \mathbb{R}^{d_A}$, and the expected reward is $\mathbb{E}[r \mid x, a] = \phi(x, a)^\top \boldsymbol{\beta}$ for an unknown parameter $\boldsymbol{\beta} \in \mathbb{R}^{d_A}$ [Abbasi-Yadkori et al., 2011, Li et al., 2010, Lattimore and Szepesvári, 2020]. Linearity makes it possible to aggregate information across different context-action pairs, since every observation provides a linear constraint on the shared parameter $\boldsymbol{\beta}$. This enables the construction of ellipsoidal confidence sets

$$\mathcal{C}_t = \left\{\boldsymbol{\beta} : \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_t\|_{\mathbf{V}_t} \le \alpha_t\right\}$$

using regularized least-squares estimates $\hat{\boldsymbol{\beta}}_t$, where $\mathbf{V}_t = \mu\mathbf{I} + \sum_{s=1}^{t-1} \phi_s \phi_s^\top$ is the regularized design matrix. At each round t, the LinUCB algorithm, also called OFUL (Optimism in the Face of Uncertainty for Linear bandits) [Abbasi-Yadkori et al., 2011], computes the regularized least-squares estimate

$$\hat{\boldsymbol{\beta}}_t = \mathbf{V}_t^{-1} \sum_{s=1}^{t-1} \phi_s r_s,$$

constructs the confidence set $\mathcal{C}_t$, and selects the action by jointly maximizing over arms and plausible parameters:

$$a_t = \arg\max_a \max_{\boldsymbol{\beta}\in\mathcal{C}_t} \phi(x_t, a)^\top \boldsymbol{\beta} = \arg\max_a \ \phi(x_t, a)^\top \hat{\boldsymbol{\beta}}_t + \alpha_t \|\phi(x_t, a)\|_{\mathbf{V}_t^{-1}}.$$

The second equality follows because the maximum of a linear function over an ellipsoid is attained at the boundary in the direction of the feature vector. The confidence radius α_t scales as $O(d_A \log t)$, which follows from the self-normalized martingale bound of Abbasi-Yadkori et al. [2011]. The

resulting algorithm achieves regret $\tilde{O}(d_A\sqrt{T})$ [Abbasi-Yadkori et al., 2011]. In Chapter 6, we study how the effective dimension can be reduced from d_A to a latent dimension $d_K \ll d_A$ by leveraging offline data to learn low-dimensional structure in the reward parameters.

2.3 Offline Reinforcement Learning and Policy Evaluation

What if the agent cannot interact with the environment at all? In offline (batch) RL, the learner is given a fixed dataset of trajectories $\{(s_h^n, a_h^n, r_h^n, s_{h+1}^n)\}$ collected by a behavior policy π_b, and must learn a good policy without further interaction [Levine et al., 2020]. A key subtask is *offline policy evaluation* (OPE): estimating the value $V_1^{\pi_e}$ of a target evaluation policy π_e from offline data collected under π_b. OPE is important both as a standalone problem (e.g., deciding whether to deploy a new treatment policy) and as a subroutine for policy optimization.

The central challenge of the offline setting is *risk*. Unlike in online learning, the algorithm cannot explore to reduce uncertainty, and errors compound across the horizon H through the Bellman recursion. The difficulty of OPE depends on the relationship between π_b and π_e: when π_e visits state-action pairs that π_b rarely or never visits, the data provides little information about the evaluation policy's performance. When the behavior policy is known and "covers" the evaluation policy, importance-sampling and Fitted Q-Evaluation (FQE) methods can provide consistent point estimates [Yin and Wang, 2020, Duan and Wang, 2020a]. There are two principal families of OPE estimators.

2.3.1 Importance Sampling

The most direct approach to OPE is *importance sampling* (IS), which reweights the observed trajectories to correct for the distribution mismatch between π_b and π_e. The trajectory-level IS estimator takes the form

$$\hat{V}_{\mathrm{IS}} = \frac{1}{N}\sum_{n=1}^{N}\prod_{h=1}^{H}\frac{\pi_e(a_h^n \mid s_h^n)}{\pi_b(a_h^n \mid s_h^n)}\sum_{h=1}^{H} r_h^n.$$

When π_b is known, this estimator is unbiased: the importance weight $\prod_{h=1}^{H}\pi_e(a_h^n \mid s_h^n)/\pi_b(a_h^n \mid s_h^n)$ corrects the trajectory distribution from π_b to π_e. However, because the weight is a product of H ratios, its variance grows exponentially in H. This phenomenon is known as the *curse of horizon* [Precup et al., 2000].

To mitigate this, *per-step importance sampling* decomposes the value as a sum over timesteps

and applies only the importance weights accumulated up to each step h:

$$\hat{V}_{\text{step}} = \frac{1}{N} \sum_{n=1}^{N} \sum_{h=1}^{H} \prod_{l=1}^{h} \frac{\pi_e(a_l^n \mid s_l^n)}{\pi_b(a_l^n \mid s_l^n)} \, r_h^n.$$

At step h, the maximum importance weight involves a product of only h ratios rather than H, which can substantially reduce variance when the horizon is long. Per-step IS remains unbiased by the Markov property, since the future importance ratios are conditionally independent of the reward r_h^n given the history up to step h.

A fundamental limitation of all IS methods is that they require knowledge of the behavior policy π_b (or at least an accurate estimate of it) and they suffer from high variance whenever π_e and π_b differ substantially. In particular, if π_e places positive probability on actions where π_b is near zero, the importance ratios become large and unstable. This motivates model-based alternatives that estimate the transition dynamics directly, to which we now turn.

2.3.2 Fitted Q-Evaluation and Fitted Q-Iteration

Fitted Q-Evaluation (FQE) is a model-based approach to OPE that estimates the action-value function of the evaluation policy by iterating empirical Bellman backups [Yin and Wang, 2020, Duan and Wang, 2020a]. Given an estimate $\hat{\mathbb{P}}_h$ of the transition kernel constructed from the offline data, FQE proceeds backward from $h = H$ to $h = 1$, computing

$$\hat{Q}_h(s,a) \leftarrow r_h(s,a) + \sum_{s' \in \mathcal{S}} \hat{\mathbb{P}}_h(s' \mid s,a) \, \hat{V}_{h+1}(s'), \qquad \hat{V}_{h+1}(s) = \sum_{a \in \mathcal{A}} \pi_{e,h+1}(a \mid s) \, \hat{Q}_{h+1}(s,a),$$

with boundary condition $\hat{V}_{H+1}(s) = 0$. The final estimate of the policy value is $\hat{V}_1^{\pi_e} = \mathbb{E}_{s \sim d_0}[\hat{V}_1(s)]$. Unlike IS, FQE does not require knowledge of the behavior policy; it only requires that π_b provides sufficient coverage so that the empirical transition estimates $\hat{\mathbb{P}}_h$ are accurate at the state-action pairs visited by π_e. Moreover, the estimation error of FQE grows only *polynomially* in H (through the H-step Bellman recursion) rather than exponentially, making it substantially more robust for long-horizon problems.

Fitted Q-Iteration (FQI) is the optimization analogue of FQE [Munos and Szepesvári, 2008]. Rather than evaluating a fixed policy π_e, FQI seeks the optimal policy by replacing the averaging step under π_e with a maximization over actions:

$$\hat{Q}_h(s,a) \leftarrow r_h(s,a) + \sum_{s' \in \mathcal{S}} \hat{\mathbb{P}}_h(s' \mid s,a) \, \max_{a' \in \mathcal{A}} \hat{Q}_{h+1}(s',a'),$$

again proceeding backward from $h = H$ to $h = 1$. The output is the greedy policy $\hat{\pi}_h(s) = \arg\max_{a \in \mathcal{A}} \hat{Q}_h(s, a)$. FQI requires stronger coverage conditions than FQE: while FQE only needs the behavior policy to cover the state-action pairs visited by the *evaluation* policy π_e, FQI needs coverage of the state-action pairs visited by the *optimal* policy π^*, which is unknown a priori. This makes offline policy optimization fundamentally harder than offline policy evaluation, and the gap between the two is a recurring theme in the offline RL literature.

In many safety-critical domains, point estimates are not enough. A hospital deciding whether to deploy a new treatment policy may prefer a *conservative lower bound* on the policy's value over a point estimate that could be optimistically wrong. This motivates algorithms that produce guaranteed lower (or upper) bounds on $V_1^{\pi_e}$ rather than point estimates. Such conservative approaches become especially important when the data-generating process involves hidden confounders, i.e., latent variables that affect both transitions and the behavior policy, since confounding can make consistent point estimation fundamentally impossible. In Chapter 4, we study this landscape systematically, showing when point estimation is and is not possible, and providing algorithms for both consistent estimation and conservative bound estimation depending on the structure of confounding.

2.4 Reinforcement Learning from Human Feedback

What if the reward must itself be learned from human judgments? Reinforcement learning from human feedback (RLHF) is concerned with learning a policy that maximizes an objective defined through human evaluations rather than a pre-specified reward function [Wirth et al., 2017, Christiano et al., 2017]. A distinguishing feature of RLHF is that feedback is provided at the *trajectory level*. Rather than observing a reward after each action, the learner receives a single human evaluation after an entire episode of interaction.

There are two dominant kinds of trajectory-level feedback. In *cardinal* feedback, a human provides a rating or score for a trajectory. In *dueling* feedback, a human specifies a preference between two trajectories. Much of the theoretical literature models dueling feedback using the Bradley-Terry model [Chatterji et al., 2021, Saha et al., 2023]: given two trajectories τ_1, τ_2 with underlying rewards $r(\tau_1), r(\tau_2)$, the probability that the human prefers τ_1 is $\sigma(r(\tau_1) - r(\tau_2))$ for a link function σ (typically the logistic function). In the dominant practical paradigm, a reward model is first learned from human feedback, and then a standard RL algorithm is used to optimize this learned reward [Ouyang et al., 2022]. This approach has been pivotal in the development of aligned large language models [Rafailov et al., 2023].

2.4.1 Current Approaches

The dominant practical approach to RLHF in the language model setting follows a three-stage pipeline [Ouyang et al., 2022, Christiano et al., 2017, Ziegler et al., 2019].

Stage 1: Supervised fine-tuning. A pretrained language model is first fine-tuned on a dataset of high-quality demonstration data using standard maximum likelihood. The resulting model π_{ref} serves both as the starting point for further optimization and as the reference policy against which subsequent updates are regularized.

Stage 2: Reward modeling. Given a dataset of human preference comparisons, consisting of pairs of responses (y_w, y_l) to a prompt x where y_w is preferred over y_l, a reward model r_ψ is trained by maximizing the Bradley-Terry likelihood:

$$\mathcal{L}(\psi) = -\sum_{(y_w, y_l)} \log \sigma\big(r_\psi(y_w) - r_\psi(y_l)\big).$$

The reward model is typically initialized from the SFT model with a scalar-valued head replacing the language modeling head.

Stage 3: RL fine-tuning. The policy is then optimized against the learned reward model using Proximal Policy Optimization (PPO) [Schulman et al., 2017], subject to a KL divergence penalty that prevents the policy from straying too far from the reference model π_{ref}:

$$\max_{\pi} \; \mathbb{E}_{x\sim\mathcal{D}}\big[\mathbb{E}_{y\sim\pi(\cdot|x)}\big[r_\psi(y)\big] - \beta \; \mathrm{KL}\big(\pi(\cdot \mid x) \,\|\, \pi_{ref}(\cdot \mid x)\big)\big] \,.$$

The KL penalty, weighted by $\beta > 0$, serves two purposes: it mitigates reward hacking (the phenomenon where the policy exploits inaccuracies in the learned reward model) and it preserves the fluency and coherence acquired during pretraining.

Direct Preference Optimization. Rafailov et al. [2023] observed that the three-stage pipeline can be simplified by eliminating the reward model entirely. The key insight is that the optimal policy for the KL-regularized objective above satisfies

$$\pi^*(y \mid x) \propto \pi_{ref}(y \mid x) \exp\big(r(y, x)/\beta\big),$$

which can be rearranged to express the reward as a function of the policy:

$$r(y, x) = \beta \log \frac{\pi(y \mid x)}{\pi_{ref}(y \mid x)} + \beta \log Z(x),$$

where $Z(x)$ is the partition function. Substituting this reparameterization into the Bradley-Terry preference model yields a loss that depends only on the policy, not a separate reward model:

$$\mathcal{L}_{\text{DPO}}(\pi) = -\mathbb{E}_{(y_w, y_l)} \left[\log \sigma \left(\beta \log \frac{\pi(y_w \mid x)}{\pi_{ref}(y_w \mid x)} - \beta \log \frac{\pi(y_l \mid x)}{\pi_{ref}(y_l \mid x)} \right) \right].$$

Direct Preference Optimization (DPO) thus trains the policy directly on preference data using a single supervised learning objective, avoiding the instabilities of PPO and the cost of training a separate reward model. More recently, iterative and online variants of DPO that collect fresh preference data during training have been developed, further improving upon the offline single-pass approach [Xiong et al., 2024].

However, the Bradley-Terry model and its variants assume that human feedback depends on the trajectory only through a fixed, known reward function. In reality, human evaluators have partially-observed, evolving internal states (such as sentiment, fatigue, or engagement) that can affect their feedback in much more arbitrary ways. It is more reasonable to assume that human feedback is generally *non-Markovian*, in that a human's reaction at step h may depend on the entire history of states and actions, not just the current state. While one can handle this by treating the full trajectory as the state, this naive approach leads to state spaces of size $(SA)^{\Omega(H)}$ and correspondingly poor guarantees. In Chapter 5, we introduce PORMDPs, a framework that explicitly models internal states and intermediate feedback, and design algorithms whose guarantees scale polynomially rather than exponentially in the relevant dimensions.

2.5 Structured Partial Observability

As noted in the introduction, learning in fully general POMDPs is statistically intractable [Krishnamurthy et al., 2016, Jin et al., 2020]. This thesis makes progress by studying structured sub-cases of partial observability, distinguished by *what* the latent variable affects and *how* it evolves over time. We briefly describe this taxonomy here; precise definitions are given in each chapter.

2.5.1 Partially Observable MDPs

A *partially observable Markov decision process* (POMDP) generalizes the MDP by introducing an observation space and restricting the agent's access to the underlying state [Kaelbling et al., 1998a].

Formally, a POMDP is specified by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, H, \mathbb{P}, r, \mathcal{Z}, d_0)$, where $\mathcal{S}$, $\mathcal{A}$, H, $\mathbb{P}$, r, and d_0 are as in the MDP definition, $\mathcal{O}$ is a finite observation space, and $\mathcal{Z}_h(o \mid s)$ is the observation emission probability at step h. At each step h, the environment transitions to state s_h according to the transition kernel $\mathbb{P}_h$, but the agent does not observe s_h directly. Instead, the agent observes $o_h \sim \mathcal{Z}_h(\cdot \mid s_h)$. Because the state is hidden, a policy in a POMDP must map observation–action histories to distributions over actions:

$$\pi_h : (\mathcal{O} \times \mathcal{A})^{h-1} \times \mathcal{O} \to \Delta(\mathcal{A}).$$

Learning and planning in POMDPs are statistically and computationally challenging. Even when the model is fully known, planning requires maintaining a *belief state* $b_h \in \Delta(\mathcal{S})$, the posterior distribution over the hidden state given the observation history, updated via Bayes' rule:

$$b_{h+1}(s') \propto \mathcal{Z}_{h+1}(o_{h+1} \mid s') \sum_{s \in \mathcal{S}} \mathbb{P}_h(s' \mid s, a_h)\, b_h(s).$$

The belief state is a sufficient statistic for optimal decision-making, but it lives in a continuous $(S-1)$-dimensional simplex, and the effective state space grows exponentially with the horizon. When the model is unknown, the sample complexity of learning a near-optimal policy can be exponential in the horizon H even for POMDPs with small state and observation spaces [Krishnamurthy et al., 2016, Jin et al., 2020]. This fundamental hardness motivates the study of structured sub-cases of partial observability, which is the focus of this section and of this thesis more broadly.

2.5.2 PORMDPs

A *Partially Observed Reward MDP* (PORMDP) is an MDP augmented with an unobserved internal state $u_h \in \mathcal{U}$ that affects only the reward function. The transitions of the observable state remain Markovian: $s_{h+1} \sim \mathbb{P}_h(\cdot \mid s_h, a_h)$, and the agent observes s_h directly at each step. However, the reward depends on the internal state:

$$r_h = r_h(s_h, a_h, u_h),$$

where the internal state evolves according to its own dynamics,

$$u_{h+1} \sim \mathbb{P}_h^u(\cdot \mid s_h, a_h, u_h).$$

Since u_h is unobserved, the effective reward at step h depends on the entire trajectory history $\tau[h] = (s_1, a_1, \ldots, s_h, a_h)$, rendering the reward function non-Markovian from the agent's perspective.

A PORMDP is a structured POMDP in which partial observability is confined to the reward channel: the agent has full access to the state relevant for transitions, but must contend with a hidden component that influences only the reward. The formal definition and algorithms for PORMDPs are developed in Chapter 5.

2.5.3 Latent Time Series and Mixture Models

Several forms of partial observability studied in this thesis can be understood through the lens of latent variable models for sequential data. We describe three progressively richer models that recur across the chapters.

Hidden Markov Models (HMMs). A *Hidden Markov Model* is a classical latent time series model in which a discrete latent state $z_h \in [K]$ evolves as a Markov chain with transition matrix $\mathbf{T}_h$, and at each step an observation o_h is emitted conditionally on the latent state according to an emission distribution $\mathcal{Z}_h(\cdot \mid z_h)$. Formally, the generative process is

$$z_1 \sim w, \qquad z_{h+1} \sim \mathbf{T}_h(\cdot \mid z_h), \qquad o_h \sim \mathcal{Z}_h(\cdot \mid z_h),$$

where $w \in \Delta_K$ is the initial distribution over latent states. HMMs are a special case of POMDPs without actions. The latent state sequence can be recovered (or marginalized over) via the Baum–Welch algorithm (an instance of EM) or via spectral methods that exploit the low-rank structure of observable moment matrices [Hsu et al., 2012, Anandkumar et al., 2014].

Mixtures of Markov chains. A *mixture of Markov chains* consists of K Markov chains with distinct transition kernels $\mathbb{P}^{(1)}, \ldots, \mathbb{P}^{(K)}$ over a shared state space $\mathcal{S}$. Each trajectory is generated by first drawing a component label $k \sim w$ from a mixing distribution $w \in \Delta_K$ and then running the Markov chain with transition kernel $\mathbb{P}^{(k)}$ for the duration of the episode. The component label is unobserved and remains fixed throughout the trajectory. This model is a special case of an HMM in which the latent state does not change within a trajectory (i.e., $\mathbf{T}_h = \mathbf{I}$ for all h). Equivalently, it is an instance of global confounding: the unobserved component label acts as a confounder that is sampled once and persists for the entire episode. Mixtures of Markov chains are studied in Chapter 3.

Mixtures of MDPs. A *mixture of MDPs* extends the mixture of Markov chains by incorporating actions. The model consists of K MDPs sharing a common state space $\mathcal{S}$ and action space $\mathcal{A}$, but with distinct transition kernels $\mathbb{P}^{(1)}, \ldots, \mathbb{P}^{(K)}$. Each trajectory is generated by drawing a component label $k \sim w$ and then interacting with the MDP $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}^{(k)}, r, d_0)$ for the duration of the episode.

The component label is again unobserved, making this a controlled extension of the mixture of Markov chains. This model arises naturally as the global confounder model for offline policy evaluation: the unobserved component determines which MDP generated each trajectory in the offline dataset. Mixtures of MDPs are studied in Chapters 3 and 4.

2.5.4 Taxonomy of Structured Partial Observability

What forms of structured partial observability does this thesis study? The settings are distinguished by what the latent variable affects; precise definitions are given in each chapter.

Confounding: latent variables that affect transitions. An unobserved *confounder* is a latent variable that influences the transition dynamics and possibly the behavior policy, but not the reward function [Kallus and Zhou, 2020, Bruns-Smith, 2021]. This is a natural model when rewards are designed by the user based on observed quantities (e.g. patient vitals), but unrecorded factors (e.g. genetic conditions) affect both the clinician's decisions and the patient's state evolution. Within confounding, the *memory structure* of the confounder plays a crucial role:

- **Memoryless confounders** are sampled afresh at each timestep, independently of past confounders, states, and actions. An example is an accident or supply shock encountered during treatment.
- **Global confounders** are sampled once at the beginning of each trajectory and remain fixed throughout the episode. An example is an unrecorded patient demographic or genetic condition.

As we show in Chapter 4, this distinction has sharp consequences for what is estimable: consistent offline policy evaluation is impossible under memoryless confounding (even with sensitivity constraints), but becomes possible under global confounding by leveraging the mixture-of-MDPs structure developed in Chapter 3.

Partially observed reward states: latent variables that affect rewards. Complementary to confounding, a latent variable may affect the *reward function* without influencing transitions. This arises naturally in RLHF, where the human evaluator's internal state (sentiment, fatigue, engagement) shapes their feedback but does not change the environment dynamics. Unlike confounders, partially observed reward states induce *non-Markovian* reward functions, since the internal state at step h may depend on the entire trajectory history. This setting is studied in Chapter 5.

Latent mixture structure. As described in the preceding subsection, latent mixture models (including mixtures of Markov chains, mixtures of MDPs, and latent bandits) provide a cross-cutting form of partial observability in which latent heterogeneity induces low-dimensional structure

recoverable via spectral methods. This structure is exploited in Chapters 3, 4, and 6.

2.6 Notation

We collect notation that is used across multiple chapters. Chapter-specific notation is introduced as needed.

Symbol	Meaning
$\mathcal{S}, S$	State space and its cardinality
$\mathcal{A}, A$	Action space and its cardinality
H	Horizon length
K	Number of mixture components or latent states
T	Number of episodes, trajectories, or online rounds
N	Number of offline trajectories (Chapter 6)
π, π_b, π_e	Policy, behavior policy, evaluation policy
$\mathbb{P}_h(s' \mid s, a)$	Transition probability at step h
$r_h(s, a)$	Reward function at step h
$V_h^\pi(s)$	Value function of π at step h and state s
$Q_h^\pi(s, a)$	Action-value function of π at step h, state s, action a
t_{mix}	Mixing time of the induced Markov chain
Γ	Confounding sensitivity parameter (Chapter 4)
d_A, d_K	Ambient and latent dimensions (Chapter 6)
$\mathbb{P}, \mathbb{E}$	Probability and expectation
$\mathbb{1}\{\cdot\}$	Indicator function
$\|\cdot\|_1, \|\cdot\|_2$	ℓ_1 and ℓ_2 norms
$\tilde{O}(\cdot)$	$O(\cdot)$ hiding logarithmic factors

CHAPTER 3

Learning Mixtures of Markov Chains and MDPs

When working with offline data from heterogeneous populations, a fundamental challenge is that trajectories may come from different underlying models whose identities are unknown. In the running example of Section 1.2.2, medical interns fall into latent profile types with distinct dynamics, but the type is never labeled. More broadly, this problem arises naturally in several settings studied later in this thesis: when an unobserved confounder determines which MDP generates each trajectory (Chapter 4), or when latent user types determine reward distributions (Chapter 6). In the language of Section 1.4, the latent variable here is confined to the transition kernel and acts through a K-dimensional subspace of the probability simplex, making it possible to recover the mixture structure from population data without ever observing which model generated any given trajectory. In this chapter, we develop the spectral and clustering methods for learning such mixture structure from unlabeled trajectories, providing finite-sample guarantees. The methods developed here will be directly applied in Chapter 4 to handle the global confounder setting, and will inspire the offline spectral algorithm SOLD in Chapter 6.

3.1 Introduction

This chapter is a lightly edited version of Kausik et al. [2023].

Efficiently clustering a mixture of time series data, especially with access to only short trajectories, is a problem that pervades sequential decision making and prediction (Liao [2005], Huang et al. [2021], Maharaj [2000]). This is motivated by various real-world problems, ranging through psychology (Bulteel et al. [2016]), economics (McCulloch and Tsay [1994]), automobile sensing (Hallac et al. [2017]), biology (Wong and Li [2000]), neuroscience (Albert [1991]), to name a few. One natural and important time series model is that of a mixture of K MDPs, which includes the case of a mixture of K Markov chains. We want to cluster from a set of short trajectories where (1) one does not know which MDP or Markov chain any trajectory comes from and (2) one does not know the transition structures of any of the K MDPs or Markov chains. Previous literature like

Kwon et al. [2021] and Gupta et al. [2016] has stated and underlined the importance of this problem, but so far, the literature on methods to solve it with theoretical guarantees and empirical results has been sparse.

Broadly, there are three threads of literature on problems related to ours. Within reinforcement learning literature, there has been a sustained interest in frameworks very similar to mixtures of MDPs – latent MDPs (Kwon et al. [2021]), multi-task RL (Brunskill and Li [2013]), hidden model MDPs (Chades et al. [2021]), to name a few. However, most effort in this thread has been towards regret minimization in the online setting, where the agent interacts with an MDP from a set of unknown MDPs. The framework of latent MDPs in Kwon et al. [2021] is equivalent to adding reward information to ours. They have shown that one can only learn latent MDPs online with number of episodes required polynomial in states and actions to the power of trajectory length (under a reachability assumption similar to our mixing time assumption). On the other hand, our method learns latent MDPs offline with number of episodes needed only linear in the number of states (in no small part due to the subspace estimation step we make). In the meta-RL literature, Zintgraf et al. [2021] use variational inference to identify latent task parameters online and meta-learn a task-conditioned policy. While their setting is online and uses very different methods (VAEs rather than spectral methods), the underlying problem of learning and adapting to latent task structure is shared with our offline clustering approach.

The other thread of literature deals with using a "subspace estimation" idea to efficiently cluster mixture models, from which we gain inspiration for our algorithm. Vempala and Wang [2004] first introduce the idea of using subspace estimation and clustering steps, with application to learning mixtures of Gaussians. Kong et al. [2020] adapt these ideas to the setting of meta-learning for mixed linear regression, adding a classification step. Chen and Poor [2022] bring these ideas to the time-series setting to learn mixtures of linear dynamical systems. They leave open the problems of (1) adapting the method to handle control inputs (mentioning mixtures of MDPs as an important example) and (2) handling other time series models (like autoregressive models and Markov chains), and state that the former is of great importance. There are many technical and algorithmic subtleties in adapting the ideas developed so far to MDPs and Markov Chains. The most obvious one comes from the following observation: in linear dynamical systems, the deviation from the predicted next-state value under the linear model occurs with additive i.i.d. noise. In MDPs and Markov chains, we are *sampling* from the next-state probability simplex at each timestep, and this cannot be cast as a deterministic function of the current state with additive i.i.d. noise.

Gupta et al. [2016] also provide a method for learning a mixture of Markov chains using only 3-trails, and compare its performance to the EM algorithm. While the requirement on trajectory length is as lax as can be, their method needs to estimate the distribution of 3-trails using all available

data, incurring an estimation error in estimating S^3A^3 parameters, while providing no finite-sample theoretical guarantees. If the method can be shown to enjoy finite sample guarantees, the need to estimate S^3A^3 parameters indicates that the guarantees will scale poorly with S and A.

The problem that we aim to solve is the following.

Is there a method with finite-sample guarantees that can learn both mixtures of Markov chains and MDPs offline, with only data on trajectories and the number of elements in the mixture K?

3.1.1 Summary of Contributions

We provide such a method, with trajectory length requirements free from an S, A dependence. The method performs (1) subspace estimation, (2) spectral clustering, an optional step of using clusters to initialize the EM algorithm, (3) estimating models, and finally (4) classifying future trajectories.
Theorem (Informal)**.** *Ignoring logarithmic terms, we can recover all labels exactly with K^2S trajectories of length $K^{3/2}t_{mix}$, up to logarithmic terms and instance-dependent constants characterizing the models but not explicitly dependent on S, A, t_{mix} or K.*

Other contributions include:

- This is the first method, to our knowledge, that can cluster MDPs with finite-sample guarantees where the length of trajectories does not depend explicitly on S, A. The length only explicitly depends linearly on the mixing time t_{mix}, and the number of trajectories only explicitly depends linearly on S.
- We are able to provide theoretical guarantees while making no explicit demands on the policies and rewards used to collect the data, only relying on a difference in the transition structures at frequently occurring (s, a) pairs.
- Chen and Poor [2022] work under deterministic transitions with i.i.d. additive Gaussian noise, and we need to bring in non-trivial tools to analyse systems like ours, determined by transitions with non-i.i.d. additive noise. Our use of the blocking technique of Yu [1994] opens the door for the analysis of such systems.
- Empirical results in our experiments show that our method outperforms the EM algorithm by a significant margin (73.2% for soft EM and 96.6% for us on gridworld).

3.2 Background and Problem Setup

We work in the scenario where we have K unknown models, either K Markov chains or K MDPs, and data of N_{traj} trajectories collected offline. Throughout the rest of the paper, we work with the case of MDPs, as we can think of Markov chains as an MDP where there is only one action ($A = \{*\}$) and rewards are ignored by our algorithm anyway.

We have a tuple $(\mathcal{S}, \mathcal{A}, \{\mathbb{P}_k\}_{k=1}^K, \{f_k\}_{k=1}^K, p_k)$ describing our mixture. Here, $\mathcal{S}, \mathcal{A}$ are the state and action sets respectively. $\mathbb{P}_k(s' \mid s, a)$ describes the probability of an s, a, s' transition under label k. At the start of each trajectory, we draw $k \sim$ Categorical$(f_1, ..., f_K)$, and starting state according to p_k, and generate the rest of the trajectory under policies $\pi_k(a \mid s)$. We have stationary distributions on the state-action pairs $d_k(s, a)$ for π_k interacting with $\mathbb{P}_k$. We do not know (1) the parameters $\mathbb{P}_k, f_k, p_k, \pi_k(\cdot \mid s)$ of each model or the policies, and (2) k, i.e., which model each trajectory comes from. We write $f_{\min} := \min_k f_k$ for the smallest mixture weight. Our guarantees depend on $f_{\min}$; components with small mixture weight contribute less to the population data and require more trajectories to recover. We assume K is known; in practice, K can be estimated from the eigenvalue spectrum of the double estimator (see Section 3.5.1).

This coincides with the setup in Gupta et al. [2016] in the case of Markov chains ($|\mathcal{A}| = 1$). It also overlaps with the setup of learning latent MDPs offline, in the case of MDPs. However, one difference is that we make no assumptions about the reward structure – once trajectories are clustered, we can learn the models, including the rewards. It is also possible to learn the rewards with a term in the distance measure that is alike to the model separation term. However, this would require extra assumptions on reward separation that are not necessary for clustering.

Assumption 1 (Mixing)**.** The K Markov chains on $\mathcal{S} \times \mathcal{A}$ induced by the behaviour policies π_k, each achieve mixing to a stationary distribution $d_k(s, a)$ with mixing time $t_{mix,k}$. Define the overall mixing time of the mixture of MDPs to be $t_{mix} := \max_k t_{mix,k}$.

Assumption 2 (Model Separation)**.** There exist α, Δ so that for each pair k_1, k_2 of hidden labels, there exists a state action pair (s, a) (possibly depending on k_1, k_2) so that $d_{k_1}(s, a), d_{k_2}(s, a) \geq \alpha$ and $\|\mathbb{P}_{k_1}(\cdot \mid s, a) - \mathbb{P}_{k_2}(\cdot \mid s, a)\|_2 \geq \Delta$.

Assumption 2 is merely saying that for any pair of labels, at least one visible state action pair witnesses a model difference Δ. Call this the separating state-action pair. If no visible pair witnesses a model difference between the labels, then one certainly cannot hope to distinguish them using trajectories.

Remark 1. Why is there no assumption about policies? Notice that we make no explicit assumptions about policies. The nature of our algorithm allows us to work with the transition structure directly, and so we only demand that we observe a state action pair that witnesses a

difference in transition structures. The policy is implicitly involved in this assumption through the stationary distribution $d_k(s, a)$ it induces, but our results demonstrate that this is the minimal demand we need to make in relation to the policies.

Additionally, Assumption 1, which establishes the existence of a mixing time, is not a strong assumption (outside of the implicit hope that t_{mix} is small). This is because any irreducible aperiodic finite state space Markov chain mixes to a unique stationary distribution. If the Markov chain is not irreducible, it mixes to a unique distribution determined by the irreducible component of the starting distribution.

The only requirement is thus aperiodicity, which is also technically superficial, as we now clarify. If the induced Markov chains were periodic with period L, we would have a finite set of stationary distributions $d_{k,l}(s, a)$ that the chain would cycle through over a single period, indexed by $l = 1 \to L$. One can follow the proofs to verify that the guarantees continue to hold if we modify α in Assumption 2 to be a lower bound for $\min_{i,l} d_{k_i,l}(s, a)$ instead of just $\min_i d_{k_i}(s, a)$.

3.3 Algorithm

3.3.1 Setup and Notation

We have short trajectories of length T_n, divided into 4 segments of equal length. We call the second and fourth segment Ω_1 and Ω_2 respectively. We further sub-divide Ω_i into G blocks, and focus only on the first state-action observation in each sub-block and its transition (discard all other observations). We often refer to these observations as "single-step sub-blocks." See Figure 3.1 for an illustration of this. Divide the set of trajectory indices into two sets and call them $\mathcal{N}_{sub}$ and $\mathcal{N}_{clust}$ (for subspace estimation and clustering). Denote their sizes by N_{sub} and N_{clust} respectively. Let $\mathcal{N}_{traj}(s, a)$ be the set of trajectory indices where (s, a) is observed in both Ω_1 and Ω_2. Let $N_{traj}(s, a)$ be the size of this set. Denote by $N(n, i, s, a)$ the number of times (s, a) is recorded in segment i of trajectory n, and let $\mathbf{N}(n, i, s, a, \cdot)$ be the vector of next-state counts. We denote by $\mathbb{P}_k(\cdot \mid s, a)$ the vector of next state transition probabilities. We denote by Freq_β the set of all state action pairs whose occurrence frequency in our observations is higher than β.

We will call the predicted clusters returned by the clustering algorithm $\mathcal{C}_k$. For model estimation and classification, we do not use segments, and merely split the entire trajectory into G blocks, discarding all but the last observation in each block. We call this observation the corresponding single-step sub-block. The choice of G is pinned down by the theoretical analysis (see Theorem 3.4.1); in practice, we use the full segment without sub-sampling into blocks. We denote the total count of s, a observations in trajectory n by $N(n, s, a)$ and that of s', s, a triples by

$N(n, s, a, s')$.

In practice, we choose to not be wasteful and observations are not discarded while computing the transition probability estimates. To clarify, in that case $N(n, i, s, a)$ is just the count of (s, a) in segment i and similarly for $\mathbf{N}(n, i, s, a, \cdot)$, $N(n, s, a)$ and $\mathbf{N}(n, s, a, \cdot)$. Estimators in both cases, that is both with and without discarding observations, are MLE estimates of the transition probabilities. One of them maximizes the likelihood of just the single-step sub-blocks and the other maximizes the likelihood of the entire segment. We need the latter for good finite-sample guarantees (using mixing). However, the former satisfies asymptotic normality, which is not enough for finite-sample guarantees, but it often makes it a good and less wasteful estimator in practice.

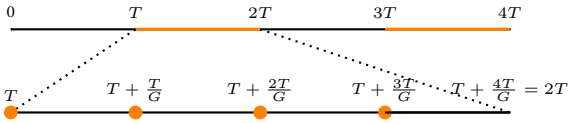

Figure 3.1: Breaking up a trajectory into 4 segments and G blocks per segment ($G = 4$) for the single-step estimator. Observations are only recorded at the orange points.

3.3.2 Overview

The algorithm amounts to (1) a PCA-like subspace estimation step, (2) spectral clustering of trajectories using "thresholded pairwise distance estimates," along with an optional step of using clusters to initialize the EM algorithm, (3) estimating models (MDP transition probailities) and finally (4) classifying any trajectories not in $\mathcal{N}_{clust}$ (for example, $\mathcal{N}_{sub}$). We provide performance guarantees for each step of the algorithm in section 3.4.

3.3.3 Subspace Estimation

The aim of this algorithm is to estimate for each (s, a) pair a matrix $\mathbf{V}_{s,a}$ satisfying $\text{rowspan}\, \mathbf{V}_{s,a}^T \approx \text{span}(\mathbb{P}_k(\cdot|s, a))_{k=1,..,K}$. That is, we want to obtain an orthogonal projector to the subspace spanned by the next-state distributions $\mathbb{P}_k(\cdot|s, a)$ for $1 \leq k \leq K$.

Summarizing the algorithm in natural language, we perform subspace estimation via 3 steps. We first estimate the next state distribution given state and action for each trajectory. We then obtain the outer product of the next state distributions thus estimated. These outer product matrices are averaged over trajectories, and the average is used to find the orthogonal projectors $V_{s,a}^T$ to the top K eigenvectors.

Remark 2. Why do we split the trajectories? We use two approximately independent segments Ω_1 and Ω_2 time separated by a multiple of the mixing time t_{mix} to estimate the next state distributions.

Algorithm 1 Subspace Estimation

Compute $N_{traj}(s,a)$ for all s,a. Initialize the $S \times S$ matrix $\hat{\mathbf{M}}_{s,a} \leftarrow 0$ and the $SA \times SA$ matrix $\hat{\mathbf{D}} \leftarrow 0$.
$\hat{\mathbf{d}}_{n,1}, \hat{\mathbf{d}}_{n,2} \leftarrow \mathbf{0} \in \mathbb{R}^{SA}$ for all $n \in \mathcal{N}_{sub}$
for $(i,s,a) \in \{1,2\} \times S \times A$ **do**
    Compute $\mathbf{N}(n,i,s,a,\cdot)$, $N(n,i,s,a), \quad \forall n \in \mathcal{N}_{sub}$
    $\hat{\mathbb{P}}_{n,i}(\cdot|s,a) \leftarrow \frac{\mathbf{N}(n,i,s,a,\cdot)}{N(n,i,s,a)} \mathbb{1}_{N(n,i,s,a)\neq 0}, \quad \forall n$
    $[\hat{\mathbf{d}}_{n,i}]_{s,a} \leftarrow \frac{N(n,i,s,a)}{G}, \quad \forall n$
    $\hat{\mathbf{M}}_{s,a} \leftarrow \hat{\mathbf{M}}_{s,a} + \sum_{n \in \mathcal{N}_{sub}} \frac{\hat{\mathbb{P}}_{n,1}(\cdot|s,a)\hat{\mathbb{P}}_{n,2}(\cdot|s,a)^T}{N_{traj}(s,a)}$
end for
$\hat{\mathbf{D}} \leftarrow \hat{\mathbf{D}} + \sum_{n \in \mathcal{N}_{sub}} \frac{1}{N_{sub}} \hat{\mathbf{d}}_{n,1} \hat{\mathbf{d}}_{n,2}^T$
Using SVD, return the orthogonal projectors $(\mathbf{V}_{s,a}^T)_{K \times S}$ to the top K eigenspaces of $\hat{\mathbf{M}}_{s,a} + \hat{\mathbf{M}}_{s,a}^T$ for each (s,a) where $N_{traj}(s,a) \neq 0$ (set the others to 0), along with the orthogonal projector $(\mathbf{U}^T)_{K \times SA}$ to the top K eigenspace of $\hat{\mathbf{D}} + \hat{\mathbf{D}}^T$.

The reduced correlation between the two estimates obtained allows us to give theoretical guarantees for concentration, despite using dependent data within each trajectory n in the estimation of the rank 1 matrices $(\mathbb{P}_{k_n}(\cdot|s,a))(\mathbb{P}_{k_n}(\cdot|s,a))^T$. The key point is that the double estimator $\hat{\mathbb{P}}_{n,1}(\cdot \mid s,a)\hat{\mathbb{P}}_{n,2}(\cdot \mid s,a)$ is in expectation very close to this matrix.

Notice that our estimator $\hat{\mathbf{M}}_{s,a}$ is in expectation then given approximately by $\sum_{k=1}^{K} f_k(\mathbb{P}_k(\cdot|s,a))(\mathbb{P}_k(\cdot|s,a))^T$. The eigenspace of this matrix is clearly $\mathrm{span}(\mathbb{P}_k(\cdot|s,a))_{k=1,..,K}$. The deviation from the expectation is controlled by the total number of trajectories, while the "approximation error" separating the expectation from the desired matrix is controlled by the separation between Ω_1 and Ω_2.

Assumption 2 ensures that for each pair of labels, at least one (s,a) pair witnesses a difference in the transition kernels. At such a separating pair, the corresponding $\mathbb{P}_k$ vectors are distinct, contributing to the rank of $\hat{\mathbf{M}}_{s,a}$. Across all frequently-occurring (s,a) pairs, the combined information ensures that the top K eigenspace of the global estimator $\hat{\mathbf{D}}$ recovers the correct subspace.

Remark 3. Why is this not PCA? This procedure has many linear-algebraic similarities to uncentered PCA on the dataset of (trajectories, next state frequencies), but statistically has a very different target. Crucially, (centered) PCA is concerned with the variance $\mathbb{E}[X^T X]$, while we are interested in a decent estimate of the target $\mathbb{E}[X^T]\mathbb{E}[X]$ above and thus use a double estimator. Our theoretical analysis also has nothing to do with analyses of PCA due to this difference in the statistical target.

3.3.4 Clustering

Using the subspace estimation algorithm's output, we can embed estimates from trajectories in a low dimensional subspace. For the clustering algorithm, we aim to compute the pairwise distances of these estimates from trajectories in this embedding. A double estimator is used yet again, to reduce the covariance between the two terms in the inner product used to compute such a distance.

This projection is crucial because it reduces the variance of the pairwise distance estimators from a dependence on SA to a dependence on K. This is the intuition for how we can shift the onus of good clustering from being heavily dependent on the length of trajectories to being more dependent on the subspace estimate and thus on the number of trajectories.

There are many ways to use such "pairwise distance estimates" for clustering trajectories. In one successful example, we use a test: if the squared distances are below some threshold (details provided later), then we can conclude that they come from the same element of the mixture, and different ones otherwise. This allows us to construct (the adjacency matrix of) a graph with vertices as trajectories, and we can feed the results into a clustering algorithm like spectral clustering. Alternatively, one can use other graph partitioning methods or agglomerative methods on the distance estimates themselves.

Algorithm 2 Clustering

Compute the set Freq_β by picking (s, a) pairs with occurrence more than β.
$\mathbf{d}_{n,1}, \mathbf{d}_{n,2} \leftarrow \mathbf{0} \in \mathbb{R}^{SA}$
for $(i, s, a) \in \{1, 2\} \times S \times A$ **do**
  Compute $\mathbf{N}(n, i, s, a, \cdot)$, $N(n, i, s, a), \quad \forall n \in \mathcal{N}_{clust}$
  $\hat{\mathbb{P}}_{n,i}(\cdot|s, a) \leftarrow \frac{\mathbf{N}(n,i,s,a,\cdot)}{N(n,i,s,a)} \mathbb{1}_{N(n,i,s,a)\neq 0}, \quad \forall n$
  $[\hat{\mathbf{d}}_{n,i}]_{s,a} \leftarrow \frac{N(n,i,s,a)}{G}, \quad \forall n$
end for
for $(n, m) \in \mathcal{N}_{clust} \times \mathcal{N}_{clust}$ **do**
  for $(i, s, a) \in \{1, 2\} \times S \times A$ **do**
    $\hat{\mathbf{\Delta}}_{i,s,a} := \mathbf{V}_{s,a}^T(\hat{\mathbb{P}}_{n,i}(\cdot \mid s, a) - \hat{\mathbb{P}}_{m,i}(\cdot|s, a))$
  end for
  $\mathrm{dist}_1(n, m) := \max_{(s,a)\in \mathrm{Freq}_\beta} \hat{\mathbf{\Delta}}_{1,s,a}^T \hat{\mathbf{\Delta}}_{2,s,a}$
  $\mathrm{dist}_2(n, m) := (\hat{\mathbf{d}}_{n,1} - \hat{\mathbf{d}}_{m,1})^T \mathbf{U}\mathbf{U}^T(\hat{\mathbf{d}}_{n,2} - \hat{\mathbf{d}}_{m,2})$
  $\mathrm{dist}(n, m) := \lambda \,\mathrm{dist}_1(n, m) + (1 - \lambda)\,\mathrm{dist}_2(n, m)$
end for
Plot a histogram of dist to determine threshold τ and cluster trajectories $\mathrm{sim}(n, m) := \mathbb{1}_{\mathrm{dist}(n,m)\leq\tau}$

Choosing β, λ and the threshold τ both involve heuristic choices, much like how choosing the threshold in Chen and Poor [2022] needs heuristics, although our methods are very different. We

describe our methods in more detail in Section 3.5.

3.3.4.1 Refinement using EM

Our guarantees in section 3.4 will show that we can recover exact clusters with high probability at the end of algorithm 2. However, in practice, it makes sense to refine the clusters if trajectories are not long enough for exact clustering. Remember that an instance of the EM algorithm for any model is specified by choosing the observations Y, the hidden variables Z and the parameters θ.

If we consider observations to be next-state transitions from $(s, a) \in \text{Freq}_\beta$, hidden variables to be the hidden labels and the parameters θ to include both next-state transition probabilities for $(s, a) \in \text{Freq}_\beta$ and cluster weights $\hat{f}_k$, then one can now refine the clusters using the EM algorithm on this setup, which enjoys monotonicity guarantees in log-likelihood if one uses soft EM. We describe the E and M steps for hard EM below; further details are in Appendix A.3.

The M-step estimates models and weights via MLE:

$$\hat{\mathbb{P}}_k(s'|s,a) \leftarrow \frac{\sum_{n \in \mathcal{N}_{clust}} \mathbb{1}_{n \in \mathcal{C}_k} N(n,s,a,s')}{\sum_{n \in \mathcal{N}_{clust}} \mathbb{1}_{n \in \mathcal{C}_k} N(n,s,a)}$$
$$\hat{f}_k \leftarrow \frac{\sum_{n \in \mathcal{N}_{clust}} \mathbb{1}_{n \in \mathcal{C}_k}}{N_{clust}} = \frac{|\mathcal{C}_k|}{N_{clust}}$$

The E-step assigns each trajectory to its most likely cluster:

$$k_m \leftarrow \arg\max_k \ell(\hat{\mathbb{P}}_k, \hat{f}_k, m) + \log(\hat{f}_i) \tag{3.1}$$

where $\ell(\hat{\mathbb{P}}_k, \hat{f}_k, n) = \log\left(f_k \prod_{s,s',a} (\hat{\mathbb{P}}_k(s' \mid s,a))^{N(n,s,a,s')}\right)$.

We hope that this is a step towards unifying the discussion on spectral and EM methods for learning mixture models, highlighting that we need not choose between one or the other – spectral methods can initialize the EM algorithm, in one reinterpretation of the refinement step.

Note that refinement using EM is not unique to our algorithm. The model estimation and classification steps in Kong et al. [2020] (under the special case of Gaussian noise) and Chen and Poor [2022] (who already assume Gaussian noise) are exactly the E-step and M-step of the hard EM algorithm as well.

3.3.5 Model Estimation and Classification

Given clusters from the clustering and refinement step, 2 tasks remain, namely those of estimating the models from them and correctly classifying any future trajectories. We can estimate the models exactly as in the M-step of hard EM.

$$\hat{\mathbb{P}}_k(s'|s,a) \leftarrow \frac{\sum_{n\in\mathcal{C}_k} N(n,s,a,s')}{\sum_{n\in\mathcal{C}_k} N(n,s,a)}$$
$$\hat{f}_k \leftarrow \frac{|\mathcal{C}_k|}{N_{clust}}$$

For classification, given a set $\mathcal{N}_{class}$ of trajectories with size N_{class} generated independently of $\mathcal{N}_{clust}$, we can run a process very similar to Algorithm 2 to identify which cluster to assign each new trajectory to. It is worth noting that we can run the classification step on the subspace estimation dataset itself and recover true labels for those trajectories, since trajectories in $\mathcal{N}_{sub}$ and $\mathcal{N}_{clust}$ are independent.

We describe the algorithm in natural language here, and present it formally as Algorithm 3 below. We first compute an orthogonal projector $\tilde{\mathbf{V}}_{s,a}$ to the subspace spanned by the now known approximate models $\hat{\mathbb{P}}_k(\cdot \mid s,a)$. For any new trajectory n and label k, we estimate a distance $\mathrm{dist}(n,k)$ between the model $\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a)$ estimated from n and the model $\hat{\mathbb{P}}_k(\cdot \mid s,a)$ for k, after embedding both in the subspace mentioned above using $\tilde{\mathbf{V}}_{s,a}$. Again, we use a double estimator as hinted at by the use of the subscript i, similar to Algorithm 2. In practice $\mathrm{dist}(n,k)$ could also include occupancy measure differences. Each trajectory n gets the label k_n that minimizes $\mathrm{dist}(n,k)$.

Previous work like Chen and Poor [2022] and Kong et al. [2020] uses the word refinement for its model estimation and classification algorithms themselves. However, we posit that the monotonic improvement in log-likelihood offered by EM makes it well-suited for *repeated application and refinement*, while in our case, the clear theoretical guarantees for the model estimation and classification algorithms make them well suited for *single-step classification*. Note that we can also apply repeated refinement using EM to the labels obtained by single-step classification, which should combine the best of both worlds.

3.4 Analysis

We have the following end-to-end guarantee for correctly classifying all data.

Theorem 3.4.1 (End-to-End Guarantee)**.** *Let both N_{sub} and N_{clust} be $\Omega\left(K^2 S \frac{\log(1/\delta)}{f_{min}^2 \alpha^3 \Delta^8}\right)$ and let*

Algorithm 3 Classification

Input: Clusters $\mathcal{C}_k \subset \mathcal{N}_{clust}$, models $\hat{\mathbb{P}}_k(\cdot \mid s,a)$ estimated from $\mathcal{C}_k$, and a set $\mathcal{N}_{class}$ of trajectories to classify.
Compute $\hat{f}_{k,s,a}$ for all k, s, a.
Compute $\tilde{\mathbf{M}}_{s,a} = \sum_{k=1}^{K} \hat{f}_{k,s,a} \hat{\mathbb{P}}_k(\cdot|s,a) \hat{\mathbb{P}}_k(\cdot|s,a)^T$ and store the orthogonal projector $\tilde{\mathbf{V}}_{s,a}^T$ to its top-K eigenspace, for each (s,a).
Compute $\hat{\mathbf{d}}_k = \frac{1}{|\mathcal{C}_k|} \sum_{n \in \mathcal{C}_k} \frac{N(n,s,a)}{G}$ for all k.
Compute $\tilde{D} = \sum_{k=1}^{K} \hat{\mathbf{d}}_k \hat{\mathbf{d}}_k^T$ and store the orthogonal projector $\tilde{\mathbf{U}}^T$ to its top-K eigenspace.
Compute the set SA_β by picking (s,a) pairs with occurrence more than β
$\mathbf{d}_{n,1}, \mathbf{d}_{n,2} \leftarrow \mathbf{0} \in \mathbb{R}^{SA}$
for $(i,s,a) \in \{1,2\} \times S \times A$ **do**
    Compute $\mathbf{N}(n,i,s,a,\cdot)$, $N(n,i,s,a), \quad \forall n$
    $\hat{\mathbb{P}}_{n,i}(\cdot|s,a) \leftarrow \frac{\mathbf{N}(n,i,s,a,\cdot)}{N(n,i,s,a)} \mathbb{1}_{N(n,i,s,a)\neq 0}, \quad \forall n$
    $[\hat{\mathbf{d}}_{n,i}]_{s,a} \leftarrow \frac{N(n,i,s,a)}{G}, \quad \forall n$
end for
for $(n,k) \in \mathcal{N}_{clust} \times \{1,2,\dots K\}$ **do**
    for $(i,s,a) \in \{1,2\} \times S \times A$ **do**
        $\hat{\mathbf{\Delta}}_{i,s,a} := (\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \hat{\mathbb{P}}_k(\cdot|s,a)) \tilde{\mathbf{V}}_{s,a}^T$
    end for
    $\mathrm{dist}_1(n,k) := \max_{s,a} \hat{\mathbf{\Delta}}_{1,s,a}^T \hat{\mathbf{\Delta}}_{2,s,a}$
    $\mathrm{dist}_2(n,k) := (\hat{\mathbf{d}}_{n,1} - \hat{\mathbf{d}}_k)^T \mathbf{U}\mathbf{U}^T (\hat{\mathbf{d}}_{n,2} - \hat{\mathbf{d}}_k)$
    $\mathrm{dist}(n,k) := \lambda \, \mathrm{dist}_1(n,k) + (1-\lambda) \, \mathrm{dist}_2(n,k)$
end for
Assign $k_n \leftarrow \arg\min_k \mathrm{dist}(n,k)$ for each n.

$T_n = \Omega\left(K^{3/2} t_{mix} \frac{\log^4((N_{clust}+N_{sub})/\delta)\log^3(1/\Delta)\log^4(1/\alpha)}{\Delta^6\alpha^3}\right)$. *If we execute algorithms 1, 2 and model estimation, and then apply algorithm 3 to* $\mathcal{N}_{sub}$ *with* $\lambda = 1$, $\alpha/3 \leq \beta < \alpha$ *and* $\Delta^2/4 \leq \tau \leq \Delta^2/2$ *for clustering and classification, then we can recover the true labels for the entire dataset* ($\mathcal{N}_{clust} \cup \mathcal{N}_{sub}$) *with probability at least* $1 - \delta$.

Proof. This follows directly from Theorems 3.4.2, 3.4.3, 3.4.4 and 3.4.5 upon combining the conditions on N_{sub}, N_{clust}, and T_n in both theorems. We also use the brief discussion after the statement of Theorem 3.4.5. □

The dependence on model-specific parameters like α, Δ and f_{min} is conservative and can be easily improved upon by following the proofs carefully. We chose the form of the guarantees in this section to present a clearer message. In one example, there are versions of these theorems that depend on both G and T_n. We choose $G = (T_n/t_{mix})^{2/3}$ to present crisper guarantees. For understanding how the guarantees would behave depending on both G and T_n, or how to improve the dependence on model-specific parameters, the reader can follow the proofs in the appendix.

3.4.1 Techniques and Proofs

We make a few remarks on the technical novelty of our proofs. As mentioned in Section 3.1, we are dealing with two kinds of non-independence. While we borrow some ideas in our analysis from Chen and Poor [2022] to deal with the temporal dependence, we crucially need new technical inputs to deal with the fact that we cannot cast the temporal evolution as a deterministic function with additive i.i.d. noise, unlike in linear dynamical systems.

We identify the blocking technique in Yu [1994] as a general method to leverage the "near-independence" in observations made in a mixing process when they are separated by a multiple of the mixing time. Our proofs involve first showing that estimates made from a single trajectory would concentrate if the observations were independent, and then we bound the "mixing error" to account for the non-independence of the observations. We first choose a distribution (often labelled as a variant of Q or Ξ) with desirable properties, and then bound the difference between probabilities of undesirable events under Q and under the true joint distribution of observations χ, using the blocking technique due to Yu [1994].

There are many other technical subtleties here. In one example, the number of (s, a) observations made in a single trajectory is itself a random variable and so our estimator takes a ratio of two random variables. To resolve this, we have to condition on the random set of (s, a) observations recorded in a trajectory and use a conditional version of Hoeffding's inequality (different from the

Azuma-Hoeffding inequality), followed by a careful argument to get unconditional concentration bounds, all under Q.

3.4.2 Subspace Estimation

For subspace estimation, we have the following guarantee.

Theorem 3.4.2 (Subspace Estimation Guarantee)**.** *Consider* 2 *models with labels* k_1, k_2 *and a state-action pair* s, a *with* $d_{min}(s,a) \geq \alpha/3$*. Consider the output* $\boldsymbol{V}_{s,a}^T$ *of Algorithm 1. Let* $f_{min} = \min(f_{k_1}, f_{k_2})$ *be the lower of the label prevalences. Remember that each trajectory has length* T_n*.*

Then given that $N_{sub} = \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$*,* $T_n = \Omega(t_{mix}\log^4(1/\alpha))$*, with probability at least* $1-\delta$*, for* $k = k_1, k_2$

$$\|\mathbb{P}_k(\cdot \mid s,a) - \boldsymbol{V}_{s,a}\boldsymbol{V}_{s,a}^T\mathbb{P}_k(\cdot \mid s,a)\|_2 \leq \epsilon_{sub}(\delta)$$

where

- *For* $T_n = \Omega\left(t_{mix}\log^3\left(\frac{f_{min}N_{sub}\alpha}{KS\log(1/\delta)}\right)\right)$

$$\epsilon_{sub}(\delta) = O\left(\sqrt{\frac{K}{f_{min}}\left(\sqrt{\frac{S}{N_{sub}\cdot\alpha^3}\log\left(\frac{1}{\delta}\right)}\right)}\right)$$

- *While for* $T_n = O\left(t_{mix}\log^3\left(\frac{f_{min}N_{sub}\alpha}{KS\log(1/\delta)}\right)\right)$

$$\epsilon_{sub}(\delta) = O\left(\left(\frac{1}{2}\right)^{\frac{1}{16}\left(\frac{T_n}{t_{mix}}\right)^{1/3}}\right)$$

Alternatively, we only need $N_{sub} = \Omega\left(\frac{K^2S\log(1/\delta)}{f_{min}^2\alpha^3\epsilon^4}\right)$ *and* $T_n = \Omega\left(t_{mix}\log^3(1/\epsilon)\log^4(1/\alpha)\right)$ *trajectories for* ϵ *accuracy in subspace estimation with probability at least* $1-\delta$*.*

Remark 4. Why are short trajectories enough? Notice that the length of trajectories only affects the bound as a multiple of t_{mix} with some logarithmic terms. This is because intuitively, the onus of estimating the correct subspace lies on aggregating information across trajectories. So, as long as there are enough trajectories, each trajectory does not have to be long.

3.4.3 Clustering

Remember that Δ is the model separation and α is the corresponding "stationary occupancy measure" from Assumption 2. We give guarantees for choosing $\lambda = 1$, which corresponds to using only model difference information instead of also using occupancy measure information. This is unavoidable since we have no guarantees on the separation of occupancy measures. See Section 3.5.2 for a discussion. Here, we provide a high-probability guarantee for exact clustering.

Theorem 3.4.3 (Exact Clustering Guarantee)**.** *Pick any pair of trajectories n, m. Then for Freq_β so that it contains (s, a) with $d_{min}(s, a) \geq \Omega(\alpha)$, $T_n = \Omega(t_{mix} \log^4(1/\delta)/\alpha^3)$, with probability at least $1 - \delta$,*

$$\left|\mathrm{dist}_1(m, n) - \|\Delta_{m,n}\|_2^2\right|$$

is

$$O\left(\sqrt{\frac{K \log(1/\delta)}{\alpha}} \left(\frac{t_{mix}}{T_n}\right)^{\frac{1}{3}}\right) + 4\epsilon_{sub}(\delta/2)$$

This means that if we choose $\lambda = 1$, then if $\epsilon_{sub}(\delta) \leq \Delta^2/32$ and $T_n = \Omega\left(K^{3/2} t_{mix} \frac{\log^4(N_{clust}/(\alpha\delta))}{\Delta^6 \alpha^3}\right)$, no distance estimate attains a value between $\Delta^2/4$ and $\Delta^2/2$. So, Algorithm 2 attains exact clustering using a threshold of say $\Delta^2/3$ with probability at least $1 - \delta$.

Since we already have high probability guarantees for exact clustering before refinement of the clusters, guarantees for the EM step analogous to the single-step guarantees for refinement in Chen and Poor [2022] are not useful here. However, we do still present single-step guarantees for the EM algorithm in our case using a combination of Theorem 3.4.4 for the M-step and Theorem A.7.1 in Appendix A.7.

3.4.4 Model Estimation and Classification

We also have guarantees for correctly estimating the relevant parts of the models and classifying sets of trajectories different from $\mathcal{N}_{clust}$.

Theorem 3.4.4 (Model Estimation Guarantee)**.** *For any state action pair (s, a) with $d_{min}(s, a) \geq \alpha/3$, and for $GN_{clust} \geq \Omega\left(\frac{\log(1/\delta)}{f_{min}^2 \alpha^2}\right)$ and $T_n \geq \Omega(G t_{mix} \log(G/\delta))$, with probability greater than $1 - \delta$,*

$$\|\hat{\mathbb{P}}_k(\cdot \mid s, a) - \mathbb{P}_k(\cdot \mid s, a)\|_1$$

is bounded above by

$$O\left(\left(\frac{t_{mix}}{T_n}\right)^{1/3} \sqrt{\frac{1}{N_{clust} f_{min} \alpha}(S + \log(\frac{1}{\delta}))}\right)$$

Note that since the 1-norm is greater than the 2-norm, the same bound holds in the 2-norm as

well. Also notice that since our assumptions do not say anything about observing all (s, a) pairs often enough, we can only given guarantees in terms of the occurrence frequency of (s, a) pairs.

Theorem 3.4.5 (Classification Guarantee)**.** *Let $\epsilon_{mod}(\delta)$ be a high probability bound on the model estimation error $\|\hat{\mathbb{P}}_k(\cdot \mid s, a) - \mathbb{P}_k(\cdot \mid s, a)\|_2$. Then there is a universal constant C_3 so that Algorithm 3 can identify the true labels for trajectories in $\mathcal{N}_{class}$ with probability at least $1 - \delta$ for $T_n = \Omega\left(K^{3/2} t_{mix} \frac{\log^4(N_{class}/(\alpha\delta))}{\Delta^6 \alpha^3}\right)$, whenever $\epsilon_{mod}(\delta/2) \leq \frac{C_3 \Delta^4 f_{min} \alpha}{K}$ and $N_{clust} \geq \Omega\left(\frac{\log(1/\delta)}{f_{min}^2 \alpha^2}\right)$.*

Note that by Theorem 3.4.4, a sufficient condition for $\epsilon_{mod}(\delta/2) \leq \frac{C_3 \Delta^4 f_{min} \alpha}{K}$ is $N_{clust} T_n^{2/3} \geq \Omega\left(K^2 t_{mix}^{2/3} S \frac{\log(1/\delta)}{\Delta^8 f_{min}^3 \alpha^3}\right)$. Under the conditions on T_n in Theorem 3.4.5, a suboptimal but sufficient condition on N_{clust} is $N_{clust} = \Omega\left(K^2 S \frac{\log(1/\delta)}{f_{min}^2 \alpha^3 \Delta^8}\right)$, which matches that for N_{sub}.

3.5 Practical Considerations

3.5.1 Subspace Estimation

Heuristics for choosing K: One often does not know K beforehand and often wants temporal features to guide the process of determining K, for example in identifying the number of groups of similar people represented in a medical study. We suggest a heuristic for this. One can examine how many large eigenvalues there are in the decomposition, via (1) ordering the eigenvalues of $\hat{\mathbf{M}}_{sa}$ $\forall s, a$ by magnitude, (2) taking the square of each to obtain the eigenvalue energy, (3) taking the mean or average over states and actions, and (4) plotting a histogram. See Figure 3.2.

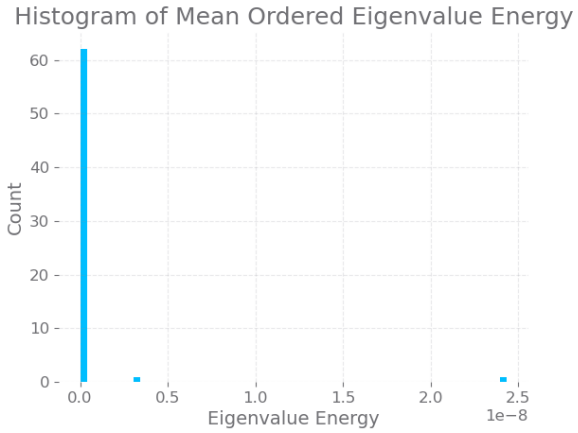

Figure 3.2: Histogram of the average ordered eigenvalue energy (the square of the eigenvalue) where the mean is taken over states and actions. There are two large eigenvalues, corresponding to $K = 2$.

One can also consider running the whole process with different values of K and choose the value

of K that maximises the likelihood or the AIC of the data (if one wishes the mixture to be sparse). However, Fitzpatrick and Stewart [2022] points out that such likelihood-based methods can lead to incorrect predictions for K even with infinite data.

3.5.2 Clustering

Picking β: Choosing β involves heuristically picking state-action pairs that have high frequency and "witness" enough model separation. We propose one method for this. For each (s, a) pair, one first executes subspace estimation and then averages the value of $\mathrm{dist}_1(m, n)$ across all pairs of trajectories. Call this estimate $\Delta_{s,a}$, since it is a measure of how much model separation (s, a) can "witness". We then compute the occupancy measure value $d(s, a)$ of (s, a) in the entire set of observations. Making a scatter-plot of $\Delta_{s,a}$ against $d(s, a)$, we want a value of β so that there are enough pairs from Freq_β in the top right.

Picking thresholds τ: The histogram of dist plotted will have many modes. The one at 0 reflects distance estimates between trajectories belonging to the same hidden label, while all the other modes reflect distance between trajectories coming from various pairs of hidden labels. The threshold should thus be chosen between the first two modes. See Figure 3.3.

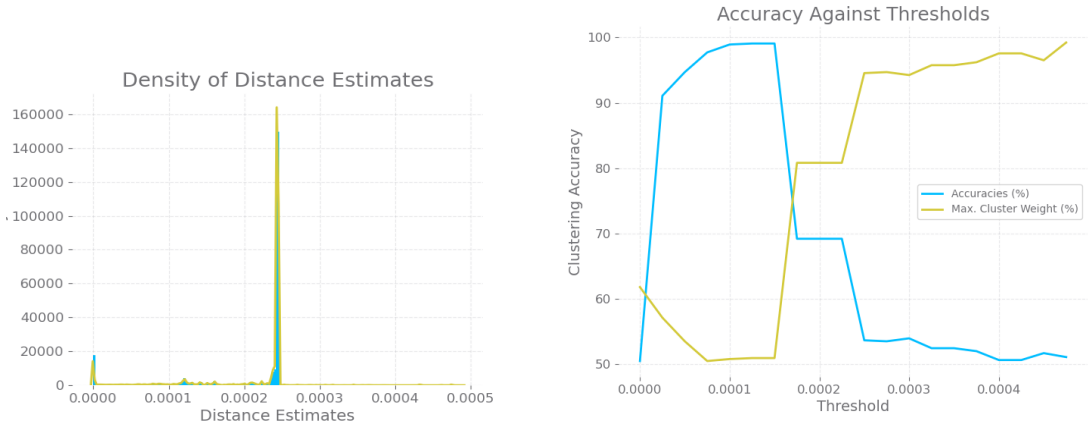

Figure 3.3: Histogram (and KDE) of pairwise squared distance estimates in projected subspace above, and accuracy against thresholds below. Note how there is a spurious mode around the 0.00015 mark, and picking any threshold past it yields a significant drop in accuracy.

Picking λ: In general, occupancy measures are different for generic policies interacting with MDPs and should be included in the implementation by choosing $\lambda < 1$. The histogram for dist_2 should indicate whether or not occupancy measures allow for better clustering (if they have the right number of well-separated modes).

Versions of the EM algorithm: In our description of the EM algorithm, we only use next-state transitions as observations instead of the whole trajectory. So, we do not learn other parameters like the policy and the starting state's distribution for the EM algorithm. This makes sense in principle, because our minimal assumptions only talk about separation in next-state transition probabilities, and there is no guarantee that other information will help with classification. In practice, one should make a domain-specific decision on whether or not to include them.

Initializing soft EM with cluster labels: We also recommend that when one initializes the soft EM algorithm with results from the clustering step, one introduces some degree of uncertainty instead of directly feeding in the 1-0 clustering labels. That is, for trajectory m, instead of assigning $\mathbb{1}(i = k_m)$ to be the responsibilities, make them say $0.8 \cdot \mathbb{1}(i \in \mathcal{C}_k) + 0.2/K$ instead. We find that this can aid convergence to the global maximum, and do so in our experiments.

3.6 Experiments

We perform our experiments for MDPs on an 8x8 gridworld with $K = 2$ elements in the mixture (from Bruns-Smith [2021]). Unlike Bruns-Smith [2021], the behavior policy here is the same across both elements of the mixture to eliminate any favorable effects that a different behavior policy might have on clustering, so that we evaluate the algorithm on fair grounds. The first element is the "normal" gridworld, while the second is adversarial – transitions are tweaked towards having a higher probability of ending up in the lowest-value adjacent state. The value is only used to adjust the transition structure in the second MDP, and has no other role in our experiments. The mixing time of this system is roughly $t_{mix} \approx 25$. We only use dist_1 for the clustering, omitting the occupancy measures to parallel the theoretical guarantees. Including them would likely improve performance. We chose to perform the experiments with 1000 trajectories, given the difficulty of obtaining large numbers of trajectories in important real-life scenarios that often arise in areas like healthcare. The mixture weights are $f_1 = f_2 = 0.5$. Several state-action pairs witness transition differences between the two MDPs, and the threshold τ is chosen from the histogram of pairwise distances as described in Section 3.5.2.

Figure 3.4 plots the error at the end of Algorithm 2 (before refinement) while either using the projectors $\mathbf{V}_{s,a}^T$ determined in Algorithm 1 ("With Subspaces"), replacing them with a random projector ("Random Subspaces") or with the identity matrix ("Without Subspaces"). The difference in performance demonstrates the importance of our structured subspace estimation step. Also note that past a certain point, between $T_n = 60$ and $T_n = 70 \sim 3t_{mix}$, the performance of our method drastically improves, showing that the dependence of our theoretical guarantees on the mixing time is reflected in practice as well. We briefly discuss the poor performance of choosing a random

Figure 3.4: Clustering error v.s. trajectory length on 1000 trajectories, with a comparison between using $\mathbf{V}_{s,a}^T$, $I_{S\times S}$ or a random projector to a K-dimensional subspace in Algorithm 2. The same threshold was used for each trajectory length. Results averaged over 30 trials. The mixing time of this system is roughly $t_{mix} \approx 25$.

subspace in Appendix A.2.

In Figure 3.6, we benchmark our method's end-to-end performance against the most natural benchmark, the randomly initialized EM algorithm. We use the version of the soft EM algorithm that considers the entire trajectory to be our observation, and thus also includes policies and starting state distributions. So, we are comparing our method against the full power of the EM algorithm. We have three different plots, corresponding to (1) soft EM with random initialization, (2) Refining models obtained from the model estimation step applied to $\mathcal{N}_{clust}$ using soft EM on $\mathcal{N}_{clust} \cup \mathcal{N}_{sub}$, and (3) Refining labels for $\mathcal{N}_{clust}$ and $\mathcal{N}_{sub}$ using soft EM (the latter obtained from applying Algorithm 3 to $\mathcal{N}_{sub}$). We report the final label accuracies over the entire dataset, $\mathcal{N}_{clust} \cup \mathcal{N}_{sub}$. Remember that we can view refinement using soft EM as initializing soft EM with the outputs of our algorithms. Note that the plot for (3), which reflects the true end-to-end version of our algorithm, almost always outperforms randomly initialized soft EM. Also, for $T_n > 60$, both variants of our method outperform randomly initialized soft EM. We present a variant of Figure 3.5 with hard EM included as Figure A.2 in the appendix.

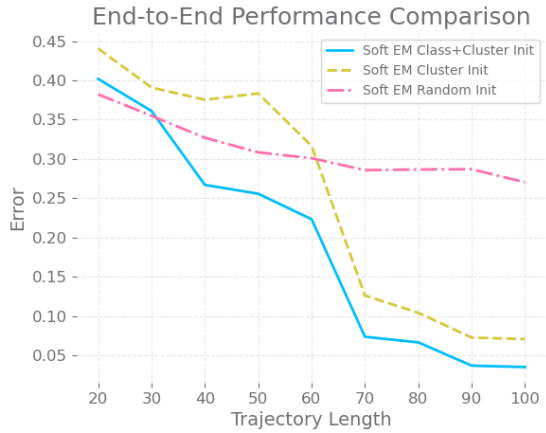

Figure 3.5: End-to-end error v.s. trajectory length on 1000 trajectories, comparing initializations of the soft EM algorithm using (1) random initializations, (2) models from $\mathcal{N}_{clust}$, and (3) classification and clustering labels from $\mathcal{N}_{clust}$ and $\mathcal{N}_{sub}$. Results averaged over 30 trials, with 30 random initializations for randomly-initialized EM within each trial.

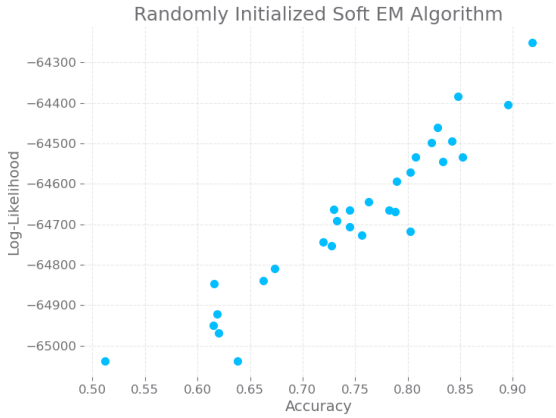

Figure 3.6: Scatter-plot of likelihoods v.s. clustering accuracy achieved by the randomly-initialized soft EM algorithm over 30 trials on gridworld. Randomly-initialized soft EM does not achieve the global maximum all of the time.

3.7 Discussion

We have shown that we can recover the true trajectory labels with (1) the number of trajectories having only a linear dependence in the size of the state space, and (2) the length of the trajectories depending only linearly in the mixing time – even before initializing the EM algorithm with these clusters (which would further improve the log-likelihood, and potentially cluster accuracy). End-to-end performance guarantees are provided in Theorem 3.4.1, and experimental results are both

promising and in line with the theory.

3.7.1 Future Work

Matrix sketching: The computation of $\mathrm{dist}_1(m, n)$ is computationally intensive, amounting to computing about $S \times A$ distance matrices. We could alternatively approximate the thresholded version of the matrix $\mathrm{dist}(m, n)$ (which in the ideal case is a rank-K binary matrix) with ideas from Musco and Musco [2016].

Function approximation: The question of the right extension of our ideas to Markov chains and MDPs with large, infinite, or uncountable state spaces is very much open (at least, those whose transition kernel is not described by a linear dynamical systems). This is important, as many applications often rely on continuous state spaces.

Other controlled processes: Chen and Poor [2022] learn a mixture of linear dynamical systems without control input. An extension to the case with control input will be very valuable. We believe that the techniques used in our work may prove useful in this, as well as for extensions to other controlled processes that may neither be linear nor Gaussian.

A natural next step is to apply these clustering methods to real-world datasets of practical importance, such as electronic health records or recommendation system logs where latent heterogeneity is a known concern.

CHAPTER 4

Offline Policy Evaluation and Optimization under Confounding

In the previous chapter, we developed spectral methods for clustering trajectories from a mixture of MDPs. This clustering machinery was in fact developed to address a specific sub-problem in the broader landscape of offline reinforcement learning under confounding. In the running example of Section 1.2.2, behavioral scientists who administered interventions had access to situational knowledge not captured in the sensor data, and this knowledge influenced which interventions were sent, biasing the historical trajectories in ways invisible to standard offline analysis. The latent variable here does not primarily touch the transitions or the reward, but rather the *data collection process*: actions in the dataset were chosen partly based on information the analyst no longer has. In the language of Section 1.4, the confounder's reach is bounded by the sensitivity parameter Γ, the low-complexity channel in this setting. When the confounder is sampled once per trajectory and remains fixed (a "global confounder"), the offline data is precisely a mixture of MDPs, and the clustering methods of the previous chapter allow us to separate this mixture and perform policy evaluation within each component. This motivates the broader question addressed in this chapter: what is the full landscape of offline policy evaluation under confounding? When is consistent estimation possible, and when is it fundamentally limited?

4.1 Introduction

This chapter is a lightly edited version of Kausik et al. [2024a].

A central problem in sequential decision making is learning from offline data, since collecting data in an online fashion is often prohibitively expensive or unsafe [Levine et al., 2020]. Since real-life data is often affected by latent variables, there has been a rise of interest in formulations of reinforcement learning problems with hidden information [Nair and Jiang, 2021, Miao et al., 2022, Wang et al., 2020]. The most general kind of latent information is considered by partially observable MDPs or POMDPs [Kaelbling et al., 1998b, Tennenholtz et al., 2019], where the latent

	With Sensitivity Constraint	Without Sensitivity Constraint
Memoryless Confounders	Consistency not possible (Theorem 4.2.1, $\Omega(\varepsilon H)$ error lower bound), $O(\varepsilon H^2)$ error upper bound with 3 methods (Theorems 4.2.2, 4.2.3, 4.2.4)	$\Omega(H)$ error lower bound (Theorem 4.2.1)
Confounders with Memory	Methods mentioned above have $\Omega(H)$ error lower bounds, even with unconfounded π_b and π_e (Theorem 4.2.6)	$\Omega(H)$ lower bound in general. For global confounders, consistency possible, sample complexity guarantees given (Theorem 4.2.7)

Table 4.1: Hardness of the OPE problem under different assumptions on the nature of confounding present. Γ is a so-called sensitivity parameter, with $\Gamma = 1 + O(\varepsilon)$. Higher ε corresponds to more confounded π_b.

information can affect both rewards and transitions. However, the reward is often *designed* by the user based only on observable variables. In medical examples, the reward could be given based on observed vitals, but unrecorded genetic conditions and socio-economic status can affect actions taken and future states. These examples motivate the important case of reinforcement learning with unobserved confounders, defined as latent information that affects transitions, but not rewards[1] [Kallus and Zhou, 2020, Bruns-Smith, 2021, Bruns-Smith and Zhou, 2023].

The hardness of learning from offline data under confounding comes from the fact that partially observed transitions can be further obscured by behavior policies that might have known the unrecorded confounder [Kallus and Zhou, 2020]. Two offline data distributions might thus be identical despite coming from different confounded MDPs, if the behavior policies accommodated for this difference (see Theorem 4.2.1).

To provide guarantees for learning from offline data, the most common assumption in previous work is that confounders are "memoryless" (Assumption 3). This assumption essentially means that they are sampled afresh at each step independently of past confounders, states, or actions [Bruns-Smith and Zhou, 2023]. In many real-life applications like healthcare and epidemiology [Daniel et al., 2013, Clare et al., 2018, Mansournia et al., 2017, Platt et al., 2009], it is more appropriate to assume that the confounders are sampled "with memory" of previous confounders, and even states and actions. A lot of work also assumes that behavior policies follow a sensitivity constraint

[1]Some papers define confounders using a kind of "memorylessness," and allow them to affect rewards [Zhang and Bareinboim, 2016, Wang et al., 2020]. We only consider unconfounded rewards.

(Assumption 5) [Kallus and Zhou, 2020, Bruns-Smith, 2021]. Motivated by these observations, we take the first step towards providing a structured view of the landscape of offline RL for confounded MDPs, distinguishing settings in terms of sensitivity assumptions and whether confounders have memory. We also introduce and study an important sub-case of confounders with memory, called global confounders (Assumption 4). Specifically, we ask the following questions for each setting:

Q.1. *If consistent offline policy estimation (OPE) is not possible, can we prove lower bounds on the error? What guarantees can we give for algorithms that instead estimate bounds on the value?*

Q.2. *If consistent OPE is possible, then what algorithms achieve this? What is their sample complexity?*

Q.3. *How can we use these insights for offline policy improvement?*

Paper Structure and Contributions. We detail our contributions below. A summary of key results is provided in Table 4.1.

OPE for Memoryless Confounders, Section 4.2.3: In Theorem 4.2.1, we give the first lower bound for OPE error that depends on a sensitivity parameter Γ and horizon length H. By choosing Γ appropriately, we show that value estimation can be *arbitrarily* bad without a sensitivity constraint. The theorem also *quantitatively* shows that the lower bound on error grows with H and consistent estimates are not possible, even under a sensitivity constraint. To provide algorithms that estimate lower bounds on the value, we modify the CFQE algorithm due to Bruns-Smith [2021] to our more general definition of memoryless confounding. We are the first to compute quantitative upper bounds on its error and the error for FQE, in Theorems 4.2.2 and 4.2.3. We further provide a new model-based algorithm that improves over CFQE for stationary transition structures, and provide guarantees for it in Theorems 4.2.4 and 4.2.5.

OPE for Confounders with Memory, Section 4.2.5: While FQE is a standard workhorse for OPE and also enjoys guarantees for memoryless confounders, it is unclear if (and how badly) FQE fails for confounders with memory. In particular, it is non-trivial to produce lower bound examples in this case. We are the first to present one in Theorem 4.2.6, where we show that FQE can have *arbitrarily* large error for confounders with memory, even for unconfounded π_b and π_e with bounded concentrability. This shows the hardness of OPE for *general* confounders with memory. In this light, we introduce and study the important sub-case of *global confounders*, where the confounder is fixed at the beginning of each trajectory. We leverage the work of Kausik et al. [2022] on clustering mixtures of MDPs to provide an algorithm for OPE under this assumption, along with sample complexity guarantees in Theorem 4.2.7. While past work on confounded RL has focused only on

consistency, we are the first to address the sample complexity of OPE under confounding.

Offline Policy Improvement, Section 4.2.7: We address offline policy improvement in Section 4.2.7, presenting policy gradient methods for memoryless confounders under a sensitivity assumption, as well as for global confounders. We prove local convergence for both.

Experiments, Section 4.3: We test and compare OPE methods for memoryless confounders in the gridworld environment provided by Bruns-Smith [2021]. Our experiments show that our model-based method gives tighter lower bounds than existing methods. We also successfully run our policy gradient method for memoryless confounders in the same environment. OPE and policy gradient methods for global confounders are tested in the sepsis simulator from Oberst and Sontag [2019], where we significantly outperform confounder-oblivious implementations of both FQE and policy gradients.

Related Work. Many specific assumptions on confounders have been studied in recent literature. Kallus and Zhou [2020], Bruns-Smith [2021], Namkoong et al. [2020] all provide algorithms that estimate bounds on the value under a sensitivity assumption. The first two assume variants of memorylessness, while the third assumes that the confounding occurs during only a single timestep. Other work like Bennett et al. [2020] uses a latent variable model for states and actions to get consistent point estimates. This is similar to work in the POMDP setting [Tennenholtz et al., 2019], and neither approach directly applies to our settings: they require either that a subset of confounders is directly observed (satisfying the backdoor criterion) or that observed mediator variables satisfy the frontdoor criterion, enabling causal identification of transition dynamics. Our setting assumes no such auxiliary variables are available. In general, a treatment of confounders with memory and a big-picture view of the OPE problem under confounding is still missing.

On the other hand, literature on offline policy *improvement* in the presence of confounders has grown more gradually. Bruns-Smith and Zhou [2023] provide robust fitted-Q-iteration methods under a sensitivity model and a memoryless assumption. This does not apply to confounders with memory, like global confounders. Other work like Wang et al. [2020], Liao et al. [2021], Fu et al. [2022] uses auxiliary variables from the data to adjust for confounding bias. However, these do not directly apply to our settings, since they require observed auxiliary variables (instrumental variables or mediators satisfying the frontdoor criterion) that are absent from our formulation.

4.2 Setup and Assumptions

4.2.1 Background

We define an episodic confounded MDP by a tuple $(\mathcal{S} \times \mathcal{U}, \mathcal{A}, H, \{\mathbb{P}_h\}_{h=1}^H, r, d_0)$, described as follows. $\mathcal{S}$ is the set of S observed states and $\mathcal{U}$ the set of U unobserved confounders; $\mathcal{A}$ is the set of A actions; H is the horizon of each episode; d_0 is the distribution for initial states $(s_1, u_1) \sim d_0$; $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ denotes the reward function; and $\mathbb{P}_h(s', u' \mid s, u, a)$ denotes the state transition probability at timestep h.

The data is collected under a behavior policy π_b specified by $\pi_{b,h}(a \mid s, u)$, which might have used the unrecorded confounders and been time-dependent. The observed behavior policy is obtained by marginalizing u over the induced distribution at timestep h, and is called $\pi_{b,h}(a \mid s)$. The goal is to estimate the value function $V_1^{\pi_e}$ of a possibly time-dependent evaluation policy π_e *that does not use confounders* [Bruns-Smith, 2021]. This is motivated by the fact that confounders can be harder to observe and account for during deployment.

4.2.2 Assumptions on Sensitivity and Memory

We consider two kinds of assumptions on unobserved confounders. The first is whether they "have memory." We define memoryless confounders below to be sampled afresh at each step [Bruns-Smith and Zhou, 2023]. A memoryless confounder in a healthcare application could be an accident encountered mid-treatment, or in an economics application could be a supply shock affecting the price of oil, as Bruns-Smith [2021] highlights.

Assumption 3 (Memoryless Confounders)**.** At each timestep h, we draw a fresh confounder $u_h \sim P_h(u \mid s = s_h)$, possibly dependent on the current state s_h, but independent of past confounders, states and actions.

On the other hand, confounders with memory could depend on all past (s, a, u) tuples. We introduce an important sub-case of this, which we call the *global confounder* assumption. This is an extreme case of confounders with memory, where the confounder is not just dependent on, but the *same* as all past confounders in the trajectory. In the example of healthcare applications, this could be an unrecorded patient demographic characteristic or genetic condition that does not change over the course of treatment.

Assumption 4 (Global Confounders)**.** A global confounder is generated by $u \sim P(u)$ at the beginning of an episode, and remains unchanged throughout the episode.

A commonly-used assumption for the effect of confounder on π_b is a sensitivity model found in [Bruns-Smith, 2021, Kallus and Zhou, 2020, Namkoong et al., 2020]. Note that $\Gamma = 1$ below

corresponds to the case where π_b is *confounder-oblivious*, that is, independent of the confounder.

Assumption 5 (Confounding Sensitivity Model)**.** Given $\Gamma \geq 1$, for all $s \in \mathcal{S}, u \in \mathcal{U}$, $h \in \{1, 2, \cdots H\}$ and $a \in \mathcal{A}$:

$$\frac{1}{\Gamma} \leq \left(\frac{\pi_{b,h}(a \mid s,u)}{1-\pi_{b,h}(a \mid s,u)}\right) / \left(\frac{\pi_{b,h}(a \mid s)}{1-\pi_{b,h}(a \mid s)}\right) \leq \Gamma,$$

where $\pi_{b,h}(a \mid s) = \sum_u P_h(u \mid s)\pi_{b,h}(a \mid s,u)$ is the marginalized (observed) behavior policy. The above inequality implies the bounds $\alpha_h(s,a) \leq \frac{\pi_{b,h}(a|s)}{\pi_{b,h}(a|s,u)} \leq \beta_h(s,a)$, where $\alpha_h(s,a) := \pi_{b,h}(a \mid s) + \frac{1}{\Gamma}(1-\pi_{b,h}(a \mid s))$ and $\beta_h(s,a) := \Gamma + \pi_{b,h}(a \mid s)(1-\Gamma)$.

We discuss OPE when confounders are memoryless. We first open with a result showing that in the absence of a sensitivity assumption like Assumption 5, we can incur an estimation error as bad as $\Omega(H)$. Note that the value functions lie in the range $[0, H]$, so the worst possible OPE error is H.

Theorem 4.2.1 (Lower Bound for Memoryless Confounders)**.** *There exists a parameter ε that determines a pair of confounded MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ with i.i.d. (and thus memoryless) confounders along with stationary policies π_{b_1}, π_{b_2} and π_e, so that data collected from $\mathcal{M}_i$ using π_{b_i} has the same distribution for $i = 1, 2$, but the values under π_e differ by $|V_1^{\pi_e}(\mathcal{M}_1) - V_1^{\pi_e}(\mathcal{M}_2)| = 2\varepsilon H$. In particular, when $\varepsilon = \frac{1}{2} - \frac{1}{H^2}$, the values under π_e differ by $\Omega(H)$.*

It can be seen from the proof of the theorem in Appendix B.2 that when $\varepsilon = \frac{1}{2} - \frac{1}{H^2}$, $\Gamma = \Omega(H^2)$. It is then clear that a bound on the sensitivity is necessary. The proof shows that for small ε in our example, $\Gamma = 1 + O(\varepsilon)$. In this light, even with a sensitivity constraint of $1 + O(\varepsilon)$, we cannot get a consistent estimate of the value of a policy. This is because by Theorem 4.2.1, even two observationally indistinguishable confounded MDPs can differ in value under a new π_e by $\Omega(\varepsilon H)$.

Thus, even with infinite data, we can only hope for *bounds* on the value, and the minimum-possible error deteriorates with horizon H. We now analyze and present algorithms for obtaining such bounds.

4.2.3 FQE and Confounded FQE

Fitted Q-Evaluation (FQE) is a standard workhorse for OPE. We present the algorithm below, adapted for memoryless systems (see also Appendix B.3).

We first present a new result on the estimation error of FQE under memoryless confounding, proved in Appendix B.4.

Theorem 4.2.2 (FQE Error)**.** *Suppose $\Gamma = 1 + \varepsilon$ in Assumption 5. Then in the limit of infinite samples, the point estimate $\hat{f}_1(s,a)$ of the Q-function produced by FQE has a worst-case error of $|V_1^{\pi_e}(s) - \sum_a \pi_{e,1}(a \mid s)\hat{f}_1(s,a)| = O(\varepsilon H^2)$ for small ε.*

Algorithm 4 FQE

input: evaluation policy π_e.
initialize: $\hat{f}_{H+1} \leftarrow 0$.
for $h = H, H-1, \ldots, 1$ **do**
    $\hat{f}_h(s,a) \leftarrow \mathbb{E}_{(s,a,s') \sim \mathcal{D}_{\pi_b},h} \left[r_h(s,a) + \sum_{a'} \pi_{e,(h+1)}(a' \mid s') \hat{f}_{h+1}(s',a') \right], \forall s, a.$
end for
return: $\sum_a \pi_{e,1}(a \mid s) \hat{f}_1(s,a)$ for $\forall s$.

Note that FQE gives a point estimate instead of a lower bound on the value function. For many safety-critical applications, it is important to have conservative lower bounds for policy estimation. Using the proof of Theorem 4.2.2, we can produce a straightforward lower bound of $\sum_a \pi_{e,1}(a \mid s) \hat{f}_1(s,a) - k\varepsilon H^2$ on the value function, for some k depending on ε. However, this is a worst-case, data-oblivious lower bound. We note that we can get a sharper lower bound using confounded FQE (CFQE), introduced by Bruns-Smith [2021] for i.i.d. confounders. Confounded FQE gives a lower bound on the value by sequentially searching for the *worst possible policies* that are consistent with the data and the sensitivity assumption. We adapt it to general memoryless confounders below.

Let $\hat{\pi}_{b,h}(a \mid s)$ and $\hat{\mathbb{P}}_h(s' \mid s,a)$ be empirical estimates from finite data $\mathcal{D}_{\pi_b,h}$. Let $\mathbb{P}_h^{\pi_b}(s' \mid s,a)$ be the limit of $\hat{\mathbb{P}}_h(s' \mid s,a)$ under infinite data. We then define the following uncertainty sets.

Definition 4.2.1 (Valid Behavior Policy Set)**.** Under a memoryless confounder, for all s, a, s', define $\mathcal{B}_{sa,h}$ to be the set of all $\pi(a \mid s, \cdot)$ that satisfy Assumption 5 and the two equations below.

$$\sum_{u \in \mathcal{U}} P_h(u \mid s) \pi_{b,h}(a \mid s,u) = \pi_{b,h}(a \mid s)$$
$$\sum_{u \in \mathcal{U}} P_h(u \mid s) \pi_{b,h}(a \mid s,u) P(s' \mid s,u,a) = \pi_{b,h}(a \mid s) \mathbb{P}_h^{\pi_b}(s' \mid s,a).$$

Now we define the following quantity using the posteriors $P_h^{\pi_b}(u \mid s,a)$, a confounded analog to inverse propensity weights.

$$\begin{aligned} g_h(s,a,s') &:= \sum_u \left(\frac{P_h^{\pi_b}(u \mid s,a) \mathbb{P}_h(s' \mid s,a,u)}{\hat{\mathbb{P}}_h^{\pi_b}(s' \mid s,a)} \right) \frac{1}{\pi_{b,h}(a \mid s,u)} \\ &= \sum_u \left(\frac{P_h(u \mid s) \mathbb{P}_h(s' \mid s,a,u)}{\hat{\mathbb{P}}_h^{\pi_b}(s' \mid s,a)} \right) \frac{1}{\pi_{b,h}(a \mid s)} \end{aligned}$$

Theorem 1 and the discussion following that in Bruns-Smith [2021] shows that we can reflect

the same uncertainty using the set $\tilde{\mathcal{B}}_{sa,h}$ of possible values of $g_h(s,a,\cdot)$.

$$\tilde{\mathcal{B}}_{sa,h} := \{g_h(s,a,\cdot) \mid \alpha_h(s,a) \leq \pi_{b,h}(a \mid s) g_h(s,a,s') \leq \beta_h(s,a), \sum_{s'} \pi_{b,h}(a \mid s) g_h(s,a,s') \mathbb{P}_h^{\pi_b}(s' \mid s,a) = 1\} \tag{4.1}$$

$\tilde{\mathcal{B}}_{sa,h}$ presents a reparameterization of the uncertainty that allows us to get rid of the explicit presence of the unknown variable u while optimizing over the uncertainty set. Let $\hat{\mathcal{B}}_{sa,h}$ and $\hat{\tilde{\mathcal{B}}}_{sa,h}$ be the version of these sets determined by the point estimates $\hat{\pi}_b$ and $\hat{\mathbb{P}}(s' \mid s,a)$ under finite data, instead of by their true values.

Algorithm 5 Confounded FQE (adapted from Bruns-Smith [2021])

1: **input:** evaluation policy π_e.
2: **initialize:** $\hat{f}_{H+1} \leftarrow 0$.
3: **for** $h = H, H-1, \ldots, 1$ **do**
4: Compute

$$\hat{f}_h(s,a) := \min_{g_h(s,a,\cdot) \in \hat{\tilde{\mathcal{B}}}_{sa,h}} \mathbb{E}_{(s,a,s') \sim \mathcal{D}_{\pi_b,h}} \Big[\hat{\pi}_{b,h}(a \mid s) g_h(s,a,s') \left(r_h(s,a) + \sum_{a'} \pi_{e,h}(a' \mid s') \hat{f}_{h+1}(s',a') \right) \Big]$$

5: **end for**
6: **return:** $\sum_a \pi_e(a \mid s) \hat{f}_1(s,a)$ for $\forall s$.

We also provide a new theoretical guarantee for the worst-case error of CFQE below, proved in Appendix B.4.

Theorem 4.2.3 (CFQE Error)**.** *Suppose $\Gamma = 1 + \varepsilon$ in Assumption 5. Then the worst-case error for the lower bound $\hat{f}_1(s,a)$ generated by CFQE in the infinite-sample case is $|V_1^{\pi_e}(s) - \sum_a \pi_{e,1}(a \mid s) \hat{f}_1(s,a)| = O(\varepsilon H^2)$ for any range of ε.*

Although it has the same *worst-case* error as FQE, we note that CFQE provides an *instance-dependent* lower bound that is sharper than the naive one mentioned above. We confirm in experiments that the naive FQE lower bound and the CFQE lower bound are in fact at different orders of magnitude.

4.2.4 Model-Based Method For Stationary Transition Kernels

While CFQE searches for the worst-possible *policies*, we discuss a method here that searches for the worst possible *transition dynamics* that are consistent with the data. Note that since π_e

is confounder-oblivious, the induced transitions $\mathbb{P}_h^{\pi_e}(s' \mid s)$ are determined by the marginalized transition dynamics defined as $\mathbb{P}_h(s' \mid s, a) := \sum_u P_h(u \mid s)\mathbb{P}_h(s' \mid s, a, u)$. This is clear from the following computation: $\mathbb{P}_h^{\pi_e}(s' \mid s) = \sum_{u,a} \pi_{e,h}(a \mid s)P_h(u \mid s)\mathbb{P}_h(s' \mid s, a, u) = \sum_a \pi_{e,h}(a \mid s) \left(\sum_u P_h(u \mid s)\mathbb{P}_h(s' \mid s, a, u)\right) = \sum_a \pi_{e,h}(a \mid s)\mathbb{P}_h(s' \mid s, a)$.

We note that CFQE optimizes separately over the data at each timestep h. In particular, if the marginalized transition kernel were stationary, then the method would not leverage its stationarity. Our model-based method can leverage this, and we therefore assume the stationarity of transition dynamics and of $P(u \mid s)$ in this section. For ease of exposition, we also assume that π_b and π_e are stationary. The method can be modified to work for potentially time-dependent π_b and π_e, which we do in Appendix B.5.

We now describe the method. Let the empirically observed transitions be $\hat{\mathbb{P}}^{\pi_b}(s' \mid s, a)$, and denote its value in the limit of infinite data by $\mathbb{P}^{\pi_b}(s' \mid s, a)$. We know that the latter is stationary under our expository simplification. Let $\hat{\alpha}(s, a)$ and $\hat{\beta}(s, a)$ be obtained using the estimate $\hat{\pi}_b(s, a)$ Denote by $\mathcal{G}$ the set of marginalized transitions $\mathbb{P}(s'|s, a)$ that fall between $\hat{\alpha}(s, a)(\hat{\mathbb{P}}^{\pi_b}(s'|s, a))$ and $\beta(\hat{s}, a)(\hat{\mathbb{P}}^{\pi_b}(s'|s, a))$ for each s', a, s. Our model-based method amounts to solving the following optimization problem:

$$\begin{aligned} &\min_{V_1(s_0), V_2, ..., V_H, V_{H+1}=0, \mathbb{P}} V_1(s_0) \\ &\text{s.t. } \mathbb{P} \in \mathcal{G}, \ \sum_{s'} \mathbb{P}(s' \mid s, a) = 1 \ \ \forall s, a. \\ &V_h(s) = \pi_e(\cdot \mid s)^T (R_s + \mathbb{P}_s V_{h+1}(\cdot)) \ \ \forall h \in \{1, ..., H\}, s \end{aligned} \tag{4.2}$$

where $V_{H+1} = 0$ and $\mathbb{P}_s \in \mathbb{R}^{A \times S}$ is the matrix whose rows are $\mathbb{P}(\cdot \mid s, a)$ for each a, $R_s \in \mathbb{R}^A$ and $V_{h+1}(\cdot) \in \mathbb{R}^S$. This corresponds to minimizing the value function $V_1(s_0)$ over the set $\mathcal{G}$ of state transition probabilities, using $H \cdot S$ Bellman backup constraints to encode the Bellman equation.

While this method is similar to the model-based method in [Bruns-Smith, 2021] inspired by robust MDP literature, it is important to note that unlike Bruns-Smith [2021], we look at uncertainty sets for each s, a (instead of just one for each s) and make no additional assumption on model-sensitivity. In particular, model sensitivity and the uncertainty sets for the true marginalized transition kernel are completely determined by Γ. This method possesses several theoretical guarantees, proved in Appendix B.5.

Theorem 4.2.4 (Error for the Model-Based Method)**.** *Suppose $\Gamma = 1 + \varepsilon$ in Assumption 5. Then the value estimation from solving* (4.2) *with infinite data, denoted by $\tilde{V}_1$, provides a lower bound no looser than CFQE and satisfies that $|V_1^{\pi_e}(s_0) - \tilde{V}_1(s_0)| = O(\varepsilon H^2)$ for any range of ε.*

We will find in experiments that the lower bound produced by the model-based method is in fact tighter in some scenarios. In the finite-sample setting, we use point estimates $\hat{\mathbb{P}}^{\pi_b}$ to construct $\mathcal{G}$. In another version for finite samples, one can account for estimation error of $\hat{\mathbb{P}}^{\pi_b}$ by constructing a Hoeffding confidence interval for the state transition probabilities, and using it to construct $\mathcal{G}$ instead. We discuss this in Appendix B.5. Denoting the output of either version by $\hat{V}_1$, the theorem below guarantees that $\hat{V}_1$ is a consistent estimate for the infinite-sample lower bound $\tilde{V}_1$. We prove it in Appendix B.5, and the Hausdorff-distance-based technique developed for the proof can be used to provide similar guarantees for FQE and CFQE.

Theorem 4.2.5 (Consistent Estimation of the Lower Bound)**.** *The estimated lower bound from the model-based method is strongly consistent for the lower bound $\tilde{V}_1$, where $\tilde{V}_1$ is the lower bound estimate of the value function from solving* (4.2) *with infinite data. That is, $\hat{V}_1 \xrightarrow{a.s.} \tilde{V}_1$.*

A Computationally Efficient Method. Although the non-convex optimization problem in (4.2) is solvable with off-the-shelf solvers, such problems can be difficult to solve efficiently. We provide Algorithm 6 below for quicker computation of lower bounds. This method approximately solves the model-based optimization problem in (4.2) via projected gradient descent, optimizing over $\mathbb{P}$ while maintaining the Bellman constraints.

Algorithm 6 Projected Gradient Descent for Model-Based Lower Bound

1: **input:** evaluation policy π_e, empirical estimate of $\mathbb{P}$, decaying learning rate η_t, starting state s_0.
2: **initialize:** $V_{H+1} \leftarrow 0$.
3: **for** $t = 1, ..., N$ **do**
4: **for** $h = H, H-1, \ldots, 1$ **do**
5:

$$V_h(s) := \sum_a \pi_e(a \mid s) \left[R(s,a) + \sum_{s'} \mathbb{P}(s' \mid s, a) V_{h+1}(s') \right]$$
$$= \pi_e(\cdot \mid s)^T (R_s + \mathbb{P}_s V_{h+1}(\cdot)).$$

6: **end for**
7: $\mathbb{P} \leftarrow \text{Proj}_{\mathcal{G}}(\mathbb{P} - \eta_t \nabla_{\mathbb{P}} V_1(s_0))$
8: **end for**
9: **return** the lowest $V_1(s_0)$ encountered.

Non-Stationary Model-Based Method. To handle non-stationary settings, we provide Algorithm 17 in Appendix B.6. This relaxes the Bellman backup constraints in (4.2) by sequentially

solving H efficiently solvable quadratic programs. This is essentially the model-based analogue to CFQE.

4.2.5 Hardness of OPE for Confounders with Memory

Sensitivity constraints do not alone contribute to the error upper bounds in Section 4.2.3 – the memorylessness of confounders is an important ingredient. We demonstrate below that OPE under confounders with memory is hard even for π_b with the best-case sensitivity, $\Gamma = 1$. Recall that $\Gamma = 1$ corresponds to confounder-oblivious behavior policies. Specifically, the theorem below shows FQE and any method that lower bounds FQE will have $\Omega(H)$ worst-case error for confounders with memory, even for unconfounded π_b and π_e with bounded concentrability and given infinite data. We prove it in Appendix B.7.

Theorem 4.2.6 (Lower Bound for Confounders with Memory)**.** *There exists an MDP $\mathcal{M}$ having confounders with memory, a stationary unconfounded behavior policy π_b with sensitivity $\Gamma = 1$, a stationary evaluation policy π_e with $\frac{\pi_e(a|s)}{\pi_b(a|s)} \leq 2\ \forall s, a$, and a state s_1, so that $V_1^{\pi_e}(s_1) = \Omega(H)$ while the output of FQE for π_e is $O(\log H)$, even with infinite data.*

Proof sketch. The construction uses $S = \{s_1, s_2\}$, $A = \{a_1, a_2\}$, and a confounder with memory: it starts at u_{a_1} and remains there only if action a_1 is consistently taken, otherwise transitioning to u_0. Under $\pi_e(a_1 \mid s) = 1$, the system stays in u_{a_1} and s_1 permanently, achieving $V_1^{\pi_e}(s_1) = H$. Under the behavior policy $\pi_b(a \mid s, u) = 1/2$, the probability of being in u_{a_1} at step h decays as $1/2^{h-1}$, so FQE — which computes expectations over the behavior-policy-induced confounder distribution — effectively sees a mixture dominated by u_0 for most timesteps. The FQE estimate is at most $O(\log H)$, yielding $\Omega(H)$ error. The full proof is in Appendix B.7.

While the challenges of FQE for POMDPs in general are qualitatively understood [Uehara et al., 2022], we show that it can be *arbitrarily* bad even in the much milder setting of confounded MDPs with unconfounded π_b and π_e. This suggests that making more specific assumptions about confounders with memory is necessary for designing OPE algorithms with theoretical guarantees. One example of such an assumption is the global confounder assumption, discussed below.

4.2.6 Clustering-Based OPE for Global Confounders

The main message of this section is that the dependence of confounders across timesteps can make it possible to pin down the effect of confounding and achieve consistent OPE, given enough structure to the dependence. We bring our focus to global confounders (Assumption 4) in the case where transition dynamics are stationary, and so are the behavior and evaluation policies. Notice that in

the stationary setting, global confounders exactly describe a mixture of MDPs. Let the value of the evaluation policy π_e under the dynamics induced by confounder u be $V_1(s_0; u, \pi_e)$. If one can estimate this value and $P(u)$ for each u, then one can provide point estimates of the policy value $V_1^{\pi_e}(s_0) = \sum_u P(u) V_1(s_0; C_u, \pi_e)$.

The approach of decomposing by confounder type and averaging per-component value estimates is conceptually related to mixture importance sampling in the Monte Carlo literature; see Owen [2013], Section 9.11. Here the mixture components correspond to latent confounder types, and the key insight is that separating heterogeneous data into homogeneous components before applying OPE avoids the confounding bias that would arise from treating the mixture as a single population.

We use Algorithm 7 as a broad meta-algorithm that takes a clustering algorithm and an OPE algorithm as input. We cluster the data and apply the OPE algorithm separately to each cluster to obtain a consistent final policy value estimate $\hat{V}_1(s_0; \pi_e)$. The crucial intuition behind this algorithm is the fact that the value estimate is a weighted average of value estimates over each confounder.

Algorithm 7 Clustering-Based OPE

1: **input:** Number of clusters U, evaluation policy π_e, clustering algorithm `cluster()`, OPE estimator `ope()`.
2: **run subroutine:** Use `cluster()` to obtain clusters $C_1, ..., C_U$.
3: Obtain cluster weight estimates $\hat{P}(u) := \frac{|C_u|}{N_{traj}}$.
4: **run subroutine:** Estimate $\hat{V}_1(s_0; C_u, \pi_e)$ for each cluster C_u using `ope()`.
5: **return:** Output the final policy value estimate $\hat{V}_1(s_0; \pi_e) = \sum_{u=1}^{U} \hat{P}(u_i) \hat{V}_1(s_0; C_u, \pi_e)$.

To present an end-to-end theoretical guarantee, we instantiate the meta-algorithm using the recent work of Kausik et al. [2022] as our clustering algorithm and the data-splitting tabular-MIS (marginalized importance sampling) estimator from Yin and Wang [2020] as our OPE estimator. To satisfy the assumptions of Kausik et al. [2022] and Yin and Wang [2020], we require 3 additional assumptions, discussed in their papers.

Assumption 6 (Mixing, from Kausik et al. [2022])**.** Let the U Markov chains on $\mathcal{S} \times \mathcal{A}$ induced by the various behavior policies $\pi(a \mid s, u)$, each achieve mixing to a stationary distribution $d_u(s, a)$ with mixing time $t_{mix,u}$. Define the overall mixing time of the mixture of MDPs to be $t_{mix} := \max_u t_{mix,u}$.

Assumption 7 (Model Separation, from Kausik et al. [2022])**.** There exist $\alpha, \Delta > 0$ so that for each pair u_1, u_2 of confounders, there exists a state action pair (s, a) (possibly depending on u_1, u_2) so that the stationary distributions under each confounder $d_{u_1}(s, a), d_{u_2}(s, a) \geq \alpha$ and $\|\mathbb{P}^{(u_1)}(\cdot \mid s, a) - \mathbb{P}^{(u_2)}(\cdot \mid s, a)\|_2 \geq \Delta$.

Assumption 8 (Concentrability and Exploration, from Yin and Wang [2020])**.** For $d_m := \min\{d_h^{\pi_b}(s) \mid d_h^{\pi_e}(s) > 0\}$, $d_m > 0$, and there exist constants τ_a and τ_s so that for all s, a, h

$\frac{d_h^{\pi_e}(s)}{d_h^{\pi_b}(s)} \leq \tau_s$ and $\frac{\pi_e(a|s)}{\pi_b(a|s)} \leq \tau_a$.

We can therefore leverage the work of Kausik et al. [2022] to achieve exact clustering with enough data under Assumptions 4, 6, and 7, recovering the unobserved global confounder u_n in each trajectory up to permutation[2]. Then, when using the estimator from Yin and Wang [2020] under Assumption 8, we obtain the following guarantee.

Theorem 4.2.7 (Sample Complexity for OPE under Global Confounding)**.** *Under Assumptions 4, 6, 7, 8, there are constants H_0, N_0 depending polynomially on $\frac{1}{\alpha}, \Delta, \frac{1}{\min_u P(u)}, \log(1/\delta)$, so that for n trajectories of length $H \geq H_0 t_{mix} \log(n)$, we have that $|\hat{V}_1(s_0;\pi_e) - V_1(s_0;\pi_e)| < \epsilon$ with probability at least $1-\delta$ if $n \geq \Omega(\max(n_1, n_2, n_3, n_4))$, where*

$$n_1 := U^2 S N_0 \log(1/\delta), n_2 := \frac{\log(U/\delta)}{\min(\epsilon^2/H^2, \min_u P(u)^2)}$$
$$n_3 := \frac{H^2 \tau_a \tau_s S A \log(U/\delta)}{\epsilon^2}, n_4 := \frac{\tau_a H}{d_m}$$

The first term represents the sample complexity for exact clustering [Kausik et al., 2022], the second term corresponds to estimating $P(u)$ accurately and the third and fourth come from the sample complexity of the OPE estimator [Yin and Wang, 2020]. In Appendix B.8, we prove a more general version of this theorem, where the OPE estimator makes an assumption $A(b)$ depending on a parameter vector b and has sample complexity $N_2(\delta, \epsilon, b)$. Results analogous to Theorem 4.2.7 can thus be produced using Corollary 1 of Duan and Wang [2020a], or other off-policy estimators listed in section 2 of Zhang et al. [2022] viewed in a tabular setting. This is the first result that provides sample complexity guarantees for consistent point estimates under confounding. Theorem B.9.1 in Appendix B.9 shows that requiring that $H \geq \Omega(t_{mix})$ in Theorem 4.2.7 is unavoidable, even for small $t_{mix} = O(\log(S))$.

4.2.7 Policy Optimization under Confounding

We first make an elementary observation that given a bound on the OPE error $|\hat{V}_1(\pi) - V_1(\pi)|$ and an optimizer for the value estimate $\hat{\pi}^* \in \arg\max \hat{V}_1(\pi)$, we can obtain a sub-optimality bound for $\hat{\pi}^*$. We show this explicitly in Appendix B.10, noting that this is agnostic to the existence and the nature of confounding.

Policy Gradients on Lower Bounds under Memoryless Confounding. Recall that in Section 4.2.3, we produced lower bounds on the value function under memoryless confounding with

[2]They recover clusters, which is sufficient as we only need to know confounders up to renaming the labels.

a sensitivity model. In lieu of optimizing a point estimate of the policy's value, we can instead improve this lower bound.

Recall that Algorithm 6 computes a lower bound on $V_1(s_0)$ by projected gradient descent. We can backpropagate gradients relative to the evaluation policy, improving the lower bound on $V_1(s_0)$, and therefore the policy, with gradient ascent. We present the case with stationary transition structures in the max-min formulation below in the interest of lucidity, noting that it immediately generalizes to non-stationary transition structures as well.

$$\max_{\theta \in \Theta} \min_{\mathbb{P} \in \mathcal{G}} V_1(s_0; \pi_\theta, \mathbb{P}) \tag{4.3}$$

We repeat the alternating process of finding $\mathbb{P} \in \mathcal{G}$ to minimize $V_1(s_0)$ given an evaluation policy π_θ and then performing a gradient ascent update on π_θ. This is illustrated in Algorithm 8 below.[3] We discuss local convergence guarantees for the method in Appendix B.10.

Algorithm 8 Gradient Ascent on Differentiable Lower Bounds for Policy Improvement under Confounding

1: **input:** decaying learning rate η_t, π_θ.
2: **for** $t = 1, ..., N$ **do**
3: **run subroutine:** obtain differentiable lower bound $V_1(s_0; \pi_\theta)$ on π_θ via Alg. 6, Alg. 17, or Alg. 5
4: **update:** $\theta \leftarrow \theta + \eta_t \cdot \nabla_\theta V_1(s_0; \pi_\theta)$
5: **end for**
6: **return** π_θ

Policy Gradients under Global Confounding. Recall that we hope to solve $\arg\max_{\pi_e} V_1(s_0; \pi_e)$, where $V_1(s_0; \pi_e) = \sum_u P(u) V_1(s_0; u; \pi_e)$, for confounder-unaware evaluation policy π_e. This is the Weighted-Value Problem in [Steimle et al., 2021], which is NP-hard according to Proposition 2 in their paper.

We discuss a policy gradient method for this problem. Let $Z(\theta) := \nabla_\theta V_1(s_0; \pi_\theta)$. By Assumption 4, $Z(\theta) = \nabla_\theta \mathbb{E}_u[V_1(s_0; u, \pi_\theta)] = \nabla_\theta \sum_u P(u) V_1(s_0; u, \pi_\theta) = \sum_u P(u) \nabla_\theta V_1(s_0; u, \pi_\theta)$. Therefore, if we have gradient estimates $\hat{Z}_i(\theta)$ of $Z_i(\theta) = \nabla_\theta V_1(s_0; u_i, \pi_\theta)$ for each cluster, we can obtain the final policy gradient estimate as a weighted sum, given by $\hat{Z}(\theta) = \sum_{u=1}^{U} \hat{P}(u_i) \hat{Z}_i(\theta)$. We present this as Algorithm 9 below.

[3] Given libraries like `cvxpylayers`, we can also perform gradient ascent on any lower bound from differentiable convex optimization. This includes the lower bounds generated by the relaxation of the model-based algorithm (Alg. 17) and CFQE (Alg. 5). We state general lemmas that back our claims.

Algorithm 9 Clustering-Based Policy Gradient

input: Number of clusters U, clustering algorithm `cluster()`, offline policy gradient estimator `gradient()`, learning rate η, initial policy parameters θ_0.
run subroutine: Perform clustering on trajectories with clustering algorithm `cluster()`, obtain clusters $C_1, ..., C_K$.
Obtain cluster weight estimates $\hat{P}(u) := \frac{|C_u|}{N_{traj}}$.
for $t = 1, ..., T$: **do**
    run subroutine: Use offline policy gradient estimator `gradient()` to estimate $Z_i(\theta_t) = \nabla_\theta V_1(s_0; u_i, \pi_{\theta_t})$ for each cluster C_i, obtaining $\hat{Z}_i(\theta_t)$.
    Obtain gradient estimate of $Z(\theta_t) = \nabla_\theta V_1(s_0; \pi_{\theta_t})$ with $\hat{Z}(\theta_t) = \sum_{u=1}^{U} \hat{P}(u_i)\hat{Z}_i(\theta_t)$.
    Update $\theta_{t+1} := \theta_t - \eta\hat{Z}(\theta_t)$.
end for
return: Output the final policy $\pi_{\theta_{T+1}}$.

We then perform standard gradient descent for T iterations on the policy parameters θ, with the update rule given by $\theta_{t+1} = \theta_t - \eta\hat{Z}(\theta_t)$. In analyzing this procedure, we instantiate $\hat{Z}_i$ using the (statistically) Efficient Off-Policy Policy Gradient (EOPPG) estimator from Kallus and Uehara [2020], which enjoys an $\Theta(H^4/n)$ MSE guarantee instead of the $2^{\Theta(H)}\Theta(1/n)$ worst-case sample complexity of REINFORCE [Kallus and Uehara, 2020]. We assume that the gradient of V_1 is bounded by L, which holds if V_1 is L-Lipschitz. Additionally, let assumptions for Theorem 12 in Kallus and Uehara [2020] hold. We obtain a bound on the norm of the policy gradient that shows convergence to a stationary point in Theorem 4.2.8 below. It is proved in Appendix B.11.

Theorem 4.2.8. *Let us have large enough* $\beta > 1$ *and* $T = n^\beta$, *for* $n \geq \Omega\left(\max\left(U^2SN_0\log(1/\delta), \frac{\log(U/\delta)}{\min_u P(u)^2}\right)\right)$. *Also let* $H \geq H_0 t_{mix}\log n$, *for* H_0, N_0 *as in Theorem 4.2.7. Then we have that* $\frac{1}{T}\sum_{t=1}^{T}||\nabla_\theta V_1(s_0;\pi_{\theta_t})||^2 = O(\max(\epsilon_{MSE}, \epsilon_{freq})$, *where* $\epsilon_{MSE} = \frac{H^4\log(nU/\delta)}{n\min_u P(u)}$, *and* $\epsilon_{freq} = \frac{L^2\log(U/\delta)}{n}$

4.3 Numerical Experiments

We detail our experiments for memoryless and global confounders below. The code for all experiments can be found at https://github.com/hetankevin/off-policy.

Gridworld for Memoryless Confounders. We examine the performance of the methods in Section 4.2.3 on the 4x4 gridworld environment used by Bruns-Smith [2021], with i.i.d. (and thus memoryless) confounders. We implement the model-based method and its variations using the point estimates $\hat{\mathbb{P}}^{\pi_b}$ instead of Hoeffding confidence intervals for $\mathbb{P}^{\pi_b}$, for a fair comparison with CFQE. The horizon is $H = 8$, and Γ ranges from 1 to 50. The sensitivity parameter Γ is a global bound applied uniformly across all state-action pairs, as in Assumption 5. The confounders are drawn

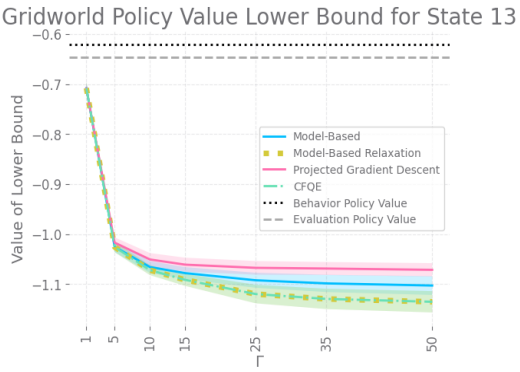

Figure 4.1: OPE for Memoryless Confounders. Comparison of our model-based method, its non-stationary relaxation (Alg. 17), its projected gradient descent variant (Alg. 6), and CFQE on state 13 in a 16-state gridworld. Confidence intervals (CIs) are one standard deviation wide and computed over 30 trials. $H = 8$.

i.i.d. at each timestep and affect the transition kernel but not the reward. The behavior policy π_b and evaluation policy π_e are the same as in Bruns-Smith [2021]; we refer the reader there for their precise construction. We plot the policy values against Γ in Figure 4.1. Across all 16 states, the model-based method's lower bound is always either as good as or tighter than that of CFQE, but the gap in performance is seen most starkly in state 13 (which we display in Figure 4.1). The output of FQE is obtained at $\Gamma = 1$ and is at most -0.7. By the remark after the proof of Theorem 4.2.2, the naive lower bound obtained using FQE is less than $-0.7 - \frac{\varepsilon H^2}{2} = -0.7 - 32\varepsilon$. This is quite literally "off-the-chart" here, showing that using FQE for lower bounds would be ineffective in practice. Note that our model-based method gives the closest lower bound after projected gradient descent. Projected gradient descent only *approximately* solves the appropriate optimization problem, and it is thus not guaranteed to return a true lower bound. So, our model-based method is empirically the best method here with guarantees.

We also study policy improvement. Figure 4.2 displays the training dynamics and convergence of Algorithm 8, where we perform gradient ascent on a lower bound obtained by Algorithm 6. We visualize the learned policy, which is appropriately conservative: on a horizon of 8, the agent will likely not reach the goal state from the first few states and move to the top left corner appropriately. Finally, we plot the increase in the lower bound on policy value against progressing gradient ascent iterations, starting at π_e. Note that even our lower bounds all eventually exceed the true (ground truth) values of π_b and π_e, displaying improvement.

Sepsis Simulator for Global Confounders. We examine the performance of the method of Algorithm 7 on the sepsis simulator of Oberst and Sontag [2019], especially in terms of the choice

Figure 4.2: Policy Improvement for Memoryless Confounders. Top Left: Loss curve dynamics of max-min gradient descent. Top Right: Resulting policy $\hat{\pi}^*$ for $\Gamma = 10$ in 4x4 gridworld with actions indexed by WENS. Brighter colors indicate higher $\hat{\pi}^*(a \mid s)$. Bottom: Increase in the lower bound on $V_1^{\pi_\theta}$ as gradient ascent iterations progress. $H = 8$.

of the clustering algorithm. Once we hide the diabetes status of each patient, it becomes a global confounder. The confounder-aware behavior policy is the same behavior policy in [Oberst and Sontag, 2019], and the evaluation policy is $\pi_e := \frac{1}{U} \sum_u \pi_b(a|s,u)$. In the simulator, glucose levels are generated i.i.d, with their distribution determined by the presence or absence of diabetes. This makes them easy proxies for diabetes, so we hide glucose levels during the clustering phase to make the clustering problem harder.

On the top left of Figure 4.3, we compare the clustering error for the method of Kausik et al. [2022] with that of classical soft EM with random initialization. In the top right, we plot a measure of the relative error in OPE against trajectory length. The relative error is computed as $\frac{\max_s |\hat{V}_1^{\pi_e}(s) - V_1^{\pi_e}(s)|}{\max_s |V_1^{\pi_e}(s)|}$. The plot compares the performance of Algorithm 7 instantiated with FQE coupled with either soft EM with random initialization or the method of Kausik et al. [2022]. At the bottom, we show the convergence of Algorithm 9, instantiated using the off-policy policy gradient variant that Kallus and Uehara [2020] attributes to Degris et al. [2013]. We compare the same possibilities for clustering as above. We observe that in general, the method of Kausik et al. [2022] outperforms randomly initialized soft EM, allowing for both OPE and policy improvement. Our experimental results highlight the effectiveness of our method as well as the importance of the clustering algorithm.

4.4 Conclusion and Future Work

We have provided a broad, structured view of the landscape of confounded MDPs, studying the OPE and OPI problems under various confounding assumptions. The paper has discussed existing

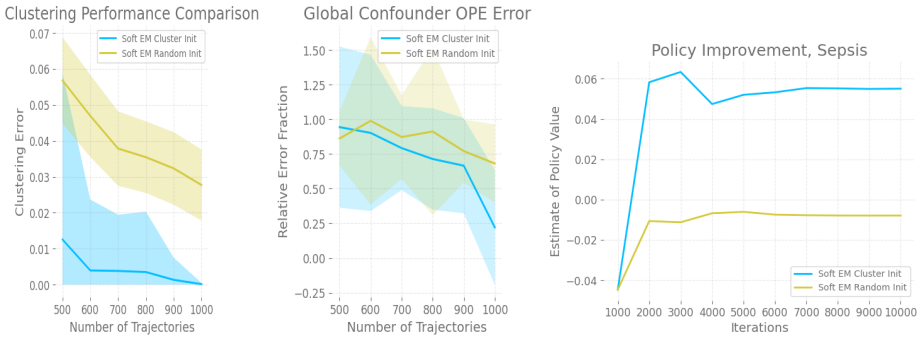

Figure 4.3: Top Left: Average performance of the clustering method from Kausik et al. Top Right: Average relative error of clustering-based OPE with different clustering algorithms. Bottom: Improvement in estimates of policy values under gradient ascent coupled with different clustering algorithms, see Appendix B.1 for details. We average over 30 trials, confidence intervals are 1 standard deviation wide. $H = 60$.

methods, presented new ones and provided theoretical and empirical grounding for the methods. We hope that the insights here will springboard further work on confounded MDPs. In particular, while we address the sensitivity assumption, a big-picture view of other assumptions like bridge functions and instrumental variables is needed. For general confounders with memory, note that while Theorem 4.2.6 rules out FQE and related methods, other methods must be explored. There are also specific structures on confounders with memory, besides global confounders, that can be formulated and studied. Finally, many of our methods (such as the gradient-based methods presented) can be extended to handle continuous state spaces via function approximation. Shi et al. [2021] provide methods under assumptions on the existence and learnability of bridge functions, being one of the first works to address this. However, work on confounding with continuous state and action spaces is still relatively sparse, and is an exciting setting to explore.

CHAPTER 5

A Theoretical Framework for Partially-Observed Reward States in RLHF

The previous chapter studied offline RL in settings where an unobserved latent variable affects the *transition dynamics* of the environment. A complementary and equally important case arises when latent variables instead affect *rewards*. This is precisely what occurs in reinforcement learning from human feedback (RLHF): the human evaluator's internal state, such as their mood, sentiment, or fatigue, is a latent variable that influences the feedback they provide but does not change the environment's transitions. In the running example of Section 1.2.2, this corresponds to the setting where an LLM generates personalized messages for interns, and the intern's internal state at the moment of receiving a message shapes the rating they give. In the language of Section 1.4, the latent variable is confined to the reward channel, and within that channel its complexity is captured by the eluder and coverability dimensions of the reward function class rather than the full $(SA)^{\Omega(H)}$ history space. In this chapter, we introduce a framework that generalizes current RLHF models by explicitly incorporating partially-observed internal states and intermediate feedback, and design algorithms whose guarantees exploit this confinement.

5.1 Introduction

This chapter is a lightly edited version of Kausik et al. [2024b].

As automated systems become more ubiquitous, the need to understand how to align their objectives with the needs of humans that interact with them has become increasingly important [Ji et al., 2023]. The development and study of reinforcement learning from human feedback (RLHF) has been an important way of formalizing these problems and design methods for alignment [Wirth et al., 2017]. RLHF is concerned with the study of how to find a policy that maximizes an objective defined in terms of human labeled data in an RL domain [Christiano et al., 2017].

Many RLHF methods entail learning a reward function from human data, and then using

the learned reward function as an input to a traditional reinforcement learning algorithm such as PPO [Schulman et al., 2017]. These reward-based RLHF methods have been pivotal in the development of several technologies such as robotics [Brown et al., 2019, Shin et al., 2023], recommender systems [Xue et al., 2022], and the training of large language models (LLMs) [Ouyang et al., 2022].

It is important to emphasize that reward-based RLHF is not limited to preferential feedback. In fact, there exist two dominant kinds of feedback, namely *cardinal* and *dueling* feedback. Cardinal feedback requires the human labeler to provide a single label over an entire trajectory of interaction between the agent and the environment. Dueling feedback requires the human to specify a preference between two trajectories. In practice, cardinal feedback has been used for LLM alignment algorithms like KTO [Ethayarajh et al., 2024], while dueling feedback has been used in algorithms like DPO [Rafailov et al., 2023] and PPO-RLHF [Ouyang et al., 2022]. Past theoretical work [Chatterji et al., 2021, Wang et al., 2023b, Saha et al., 2023] has designed algorithms for both cardinal and dueling feedback under various metrics – standard/cardinal regret, sample complexity or dueling regret.

We observe that current models of reward-based RLHF assume a very specific model of non-Markovian rewards. Modeling rewards as non-Markovian is natural, since human responses to stimuli are known to be affected by partially-observed and evolving "internal states" [Flavell et al., 2022]. For example, when a human reads a piece of text (possibly generated by an LLM), their assessment may oscillate between opposing sentiments in different parts of the text. Unfortunately, current models do not explicitly incorporate such "internal states" that affect rewards, and are limited to a specific linear model of rewards. While one can incorporate internal states using naive history-summarization, i.e. by treating the entire trajectory $\tau[h]$ so far as the state, we show below that better general algorithms can be designed with improved guarantees.

Additionally, current models assume that feedback is received only once at the end of an episode or pair of episodes. In many applications such as robot motion [Lee et al., 2021] and mathematical reasoning [Uesato et al., 2022], correctly incorporating intermediate or "snippet-level" feedback can speed up learning as well as improve alignment. With this in mind, we ask the following questions:

How do we generalize the RLHF setting to incorporate internal states and intermediate feedback? What algorithms and guarantees can improve over naive history-summarization here?

Contributions:

- **Introducing PORRL:** In Section 5.2, we introduce PORRL, which generalizes current RLHF models to incorporate "internal states" and intermediate feedback.
- **Improving over naive history-summarization (model-based algorithms):** In Section 5.3.1,

we design model-based optimistic algorithms that, POR-UCRL and POR-UCBVI, achieving a regret of $\widetilde{\mathcal{O}}((poly(H,S,A) + p\sqrt{d_E d_C})\sqrt{T})$ and a sample complexity of $\widetilde{O}((poly(H,S,A)/\epsilon^2 + p^2 d_E d_C/\epsilon^2)$ under minimal assumptions.[1] The $poly(H,S,A)$ term would be $(SA)^{\Omega(H)}$ under naive history-summarization. We show that our guarantees subsume and improve over past results.

- **Leveraging recursive structure on internal states (model-free algorithms):** In Section 5.3.2, we study the model-free algorithm GOLF, applied using history-summarization. We define a new "history-aware" notion of dimension, d_{HABE} and show that GOLF has regret $\widetilde{\mathcal{O}}(pH\sqrt{d_{\mathrm{HABE}} d_C T})$. We show using an example that when internal states have a recursive structure, our guarantee can be exponentially smaller than existing guarantees and guarantees for our model-based methods.
- **Reduction from Dueling to Cardinal PORRL:** In section 5.4, we show that a naive blackbox reduction from dueling to cardinal PORRL always fails. We design a whitebox reduction from dueling PORRL to a large class of optimistic algorithms for cardinal PORRL. To the best of our knowledge, this is the first explicit reduction from cardinal to dueling regret guarantees for MDPs.
- **Practical Implications:** While the aim of our work is largely theoretical, we extract practical insights from our results throughout the text. These are summarized in section 5.5.

5.1.1 Related Work

RLHF. RL with human preferences has a long history [Akrour et al., 2012, Busa-Fekete and Hüllermeier, 2014, Sadigh et al., 2017]. It has been successfully used in disparate domains such as robotics, games, and LLMs. The problem of learning from cardinal feedback has been theoretically studied in [Efroni et al., 2021, Chatterji et al., 2021]. Theoretical guarantees for utility-based preferential (dueling) feedback can be found in [Novoseller et al., 2020, Saha et al., 2023, Chen et al., 2022b, Zhan et al., 2023]. The non-Markovian nature of the optimal policy under these RLHF models contributes greatly to why the problem is harder than traditional RL. Recent work has studied RLHF with heterogeneous user populations, where a latent user type determines preferences [Poddar et al., 2024]. When the latent type is fixed per episode and affects rewards but not transitions, this falls within our framework: it is an instance of the latent bandit model of Chapter 6 extended to dueling preferences.

Internal states and intermediate feedback. There is evidence in neuroscience research indicating that human responses to stimuli are affected by "internal states" — partially hidden variables that profoundly shape perception, cognition, and action" [see Flavell et al., 2022]. Despite

[1] d_E is a relevant eluder dimension and d_C is a relevant covering dimension. We are working under general function-approximation for the *reward* model. It is straightforward to also add general function-approximation for the *transition* model, abstracting out the S, A dependence. See remark 8 and Appendix C.6.

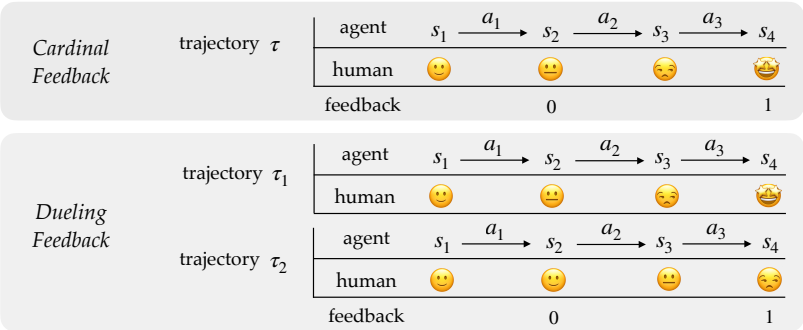

Figure 5.1: Illustrating how a human's internal states (represented by emojis) affect their feedback to an agent or LLM. Top: Cardinal or good/bad feedback. Bottom: Dueling or preferential feedback. In line with Definition 5.2.1, $u_h \in \mathcal{U}$ are represented by the emojis, $p = 2$ and $\mathcal{H}_p = \{2, 4\}$ in both cases.

not explicitly recognizing the phenomenon of human internal states, several works in RLHF incorporate richer forms of feedback. For example, Wu et al. [2023] consider human labeling over sub-sections of the text. In work on process supervision [Uesato et al., 2022, Lightman et al., 2023], humans give feedback on intermediate steps. Motivated by these, our work aims to lay out the groundwork for a theoretical treatment of internal human states and intermediate feedback in RLHF, using partially observed reward-states. The interaction-grounded learning framework [Xie et al., 2021] studies settings where the agent observes feedback but not rewards directly, requiring a "reward-decodable" condition: feedback is conditionally independent of the context and action given the reward. This is structurally related to Assumption 9, where feedback depends on reward through deterministic decoder functions g_h and emission distributions e_h.

Partial observability in RL. The problem of partial observability in RL is not new. Although learning in POMDPs [Åström, 1965] is known to be statistically intractable in general [Krishnamurthy et al., 2016, Jin et al., 2020], a flurry of recent works have studied POMDPs under various structural assumptions [Du et al., 2019, Liu et al., 2022a,b, Golowich et al., 2022, Zhan et al., 2022, Cai et al., 2022, Chen et al., 2022a, 2023, Wang et al., 2023a, Zhong et al., 2023]. Our model is distinct from POMDPs since our results do not require the latent state evolution to be Markovian, but assumes Markovian transitions for observed states. See Section 5.2.3 for a discussion.

5.2 Defining RL with Partially-Observed Reward States (PORRL)

In this paper, we consider an episodic reinforcement learning setting in which a learner interacts with an MDP having a state space $\mathcal{S}$, an action space $\mathcal{A}$, transitions dynamics $\mathbb{P}$, and episode length

H. At each time-step $h \in [H]$ of an episode, the learner observes the state s_h and takes an action a_h, generating a *trajectory* $\tau = (s_1, a_1, \cdots, s_H, a_H) \in \Gamma$, where Γ denotes the space of trajectories.[2] In a typical RLHF setting, the learner observes a human feedback $o_H \in \mathcal{O}$ at the end of the episode, which is associated to but potentially different from a reward $r : \Gamma \to \mathcal{R}$ encoding the task. We now describe how internal states and intermediate feedback shall be incorporated in the latter RLHF framework through a guiding example, and we use this to formally introduce the PORMDP model.

5.2.1 PORMDPs

Let us consider the example of a human interacting with a language model, as in Figure 5.1. Here, an action is a token, the state is the text so far, and the reward is some score representing the human's satisfaction, which induces stochastic feedback. The internal states could be the human's emotional reaction to the text (e.g., happy, frustrated, or amused), or numbers in $[0, 1]$ encoding a confidence level that the text is progressing towards a coherent response. While an agent goes through a sequence of states and actions, the system (i.e., the human) progresses through internal states, which inevitably affect, together with agent's actions and the state of the process, the human's satisfaction.

Formally, this can be modeled by introducing internal states $u \in \mathcal{U}$ and defining the set of *underlying* histories Γ^u_{h-1} that incorporate internal states by $\Gamma^u_{h-1} := \{\tau^u[h-1] = \{(s_l, u_l, a_l)\}_{l=1}^{h-1} \mid s_l \in \mathcal{S}, a_l \in \mathcal{A}, u_l \in \mathcal{U}\}$. We model the dynamics of internal states by saying that there exists an internal state generator $w_h : \Gamma^u_{h-1} \times \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{U})$ so that the human's internal state u_h is sampled from the distribution defined by $w_h(\tau^u[h-1], s_h, a_h)$. The human's satisfaction at time h should then be a function of the current state and action, but also the current internal state, given by $r_h(s_h, u_h, a_h)$.

The agent does not observe the reward r_h directly, but a feedback o_h depending on r_h. Typically, o_h will be $\{0, 1\}$ feedback reflecting whether the human says that they are satisfied or not. In general, this could be stochastic. For instance, this could be $Ber(\sigma_h(r_h))$ for some function σ_h. So, $o_h \sim e_h(r_h)$ for some distribution $e_h(r_h)$. This leads to the general definition below, where we have introduced new objects $\mathcal{U}, \mathcal{H}_p, w, e$ not seen in traditional RL:

Definition 5.2.1. A PORMDP $\mathcal{M}$ with *cardinal feedback* is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{U}, \mathcal{O}, \mathbb{P}, \mathcal{H}_p, r, w, e)$, where:

- $\mathcal{S}, \mathcal{A}$ are fully observable states and actions, $\mathcal{U}$ are *unobserved internal reward-states*, $\mathcal{O}$ is a space of feedback, $\mathbb{P}(\cdot \mid s, a)$ is a Markovian transition matrix, $s_1 \in \mathcal{S}$ is an initial state.[3]

[2] We will further denote $\tau[h] = (s_1, a_1, \ldots, s_h, a_h)$ the sub-trajectory of τ of length h and Γ_h the corresponding space of sub-trajectories of length h.

[3] Recall that choosing a formal state s_1 to serve as a placeholder initial state is not restrictive.

- $\mathcal{H}_p \subset [H]$ is a set of timesteps where reward and feedback is obtained with size $|\mathcal{H}_p| = p$.
- $r := \{r_h\}_{h \in \mathcal{H}_p}$ so that $r_h : \mathcal{S} \times \mathcal{U} \times \mathcal{A} \to \mathcal{R}$ are reward functions at time h.
- $w := \{w_h\}_{h \in \mathcal{H}_p}$ so that $w_h : \Gamma^u_{h-1} \times \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{U})$ are *internal state generators* that map underlying histories of (s, a, u) tuples to distributions over $\mathcal{U}$.
- $e := \{e_h\}_{h \in \mathcal{H}_p}$ are *feedback functions* so that the feedback $o_h \sim e_h(r_h)$ is sampled from an η_h-subgaussian distribution e_h with mean $\sigma_h(r_h)$ for some activation function $\sigma_h : \mathcal{R} \to \mathcal{R}$.[4]

In some relevant RLHF applications, the human is presented with two trajectories and they provide feedback based on the pair. In most cases, this involves indicating a 0-1 preference between trajectories. To accommodate this setting, we extend the framework to dueling feedback.

Definition 5.2.2. A PORMDP $\mathcal{M}$ with *dueling feedback* is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{U}, \mathcal{O}, \mathbb{P}, \mathcal{H}_p, r, w, e)$, where everything is identical to Definition 5.2.1, except that every episode now involves running two trajectories τ_1, τ_2 that produce rewards $r_{h,1}, r_{h,2}$ $\forall h \in \mathcal{H}_p$, and feedback is distributed as $o_h \sim e_h(r_{h,1} - r_{h,2})$. We note that PORMDPs subsume and model a wide class of RL settings, including RLHF. A brief list of settings that PORMDPs subsume is as follows: (i) traditional MDPs, by setting $\mathcal{U} = \{\star\}$; (ii) existing linear models of RLHF, setting $\mathcal{U} = \{\phi(\tau)^\top \mathbf{w}\}$ for a known feature map ϕ and unknown $\mathbf{w}$ [Chatterji et al., 2021, Efroni et al., 2021, Saha et al., 2023, Wang et al., 2023b]; (iii) learning reward models with stochastic feedback by setting $\mathcal{U}$ to be the set of reward states [Icarte et al., 2019, 2018, 2022, Icarte, 2022]. By using $\mathcal{U}$ to model implicit intentions, PORMDPs can also model learning from the following feedback: (iv) process supervision [Lightman et al., 2023, Uesato et al., 2022], (v) fine-grained feedback [Wu et al., 2023] and (vi) snippet-level feedback [Lee et al., 2021]. Further, one can show that in all these settings, we can define the $\mathcal{U}$ generators w_h to be deterministic.

A natural attempt is to avoid modeling $\mathcal{U}$ explicitly by marginalizing out the internal states, defining a non-Markovian reward $\bar{r}_h(\tau[h], a_h) = \mathbb{E}_{u_h \sim g_h(\tau[h])}[r_h(s_h, u_h, a_h)]$. The resulting reward depends on the full history through the induced distribution over u_h, but this is precisely the history-summarization approach whose policy class has size $(SA)^{\Omega(H)}$. Explicitly modeling the internal state structure, as PORMDPs do, is what allows the complexity to scale with the eluder or HABE dimension of the reward class rather than the full history space.

One illustrative hard example of PORRL is that of a *combination lock,*[5] which we will also use later in the paper. Consider an H-digit numerical lock with a set $\mathcal{A}$ of options at each digit. Let the true combination be $a_1^\star, \dots a_H^\star$. An agent tries to unlock it by listening for "clicks" while rotating

[4]This subsumes and generalizes the example of Bernoulli feedback in RLHF.

[5]This is a variant of a common example used to generate lower bounds in POMDPs [Krishnamurthy et al., 2016, Jin et al., 2020]. In contrast, we will use it to illustrate the power of our upper bounds.

the dial at each digit h. Naturally, we only hear clicks at digit h if the entire combination so far is correct. We thus model this as a PORMDP with non-Markovian rewards, $\mathcal{S} = \{\star\}, \mathcal{U} = \{\bigcup_h \mathcal{A}^h\}$ and the appropriate dynamics. Arguing that the click might sometimes be too faint, we consider stochastic rewards. Specifically, we model this as $r_h(s_h, u_h, a_h) = Ber(q\mathbb{1}_{a_1^\star, \ldots a_h^\star}(u_h))$ for some uncertainty parameter q. Notice that the internal states have a recursive structure here, and they evolve in a Markovian way. This is a toy model for the problem of learning to take desirable sequences of actions using intermediate feedback. It can be viewed as a simplified version of many such tasks – navigating mazes, writing structured essays with guidance, writing a proof with feedback on correctness.

5.2.2 Reinforcement Learning in PORMDPs (PORRL) with Cardinal and Dueling Feedback

Due to the complex nature of observability in our problem, we will use this subsection to carefully set up a meaningful set of RL problems, in which an agent interacts with a PORMDP to optimize a policy. At each step h, the agent observes a history $\tau[h-1] \in \Gamma_{h-1}$ and takes an action $a_h \sim \pi(\tau[h-1], s_h)$. The agent does not observe the reward r_h, but receives an observation $o_h \sim e_h(r_h)$.

Defining the learning objective. Since rewards are partially observed and dependent on the entire history, there is a subtlety in defining value functions. We first choose and fix some subclass Π of history-dependent policies and we define the total expected reward of a policy $\pi \in \Pi$ as

$$V_w(\mathcal{M}, \pi) := \mathbb{E}_{\tau^u \sim \mathbb{P}^{w,\pi}}\left[\sum\nolimits_{h \in \mathcal{H}_p} r_h(s_h, u_h, a_h)\right]$$

$V_w(\mathcal{M}, \pi)$ is taking an expectation over the dynamics of *underlying* trajectories $\tau^u = \{(s_h, u_h, a_h)\}_{h=1}^H \sim \mathbb{P}^{w,\pi}$. Since the states u are never revealed, these dynamics can never be learnt, making V_w hard to directly deal with. In this light, we introduce stochastic functions $g_h : \Gamma_h \to \Delta(\mathcal{U})$ that marginalize the internal state generator w_h over the sequence $u_1, \ldots u_{h-1}$. That is, given an (s, a) history $\tau[h]$, we can define[6] $g_h(\tau[h]) \sim u_h \mid \tau[h]$. Now define

$$V_g(\mathcal{M}, \pi) := \mathbb{E}_{\tau \sim \mathbb{P}^\pi}\left[\sum\nolimits_{h \in \mathcal{H}_p} \mathbb{E}_{u_h \sim g_h(\tau[h])}[r_h(s_h, u_h, a_h)]\right]$$

$V_g(\mathcal{M}, \pi)$ is a much more tractable object, where the outer expectation is taken over the dynamics

[6]More technically, define $g_h(\tau[h])$ to be the *regular conditional distribution* of the random variable $w_h((\tau[h-1], u_1, \ldots u_{h-1}), s_h, a_h)$, conditioned on $\tau[h]$.

of the *observed* trajectories τ. The following result establishes that as one would hope, $V_w = V_g$.

Lemma 5.2.1 (Replacing w with g)**.** *For any history-dependent policy π that selects an action $a_h \sim \pi(\tau[h-1], s_h)$, $V_w(\mathcal{M}, \pi) = V_g(\mathcal{M}, \pi)$ holds for any $\mathcal{M}$.*

For the purposes of value functions, $\mathcal{M}$ is fully specified by $(\mathcal{S}, \mathcal{A}, \mathcal{U}, \mathcal{O}, \mathbb{P}, \mathcal{H}_p, r, g, e)$. Henceforth we replace w with g and denote the value function $V_g(\mathcal{M}, \pi)$ by $V(\mathcal{M}, \pi)$. Define the optimal policy as $\pi_\star := \arg\max_{\pi \in \Pi} V(\mathcal{M}, \pi)$.

Cardinal PORRL. Consider an algorithm producing a sequence of policies $\pi_1, \ldots, \pi_T \in \Pi$, where π_t is chosen only using trajectories $\{\tau_i\}_{i=1}^{t-1}$ generated by $\{\pi_i\}_{i=1}^{t-1}$. We measure the performance of such an algorithm by its *cardinal regret* under model $\mathcal{M}_\star$:

$$\mathrm{Regret}(T) = \sum_{t=1}^{T} V(\mathcal{M}_\star, \pi_\star) - V(\mathcal{M}_\star, \pi_t)$$

One can also ask for the sample complexity of learning a good policy. Given a randomized algorithm that completes N episodes of interaction and outputs π_N, the *sample complexity* $N(\epsilon, \delta)$ of the algorithm is the minimum N so that $V(\mathcal{M}_\star, \pi_\star) - V(\mathcal{M}_\star, \pi_N) \leq \epsilon$ with probability at least $1 - \delta$ over the randomness of the feedback and the algorithm. It makes sense to study cardinal regret and sample complexity in two RLHF settings:

- Using a learnt reward model: In most deployments of offline RLHF, an offline dataset of dueling feedback from humans is typically used to create a cardinal feedback oracle (a reward model), which is then used to train the policy using RL. In fact, Lightman et al. [2024] do exactly this under our model. The sample complexity of the algorithm is important in this setting.
- Improving a deployed model with batched feedback: One can learn from batches of interaction with humans and hope to improve the model/policy adaptively over multiple batches. This is compatible with deploying LLMs or recommender systems to users, collecting a batch of good/bad feedback, and then fine-tuning the model offline using this batch. This approach is also discussed in [Swamy et al., 2024, Dong et al., 2024]. Regret is a better metric than sample complexity here, since we want users to be satisfied (exploiting) while improving the model (exploring). Instead of good/bad feedback, we can also ask for dueling feedback against a fixed π_0 and treat it as cardinal feedback.[7]

Dueling PORRL. In dueling PORRL, we play a *duel* by running two policies $(\pi_1, \pi_2) \in \Pi \times \Pi$ in parallel to obtain trajectories (τ_1, τ_2) and receive feedback $\{o_h\}_{h \in \mathcal{H}_p}$. Again, note that the rewards of the policies are not observed. While the definitions of $V(\mathcal{M}, \pi)$ and $\pi_\star$ are the same as before,

[7] If the activation function is Lipschitz and monotone, then we can get cardinal regret guarantees for this problem by using the difference function class.

we define a new measure of regret accordingly. If we play T duels $(\pi_{1,1}, \pi_{2,1}), \ldots, (\pi_{1,T}, \pi_{2,T})$ according to an algorithm, we aim to minimize the *dueling regret* given by

$$\text{Regret}_D(T) = \sum_{t=1}^{T} V(\mathcal{M}_\star, \pi_\star) - \frac{V(\mathcal{M}_\star, \pi_{1,t}) + V(\mathcal{M}_\star, \pi_{2,t})}{2}$$

It makes sense to consider this metric when improving a deployed model with batched dueling feedback. We can do the same batching as the batched feedback example above, but instead compare our model/policy π_t to a fixed base policy π_0 and ask for dueling feedback. The induced feedback can be treated as cardinal feedback. This is similar to the ideas in Wang et al. [2023b], who consider this setting and give cardinal regret/sample complexity guarantees. However, when deploying a model, we typically want humans to be satisfied with *both* the options they are given. Cardinal regret only accounts for one of the options being good. Dueling regret demands that *both* policies used are good.

Remark 5. PORRL subsumes the settings of [Saha et al., 2023, Chatterji et al., 2021], which in turn subsume the feedback models of RLHF [Wang et al., 2023b]. Crucially, Wang et al. [2023b], Chatterji et al. [2021] measure performance using only sample complexity or cardinal regret, while Saha et al. [2023] only study dueling regret. We have discussed above why both metrics are important.

5.2.3 A General Yet Tractable Case

The nature of the feedback in PORMDPs , which depends on a reward that is function of the entire history, signals that PORRL may be intractable in general. We now instantiate the model into a statistically tractable sub-class that still subsumes most existing work on RLHF and all the examples provided at the end of Section 5.2.1. Specifically, we assume that the internal reward-state functions g_h are deterministic and the feedback is emitted according to a Bernoulli distribution depending on the reward. We will work under this assumption in the remainder of the paper.

Assumption 9. We work in a realizable setting. That is, the unknown transition kernel $\mathbb{P}$ lies in a known class $\mathcal{P}$, and the unknown reward function $r_h : \mathcal{S} \times \mathcal{U} \times \mathcal{A} \to \mathcal{R}$ lies in a known class $\mathcal{R}_h$ with $|r_h| \leq B$ for all h. Assume that g_h is deterministic[8] (but unknown) and belongs to a known class of "decoder functions" $\mathcal{G}_h$. Let $\mathcal{O} = \{0, 1\}$ and let e_h only depend on the rewards. For dueling feedback, let $e_h(r_{h,1} - r_{h,2})$ be η_h-subgaussian with unknown mean $\sigma_h(r_{1,h} - r_{2,h})$. Also assume that σ_h and σ_h^{-1} are Lipschitz with unknown Lipschitz constants $\kappa_{1,h}$ and $\kappa_{2,h}$ respectively. Call the resulting class of PORMDPs $\mathcal{M}$.

[8]We make this assumption for simplicity of exposition, it is not necessary. As long as f_h is η_h subgaussian conditioned on $\tau[h]$, all our theory follows verbatim irrespective of whether g_h is deterministic or stochastic.

We also define a function class induced by $\mathcal{R}_h$ and $\mathcal{G}_h$.

Definition 5.2.3. Let us then consider the decoder-induced function classes $\mathcal{F}_h$ given by

$$\mathcal{F}_h := \left\{ f_h : \Gamma_h \to \mathcal{R} \;\middle|\; \exists g_h \in \mathcal{G}_h, r_h \in \mathcal{R}_h \quad \text{s.t.} \quad f_h(\tau[h]) = r_h(s_h, g_h(\tau[h-1]), a_h),\ \forall \tau \right\}$$

Also define $\mathcal{F} := \prod_{h \in \mathcal{H}_p} \mathcal{F}_h$ so that $f = \{f_h\}_{h \in \mathcal{H}_p} \in \mathcal{F}$. A model $\mathcal{M}$ is then fully determined by $(\mathbb{P}, f)$, so we denote $V(\mathbb{P}, f, \pi) := V(\mathcal{M}, \pi)$. Note that $V(\mathbb{P}, f, \pi) = \mathbb{E}_{\tau \in \mathbb{P}^\pi}\left[\sum_{h \in \mathcal{H}_p} f_h(\tau[h])\right]$.

Remark 6. We note that all examples from Section 5.2.1 work with deterministic dynamics for $\mathcal{U}$ and satisfy Assumption 9.

Giving statistically efficient algorithms for this framework comes with numerous challenges:

- **Traditional RL incurs linear regret:** Any method that outputs Markovian (possibly time-dependent) policies can incur linear regret in a PORMDP.

  Lemma 5.2.2 (Markovian policies are not enough)**.** *There is a* PORMDP *where POR-UCRL and POR-UCBVI achieve $poly(H, S, A)\sqrt{T}$ regret, but any Markovian policy is at least $\frac{1}{4}$-suboptimal and so any method that outputs Markovian (possibly time-dependent) policies will lead to linear regret.*

  The construction uses $\mathcal{S} = \{s_1, s_2\}$, $\mathcal{A} = \{a_1, a_2\}$ with uniform transitions and a reward function that requires following a specific non-Markovian action pattern — play a_2 until s_2 appears, then switch to a_1. The optimal history-dependent policy achieves value 1, but any Markovian policy achieves at most $3/4$ because it cannot condition on whether s_2 has previously appeared. The proof is in Appendix C.1.2.

- **POMDP results do not apply:** PORMDPs cannot be viewed as a subcase of POMDPs with latent states $\mathcal{S} \times \mathcal{U}$ since $s, u, a \to s', u'$ is not Markovian.[9] Even if we considered the subclass of PORMDPs where $s, u, a \to s', u'$ is Markovian, which would be a subclass of reward machines, this is a specific kind of overcomplete POMDP . Literature on overcomplete POMDPs is much more scarce than their undercomplete counterpart. The only paper that gives guarantees for overcomplete POMDPs to our knowledge is Liu et al. [2022a], where they assume that the reward function is *fully* observable and only depends on *observed* states. This cannot apply to our setting, since our rewards have to be *partially* observable, and fundamentally depend on latent states too. Also, this is not a minor difference, since the number of latent states can be $(SA)^{\Omega(H)}$.

- **Naive history-summarization is inefficient:** It is overkill to use naive history-summarization – where one treats the history $\tau[h]$ as the state s_h and executes traditional RL. This is because while policies are non-Markovian, state transitions are Markovian. It is unclear if we can leverage this

[9] Since observed state transitions are Markovian, PORMDPs are also not more general than POMDPs .

structure without running into explicit exponential dependence on H. Moreover, most work on MDPs works with known rewards, but not knowing the rewards is a truly non-trivial problem here, since exploring the reward at each latent state could take $(SA)^{\Omega(H)}$ steps.

- **Ensuring satisfactory utilization of additional structure:** Examples like the combination lock signal that there are intuitive ways to leverage a recursive structure on the internal states. In the combination lock, one should wait for the "click" at each digit before moving onto the next digit, giving us a polynomial dependence on A, H in sample complexity. It is unclear if *general* algorithms for PORRL can implicitly leverage such structure to achieve polynomial guarantees.

If reward and feedback were received only at the final step and depended only on the observed state s_H, the problem would reduce to a standard MDP. It is the dependence of intermediate feedback on the unobserved internal state u_h that couples the reward structure to the full trajectory history and makes the problem hard.

5.3 Optimistic Algorithms for Cardinal PORRL

5.3.1 Improving over Naive History-Summarization with Model-Based Methods

In this section, we present two optimistic methods that leverage Markovian transitions in PORMDPs – POR-UCRL and POR-UCBVI. The methods explicitly learn both the unknown reward model and the unknown transition model, while still accounting for the Markovian nature of transitions. We describe them below and provide formal versions in Appendix C.4, C.5.

- **POR-UCRL:** At each timestep t, we maintain a least squares estimate $\hat{f}^{t+1}$ of f and an MLE estimate $\hat{\mathbb{P}}_t$ and define confidence sets $\mathcal{C}_h^t(\delta)$ that consider all f_h with a small mean squared error against $\hat{f}_h^{t+1}$, such that $\mathcal{C}_{\mathcal{F}}^t(\delta) = \prod_{h=1}^H \mathcal{C}_h^t(\delta)$. The probability transition confidence sets $\mathcal{C}_{\mathcal{P}}^t(\delta)$ are the same as UCRL [Jaksch et al., 2010]. At timestep t, following confidence-set optimism, we play an optimistic policy $\widetilde{\pi}_t$ that maximizes its highest value $V(\mathcal{M}, \pi)$ over all models $\mathcal{M} \in \mathcal{C}_{\mathcal{F}}^t \times \mathcal{C}_{\mathcal{P}}^t$.
- **POR-UCBVI:** It is trickier to adapt ideas from UCBVI [Azar et al., 2017]. Yet again, we maintain a least squares estimate $\hat{f}^t$ and an MLE estimate $\hat{\mathbb{P}}_t$. Instead of confidence sets, we define trajectory-dependent bonuses for $\mathcal{F}$ as $b_{\mathcal{F}}^t(\tau, \delta) := \sum_{h \in \mathcal{H}_p} \max_{f_h, f_h' \in \mathcal{C}_h^t(\delta)} f_h(\tau[h]) - f_h'(\tau[h])$. We use these to define policy-level bonuses for $\mathcal{F}$ as $b_{\mathcal{F}}^t(\mathbb{P}, \pi, \delta) := \mathbb{E}_{\tau \sim \mathbb{P}^\pi}[b_{\mathcal{F}}^t(\tau, \delta)]$. Then, the standard UCBVI bonuses provide policy-level bonuses for $\mathcal{P}$. At timestep t, following bonus-based optimism, we play an optimistic policy $\tilde{\pi}_t$ that maximizes its bonus-boosted value under $\hat{f}^t, \hat{\mathbb{P}}^t$. POR-UCBVI bonuses are in fact computable for many $\mathcal{U}$ and $\mathcal{F}$, such as those in remark 7.

We show that POR-UCRL enjoys the guarantee below.

Theorem 5.3.1 (POR-UCRL Regret)**.** *Under Assumption 9, the regret* $\mathrm{Regret}(T)$ *of POR-UCRL is bounded by the following with probability at least* $1-\delta$

$$\widetilde{\mathcal{O}}\left(\left(pS\sqrt{HA}+\sum\nolimits_{h\in\mathcal{H}_p}\sqrt{d_{E,h}d_{C,h}}\right)\sqrt{T}\right)$$

where $d_{E,h}=\dim_E\left(\mathcal{F}_h,\frac{B}{T}\right)$ *and* $d_{C,h}=\log(\mathcal{N}(\mathcal{F}_h,1/T,\|\cdot\|_\infty))$. Here, the first term comes from uncertainty in $\mathbb{P}$. Under naive history-summarization, the first term would be exponential in H since the modified state space of trajectories would have size $\Omega((SA)^H)$. Similar regret guarantees are given for POR-UCBVI in Theorem C.5.1. Both guarantees are proved by viewing each algorithm as a specific instance of a generic optimistic algorithm for PORRL (see Appendix C.3, C.4, C.5). By a simple regret-to-PAC conversion, we also show that POR-UCRL has sample complexity of $\widetilde{\mathcal{O}}\left(\frac{p^2HS^2A}{\epsilon^2}+\frac{p^2d_Ed_C}{\epsilon^2}\right)$, where $d_E:=\max_{h\in\mathcal{H}_p}d_{E,h}$, and $d_C:=\max_{h\in\mathcal{H}_p}d_{C,h}$. POR-UCBVI has sample complexity $\widetilde{\mathcal{O}}\left(\frac{p^2HSA\max(H,S)}{\epsilon^2}+\frac{p^2d_E\max(d_C,H)\log(1/\delta)}{\epsilon^2}\right)$.

Challenges: There are three main technical challenges in proving these guarantees. First, we have to handle non-Markovian reward functions with Markovian transitions. Second, in POR-UCBVI, we have the added challenge of ensuring that the bonus is uniformly optimistic over all history-dependent policies. This is typically a doubly exponential set ($A^{(SA)^H}$), so a union bound does not help us. Third, we are working with general function approximation for reward functions using $\mathcal{F}$.

Remark 7 (Comparison to past results)**.** Notice that with $\mathcal{U}=\phi(\tau)^\top\mathbf{w}$ with $\mathbf{w}\in\mathcal{R}^d$ and $\mathcal{H}_p=\{H\}$, we are in the setting of Chatterji et al. [2021]. Here, $d_{E,H}=d_{C,H}=d$, so POR-UCRL and POR-UCBVI both improve over their regret guarantees. With respect to sample complexity guarantees, we compare to Wang et al. [2023b]. While they use dueling feedback, our methods use cardinal feedback. In their setting, $\mathcal{U}$ is the set of all histories and $\mathcal{H}_p=\{H\}$. Their best guarantee is from P-OMLE, which makes $\widetilde{\mathcal{O}}\left(\frac{H^2S^2A}{\epsilon^2}+\frac{H^2d_{E,H}d_{C,H}}{\epsilon^2}\right)$ dueling oracle queries for tabular $\mathcal{P}$. Both POR-UCRL and POR-UCBVI have a smaller complexity for cardinal feedback queries.

Remark 8 (General function approximation for $\mathcal{P}$)**.** For clearer exposition, we have assumed that $\mathcal{P}$ is a tabular class with finite $\mathcal{S},\mathcal{A}$ in the results stated above. This is because handling general function approximation for $\mathcal{F}$ is the non-trivial part of this work. We provide straightforward extensions to general function approximation for $\mathcal{P}$ with continuous $\mathcal{S},\mathcal{A}$ in Appendix C.6, using existing work.

Concretely, in Appendix C.6, we extend both POR-UCRL and POR-UCBVI to general function approximation for the transition model, replacing the tabular S,A dependence with complexity

measures of $\mathcal{P}$. Under the distributional eluder dimension, POR-UCRL achieves regret

$$\widetilde{\mathcal{O}}\left(\left(p\sqrt{d_{E,\mathcal{P}}d_{C,\mathcal{P}}}+\sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}d_{C,h}}\right)\sqrt{T}\right)$$

where $d_{E,\mathcal{P}}$ and $d_{C,\mathcal{P}}$ are the distributional eluder and bracketing dimensions of $\mathcal{P}$. Under the strong SAIL condition, a similar bound holds with the SAIL dimension replacing the distributional eluder dimension. The dueling reduction (Theorem 5.4.2) extends analogously.

5.3.2 Leveraging Recursive Structures Using Model-Free Methods

We have established that the model-based methods POR-UCRL and POR-UCBVI improve over naive history-summarization and have a $poly(S, A, H)$ guarantee in terms of transition function estimation. However, we recall the last challenge mentioned in Section 5.2.3 – can they adapt to examples like the combination lock, where there is a recursive structure on the internal states? Disappointingly, we will see in Proposition 5.3.3 that the answer is no – they are *exponentially* worse than the ideal solution. Intuitively, learning the reward and transition models separately is needlessly expensive here. At the more technical level, since POR-UCRL and POR-UCBVI decouple the learning of reward functions across timesteps, they are unable to incorporate a recursive structure on the reward functions.

In this light, we consider model-free methods. Unlike model-based methods that have to account for Markovian transitions, we can simply use naive history-summarization here and treat $\tau[h]$ as the state for Q-functions Q_h. However, under history-summarization, there is a subtlety involved in choosing the class $\mathcal{Q}$ of Q-functions given a known class $\mathcal{M}$ of models. Using product classes $\mathcal{Q}_1 \times \cdots \times \mathcal{Q}_H$ is wasteful, since often exponentially many tuples in a product class cannot be realized by any model $\mathcal{M}$.[10] Instead, one should consider the class of only the tuples $(Q_1, \ldots, Q_H)$ that can be *realized* by a model $\mathcal{M}$. In practice, this translates to the problem of good representation learning – one should use a shared network for all Q-functions instead of using a different network for each timestep. This is reflected in the experimental choices of Lightman et al. [2024].

Model-free methods rely on the Bellman error, which relates consecutive Q-functions and couples their learning. Recall that GOLF [Jin et al., 2021a] maintains a version space $\mathcal{Q}_t$ of Q-functions with small empirical Bellman errors and plays the most optimistic element at each episode; its regret scales with the Bellman eluder dimension of the function class. It is thus natural to expect model-free methods like GOLF [Jin et al., 2021a] to adapt to a recursive structure on internal states and perform better than model-based methods. However, existing guarantees do not reflect this. It

[10] The reader can use the example of the combination lock to convince themselves of this.

turns out from Proposition 5.3.3 below that the Bellman-eluder (BE) dimension of the combination lock problem is A^H, even with the minimal Q-function class.

The issue is that the proof of GOLF bounds the h-step Bellman errors in a decoupled manner, which is why it still fails to incorporate a recursive structure on internal states. Intuitively, one wants to *wait* for Bellman errors at timesteps $1, \dots, h-1$ to become small before bounding the Bellman error at h. In this light, given a parameter α, we define the function class

$$\mathcal{Q}(\alpha, h) := \left\{Q \in \mathcal{Q} \mid |\mathbb{E}_{\mu_l(Q)}[Q_l - \mathcal{T}_l Q_{l+1}]| \le \alpha,\ \forall 1 \le l \le h\right\}$$

that considers all tuples $(Q_1, \dots, Q_H)$ where the Bellman errors until step h are already low. We can use this class to define the α-history aware Bellman-eluder dimension (HABE) of $\mathcal{Q}$ as follows. Recall that π_Q is the policy that acts greedily according to $Q = (Q_1, \dots Q_H)$.

Definition 5.3.1. Consider the Bellman errors $\Phi_h := \left\{Q_h - \mathcal{T}_h Q_{h+1} \middle| Q \in \mathcal{Q}(\alpha, h-1)\right\}$. Denote $\mu_h(Q)$ the distribution induced on $\tau[h-1], a_h$ by π_Q and let $\mathcal{D}_{h,\mathcal{Q}} := \{\mu_h(Q) \mid Q \in \mathcal{Q}\}$. Let $\dim_{DE}$ the distributional eluder dimension and define $\dim_{\mathrm{HABE}}(\mathcal{Q}, \alpha, \epsilon) := \max_h \dim_{DE}(\Phi_h, \mathcal{D}_{h,\mathcal{Q}(\alpha,h-1)}, \epsilon)$. Intuitively, α-HABE dimension measures how hard it is to reduce the Bellman error at timestep h if the errors at *previous timesteps* $1, \dots, h-1$ are already small. This captures the hardness of adapting to the recursive structure on internal states one/a few timesteps at a time. We discuss in Appendix C.7.1 how the α-HABE dimension compares to the Bellman-eluder dimension in general. We now give a new guarantee for GOLF using the α-HABE dimension.

Theorem 5.3.2 (Modified GOLF Regret)**.** *Let Assumption 9 hold, let $\mathcal{Q}$ be Bellman complete, and let $d_{\mathrm{HABE}} = \dim_{\mathrm{HABE}}(\mathcal{Q}, \alpha, \min(\alpha, \sqrt{1/T}))$. Choose hyperparameter $\beta = c\log(HT\mathcal{N}(\mathcal{Q} \cup \mathcal{G}, 1/T, \|\cdot\|_\infty))$ for some universal constant c and the auxiliary function class $\mathcal{G}$ used in GOLF, and define $d_{C,\mathcal{Q}} := \log(\mathcal{N}(\mathcal{Q} \cup \mathcal{G}, 1/T, \|\cdot\|_\infty))$. Then, GOLF satisfies $\mathrm{Regret}(T) = \mathcal{O}\left(pH\sqrt{d_{\mathrm{HABE}} d_{C,\mathcal{Q}} T}\right)$.*

Using a regret-to-PAC conversion, we also show in Corollary C.7.2 that the sample complexity of GOLF is $\widetilde{\mathcal{O}}\left(\frac{p^2H^2 d_{\mathrm{HABE}} d_{C,\mathcal{Q}}}{\epsilon^2}\right)$. As foreshadowed above, we now show in Proposition 5.3.3 that these guarantees can be polynomial even when the the usual guarantees for GOLF as well as guarantees for our model-based algorithms are exponential. Note that this improvement is achieved only given dense intermediate feedback. Under sparse intermediate feedback, one cannot adapt to internal states "a few timesteps at a time," and we in fact have $\Omega(\sqrt{A^H T})$ regret under *any* algorithm. However, dense feedback case is quite realistic for many applications, such as automated mathematical reasoning.

Proposition 5.3.3 (Dimensions for the Combination Lock)**.** *Consider the combination lock problem with model class $\mathcal{M} = \mathcal{P} \times \mathcal{F}$ and induced Q-function class $\mathcal{Q}$.*

- *Under dense intermediate feedback with $\mathcal{H}_p = [H]$, $\dim_{\mathrm{HABE}}(\mathcal{Q}, \alpha) = A$ for all $\alpha < q$, while its BE dimension is at least $A^H - 2$. The eluder dimension for reward functions $\dim_E(\mathcal{F}_h, \frac{B}{T})$ is at least A^h for any $h \le H$.*
- *For sparse intermediate feedback with $\mathcal{H}_p = \{H\}$ and any $\alpha > 0$, the α-HABE dimension, the BE dimension and the eluder dimension of $\mathcal{F}_H$ are all at least $A^H - 2$. Moreover, any algorithm in this setting will have regret $\Omega(\sqrt{A^H T})$.*

The α-HABE dimension measures how hard it is to reduce the Bellman error at timestep h *if the errors at previous timesteps* $1, \dots, h-1$ *are already at most* α. For the combination lock with dense feedback ($\mathcal{H}_p = [H]$), the Bellman error at each step is binary: either 0 (correct action sequence so far) or q (wrong). For any $\alpha < q$, the restriction to Q-functions with errors at most α at steps $1, \dots, h-1$ forces the first $h-1$ actions to match the true combination exactly, leaving only A candidate Q-functions at step h — one per choice of the free action a_h. The HABE dimension is thus A. The standard BE dimension does not condition on earlier errors, so it must discriminate among all A^H possible action sequences simultaneously, giving dimension $A^H - 2$. With sparse feedback ($\mathcal{H}_p = \{H\}$), no intermediate errors are available to constrain earlier actions, so $\mathcal{Q}(\alpha, H-1) = \mathcal{Q}$ for all α and the HABE dimension matches the BE dimension at $A^H - 2$. The full computation is in Appendix C.7.2.

We discuss in Appendix C.7.1 that in general, we do not have an inequality in either direction. However, the α-HABE dimension is typically smaller.

5.4 Dueling to Optimism Reduction

The dueling and cardinal feedback models are intimately related. It is thus tempting to use algorithms for cardinal PORRL to solve dueling PORRL. However, we detail why the "obvious" reduction from dueling feedback to cardinal feedback fails. This both demonstrates the hardness of the problem and motivates our reduction.

5.4.1 The Naive Reduction Always Fails

Consider a modified PORMDP $\overline{\mathcal{M}}$ with $\overline{\mathcal{S}} := \mathcal{S} \times \mathcal{S}$, $\overline{\mathcal{A}} := \mathcal{A} \times \mathcal{A}$, $\overline{\mathbb{P}} := \mathbb{P} \otimes \mathbb{P}$, where we run the pair of policies $\overline{\pi} := (\pi_1, \pi_2)$ and obtain observations based on the decoder-induced function $\overline{f}_h(\tau_1[h], \tau_2[h]) := f_h(\tau_1[h]) - f_h(\tau_2[h])$. Consider the space of all such PORMDPs induced by $\mathcal{M}$, and denote it by $\overline{\mathcal{M}}$. Since cardinal feedback in $\overline{\mathcal{M}}$ exactly corresponds to dueling feedback in $\mathcal{M}$, it is tempting to restrict to searching over $\Pi \times \Pi$ and run any algorithm for cardinal PORRL on this modified PORMDP $\overline{\mathcal{M}}$ to achieve low dueling regret.

This fails because the feedback model and regret metric are fundamentally non-aligned in dueling feedback, unlike in cardinal feedback. While the agent receives dueling feedback over the duel for $(\pi_{1,t}, \pi_{2,t})$, dueling regret is instead concerned with duels for $(\pi_\star, \pi_{1,t})$ and $(\pi_\star, \pi_{2,t})$. Running an algorithm for cardinal PORRL on the modified MDP will maximize the dueling *feedback* itself. This is achieved by playing one good and one really bad policy, unlike the two good policies needed for low dueling regret. We formalize this below, showing that the naive reduction leads to linear dueling regret for *any* PORMDP and *any* cardinal PORRL algorithm with sublinear regret.

Lemma 5.4.1 (Naive Reduction Lower Bound)**.** *Using any algorithm for cardinal PORRL with sublinear cardinal regret on $\overline{\mathcal{M}}$ with policy class $\Pi' := \Pi \times \Pi$ to get a sequence $(\pi_{1,1}, \pi_{2,1}), \dots, (\pi_{1,T}, \pi_{2,T})$ leads to linear dueling regret for $\mathcal{M}$ whenever all policies π do not have the same value $V(\mathcal{M}, \pi)$.*

The key point is that sublinear cardinal regret forces $\pi_{2,t} \to \pi_{\min}$ (the worst policy), since the algorithm maximizes the dueling feedback $V(\pi_1) - V(\pi_2)$. This produces one good and one bad policy, rather than the two good policies that low dueling regret demands. The proof is in Appendix C.1.2.

5.4.2 Reducing Dueling to Optimistic Cardinal PORRL

The naive reduction fails because maximizing dueling feedback can lead to bad policies being played. In this subsection, we present a white-box reduction where we ensure that we only play potentially good policies for *both* $\pi_{1,t}$ and $\pi_{2,t}$. We detail here how we can obtain an algorithm for the dueling feedback problem from *any* optimistic algorithm for cardinal PORRL. We will focus on the case of confidence sets here for smoother exposition, the much harder case of bonuses is treated in Appendix C.8.2. A *generic optimistic algorithm using confidence sets* maintains confidence sets $\mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$ using the collected dataset $\mathcal{D}_t$ of trajectories and feedback. We define it formally in Appendix C.3.1. For the reduction to work, we require that the confidence sets are well-designed, as demanded by Assumption 10. This assumption is satisfied for confidence sets used by POR-UCRL.

Assumption 10 (Controlling Value Error due to Confidence Sets)**.** $\mathcal{M}_\star \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$ for arbitrary sequences $(\mathbb{P}_t, f^t) \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$, both $\left|\sum_{t=1}^T V(\mathbb{P}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t)\right| = \widetilde{\mathcal{O}}(C_P(\mathcal{M}, T, \delta))$ and $\left|\sum_{t=1}^T V(\mathbb{P}_\star, f^t, \pi_t) - V(\mathbb{P}_\star, f_\star, \pi_t)\right| = \widetilde{\mathcal{O}}(C_F(\mathcal{M}, T, \delta))$ hold with probability $1 - \delta/2$ each.

The key insight is to use confidence sets from cardinal PORRL to search for $\pi_{1,t}$ and $\pi_{2,t}$ only among policies that *both* have a chance of being optimal. Then one plays the *most uncertain* duel among all possible choices for $\pi_{1,t}$ and $\pi_{2,t}$. This generalizes and abstracts out ideas in Pacchiano et al. [2021], which presents a specific algorithm to achieve low dueling regret in their model. We present the reduction to optimism over confidence sets in Algorithm 10, the version for bonuses is in Appendix C.8.2. Define $V_D(\overline{\mathcal{M}}, \pi, \pi') = V(\mathcal{M}, \pi) - V(\mathcal{M}, \pi')$. We compute the confidence sets

$\mathcal{C}_{\overline{\mathcal{P}}}(\mathcal{D}, \delta)$ as the image of $\mathcal{C}_{\mathcal{P}}(\mathcal{D}, \delta)$ under $\mathbb{P} \mapsto \overline{\mathbb{P}}$. We compute $\mathcal{C}_{\overline{\mathcal{F}}}(\mathcal{D}, \delta)$ by treating $\{o_h\}_{h \in \mathcal{H}_p}$ as cardinal feedback in $\overline{\mathcal{M}}$. As an example, for POR-UCRL, we perform a least squares fit for $\overline{f}$ and use Lemma C.4.1 to define our confidence sets again. We then get the following regret guarantee.

Algorithm 10 Reduction from Dueling to Cardinal Confidence-Set Optimism

1: **Input** Known reward function $\{r_h\}_{h=1}^H$, method to compute $\mathcal{C}_{\overline{\mathcal{M}}}(\mathcal{D}, \delta) \leftrightarrow \mathcal{C}_{\overline{\mathcal{P}}}(\mathcal{D}, \delta) \times \mathcal{C}_{\overline{\mathcal{F}}}(\mathcal{D}, \delta)$
2: **Initialize** dataset $\mathcal{D}_1 \leftarrow \{\}$, $\mathcal{C}_{\overline{\mathcal{M}}}(\mathcal{D}_1, \delta) := \overline{\mathcal{P}} \times \overline{\mathcal{F}}$
3: **for** $t = 1, ..., T$ **do**
4: **Compute** $\Pi_t = \left\{ \pi \in \Pi \middle| \exists \overline{\mathcal{M}} \in \mathcal{C}_{\overline{\mathcal{M}}}(\mathcal{D}_t, \delta) \text{ s.t. } V(\overline{\mathcal{M}}, \pi, \pi_1) \geq 0 \; \forall \pi_1 \in \Pi \right\}$ {Candidates $\pi_\star$}
5: **Play** $(\pi_{1,t}, \pi_{2,t}) \in \arg\max_{\pi, \pi' \in \Pi_t} \max_{\overline{\mathcal{M}}, \overline{\mathcal{M}}' \in \mathcal{C}_{\overline{\mathcal{M}}}(\mathcal{D}_t, \delta)} V_D(\overline{\mathcal{M}}, \pi, \pi') - V_D(\overline{\mathcal{M}}', \pi, \pi')$ {Most uncertain duel}
6: **Observe** trajectories $\tau_{i,t} = \left\{(s_{i,h}^t, a_{i,h}^t)\right\}_{h=1}^H$ along with feedback $\{o_h\}_{h \in \mathcal{H}_p}$
7: **Update** $\mathcal{D}_t$ to $\mathcal{D}_{t+1}$ using the data and compute $\mathcal{C}_{\overline{\mathcal{P}}}(\mathcal{D}_{t+1}, \delta)$, $\mathcal{C}_{\overline{\mathcal{F}}}(\mathcal{D}_{t+1}, \delta)$
8: **end for**

Theorem 5.4.2 (Reduction from Dueling to Confidence-Set-Based Optimism)**.** *If the confidence sets $\mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$ satisfy Assumption 10, then the dueling regret $\mathrm{Regret}_D(T)$ of Algorithm 10 is given by*

$$\mathrm{Regret}_D(T) = \widetilde{\mathcal{O}}(C_P(\mathcal{M}, T, \delta) + C_F(\overline{\mathcal{M}}, T, \delta))$$

Note that complexity parameter C_F depends on $\overline{\mathcal{M}}$. It is a priori unclear how the complexity of $\overline{\mathcal{M}}$ relates to that of $\mathcal{M}$. Fortunately, Lemma 5.4.3 below settles this, and we can then use our results for POR-UCRL to get Corollary 5.4.4 below. See Appendix C.6.3 for a straightforward extension to general function approximation for $\mathcal{P}$, abstracting out the S, A dependence.

Lemma 5.4.3 (Relating $\mathcal{F}$ and $\overline{\mathcal{F}}$)**.** *For any function class $\mathcal{F}$, $\dim_E(\overline{\mathcal{F}}, \epsilon) \leq 9 \dim_E(\mathcal{F}, \epsilon/2)$.*

Corollary 5.4.4 (Dueling Regret using POR-UCRL Confidence Sets)**.** *The confidence sets from POR-UCRL satisfy Assumption 10 and using them in Algorithm 10 leads to the following regret bound* $\mathrm{Regret}_D(T) = \widetilde{\mathcal{O}}\left(\left(pS\sqrt{HA} + \sum_{h \in \mathcal{H}_p} \sqrt{d_{E,h} d_{C,h}}\right)\sqrt{T}\right)$.

5.5 Conclusions and Future Work

In this work, we have introduced PORMDPs and their analysis as a way to better model internal states of humans and intermediate feedback in RLHF. We have introduced two statistically efficient algorithms for handling partially observed reward-states and have shown that they improve over naive history summarization. We have noted that these methods subsume as well as improve over a lot of past work in RLHF. We have studied how one can further leverage a recursive structure

over internal states using model-free methods. For this purpose, we have defined a new notion of dimension, the α-HABE dimension, that captures the hardness of utilizing the recursive structure. Finally, we have also provided a novel reduction from dueling regret to optimistic algorithms for cardinal regret.

Besides our theoretical contributions, we would like to note the practical implications of our work.

- When the feedback is suspected to have a recurrent structure, we conclude in sections 5.3.1 and 5.3.2 that it can be exponentially more statistically efficient to use practical model-free methods like learning history-dependent Q-functions or using actor-critic methods. In the absence of such a structure, a model-based approach learning $f_\star$ and $\mathbb{P}_\star$ explicitly will also suffice.
- We note in section 5.3.2 that in practice, using a single network for the Q-function or critic across timesteps h is important for the "exponential improvement" mentioned above, as opposed to using a different network for the Q-function or critic for each timestep.
- We are hoping that our work inspires new practical algorithms for PORRL. Theoretical advances in optimistic algorithms are known to inspire practical versions of the algorithms, such as perturbative ones. Some classic examples include the analysis of PSRL inspired by UCRL, and bootstrapped DQN inspired by optimistic versions of Q-learning.
- The dueling to optimism reduction in section 5.4 can inspire future practical algorithms that achieve low dueling regret, which we have established in section 5.2.2 as an important metric for online (and online iterative) RLHF applications, like in Dong et al. [2024], Xiong et al. [2024].

We hope that our ideas lay the groundwork for further understanding of both statistical and algorithmic aspects of learning good policies when interacting with "stateful" feedback, such as that of humans. Our algorithms and proofs are presented in high generality and modularity in the appendix, and we hope that they can be used to provide novel algorithms and bounds in the future.

CHAPTER 6

Leveraging Offline Data in Linear Latent Contextual Bandits

In the previous chapter, we studied how latent variables in reward models affect learning, focusing on the online RLHF setting. A natural follow-up question concerns combining offline data with online interactions: when a wealth of historical trajectories is available and the reward model is determined by an unobserved latent state, can we leverage offline data to accelerate online learning? In the running example of Section 1.2.2, this corresponds to personalizing an LLM across the intern population, where each intern's preference direction $\boldsymbol{\beta} \in \mathbb{R}^{d_A}$ lies in a d_K-dimensional subspace with $d_K \ll d_A$. In the language of Section 1.4, the latent variable is again confined to the reward channel and acts through a literal low-dimensional subspace, the same kind of low-complexity channel seen in Chapter 3 but now for reward parameters rather than transitions. This question connects back to the spectral methods developed in Chapter 3, but in a different setting: rather than clustering trajectories to identify mixture components, we now use offline data to estimate the low-dimensional subspace of reward parameters, which then speeds up online exploration. We focus on the bandit setting, since the combined challenges of offline subspace estimation, uncertainty quantification over the estimated subspace, and online exploration are already quite ambitious.

6.1 Introduction

This chapter is a lightly edited version of Kausik et al. [2025].

Many sequential-decision making problems can be effectively modeled using the bandit framework. This can span domains as diverse as healthcare Lu et al. [2021b], randomized clinical trials Press [2009], search and recommendation Li et al. [2010], distributed networks Kar et al. [2011], and portfolio design Brochu et al. [2011]. There is often a wealth of offline data in such domains, which has led to a growing interest in using offline data to accelerate online learning. However, there often also exist unobserved contexts in the population that influence the distribution of rewards, making it non-trivial to leverage offline data. In Hong et al. [2020], it is shown that this uncertainty

can be modeled by a *latent bandit* (or mixture of bandits). This is a bandit where an unobserved latent state determines the reward model for the trajectory. For example, a patient's underlying genetic conditions in healthcare and a user's tastes in recommendation systems are both examples of latent states in sequential decision making. Typically, these latent states are less complex than the actual models underlying users or patients, making it valuable to reduce the online task to learning the latent state Hong et al. [2020, 2022].

The latent bandit framework therefore has high practical value, and efficient principled algorithms are needed for using offline data to speed up online learning. Using traditional bandit algorithms in this setting does not leverage the offline data that is often available to the agent. Naturally, one also cannot treat the offline data as coming from a single bandit. For example, different user tastes or different underlying genetic conditions require modeling the offline data as coming from a latent bandit. So, we have to develop algorithms *specific* to the latent bandit setting that leverage the offline data to improve online performance.

We note that in bandit literature, it is common to impose a structure on bandit rewards when designing algorithms, the most popular one being a linear structure Li et al. [2010], Abbasi-Yadkori et al. [2011]. In this light, we study a linear contextual bandit setting where each user has its own high-dimensional reward parameter, but reward parameters across users lie in a low-rank subspace. This is a *linear latent contextual bandit*[1], and is much more general than existing models that restrict themselves to finitely many latent states Hong et al. [2020, 2022]. We design a two-pronged algorithm to tackle this setting. First, we provide a method to approximate the low-dimensional subspace spanned by latent states from an offline dataset of unlabeled trajectories collected under some behavior policy π_b. This is non-trivial since the trajectories are unlabeled, and standard unsupervised learning methods fail. Second, we use this subspace to speed up online learning. However, since the subspace is only learnt *approximately,* we also tackle the non-trivial task of accounting for the uncertainty in the subspace. We design two methods for the latter, facing a trade-off between computational tractability and tightness of guarantees. Experiments show the efficacy of our methods.

While latent bandits have thus shown to be a powerful and tractable framework for accounting for uncertainty in reward models, the extent of their *generality* is unclear. Are there other stateless decision processes that generalize over latent bandits? We end by theoretically demonstrating that under very reasonable assumptions, the answer is no. We show a de Finetti theorem for decision processes, demonstrating that *every* "coherent" and "exchangeable" stateless (contextual) decision process is a latent (contextual) bandit. With this in mind, we outline our contributions below:

[1]This can also be thought of as a continuous mixture of bandits.

- **Offline method:** We present SOLD, a novel offline method for learning low-dimensional subspaces of reward parameters with guarantees, inspired by the novel spectral methods in Kausik et al. [2023].
- **Tight online algorithm:** We present LOCAL-UCB, an online algorithm leveraging the subspace estimated offline to sharpen optimism, achieving $\tilde{O}(\min(d_A\sqrt{T}, d_K\sqrt{T}(1+\sqrt{d_AT/d_KN})))$ regret.
- **Lower Bound:** We establish a matching lower bound showing that LOCAL-UCB is minimax optimal. To the best of our knowledge, this is the first lower bound in a hybrid (offline-online) sequential decision-making setting.
- **Tractable online algorithm:** Finally, we present ProBALL-UCB, a practical and computationally efficient online algorithm with a slightly looser regret guarantee. This also illustrates a general algorithmic idea for integrating offline subspace estimation into optimistic algorithms.
- **Experiments:** We establish the efficacy of our algorithms outlined above through a simulation study and a demonstration on a real recommendation problem with the MovieLens-1M Harper and Konstan [2015] dataset.
- **Theoretical generality:** We are the first, to our knowledge, to prove a de Finetti theorem for decision processes. This establishes the generality of the latent bandit model.

Related work. There are three main threads of related work.

- **Latent Bandits.** The line of work most relevant to us has been on latent bandits. The work of Hong et al. [2020, 2022] studies the latent bandit problem under finitely many states. However, they black-box the offline step and do not provide end-to-end guarantees, and their ideas do not extend to infinitely many states. Our work seeks to provide end-to-end guarantees for both the offline and online component under infinitely many latent states.
- **Meta learning, multi-task learning and mixture learning.** A long line of work studies learning with multiple underlying tasks or models. For example, the work of Vempala and Wang [2004], Kong et al. [2020], Anandkumar et al. [2014], Tripuraneni et al. [2021] study learning under latent variable or multi-task models in a supervised setting. The work of Kausik et al. [2023], Chen and Poor [2022] extend some of these ideas to unsupervised but purely offline learning in a time-series setting. On the other hand, work like Yang et al. [2022], Cella et al. [2022] instead focuses on the purely online setting of learning the low-rank structure while simultaneously interacting with multiple finitely many bandit instances. Finally, Zhou et al. [2024], Lu et al. [2021a] work with multiple underlying models in MDPs but in a purely offline and generative model setting respectively. Unlike these papers, we crucially combine the offline and online

settings for sequential data and study the problem of using offline data to accelerate online learning in bandits. Multi-task linear bandits [Soare et al., 2014] study sequential interaction with multiple related tasks whose reward parameters share low-rank structure. Unlike our setting, these methods assume simultaneous online access to multiple tasks; we work with a single online task and use offline data from a heterogeneous population to learn the shared structure. A direct experimental comparison is not possible due to this difference in setting, though adapting multi-task methods to the offline-online hybrid setting is an interesting direction.

- **Hybrid (offline-online) RL.** Work in hybrid RL studies the use of offline data to accelerate online RL, first proposed by Song et al. [2023], with extensions to linear MDPs by Wagenmaker and Pacchiano [2023], Tan et al. [2024]. Cai et al. [2024] studies the same problem for contextual bandits. However, all work so far assumes that the offline data is generated by a single model, and does not account for latent states. Our work explores a hybrid offline-online setting while also accounting for the offline data being generated by multiple underlying models.

6.2 Linear Bandits With Latent Structure

We will first introduce latent contextual bandits, and then specialize to our linear model later in this section. A latent contextual bandit is a decision process with contexts $\mathcal{X}$, actions $\mathcal{A}$ and an random latent state θ that is sampled independently at the beginning of each trajectory. Given a sequence of contexts and actions, the rewards at all steps are independent *conditioned* on θ, and depend on the latent state θ, the context and the action. Since we often have access to an offline dataset of trajectories coming from different kinds of users or patients, it is important to account for a changing latent state θ between trajectories.

As we will see in Section 6.8, latent bandits are a powerful and general framework for encoding uncertainty in reward models. However, this generality is both a blessing and a curse. It is hard to design concrete algorithms without further assumptions on the structure and effect of the latent state θ. We therefore focus on a linear structure here. We consider the natural generalization of the linear contextual bandit to the latent bandit setting, where we impose a linear structure on the effect of low-dimensional latent states. We further justify this by noting that in most application domains, it is reasonable to assume that a parsimonious, low-dimensional latent state affects the reward distribution. This motivates the following definition.

Definition 6.2.1. A *linear latent contextual bandit* is a linear bandit equipped with a feature map for context-action pairs $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{d_A}$, a latent random variable $\boldsymbol{\theta} \in \mathbb{R}^{d_K}$ with distribution $\mathcal{D}_\theta$ and a map $\mathbf{U}_\star : \mathbb{R}^{d_K} \to \mathbb{R}^{d_A}$ such that for any H and context-action sequence $((x_1, a_1), \dots (x_H, a_H))$, the rewards $(Y_1, \dots Y_H)$ are independent conditioned on $\boldsymbol{\theta}$. Moreover, $Y_h \mid \boldsymbol{\theta} \sim \phi(x_h, a_h)^\top \boldsymbol{\beta} + \epsilon$,

where ϵ is subgaussian noise independent of all actions and all other observations, and $\boldsymbol{\beta} = \mathbf{U}_\star \boldsymbol{\theta}$.

Further, we note that WLOG $\mathbf{U}_\star$ has orthonormal columns: $\mathbf{U}_\star^\top \mathbf{U}_\star = \mathbf{I}_{d_A}$. This is because for any invertible map $A : \mathbb{R}^{d_K} \to \mathbb{R}^{d_K}$, the observation distribution does not change upon replacing $\boldsymbol{\theta}$ with $A\boldsymbol{\theta}$ and $\mathbf{U}_\star$ with $\mathbf{U}_\star A^{-1}$. One can see this as a generalization of the fact that with finitely many latent states, the observations are not changed by permuting the latent states. That is, the observations are not changed by permuting latent trajectory labels while keeping trajectories with the same label together.

Let us now assume that we have access to a dataset $\mathcal{D}_{\text{off}}$ of N *short* trajectories $\tau_n = ((x_{n,1}, a_{n,1}, r_{n,1}), \ldots, (x_{n,H}, a_{n,H}, r_{n,1}))$ of length H, collected by some behavior policy π_b. The trajectories are short in the sense that in most relevant domains, individual trajectories are not long enough to learn the underlying reward model. Each trajectory τ_n has a different $\boldsymbol{\beta}_n = \mathbf{U}_\star \boldsymbol{\theta}_n$. In online deployment, a single latent label $\boldsymbol{\theta}_\star$ is chosen and rewards are generated using $\boldsymbol{\beta}_\star = \mathbf{U}_\star \boldsymbol{\theta}_\star$. At each timestep t, an agent observes contexts x_t and uses both the offline data and the online data at time t to execute a policy π_t. Define the optimal action at time t by $a_t^\star := \max_a \phi(x_t, a_t)^\top \boldsymbol{\beta}_\star$ We tackle the problem of minimizing the *frequentist* regret in linear latent contextual bandits, given by

$$\text{Reg}_T := \sum_{t=1}^{T} \phi(x_t, a_t^\star)^\top \boldsymbol{\beta}_\star - \mathbb{E}_{a \sim \pi_t}[\phi(x_t, a)^\top \boldsymbol{\beta}_\star].$$

For example, in medical applications, data from short randomized controlled trials can be used to help an agent suggest treatment decisions for a new patient online. In this case, we would like the algorithm to administer the correct treatments for *each* patient. This means that the *frequentist* regret is the relevant performance metric here, and not the Bayesian regret over some prior. Additionally, any worst-case bound on the frequentist regret is a bound on the Bayesian regret for arbitrary priors.

Challenges with latent bandits. Despite the linear assumption, and the dimension reduction obtained in the common case when $d_K \ll d_A$, significant challenges remain. First, the value of the latent state $\boldsymbol{\theta}$ and the map $\mathbf{U}_\star$ are both unknown a priori, making it hard to leverage the low-dimensional structure of the problem. Second, a good choice of dimension d_K is itself unknown a priori, and must be determined from data in a principled manner. Third, even if we learn the low-dimensional structure, our learning will be *approximate*, and the online procedure must account for this uncertainty. In the following sections, we will provide a method to estimate and use latent subspaces given offline data that allows us to overcome these challenges.

Additional Notation. We use $\mathbf{V}$ to denote regularized design matrices given by $\mu\mathbf{I} + \sum_{(x,a)} \phi(x,a)\phi(x,a)^\top$. We define $\mathbf{D}_{n,i} = I - \mu \mathbf{V}_{n,i}^{-1}$ and $\overline{\mathbf{D}}_{N,i} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{D}_{n,i}$. Denote by $\hat{\beta}_{n,1}, \hat{\beta}_{n,2}$

independent estimates of β_n from τ_n. Let $\mathbf{M}_n = \frac{1}{2}(\hat{\boldsymbol{\beta}}_{n,1}\hat{\boldsymbol{\beta}}_{n,2}^\top + \hat{\boldsymbol{\beta}}_{n,2}\hat{\boldsymbol{\beta}}_{n,1}^\top)$ and $\overline{\mathbf{M}}_N \leftarrow \frac{1}{N}\sum_{n=1}^N \mathbf{M}_n$.

6.3 Estimating Latent Subspaces Offline

Although we do not have access to the values of the latent states $\boldsymbol{\theta}$ or to the map $\mathbf{U}_\star$, we can still extract useful information from data. To that effect, recall that we have access to a dataset $\mathcal{D}_{\text{off}}$ of N trajectories $\tau_n = ((x_{n,h}, a_{n,h}, r_{n,h}))_{h=1}^H$ of length H, collected by some behavior policy π_b.

How can offline data help us in online deployment? To minimize the regret, one must learn the reward parameter $\boldsymbol{\beta}_\star$ online. However, it is much easier to search among all latent states $\boldsymbol{\theta} \in \mathbb{R}^{d_K}$ than to search among all possible reward parameters $\boldsymbol{\beta} \in \mathbb{R}^{d_A}$ since typically, $d_K \ll d_A$. So, it will help to learn some projection matrix $\hat{\mathbf{U}}^\top \approx \mathbf{U}_*^\top : \mathcal{R}^{d_A} \to \mathcal{R}^{d_K}$ offline so that for any estimate $\hat{\boldsymbol{\beta}}_t$ of $\boldsymbol{\beta}_\star$, $\hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_t$ is an estimate of $\hat{\boldsymbol{\theta}}_t \in \mathbb{R}^{d_K}$. This amounts to *learning a subspace* of the feature space from logged bandit data. We therefore provide a method for Subspace estimation from Offline Latent bandit Data (SOLD) in Algorithm 11. Recall that since the learnt subspace is approximate, we also need to compute the uncertainty over the subspace to get a subspace confidence set that we can use online.

Algorithm 11 Subspace estimation from Offline Latent bandit Data (SOLD)

1: **Input:** Dataset $\mathcal{D}_{\text{off}}$ of collected trajectories $\tau_n = ((x_{n,1}, a_{n,1}, r_{n,1}), ..., (x_{n,H}, a_{n,H}, r_{n,1}))$ under a behavior policy π_b, dimension of latent subspace d_K.
2: **Divide** each τ_n into odd and even steps, giving trajectory halves $\tau_{n,1}$ and $\tau_{n,2}$.
3: **Estimate** reward parameters $\hat{\boldsymbol{\beta}}_{n,i} \leftarrow \mathbf{V}_{n,i}^{-1}\boldsymbol{b}_{n,i}$, where $\mathbf{V}_{n,i} \leftarrow \mu\mathbf{I} + \sum_{(x,a,r)\in\tau_{n,i}} \phi(x,a)\phi(x,a)^\top$ and $\boldsymbol{b}_{n,i} \leftarrow \sum_{(x,a,r)\in\tau_{n,i}} \phi(x,a)r$ for $i = 1, 2$.
4: **Compute** $\mathbf{M}_n \leftarrow \frac{1}{2}(\hat{\boldsymbol{\beta}}_{n,1}\hat{\boldsymbol{\beta}}_{n,2}^\top + \hat{\boldsymbol{\beta}}_{n,2}\hat{\boldsymbol{\beta}}_{n,1}^\top)$ and compute $\overline{\mathbf{M}}_N \leftarrow \frac{1}{N}\sum_{n=1}^N \mathbf{M}_n$.
5: **Compute** $\overline{\mathbf{D}}_{N,i} \leftarrow \frac{1}{N}\sum_{n=1}^N (I - \mu\mathbf{V}_{n,i}^{-1})$, $i = 1, 2$.
6: **Obtain** $\hat{\mathbf{U}}$, the top d_K eigenvectors of $\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1}$.
7: **return** Projection matrix $\hat{\mathbf{U}}\hat{\mathbf{U}}^\top$, Δ_{off} as in Theorem 6.3.2

On trajectory splitting and corrections. To extract the d_K-dimensional subspace, we aim to estimate $\mathbb{E}[\boldsymbol{\beta}\boldsymbol{\beta}^\top] = \mathbf{U}_\star\mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^\top]\mathbf{U}_\star^\top$, which has the same d_K-dimensional span as $\mathbf{U}_\star$. This cannot be achieved by using a single estimator $\hat{\boldsymbol{\beta}}_n$ for each trajectory τ_n and averaging the outer products $\hat{\boldsymbol{\beta}}_n\hat{\boldsymbol{\beta}}_n^\top$ across all n. That is because the per-reward noise ϵ will be shared by both copies of $\hat{\boldsymbol{\beta}}_n$, and so the variance of ϵ will make $\mathbb{E}[\hat{\boldsymbol{\beta}}_n\hat{\boldsymbol{\beta}}_n^\top]$ full rank. We therefore split each trajectory τ_n to obtain two independent estimators $\hat{\boldsymbol{\beta}}_{n,1}, \hat{\boldsymbol{\beta}}_{n,2}$, compute the outer products $\hat{\boldsymbol{\beta}}_{n,1}^\top\hat{\boldsymbol{\beta}}_{n,2}$, and obtain the top d_K eigenvectors of the mean outer product across trajectories.

However, there is a further wrinkle here. We cannot simply take the top d_K eigenvectors of the mean outer product $\overline{\mathbf{M}}_N$. One can compute that $\mathbb{E}[\overline{\mathbf{M}}_N] = \mathbb{E}[\mathbf{D}_{n,1}\boldsymbol{\beta}_n\boldsymbol{\beta}_n^\top\mathbf{D}_{n,2}] = \mathbb{E}[\mathbf{D}_{n,1}\mathbf{U}_\star\boldsymbol{\theta}_n\boldsymbol{\theta}_n^\top\mathbf{U}_\star^\top\mathbf{D}_{n,2}]$. To separate $\mathbb{E}[\boldsymbol{\beta}_n\boldsymbol{\beta}_n^\top]$ from this, we need $\mathbf{D}_{n,1}, \boldsymbol{\beta}_n, \mathbf{D}_{n,2}$ to be independent. If π_b does not use $\boldsymbol{\theta}$ and contexts are generated independently of each other and of $\boldsymbol{\theta}$, then this is satisfied. Intuitively, we need the offline trajectories to be non-adaptive. In fact, we show in the lemma below that if any of these three conditions is violated, then it is in fact impossible to determine the latent subspace $\mathbf{U}^\star$ using *any* method, even with infinitely many infinitely long trajectories.

Lemma 6.3.1 (Contexts, $\boldsymbol{\theta}$, and π_b cannot be dependent)**.** *For each of these conditions:*

1. *Contexts in a trajectory are dependent but do not depend on $\boldsymbol{\theta}$, and π_b also does not use $\boldsymbol{\theta}$,*
2. *Contexts are generated independently using $\boldsymbol{\theta}$, while π_b does not use $\boldsymbol{\theta}$,*
3. *Contexts are generated independently without using $\boldsymbol{\theta}$, while π_b uses $\boldsymbol{\theta}$,*

there exist two different linear latent contextual bandits with orthogonal latent subspaces satisfying the condition, and a behavior policy π_b so that the offline data distributions are indistinguishable and cover all (x, a) pairs with probability at least $1/4$. Since the latent subspaces are orthogonal, an action that gives the maximum reward on one latent bandit gives reward 0 on the other.

To estimate the latent subspace, one is thus forced to make the following assumption.

Assumption 11 (Unconfounded Offline Actions)**.** The offline behavior policy π_b does not use $\boldsymbol{\theta}$ to choose actions, and contexts $x_{n,h}$ are stochastic and generated independently of each other and of $\boldsymbol{\theta}$.

This is satisfied when the offline data comes from randomized controlled trials or A/B testing, which are common sources of offline datasets. Even if this is not satisfied, Algorithm 11 can learn a good subspace whenever $\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1}$ has eigenspace close to the span of $\mathbf{U}_\star$, e.g. when high-reward actions contribute heavily to $\mathbf{D}_{n,i}$. This can happen if offline trajectories were collected to maximize rewards.

This assumption plays a different role from the model separation condition (Assumption 2) in Chapter 3. There, separation makes sense because there are finitely many mixture components, and it ensures their effects on the transition structure are distinguishable. The analogue in our continuous setting is the coverage condition $\lambda_\theta > 0$ (Assumption 12), which ensures the latent population is diverse enough for the subspace to be recoverable. Unconfoundedness is an additional requirement: even with ample coverage, if π_b depends on $\boldsymbol{\theta}$, the offline data can be systematically biased in ways that make the latent subspace unrecoverable (Lemma 6.3.1).

Returning to our scrutiny of $\overline{\mathbf{M}}_N$, let the covariance matrix of $\boldsymbol{\theta}$ be Λ and let its mean be μ_θ. Then we have that $\mathbb{E}[\overline{\mathbf{M}}_N] = \mathbb{E}[\mathbf{D}_{n,1}\boldsymbol{\beta}_n\boldsymbol{\beta}_n^\top\mathbf{D}_{n,2}] = \mathbb{E}[\mathbf{D}_{n,1}]\mathbf{U}_\star(\Lambda + \mu_\theta\mu_\theta^\top)\mathbf{U}_\star^\top\mathbb{E}[\mathbf{D}_{n,2}]$. So, we still

cannot merely consider the top d_K eigenvectors of $\overline{\mathbf{M}}_N$ without accounting for $\mathbf{D}_{n,1}$. Intuitively, $\mathbf{D}_{n,1}$ captures the distortion in reward estimation caused by regularization in ridge regression[2]. We therefore construct correction matrices $\overline{\mathbf{D}}_{N,i}$ and use them to "neutralize" the distortion from regularization. In particular, $\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1}$ is an estimator for $\mathbf{U}_\star(\Lambda + \mu_\theta\mu_\theta^\top)\mathbf{U}_\star^\top$. Crucially, this allows us to aggregate information across many trajectories to overcome the challenge of learning from short trajectories. We can now take the top d_K eigenvectors of $\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1}$ to estimate the subspace determined by $\mathbf{U}_\star$. To give guarantees, we must make a coverage assumption. Unlike in standard offline RL, where only coverage along actions is needed, we also need coverage along latent states.

Assumption 12 (Boundedness and Coverage)**.** Rewards $|r_{n,h}| \leq R$ [3] for all n, h, $\|\phi(x,a)\|_2 \leq 1$ and $\|\boldsymbol{\beta}\|_2 \leq R$. Also, $\lambda_A := \min_{i=1,2}\lambda_{\min}(\mathbb{E}[\mathbf{D}_{n,i}]) > 0$ and $\lambda_\theta := \frac{1}{R^2}\lambda_{\min}(\Lambda) > 0$.

Intuitively, λ_A measures coverage along actions, while λ_θ measures coverage along latent states $\boldsymbol{\theta}$. Both must be non-zero to expect satisfactory estimation of the subspace. Unlike the setting of Yang et al. [2022], whose setting is purely online, we work with an offline dataset of trajectories spanning multiple bandit instances. The learner has no control over the behavior policy that collected the data. Without structural assumptions on the dataset, estimating a useful subspace becomes infeasible, and the regret degenerates to the standard $d_A\sqrt{T}$. Similar coverage assumptions are commonplace within the offline linear MDP literature [Jin et al., 2021b, Duan and Wang, 2020b].

We can then use confidence bounds for $\overline{\mathbf{M}}_N$ and $\overline{\mathbf{D}}_{N,i}$ to give a data-dependent confidence bound Δ_{off} for the projection matrix $\hat{\mathbf{U}}\hat{\mathbf{U}}^\top$, as in Theorem 6.3.2 below. In one instantiation, Propositions D.3.2 and D.3.3 in Appendix D.3 derive simple data-dependent bounds for $\overline{\mathbf{M}}_N$ and $\overline{\mathbf{D}}_{N,i}$ respectively. Under this choice, we control the growth of Δ_{off} in terms of the unknown problem parameters at the end of Theorem 6.3.2.

Theorem 6.3.2 (Computing and Bounding Δ_{off})**.** *Let $\|\overline{\boldsymbol{M}}_N - \mathbb{E}[\boldsymbol{M}_1]\|_2 \leq \Delta_M$ and $\|\overline{\boldsymbol{D}}_{N,i} - \mathbb{E}[\boldsymbol{D}_{n,i}]\|_2 \leq \Delta_D$ for $i = 1, 2$ with probability $1 - \delta/3$ each. Then, with probability $1 - \delta$, $\|\hat{\boldsymbol{U}}\hat{\boldsymbol{U}}^\top - \boldsymbol{U}_\star\boldsymbol{U}_\star^\top\|_2 \leq \Delta_{\text{off}}$, where for $B_D = \|\overline{\boldsymbol{D}}_N^{-1}\|_2$ and $\hat{\lambda} := \lambda_{d_K}(\overline{\boldsymbol{M}}_N) - \lambda_{d_K+1}(\overline{\boldsymbol{M}}_N)$,*

$$\Delta_{\text{off}} = \frac{2\sqrt{2d_K}}{\hat{\lambda}}\left(\frac{B_D^3(2 - B_D\Delta_D)}{(1 - B_D\Delta_D)^2}(R^2 + \Delta_M)\Delta_D + \left(\frac{B_D}{1 - B_D\Delta_D}\right)^2\Delta_M\right).$$

[2] This is not unique to regularization. Pseudo-inverses cause an analogous problem of distortion caused by unseen actions.

[3] R-bounded rewards are automatically R-subgaussian. We can easily extend our results to more general subgaussian rewards, but stick to bounded rewards for simplicity of proofs.

Obtaining Δ_M *and* Δ_D *from Propositions D.3.2 and D.3.3,* $\Delta_{\text{off}} = \widetilde{O}(\frac{1}{\lambda_\theta \lambda_A^3} N^{-1/2} \sqrt{d_K d_A \log(d_A/\delta)})$.

Estimating d_K offline. As our estimator $\overline{\mathbf{D}}_{N,1}^{-1} \overline{\mathbf{M}}_N \overline{\mathbf{D}}_{N,2}^{-1}$ is approximately rank-d_K, the number of nonzero eigenvalues of the estimator is a principled heuristic for determining d_K.

Insufficiency of PCA and PMF for subspace estimation. Naively performing PCA on the raw rewards or on single reward estimates $\hat{\beta}_n$ can lead to erroneous subspaces – as while the PCA target is linear-algebraically similar to $\overline{\mathbf{M}}_N$, it is statistically different. The PCA target (e.g. $\mathbb{E}[\hat{\beta}_n \hat{\beta}_n^\top]$) is typically full rank due to the variance of the per-reward noise ϵ. On the other hand, PMF Mnih and Salakhutdinov [2007b] offers neither confidence bounds on the estimated subspace, nor a principled method for determining d_K.

6.4 Offline Data Sharpens Online Optimism

Here, we motivate and describe LOCAL-UCB, a natural algorithm that accelerates LinUCB with offline data. The core idea is **sharpening optimism** by being optimistic over the intersection of two confidence sets – one obtained using offline and online data and another purely from online data.

We geometrically motivate our update rule here, and illustrate it in Figure 6.1. After any t steps, we can construct a d_K-dimensional confidence ellipsoid for every subspace in the subspace confidence set obtained from SOLD. The union of all these ellipsoids gives us our "offline confidence set"[4], called $\mathcal{C}_{\text{off}}^t(\boldsymbol{\beta})$. The usual d_A-dimensional ellipsoid forms our "online confidence set." We call this $\mathcal{C}_{\text{on}}^t(\boldsymbol{\beta})$. Since the true parameter lies in both confidence sets with high probability, being optimistic over their intersection allows us to sharpen or "further localize" optimism. Even though the offline confidence set uses both offline and online data, it will never shrink to a point due to the frozen subspace confidence set. So, we need the intersection of both sets to be sharply optimistic. This is the intuition behind LOCAL-UCB.

We formalize this intuition in Algorithm 12 by formulating the sharpened optimism as an optimization problem in step 4. The first two constraints represent the low dimensional confidence ellipsoids in the subspace spanned by a given $\mathbf{U}$, while the next two merely represent the usual high dimensional ellipsoid. The remaining constraints let $\mathbf{U}$ range over our subspace confidence set.

We provide the following guarantee for LOCAL-UCB. Notice that our guarantee shows that for enough offline data with $N \gg T$, the effective dimension of the problem is d_K. It increases to d_A as

[4]The set is not only dependent on offline data, since online data is used to construct the d_K-dimensional ellipsoids.

Algorithm 12 Latent Offline subspace Constraints for Accelerating Linear UCB (LOCAL-UCB)

Input: Projection matrix $\hat{\mathbf{U}}\hat{\mathbf{U}}^\top$, confidence bound Δ_{off} from an offline uncertainty-aware method, e.g. SOLD.
Initialize $\mathbf{V}_1 \leftarrow \mathbf{I}_{d_A}$, $\boldsymbol{b}_1 \leftarrow 0$, α_t
for $t = 1, \dots T$ **do**
    Play action a_t and receive reward r_t according to:

$$a_t, \widetilde{\boldsymbol{\beta}}_t, \widetilde{\mathbf{U}}_t \leftarrow \arg\max_{a,\boldsymbol{\beta},\mathbf{U}} \phi(x_t, a)^\top \boldsymbol{\beta} \text{ such that}$$
$$\hat{\boldsymbol{\beta}}_{1,t} \leftarrow \mathbf{U}(\mathbf{U}^T \mathbf{V}_t \mathbf{U})^{-1} \mathbf{U}^\top \boldsymbol{b}_t,$$
$$\|\mathbf{U}^\top(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{1,t})\|_{(\mathbf{U}^\top \mathbf{V}_t \mathbf{U})^{-1}} \le \alpha_{1,t}$$
$$\hat{\boldsymbol{\beta}}_{2,t} \leftarrow \mathbf{V}_t^{-1} \boldsymbol{b}_t, \|(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{2,t})\|_{\mathbf{V}_t^{-1}} \le \alpha_{2,t}$$
$$\|\boldsymbol{\beta}\|_2 \le R, \mathbf{U}^\top \mathbf{U} = \mathbf{I}_{d_K}, \mathbf{U}\mathbf{U}^\top \boldsymbol{\beta} = \boldsymbol{\beta},$$
$$\|\hat{\mathbf{U}}^\top \mathbf{U}\|_F \ge \sqrt{d_K - \Delta_{\text{off}}^2/2}$$

    Compute $\boldsymbol{b}_{t+1} \leftarrow \boldsymbol{b}_t + \phi(x_t, a) r_t$, $\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t + \phi(x_t, a)\phi(x_t, a)^\top$, α_{t+1}
end for

T gets closer to N. The quality of the offline data is reflected in the coverage constants λ_θ and λ_A.

Theorem 6.4.1 (LOCAL-UCB Regret)**.** *Under Assumptions 11 and 12, if* $\alpha_{1,t} = R\sqrt{\mu} + CR\sqrt{d_K \log(2T/\delta)}$ *and* $\alpha_{2,t} = R\sqrt{\mu} + CR\sqrt{d_A \log(2T/\delta)}$ *for a universal constant* C*, then with probability at least* $1 - \delta$ *over offline data and online rewards, LOCAL-UCB has regret* Reg_T *bounded by*

$$O\left(\min\left(Rd_A\sqrt{T}, Rd_K\sqrt{T}\left(1 + \frac{1}{\lambda_\theta \lambda_A^3}\sqrt{\frac{d_A T}{d_K N}}\right)\right)\right).$$

However, the subspace constraint $\|\hat{\mathbf{U}}^\top \mathbf{U}\|_F \ge \sqrt{d_K - \Delta_{\text{off}}^2/2}$ is *nonconvex*. In fact, we lower bound a convex function in the constraint, making us search for $\widetilde{\mathbf{U}}_t$ over a complicated star-shaped set. So, it is unclear if LOCAL-UCB can be made computationally efficient.

6.5 Lower Bound

We now establish that LOCAL-UCB is in fact minimax optimal up to the coverage constants $\lambda_A, \lambda_\theta$ defined in Assumption 12. While we provide a full statement and proof of our lower bound in Appendix D.6, we provide an informal version here. Much like how we generate families of reward parameters in lower bound proofs for purely online regret, we are now generating a family of tuples of latent bandits (for the offline data) and reward parameters represented in the latent bandit (for the

online interaction).

Theorem 6.5.1. *There exists a family of tuples $(F, \boldsymbol{\beta})$, where F is a latent bandit with a rank d_K latent subspace and $\boldsymbol{\beta}$ is a reward parameter in its support, so that for any offline behavior policy π_b and any learner, (i) λ_θ is uniformly bounded from below for all F, (ii) there exists a $(F, \boldsymbol{\beta})$ such that the regret $\mathrm{Reg}(T, \boldsymbol{\beta})$ of the learner under offline data from π_b and F and online reward parameter $\boldsymbol{\beta}$ is bounded below by*

$$\Omega\left(\min\left(d_A\sqrt{T}, d_K\sqrt{T}\left(1+\sqrt{\frac{d_A T}{d_K N}}\right)\right)\right)$$

To the best of our knowledge, this is the first lower bound in a hybrid (offline-online) sequential decision-making setting.[5] The key challenge is in selecting an instance space that yields an informative lower bound. When the offline data has insufficient coverage, one can show a trivial $d_A\sqrt{T}$ lower bound. Assuming λ_θ is uniformly bounded from below for all F models scenarios with sufficient offline coverage, and our lower bound shows that even in these non-trivial settings, no algorithm can achieve a regret better than $\min\left(d_A\sqrt{T}, d_K\sqrt{T}\left(1+\sqrt{\frac{d_A T}{d_K N}}\right)\right)$. While we analyze worst-case performance over a meaningful class of instances where the offline data is of sufficiently high quality, there remains room for future work on sharper, instance-dependent lower bounds that reflect explicit dependence on both λ_θ and λ_A.

The proof, detailed in Appendix D.6, constructs a hypercuboid $\mathcal{B} = \{\pm\Delta_{\text{in}}\}^{d_K-1} \times \{0\} \times \{\pm\Delta_{\text{out}}\}^{d_A-d_K}$ of reward parameters, where $\Delta_{\text{in}} = \Theta(\sqrt{d_K/T})$ and $\Delta_{\text{out}} = \Theta(\sqrt{d_K/N})$. For each $\boldsymbol{\beta} \in \mathcal{B}$, the latent bandit $F_{\boldsymbol{\beta}}$ places uniform weight on the 2^{d_K-1} sign-flips of the first $d_K - 1$ coordinates. The proof lower-bounds the average regret over adjacent pairs in $\mathcal{B}$ using change-of-measure inequalities, treating two cases separately. For within-subspace coordinates $i < d_K$, flipping the sign of β_i does not change $F_{\boldsymbol{\beta}}$, so the offline data distributions are identical and only online interaction distinguishes $\boldsymbol{\beta}$ from $\boldsymbol{\beta}'$. For out-of-subspace coordinates $i > d_K$, $F_{\boldsymbol{\beta}} \neq F_{\boldsymbol{\beta}'}$ and the KL divergence includes a contribution from the N offline trajectories, reflecting the additional information that offline data provides about out-of-subspace directions. Averaging over the full hypercuboid yields the minimax rate.

6.6 Practical Optimism with ProBALL-UCB

While LOCAL-UCB is minimax-optimal, it is not computationally efficient due to the non-convex constraint discussed in Section 6.4. We address this by introducing ProBALL-UCB (Algorithm 13),

[5] Pal et al. [2023] give lower bounds on the cumulative regret for a structure type of latent bandits (with hidden clusters). Their setting is purely online, although they rely on an offline matrix completion oracle during online learning.

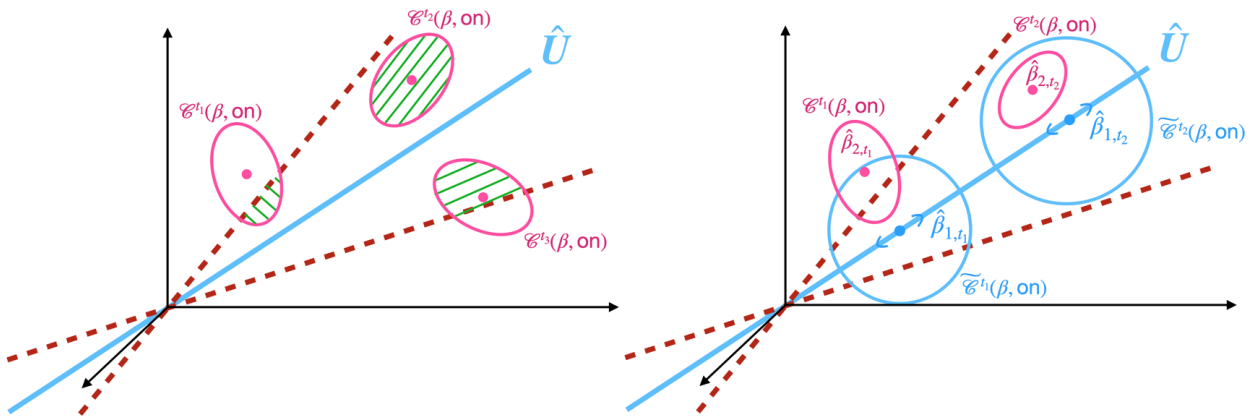

Figure 6.1: **Left:** Geometric interpretations of LOCAL-UCB. Showing $\mathcal{C}^t_{\text{on}}(\boldsymbol{\beta}) \cap \mathcal{C}^t_{\text{off}}(\boldsymbol{\beta})$ in green for three timepoints $t = t_1, t_2, t_3$. The dotted lines delineate the subspace confidence set. **Right:** Geometric interpretation of ProBALL-UCB. $\mathcal{C}^{t_1}_{\text{on}}(\boldsymbol{\beta}) \not\subset \widetilde{\mathcal{C}}^{t_1}_{\text{off}}(\boldsymbol{\beta})$, so we continue to use projections; but by time t_2, $\mathcal{C}^{t_2}_{\text{on}}(\boldsymbol{\beta}) \subset \widetilde{\mathcal{C}}^{t_2}_{\text{off}}(\boldsymbol{\beta})$, so we stop using projections.

a practical and computationally efficient algorithm. In this section, we first sketch the algorithm and then describe how it can be geometrically motivated as a relaxation of LOCAL-UCB.

ProBALL-UCB works in the subspace estimated by SOLD until the online confidence set is small enough. The algorithm maintains a low-dimensional confidence set, a high-dimensional confidence set, and swaps between them to achieve acceleration. Once the cumulative error of using the low-dimensional confidence set ($\approx \Delta_{\text{off}} T$ in ProBALL-UCB) exceeds the cumulative error of using the high-dimensional confidence set ($\approx d_A \sqrt{T}$), we stop using the former. This is instantiated with LinUCB, but the same idea can be immediately applied to other algorithms like SupLinUCB or Bayesian algorithms like Thompson sampling.

We geometrically motivate our update rule here as a relaxation of LOCAL-UCB, and illustrate it in Figure 6.1, like in Section 6.4. We go through three stages of simplification over LOCAL-UCB, which suprisingly only leads to a minor degradation in provable guarantees.

- Cruder offline confidence sets are used. We take the subspace estimated by SOLD, compute a point estimate for $\beta_\star$ within the subspace, and construct a ball that contains the LOCAL-UCB offline confidence set. The online confidence set is still the standard d_A-dimensional ellipsoid.
- We wait for the offline confidence ball to contain the online confidence set, instead of taking intersections.
- We use a computable proxy for this subset condition instead of explicitly checking it.

As a final note before presenting the regret bound proper, there is a technical challenge with analyzing ProBALL-UCB. Since $\hat{\boldsymbol{\beta}}_{1,t}$ lies in $\hat{\mathbf{U}}$ but $\boldsymbol{\beta}_\star$ might not, the d_K-dimensional confidence

Algorithm 13 Projection and Bonuses for Accelerating Latent bandit Linear UCB (ProBALL-UCB)

1: **Input:** Projection matrix $\hat{\mathbf{U}}\hat{\mathbf{U}}^\top$, confidence bound Δ_{off}. Hyperparameters $\alpha_{1,t}, \alpha_{2,t}, \tau, \tau'$.
2: **Initialize** $\mathbf{V}_1 \leftarrow I$, $\boldsymbol{b}_1 \leftarrow 0$, $\mathbf{C}_t \leftarrow 0$
3: **for** $t = 1, \dots T$ **do**
4: **if** $\Delta_{\text{off}}\tau\sqrt{t} + \Delta_{\text{off}}\tau'\sqrt{d_K \sum_{s=1}^{t} \kappa_s^2 / t} \le d_A$ **then**
5: **Compute** $\hat{\boldsymbol{\beta}}_{1,t} \leftarrow \hat{\mathbf{U}}(\hat{\mathbf{U}}^\top \mathbf{V}_t \hat{\mathbf{U}})^{-1} \hat{\mathbf{U}}^\top \boldsymbol{b}_t$
6: **Play** $a_t \leftarrow \arg\max_a \phi(x_t, a)^\top \hat{\mathbf{U}}\hat{\mathbf{U}}^\top \hat{\boldsymbol{\beta}}_{1,t} + \alpha_{1,t} \|\phi(x_t, a)^\top \hat{\mathbf{U}}\|_{(\hat{\mathbf{U}}^\top \mathbf{V}_t \hat{\mathbf{U}})^{-1}}$
7: **else**
8: **Compute** $\hat{\boldsymbol{\beta}}_{2,t} \leftarrow \mathbf{V}_t^{-1} \boldsymbol{b}_t$
9: **Play** $a_t \leftarrow \arg\max_a \phi(x_t, a)^\top \hat{\boldsymbol{\beta}}_{2,t} + \alpha_{2,t} \|\phi(x_t, a)\|_{\mathbf{V}_t^{-1}}$
10: **end if**
11: **Observe** reward r_t and update $\boldsymbol{b}_{t+1} \leftarrow \boldsymbol{b}_t + \phi(x_t, a) r_t$, $\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t + \phi(x_t, a)\phi(x_t, a)^\top$
12: **Update** $\mathbf{C}_{t+1} \leftarrow \mathbf{C}_t + \hat{\mathbf{U}}^\top \phi(x_t, a_t)\phi(x_t, a_t)^\top$, $\kappa_{t+1} \leftarrow \|\mathbf{C}_{t+1}\|_{(\hat{\mathbf{U}}^\top \mathbf{V}_{t+1} \hat{\mathbf{U}})^{-1}}$
13: **end for**

ellipsoid bound no longer applies. We therefore prove our own confidence ellipsoid bound in Lemma D.5.1, to bypass this issue.

Theorem 6.6.1 (Regret for ProBALL-UCB)**.** *Let* $\alpha_{1,t} = R\sqrt{\mu} + \tau' R \Delta_{\text{off}} \kappa_t + CR\sqrt{d_K \log(T/\delta)}$ *and let* $\alpha_{2,t} = R\sqrt{\mu} + CR\sqrt{d_A \log(T/\delta)}$*. Let* S *be the first timestep when Algorithm 13 does not play Line 6 and let* $S = T$ *if no such timestep exists. For* $\tau = \tau' = 1$ *we have that*

$$\text{Reg}_T = \widetilde{O}\left(\min\left(\text{Reg}_{on,T}, \text{Reg}_{hyb,T}\right)\right).$$

where $\text{Reg}_{on,T} = Rd_A\sqrt{T}$ *and* $\text{Reg}_{hyb,T}$ *is defined as*

$$Rd_K\sqrt{T}\left(1 + \frac{1}{\lambda_A^3 \lambda_\theta}\left(\sqrt{\frac{d_A T}{d_K N}} + \sqrt{\frac{d_A}{SN}\sum_{t=1}^{S}\kappa_t^2}\right)\right).$$

In the worst case, $\kappa_t = O(t)$ *and so* $\frac{1}{S}\sum_{t=1}^{S}\kappa_t^2 = O(T^2)$*, but if all features* $\phi(x_t, a_t)$ *lie in the span of* $\hat{\boldsymbol{U}}$ *for* $t \le S$*, then* $\frac{1}{S}\sum_{t=1}^{S}\kappa_t^2 = O(T)$. While the regret bound looks weaker in the worst case, we emphasize that the "good case" in Theorem 6.6.1 is quite common. As an illustrative example, if the feature set $\mathcal{F}_t = \{\phi(x_t, a) \mid a \in \mathcal{A}\}$ is an ℓ_2 ball, then the maximization problem in Step 6 will always choose a_t with $\phi(x_t, a_t)$ in the span of $\hat{\mathbf{U}}$. This can also approximately hold if the features are roughly isotropic or close to the span of $\hat{\mathbf{U}}$. We direct the reader to Appendix D.5.2.1 for further discussion.

Furthermore, Theorem 6.6.1 shows that ProBALL-UCB performs no worse than LinUCB, and can significantly outperform it both in theory and in practice, as we will see in the following section.

A Thompson sampling variant, ProBALL-TS, replaces the optimistic action selection with posterior sampling; we present it below and include experimental comparisons in Section 6.7.

Algorithm 14 Projection and Bonuses for Accelerating Latent bandit Thompson Sampling (ProBALL-TS)

Input: Projection matrix $\hat{\mathbf{U}}\hat{\mathbf{U}}^\top$, confidence bound Δ_{off}. Hyperparameters $\alpha_{1,t}, \alpha_{2,t}, \tau, \tau'$.
Initialize $\mathbf{V}_1 \leftarrow I$, $\boldsymbol{b}_1 \leftarrow 0$, $\mathbf{C}_t \leftarrow 0$
for $t = 1, \dots T$ **do**
  if $\Delta_{\text{off}}\tau\sqrt{t} + \Delta_{\text{off}}\tau'\sqrt{d_K \sum_{s=1}^{t} \kappa_s^2 / t} \leq d_A$ **then**
    Compute $\bar{\boldsymbol{\theta}}_{1,t} \leftarrow (\hat{\mathbf{U}}^\top \mathbf{V}_t \hat{\mathbf{U}})^{-1} \hat{\mathbf{U}}^\top \boldsymbol{b}_t$
    Sample $\hat{\boldsymbol{\theta}}_{1,t} \sim \mathcal{N}\left(\bar{\boldsymbol{\beta}}_{1,t}, \alpha_{1,t}^2 (\hat{\mathbf{U}}^\top \mathbf{V}_t \hat{\mathbf{U}})^{-1}\right)$
    Play $a_t \leftarrow \arg\max_a \phi(x_t, a)^\top \hat{\mathbf{U}} \hat{\boldsymbol{\theta}}_{1,t}$
  else
    Compute $\bar{\boldsymbol{\beta}}_{2,t} \leftarrow \mathbf{V}_t^{-1} \boldsymbol{b}_t$
    Sample $\hat{\boldsymbol{\beta}}_{2,t} \sim \mathcal{N}\left(\bar{\boldsymbol{\beta}}_{2,t}, \alpha_{2,t}^2 \mathbf{V}_t^{-1}\right)$
    Play $a_t \leftarrow \arg\max_a \phi(x_t, a)^\top \hat{\boldsymbol{\beta}}_{2,t}$
  end if
  Observe reward r_t and update $\boldsymbol{b}_{t+1} \leftarrow \boldsymbol{b}_t + \phi(x_t, a) r_t$, $\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t + \phi(x_t, a)\phi(x_t, a)^\top$
  Update $\mathbf{C}_{t+1} \leftarrow \mathbf{C}_t + \hat{\mathbf{U}}^\top \phi(x_t, a_t)\phi(x_t, a_t)^\top$, $\kappa_{t+1} \leftarrow \|\mathbf{C}_{t+1}\|_{(\hat{\mathbf{U}}^\top \mathbf{V}_{t+1} \hat{\mathbf{U}})^{-1}}$
end for

6.7 Experiments

We now establish the practical efficacy of SOLD (Algorithm 11) and ProBALL-UCB (Algorithm 13) for linear latent contextual bandits through a series of numerical experiments.[6] We perform a simulation study and a demonstration using real-life data. While specific details of the experiments and many ablation studies are in Appendix D.8, we sketch our experiments and discuss key observations in this section.

In all experiments, we obtain confidence bounds Δ_{off} using three different concentration inequalities – (1) Hoeffding as in Proposition D.3.3 (H-ProBALL), (2) empirical Bernstein as in Proposition D.3.2 (E-ProBALL), and (3) the martingale Bernstein concentration inequalities of Waudby-Smith and Ramdas [2023] (M-ProBALL). We use a simpler expression for Δ_{off}, set $\tau' = 0$, and choose a suitable value of the hyperparameter τ to adjust for overly conservative Δ_{off}[7]. We later vary τ in ablation experiments to demonstrate that our results are not a consequence of our choice of hyperparameters. Finally, for the MovieLens experiments, we additionally design a natural

[6] See https://github.com/hetankevin/probono for source code.

[7] Namely, we set $\Delta_D = 0$ in Δ_{off}. Also, Lemma D.5.1 and some thought reveal that choosing $\tau' = 0$ recovers the "good" version of ProBALL-UCB guarantees, if features are isotropic enough.

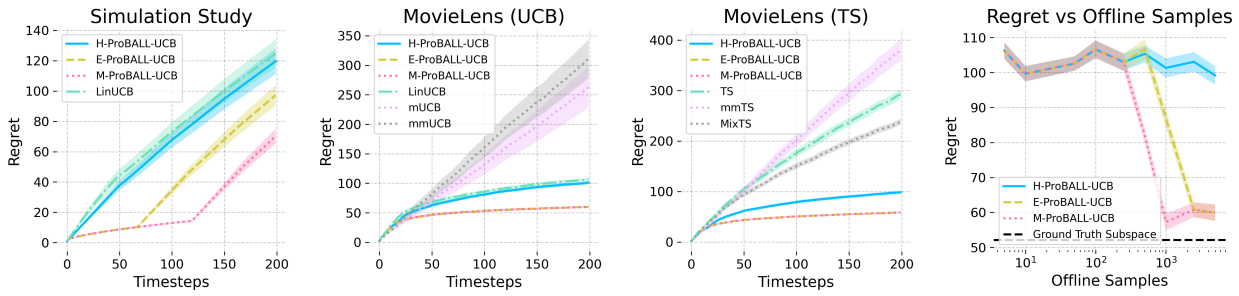

Figure 6.2: Left to Right. **First:** Simulation study comparison of ProBALL-UCB against LinUCB for $\tau = 5$. **Second/Third:** Comparison of ProBALL-UCB initialized with SOLD against {LinUCB, mUCB, and mmUCB, TS, mmTS, and MixTS}, for $\tau = 0.1$ and various confidence bound constructions. ProBALL-UCB outperforms all other algorithms, and approaches the performance of LinUCB when Hoeffding confidence sets are used. **Fourth**: ProBALL-UCB regret on MovieLens against offline samples used in SOLD, compared to LinUCB on ground-truth low-dimensional features. Here, $\tau = 0.1, T = 200$. As the number of offline samples increases, SOLD recovers a low-rank subspace almost as good as ground-truth. The shaded area in each sub-figure depicts 1-s.e. confidence intervals over 30 trials with fresh $\boldsymbol{\theta}$, accounting for the variation in frequentist regret for changing $\boldsymbol{\theta}$.

Thompson sampling version of ProBALL-UCB to highlight the applicability of the ProBALL idea called ProBALL-TS in Appendix D.7. All experiments for ProBALL-TS are in Appendix D.8.3.2.

Simulation study. We first perform a simulation study on a latent linear bandit with $d_A = 50$ and $d_K = 2$, with 5000 trajectories generated offline. Further details are provided in Appendix D.8, and the results are presented in Figure 6.2. Note that ProBALL-UCB (Algorithm 13) performs no worse than LinUCB, no matter what we choose for τ and Δ_{off}. However, we see a clear benefit from using tighter confidence bounds – as Δ_{off} gets smaller, Algorithm 13 chooses to utilize the projected estimate $\hat{\boldsymbol{\beta}}_{1,t}$ more often, resulting in better performance. Note that the kinks in the regret curves correspond to points where Algorithm 13 switches over to the higher dimensional optimism in step 7.

MovieLens dataset. In line with Hong et al. [2020], we assess the performance of our algorithms on real data using the MovieLens dataset. Like them, we filter the dataset to include only movies rated by at least 200 users and vice versa, and apply probabilistic matrix factorization (PMF) to the rating matrix to generate ground truth user preferences for online experiments. Applying PMF gives $d_K = 18$ and we choose $d_A = 200$, generating 5000 trajectories offline. For baselines, we reproduce the methods of Hong et al. [2020, 2022] and implement LinUCB with canonical hyperparameter choices.

We initialize ProBALL-UCB with a subspace estimated with an unregularized variant of SOLD

(see Appendix D.7) that uses pseudo-inverses instead of inverses. This is due to difficulties in finding an appropriate regularization parameter for this large, noisy, and high-dimensional dataset. Figure 6.2 depicts the result of this experiment. Once again, ProBALL-UCB performs no worse than LinUCB, no matter what we choose for τ and Δ_{off}, and the benefit of using tighter confidence bounds remains. With $\tau = 0.5$, ProBALL-UCB with martingale Bernstein confidence bands stops using the projected estimates at around timestep 70, but still continues to outperform LinUCB. Although mUCB and mmUCB perform slightly better than ProBALL-UCB and LinUCB at the beginning, the model misspecification incurred by discretizing the features into d_K clusters ensures that it typically suffers linear regret in this scenario. The lower initial performance of ProBALL-UCB and Lin-UCB is a consequence of their higher initial exploration.

Ablation study. While the end-to-end performance of ProBALL-UCB significantly improves over existing algorithms, we also address further questions about various components of our method in Appendix D.8. We first show that the rank d_K can be determined from offline data via the procedure outlined in Section 6.3 of using the eigenvalues of $\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1}$ to determine the rank of our subspace. Second, we study the effect of varying the hyperparameter τ and note that our method stably outperforms existing methods at all reasonable values of τ. Third, we compare different combinations of algorithms in our figures above, side by side. Finally, we evaluate the effect of offline data by plotting the online regret against the number of offline samples used to estimate the latent subspace.

6.8 How General Are Latent Bandits?

While have established that latent bandits are a powerful framework for accounting for uncertainty in reward models, the extent of their generality is unclear. Are there other stateless decision processes that generalize over latent bandits? We cap off our contributions by establishing the generality of latent bandits. In this section, we show a de Finetti theorem for decision processes, demonstrating that *every* "coherent" and "exchangeable" stateless (contextual) decision process is a latent (contextual) bandit. We first define a stateless decision process at a high level of generality.[8]

Definition 6.8.1. A *stateless decision process* (SDP) with action set $\mathcal{A}$ is a probability space $(\Omega, \mathcal{G}, \mathbb{P})$ with a family of random maps $\mathcal{F}_H : \Omega \to (\mathcal{A}^H \to \mathbb{R}^H)$ for $H \in \mathcal{N} \cup \{\infty\}$. That is, given a sequence of actions $(a_1, \dots a_H)$, an SDP generates a random sequence of rewards $(Y_1, \dots Y_H)$. As such, we abuse notation to denote by $\mathcal{F}_H(a_1, \dots a_H)$ the random variable $\omega \mapsto \mathcal{F}_H(\omega)(a_1, \dots a_H)$. Without any coherence between $\mathcal{F}_H$ across H, a stateless process can behave arbitrarily for different horizons H. We present a natural coherence condition below, essentially requiring that a given

[8] Liu et al. [2023] work with a much more restrictive notion of a generalized bandit and use the original de Finetti theorem in some of their lemmas. See Appendix D.2.1 for a discussion.

action sequence should produce consistent rewards.

Definition 6.8.2. A stateless decision process is *coherent* if for any $h \leq k \leq H, H' \in \mathcal{N} \cup \{\infty\}$ and for any two action sequences τ, τ' of lengths H and H' sharing the same actions $(a_h, \dots a_k)$ from index h to k, with $\mathcal{F}_H(\tau) = (Y_1, \dots Y_H)$ and $\mathcal{F}_{H'}(\tau') = (Y'_1, \dots Y'_{H'})$, we have $(Y_h, \dots Y_k) = (Y'_h, \dots Y'_k)$, viewed as functions of Ω. It is natural to require equality in value and not just in distribution, since after taking an extra action, the *values* of past rewards stay the same, not just their *distribution*. For example, if we pull 10 different jackpot levers and then pull a new one, the previous 10 outcomes stay the same in value, not just in distribution.

The requirement of pointwise coherence (not merely distributional) is necessary. In Appendix D.1.1, we construct an exchangeable SDP with $\theta \sim \text{Ber}(1/2)$ where all rewards are θ for even-length trajectories and $1-\theta$ for odd-length trajectories. This process satisfies distributional coherence but is not coherent, since the reward at position 1 equals θ or $1-\theta$ depending on trajectory length. It is also not a latent bandit: the law condition applied to even and odd H forces $F(a) = \text{Ber}(1/2)$ a.s., but then conditional independence requires independent $\text{Ber}(1/2)$ rewards for even-length trajectories, contradicting the perfect correlation $(Y_1, Y_2) = (\theta, \theta)$.

We also give a natural definition for exchangeability of a stateless decision process – namely that exchanging any two rewards should lead to the distribution obtained by exchanging the corresponding actions.

Definition 6.8.3. A stateless decision process is *exchangeable* if for any permutation $\pi : [H] \to [H]$ and $\mathcal{F}_H(a_1, \dots a_H) = (Y_1, \dots Y_H)$, we have $\mathcal{F}_h(a_{\pi(1)}, \dots a_{\pi(H)}) \sim (Y_{\pi(1)}, \dots Y_{\pi(H)})$. Finally, a latent bandit is an SDP that behaves like a bandit *conditioned* on a random latent state F that determines the reward distribution. As F determines a distribution, it is a random measure-valued function on $\mathcal{A}$.

Definition 6.8.4. A latent bandit is a stateless decision process equipped with a random measure-valued function $F : \Omega \to (\mathcal{A} \to \mathcal{P}(\mathbb{R}))$ so that for any H and action sequence $(a_1, \dots a_H)$, the rewards $(Y_1, \dots Y_H) := \mathcal{F}_H(a_1, \dots a_H)$ are independent conditioned on F. Moreover, the conditional distribution $\mathcal{L}[Y_h \mid F] = F(a_h)$ for all $h \leq H$.[9] As such, the latent bandit is indeed a special case of an SDP, where the function $\mathcal{F}_H$ is induced by the latent state random variable F.

While exchangeability and coherence are reasonable conditions on an SDP and are clearly satisfied by latent bandits, it is a-priori unclear if they are sufficient to ensure that the SDP is a latent bandit. As only exchanging rewards from the *same action* preserves the distribution, standard de Finetti proof ideas do not immediately apply. After all, it is possible that an SDP could be cleverly

[9]We abuse notation twice here. First, we write $F(a_h) := (\omega \mapsto F(\omega)(a_h))$. Second, as the regular conditional distribution $\mathcal{L}[Y_h \mid F]$ is a kernel that maps from $\Omega \times \mathcal{B} \to \mathbb{R}$, we view $F(a_h)$ as its curried map $(\omega, B) \mapsto F(a_h)(\omega)(B)$. A discussion of issues (measurability and well-definedness) is in Appendix D.2.2.

designed to choose rewards adaptively across time and satisfy these properties. Reassuringly, no such counterexamples exist, guaranteed by the following theorem.

Theorem 6.8.1 (De Finetti Theorem for Stateless Decision Processes)**.** *Every exchangeable and coherent stateless decision process is a latent bandit.* We show in Lemma D.1.1 in Appendix D.1 that coherence is not a consequence of exchangeability – it is a necessary condition for being a latent bandit. Finally, we analogously consider contexts and define "transition-agnostic contextual decision processes" (TACDPs) in Appendix D.2.2. We define coherence and coherence and exchangeability for TACDPs, and define latent *contextual* bandits by simply replacing $\mathcal{A}$ with $\mathcal{X} \times \mathcal{A}$ in the definitions above. We then show an analogous de Finetti theorem, as a corollary of our proof of Theorem 6.8.1. See Appendix D.2.6 for more details.

Linear latent contextual bandits and SDPs. Finally, note that this section is faithful to the rest of this paper. A linear latent contextual bandit is a latent contextual bandit where the random measure-valued function $F : \Omega \to ((\mathcal{X} \times \mathcal{A}) \to \mathcal{P}(\mathbb{R}))$ is defined by setting $F(\omega)(x, a)$ to be the distribution given by $\phi(x, a)^\top \mathbf{U}_\star \boldsymbol{\theta}(\omega) + \epsilon$, for any $\omega \in \Omega$ and $(x, a) \in (\mathcal{X} \times \mathcal{A})$. We have seen that every coherent and exchangeable stateless contextual decision process is a latent contextual bandits, of which the linear latent contextual bandit is an important special case.

6.9 Discussion, Limitations and Further Work

In this paper, we have addressed the problem of leveraging offline data to accelerate online learning in linear latent bandits. Our work has a few limitations. First, while ProBALL-UCB is practical and computationally efficient, it has a slightly weaker worse-case guarantee than LOCAL-UCB. Second, when the data is noisy, it can be hard to tune the regularization μ. We use a pseudoinverse-based version of SOLD in such a case (Appendix D.7), implemented in our code. This variant is easy to tune and performs well empirically. Third, the offline uncertainty sets computed using Δ_{off} can be overly conservative, and a discount hyperparameter τ for deciding when to switch between using the offline confidence ball and the online confidence set within ProBALL-UCB must be fine-tuned online.

Despite these limitations, our work enjoys strong theoretical guarantees and convincing empirical performance. We hope that this method opens the door for developing other efficient and scalable algorithms for sequential decision-making with continuous latent states. One can use ideas presented in this paper to design similar algorithms for MDPs, linear MDPs, and RL or bandits with general function approximation.

An alternative is to learn a Bayesian prior over $\boldsymbol{\theta}$ from the offline data — for instance, by

fitting a distribution over the latent subspace estimated by SOLD — and then run Thompson sampling or a posterior-based method online, letting the prior implicitly encode the low-dimensional structure. This could yield Bayesian regret guarantees and may be more natural in practice than the confidence-set-based approach of LOCAL-UCB.

A practical concern is that the coverage constants λ_θ and λ_A governing our guarantees are hard to compute from data. Our algorithms do not require knowledge of these quantities to run — they are fully specified given the offline data and the estimated subspace. However, computing the sample complexity needed for a desired level of statistical accuracy does require knowing these constants. Our experiments suggest that the qualitative predictions of the theory — the $1/\sqrt{N}$ subspace estimation rate and the transition from d_A to d_K scaling — hold in practice even without explicit knowledge of these constants.

CHAPTER 7

Conclusion

7.1 Summary through the Compass

This thesis began with a claim: that many tractable and realistic problems can be found in the intractable ocean of POMDPs by looking for settings where the latent state's influence is *confined* to one part of the observables and acts through a *low-complexity channel* within that part. We can now assess what this perspective bought us.

In Chapters 3 and 4, the latent variable — the road type, the confounder — affects the transition dynamics and the behavior policy, but the reward is defined from observables alone. This confinement decouples reward estimation from the latent structure entirely, and the low-complexity channel (a K-dimensional subspace of transition kernels, a small sensitivity parameter Γ) determines the statistical cost of handling the latent part. In Chapters 5 and 6, the latent variable — the internal state of the human, the latent preference vector — affects the reward, but transitions or contexts remain clean. This confinement lets us estimate the transition model with standard methods and concentrate all algorithmic novelty on the reward channel, whose complexity is governed by the eluder dimension, the history-aware Bellman eluder dimension, or the latent subspace dimension d_K.

The compass did three things. First, it gave us a *language for tractability*: rather than asking "is this POMDP learnable?" in the abstract, we ask where the latent variable enters and how complex its channel is. Second, it produced a *recipe*: decouple the clean and latent-affected parts, handle the clean part with standard methods, handle the latent-affected part by aggregating across trajectories using tools calibrated to the channel's complexity, and compose. Third, it generated *impossibility results that are as informative as the algorithms*: the $\Omega(H)$ hardness for general confounders with memory (Theorem 4.2.6), the unidentifiability of the latent subspace when confinement fails (Lemma 5.2.1), and the exponential separation between model-based and model-free methods on recursive reward structures (Proposition 5.3.3) all arise from understanding exactly which structural

assumption is absent and what breaks without it.

7.2 Cross-Cutting Themes

Several technical themes recur across chapters that are superficially about different problems.

Spectral methods and the double estimator. The double estimator — split each trajectory into two halves, estimate the relevant quantity from each, form the cross-product, and average — appears in Chapter 3 for recovering the subspace of transition kernels and again in Chapter 6 for recovering the latent preference subspace via SOLD. In both cases, the same failure mode motivates the construction: squaring a single per-trajectory estimate and averaging gives a full-rank matrix due to estimation noise, so PCA conflates signal and noise even with infinite data. The double estimator targets the second-moment matrix $\mathbb{E}[\boldsymbol{\beta}_1\boldsymbol{\beta}_2^\top]$ directly, whose rank equals the latent dimension. The technique is not new [Vempala and Wang, 2004], but in each setting it replaces PCA, EM, and probabilistic matrix factorization — all of which fail silently for the reasons discussed in Section 1.4.

The impossibility-then-algorithm pattern. Each chapter first establishes what cannot be done, then designs algorithms that reach the boundary. Chapter 4 proves that FQE incurs irreducible $\Omega(H)$ error under general confounders with memory, then shows that restricting to memoryless or global confounders restores tractability. Chapter 6 proves that the latent subspace is unidentifiable when confinement fails in any of three ways, then shows that under confinement the subspace is recoverable at the $N^{-1/2}$ rate with explicit confidence bounds. Chapter 5 shows that the naive dueling-to-cardinal reduction always fails, then provides a whitebox reduction that works by restricting both policies to the current confidence set. This pattern ensures the assumptions are necessary and the algorithms are tight.

Coverage of the latent space. As discussed in Section 1.4, the settings in this thesis require coverage of the latent space, not just of state-action pairs. The form of this requirement varies — model separation Δ in Chapter 3, the sensitivity parameter Γ in Chapter 4, the coverage constant λ_θ in Chapter 6 — but the underlying demand is the same: the offline population must be diverse enough to span the latent structure.

Aggregation across trajectories. The latent structure is invisible in any single trajectory; it becomes visible only through aggregation across many trajectories. In Chapters 3 and 6, the subspace estimation error decays at the parametric $N^{-1/2}$ rate once trajectory length exceeds a mixing or coverage threshold, shifting the statistical burden from trajectory length to trajectory

count. In Chapter 5, the analogous aggregation happens online across episodes. The latent channel is a population-level object, and recovering it requires seeing the population, not just individual instances.

7.3 Practical Considerations

The guarantees in this thesis are stated in terms of instance-dependent constants — the mixing time t_{mix}, the model separation Δ, the sensitivity parameter Γ, the latent coverage λ_θ, the eluder and coverability dimensions — that are rarely known in advance. Several of these are hard to compute or verify from data alone: t_{mix} requires knowledge of the stationary distribution; confinement (whether the latent variable truly leaves the reward or the transitions clean) is an untestable causal assumption; λ_θ depends on the diversity of latent types in the offline population, which is invisible from the observed data. The theoretical bounds themselves may be loose, since they optimize for the right *scaling* rather than tight constants.

The thesis provides partial remedies. The number of latent types K is estimable from the eigenvalue spectrum of the double estimator (Section 3.5.1), which shows a clear gap between signal and noise eigenvalues in both synthetic and real data. The model separation Δ is visible in scatter plots of pairwise trajectory distances, where well-separated clusters correspond to distinct latent types. The mixing time t_{mix} manifests as a threshold in trajectory length beyond which clustering accuracy improves sharply; the experiments in Chapter 3 show this transition clearly. Overestimating the latent dimension d_K in Chapter 6 is harmless — SOLD learns a subspace containing the true one, and the online algorithm pays for the extra dimensions but does not break. These heuristics do not replace the theoretical assumptions, but they make it possible to apply the methods in settings where the assumptions are plausible but not certifiable.

7.4 Future Directions

The four chapters open several lines of future work that cut across the individual settings.

Function approximation and continuous state spaces. Chapters 3 and 4 operate in the tabular setting. Extending the spectral clustering approach of Chapter 3 to MDPs with large or continuous state spaces — where the transition kernel is not a finite matrix but a conditional density — is open. The double estimator targets the second-moment matrix of per-trajectory statistics; the challenge is defining the right per-trajectory statistic when the state space is continuous and the mixing time is not measured in visits to individual states. For confounded MDPs, function approximation would

allow the sensitivity-based methods of Chapter 4 to handle the continuous-state settings common in healthcare and robotics. Shi et al. [2021] take steps in this direction under bridge function assumptions, but a general treatment remains to be developed.

Real-world dataset applications. The experiments in Chapters 3 and 4 use gridworld environments and the sepsis simulator of Oberst and Sontag [2019]. Applying the clustering and confounded OPE pipeline to real clinical datasets — where the latent confounder is a genuine unrecorded patient characteristic rather than a hidden simulator variable — would test whether the structural assumptions hold in practice and whether the heuristics for checking them (eigenvalue gaps, scatter plots, mixing time thresholds) are reliable outside controlled settings.

Bridging confinement assumptions. The four chapters study two complementary confinement regimes: the latent variable affects transitions but not rewards (Chapters 3 and 4), or the latent variable affects rewards but not transitions (Chapters 5 and 6). What happens in between? If the latent variable has a "strong" effect on one part and a "weak" effect on the other, the clean decoupling breaks down, but a perturbative analysis may be possible: treat the weak effect as misspecification and bound the additional error. The graceful degradation results in Chapter 4 (where the OPE error scales as $O(\varepsilon H^2)$ with the sensitivity $\varepsilon = \Gamma - 1$) and in Chapter 6 (where overestimating d_K is harmless) suggest that such an analysis is feasible, but a unified treatment across settings does not yet exist.

Bayesian and empirical Bayes approaches for latent bandits. Chapter 6 recovers the latent subspace from offline data and then runs a frequentist bandit algorithm online. An alternative is to estimate a prior distribution over the latent preference vector $\boldsymbol{\theta}$ from the offline population — empirical Bayes — and then refine the posterior online using Thompson sampling or a Bayes-UCB variant. This could exploit not just the subspace but also the *distribution* of preferences within it, potentially accelerating adaptation for new users whose preferences are close to common archetypes.

Simulations and practical algorithms for PORRLs. Chapter 5 is the most theoretical of the four chapters; it provides no experiments. Developing simulated RLHF environments with nontrivial internal state dynamics — where the human's mood, trust, or attention evolves over the episode — would test whether the theoretical separation between model-based and model-free methods (governed by the history-aware Bellman eluder dimension) manifests in practice. More broadly, the optimistic algorithms POR-UCRL and POR-UCBVI are designed for theoretical analysis; translating them into practical perturbative or posterior-sampling variants, following the path from UCRL to PSRL or from optimistic Q-learning to bootstrapped DQN, is an important next step.

Cost-aware observation of the latent state. This thesis assumes the latent variable is genuinely inaccessible. In many applications, the agent can pay a cost to observe it: running a diagnostic test, deploying an additional sensor, or asking a user to complete a preference survey. This leads to cost-aware decision-making frameworks where the agent chooses *when* to invest in observing the latent state and when to rely on the structural assumptions developed here. The value of observation depends on the quality of the low-complexity channel: if the channel is already highly informative (small d_K, large Δ), the marginal benefit of direct observation is low, and the methods of this thesis may suffice without ever paying the observation cost.

7.5 Closing Remarks

Partial observability pervades real sequential decision-making. The general POMDP is intractable, but many applications carry structure that generic frameworks ignore. This thesis has developed one systematic approach — confinement and low-complexity channels — for identifying and exploiting that structure, with algorithms and matching lower bounds across four settings spanning offline and online learning, MDPs and bandits, cardinal and dueling feedback.

APPENDIX A

Supplementary Material for Chapter 2

This appendix contains proofs and supplementary material for Chapter 3.

A.1 Additional Figures

All figures here pertain to the gridworld experiment in Section 3.6.

A.1.1 Determining K

See Figure 3.2 in Section 3.5.1.

A.1.2 Block Matrix of Raw Distance Estimates

See Figure A.1 below, which presents the raw distance matrix before thresholding, to provide a sense of the quality of the pairwise distance estimates themselves. These could also be used for agglomerative clustering, for example.

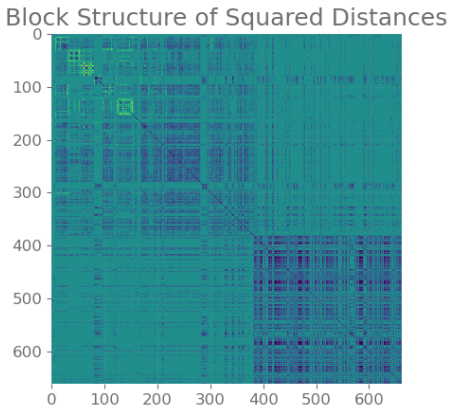

Figure A.1: Block structure of the matrix of squared pairwise distance estimates (after sorting).

A.1.3 Determining The Threshold τ

See Figure 3.3 in Section 3.5.2.

A.1.4 Local Extrema in EM

See Figure 3.6 in Section 3.6, illustrating how EM often gets stuck in suboptimal local extrema, given by the low final log-likelihood values recorded in the scatterplot.

A.1.5 Comparing End-To-End Performance Using Soft and Hard EM

We compare various initializations of EM – (1) random initializations, (2) models from $\mathcal{N}_{clust}$, and (3) classification and clustering labels from $\mathcal{N}_{clust}$ and $\mathcal{N}_{sub}$ – this time using both soft and hard EM.

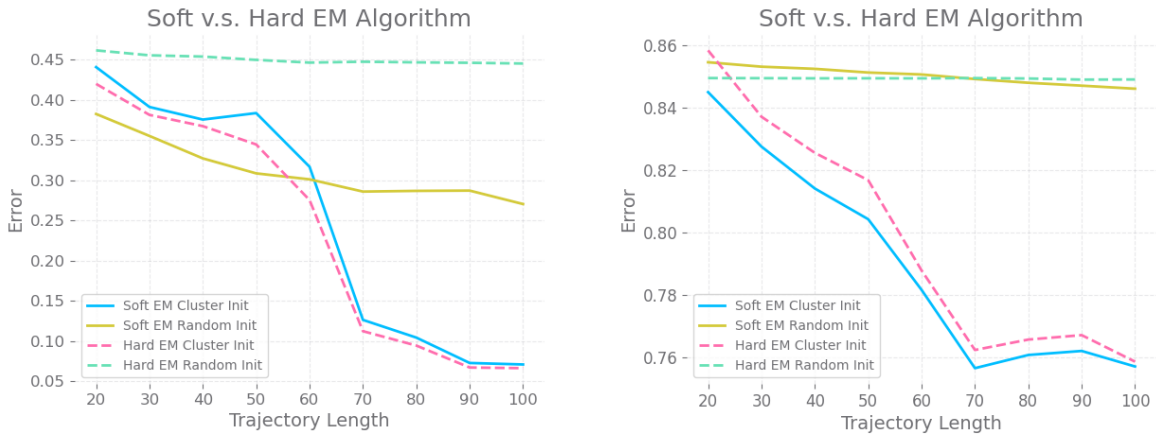

Figure A.2: End-to-end error v.s. trajectory length on (left) 1000 MDP trajectories from the gridworld dataset and (right) 750 Markov chain trajectories from the Last.fm dataset, comparing various initializations of the soft and the hard EM algorithm. Results averaged over 30 trials, with 30 random initializations for randomly-initialized EM within each trial.

A.2 Discussion on Using Random Projections

We note that those familiar with the intuition behind the Johnson-Lindenstrauss lemma would guess that a projection to a random n-dimensional subspace for low n would preserve distances with good accuracy. However, note that the bound on the dimension n needed to preserve distances between our N_{clust} estimators up to a multiplicative distortion of $1 \pm \epsilon$ is $\frac{\log(N_{clust})}{\epsilon^2}$. This bound is known to be tight, see for example Larsen and Nelson [2017]. Upon thought, this shows that to get good distortion bounds (which will contribute to the deviation between distance estimates and the

thresholds), we need a large dimension, interpreted as being affected by the $1/\epsilon^2$. In fact, as soon as $\log(N_{clust})$ exceeds 1, we will need a dimension of order $1/\Delta^2$, while K can be arbitrarily small compared to this.

In the gridworld case, $K = 2$, and we see that we don't get good performance using a random subspace until we hit dimension 50, where the maximum dimension is $S = 64$. Clearly, the $1/\epsilon^2$ term in the Johnson-Lindenstrauss lemma drastically affects the performance of using random subspaces. Using a random subspace of dimension 50 for $S = 64$ is much closer to not projecting at all than to using a subspace of dimension 2.

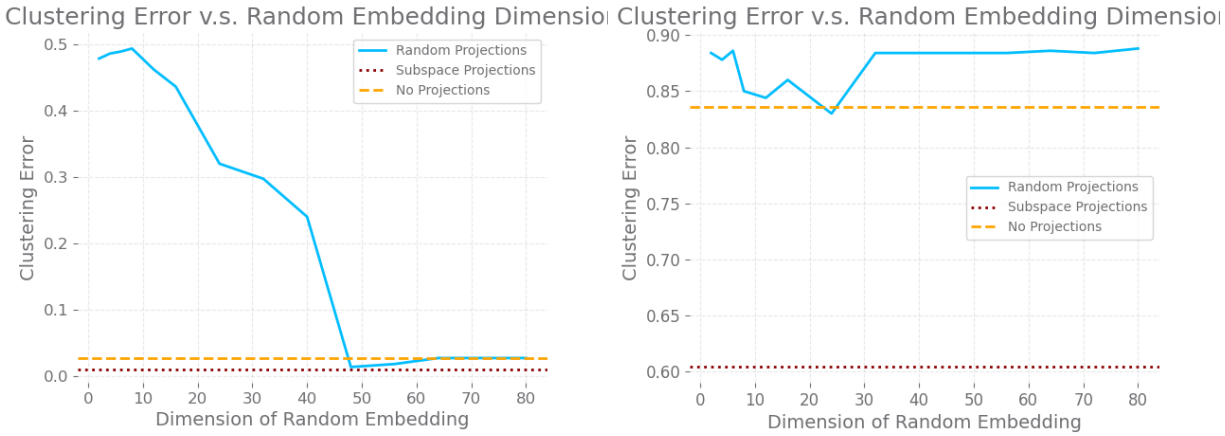

Figure A.3: Clustering error using random projections of varying dimension for a trajectory length of 100, benchmarked against the performance of the "with subspace" and "without subspace" versions. The gridworld MDP dataset is on the left, while the Last.fm Markov chain dataset is on the right.

A.3 Details of the EM Algorithm

We describe the E and M steps for hard EM below first, for simplicity.

M-step: Given the cluster labels, we can estimate each model with the MLE as:

$$\hat{\mathbb{P}}_k(s'|s,a) \leftarrow \frac{\sum_{n \in \mathcal{N}_{clust}} \mathbb{1}_{n \in \mathcal{C}_k} N(n,s,a,s')}{\sum_{n \in \mathcal{N}_{clust}} \mathbb{1}_{n \in \mathcal{C}_k} N(n,s,a)}$$

$$\hat{f}_k \leftarrow \frac{\sum_{n \in \mathcal{N}_{clust}} \mathbb{1}_{n \in \mathcal{C}_k}}{N_{clust}} = \frac{|\mathcal{C}_k|}{N_{clust}}$$

Readers can convince themselves that this is truly the MLE estimate by making the following observation. We can write the log-likelihood of the predicted clusters $\mathcal{C}_k$ and estimated models

as $\sum_{k=1}^{K}\sum_{n\in\mathcal{N}_{clust}}\mathbb{1}_{n\in\mathcal{C}_k}\ell(\hat{\mathbb{P}}_k,\hat{f}_k,n)$, where $\ell(\hat{\mathbb{P}}_k,\hat{f}_k,n)=\log\left(f_k\prod_{s,s',a}(\hat{\mathbb{P}}_k(s'\mid s,a))^{N(n,s,a,s')}\right)$. The rest of the derivation mimics the well-known and straightforward computation for Markov chains, using Lagrange multipliers to constrain the estimates to probability distributions.

E-step: On new or unseen data, assign cluster membership according to the following rule:

$$k_m \leftarrow \arg\max_k \ell(\hat{\mathbb{P}}_k,\hat{f}_k,m)+\log(\hat{f}_i) \tag{A.1}$$

where $\ell(\hat{\mathbb{P}}_k,m)$ is as above.

A.4 The Classification Algorithm

The classification algorithm is presented as Algorithm 3 in Chapter 3. Note that we define a new quantity, $\hat{f}_{k,s,a}$, which is the proportion of trajectories with label k among all trajectories in $\mathcal{N}_{clust}$ where s,a is observed. Quantities $N(n,s,a),\mathbf{N}(n,i,s,a,\cdot)$ and $N(n,i,s,a)$ carry their usual meanings with respect to either $\mathcal{N}_{clust}$ until step 5 and with respect to $\mathcal{N}_{class}$ after that.

A.5 Proof of Theorem 3.4.2

A.5.1 Proof of the theorem

We recall the theorem here.

Theorem 3.4.2 (Subspace Estimation Guarantee)**.** *Consider 2 models with labels k_1,k_2 and a state-action pair s,a with $d_{min}(s,a)\geq\alpha/3$. Consider the output $\boldsymbol{V}_{s,a}^T$ of Algorithm 1. Let $f_{min}=\min(f_{k_1},f_{k_2})$ be the lower of the label prevalences. Remember that each trajectory has length T_n.*

Then given that $N_{sub}=\Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$, $T_n=\Omega(t_{mix}\log^4(1/\alpha))$, with probability at least $1-\delta$, for $k=k_1,k_2$

$$\|\mathbb{P}_k(\cdot\mid s,a)-\boldsymbol{V}_{s,a}\boldsymbol{V}_{s,a}^T\mathbb{P}_k(\cdot\mid s,a)\|_2\leq\epsilon_{sub}(\delta)$$

where

- *For $T_n=\Omega\left(t_{mix}\log^3\left(\frac{f_{min}N_{sub}\alpha}{KS\log(1/\delta)}\right)\right)$*

$$\epsilon_{sub}(\delta)=O\left(\sqrt{\frac{K}{f_{min}}\left(\sqrt{\frac{S}{N_{sub}\cdot\alpha^3}\log\left(\frac{1}{\delta}\right)}\right)}\right)$$

- *While for* $T_n = O\left(t_{mix}\log^3\left(\frac{f_{min}N_{sub}\alpha}{KS\log(1/\delta)}\right)\right)$

$$\epsilon_{sub}(\delta) = O\left(\left(\frac{1}{2}\right)^{\frac{1}{16}\left(\frac{T_n}{t_{mix}}\right)^{1/3}}\right)$$

Alternatively, we only need $N_{sub} = \Omega\left(\frac{K^2S\log(1/\delta)}{f_{min}^2\alpha^3\epsilon^4}\right)$ *and* $T_n = \Omega\left(t_{mix}\log^3(1/\epsilon)\log^4(1/\alpha)\right)$ *trajectories for* ϵ *accuracy in subspace estimation with probability at least* $1-\delta$.

Remark 9. We can convert the α^3 in the denominator to an α at the cost of making T_n more heavily dependent on α (more than just $\log(1/\alpha)$). Intuitively, α accounts for the probability of not observing s, a, so this is just saying that we can shift the onus for that from the number of trajectories to their length. We chose not to do that since we are trying to minimize the length of trajectories needed, and assume that we have access to many trajectories.

Proof. The main input is the proposition below, proved in the next section.

Proposition A.5.1. *Consider* $L < K$ *models with labels* j_l, $1 \le l \le L$, *with* $d_{min}(s,a) := \min_l d_{j_l}(s,a)$. *Consider the output* $\boldsymbol{V}_{s,a}^T$ *of Algorithm 1. Let* $f_{min} = \min_l f_{j_l}$ *be the minimum frequency across these models in the mixture. Remember that each trajectory has length* T_n. *Then we have the guarantee that with probability at least* $1-\delta$

$$\|\mathbb{P}_j(\cdot \mid s,a) - \boldsymbol{V}_{s,a}\boldsymbol{V}_{s,a}^T\mathbb{P}_j(\cdot \mid s,a)\|_2$$

is bounded above by

$$\sqrt{\frac{4K}{f_{min}d_{min}(s,a)}\left(\sqrt{\frac{128}{N_{sub}\cdot d_{min}(s,a)}(2S\log(12)+\log(4/\delta))} + \left(\frac{1}{2}\right)^{\frac{T_n}{8Gt_{mix}}}\right)}$$

for all $j \in \{j_l \mid 1 \le l \le L\}$, *when* $N_{sub} \ge \frac{32}{d_{min}(s,a)^2}\log\left(\frac{1}{\delta}\right)$ *and* $\frac{T_n}{8t_{mix}} > \frac{G\log(48G/d_{min}(s,a))}{\log 2}$.

For a state-action pair with $d_{min}(s,a) \ge \alpha/3$, the conditions simplify to $N_{sub} \ge \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$ and $T_n \ge \Omega(Gt_{mix}\log(G/\alpha))$. We set $G = \left(\frac{T_n}{t_{mix}}\right)^{\frac{2}{3}}$ to get bounds that only depend on T_n. Note that this means a sufficient condition on T_n is $T_n \ge \Omega(t_{mix}\log^4(1/\alpha))$ (one can show this with an elementary computation). Also note that

$$\sqrt{\frac{S+\log(1/\delta)}{N_{sub}\cdot\alpha}} \le \sqrt{\frac{S\log(1/\delta)}{N_{sub}\cdot\alpha}}$$

Then with probability at least $1-\delta$, the following bound holds for any label $j = j_l$ for some l.

$$\|\mathbb{P}_j(\cdot \mid s,a) - \mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbb{P}_j(\cdot \mid s,a)\|_2 \leq O\left(\sqrt{\frac{K}{f_{min}\alpha}\left(\sqrt{\frac{S\log(1/\delta)}{N_{sub}\cdot\alpha}} + \left(\frac{1}{2}\right)^{\frac{1}{8}\left(\frac{T_n}{t_{mix}}\right)^{1/3}}\right)}\right)$$

So, there is a universal constant C_2 so that for $T_n > C_2 t_{mix}\log^3\left(\frac{f_{min}N_{sub}\alpha}{KS\log(1/\delta)}\right)$,

$$\left(\frac{1}{2}\right)^{\frac{1}{8}\left(\frac{T_n}{t_{mix}}\right)^{1/3}} \leq C'\frac{K}{f_{min}}\left(\sqrt{\frac{S}{N_{sub}\cdot\alpha^3}\log\left(\frac{1}{\delta}\right)}\right)$$

While for $T_n = O\left(t_{mix}\log^3\left(\frac{f_{min}N_{sub}\alpha}{KS\log(1/\delta)}\right)\right)$,

$$\frac{K}{f_{min}}\left(\sqrt{\frac{S}{N_{sub}\cdot\alpha^3}\log\left(\frac{1}{\delta}\right)}\right) \leq O\left(\left(\frac{1}{2}\right)^{\frac{1}{8}\left(\frac{T_n}{t_{mix}}\right)^{1/3}}\right)$$

So, combining all these, for $N_{sub} = \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$, $T_n = \Omega(t_{mix}\log^4(1/\alpha))$

- For $T_n = \Omega\left(t_{mix}\log^3\left(\frac{f_{min}N_{sub}\alpha}{KS\log(1/\delta)}\right)\right)$

$$\epsilon_{sub}(\delta) = O\left(\sqrt{\frac{K}{f_{min}}\left(\sqrt{\frac{S}{N_{sub}\cdot\alpha^3}\log\left(\frac{1}{\delta}\right)}\right)}\right)$$

- While for $T_n = O\left(t_{mix}\log^3\left(\frac{f_{min}N_{sub}\alpha}{KS\log(1/\delta)}\right)\right)$

$$\epsilon_{sub}(\delta) = O\left(\left(\frac{1}{2}\right)^{\frac{1}{16}\left(\frac{T_n}{t_{mix}}\right)^{1/3}}\right)$$

□

A.5.2 Proof of the Proposition A.5.1

We recall the proposition here.

Proposition A.5.1. *Consider $L < K$ models with labels j_l, $1 \leq l \leq L$, with $d_{min}(s,a) := \min_l d_{j_l}(s,a)$. Consider the output $\boldsymbol{V}_{s,a}^T$ of Algorithm 1. Let $f_{min} = \min_l f_{j_l}$ be the minimum frequency across these models in the mixture. Remember that each trajectory has length T_n. Then we have the guarantee that with probability at least $1-\delta$*

$$\|\mathbb{P}_j(\cdot \mid s,a) - \boldsymbol{V}_{s,a}\boldsymbol{V}_{s,a}^T\mathbb{P}_j(\cdot \mid s,a)\|_2$$

is bounded above by

$$\sqrt{\frac{4K}{f_{min}d_{min}(s,a)}\left(\sqrt{\frac{128}{N_{sub}\cdot d_{min}(s,a)}(2S\log(12)+\log(4/\delta))} + \left(\frac{1}{2}\right)^{\frac{T_n}{8Gt_{mix}}}\right)}$$

for all $j \in \{j_l \mid 1 \leq l \leq L\}$, when $N_{sub} \geq \frac{32}{d_{min}(s,a)^2}\log\left(\frac{1}{\delta}\right)$ and $\frac{T_n}{8t_{mix}} > \frac{G\log(48G/d_{min}(s,a))}{\log 2}$.

Remark 10. We should point out that we will only need $L = 2$ for subsequent theorems. Also, remember that only s, a with $d_{min}(s,a) > \alpha$ will be relevant in subsequent theorems, with α as in our assumption.

Proof. For brevity of notation, we will denote $c_{n,i} := N(n,i,s,a)$, $\mathbf{w}_{n,i} := N(n,i,s,a,\cdot)$ and suppress the (s,a) dependence. We will first need the following lemma which guarantees that we can get past mixing and concentration hurdles with our estimator, modulo actually observing s, a in both segments.

Lemma A.5.2. *Let $\mathcal{B}_n$ be the event given by $n \in \mathcal{N}_{traj}(s,a)$, which is the same as $c_{n,1}c_{n,2} \neq 0$ and let*

$$\boldsymbol{M}_{s,a} = \sum_{j=1}^{K}\mathbb{P}(k_n = j \mid \mathcal{B}_n)\mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T$$

.

Call our estimator $\hat{\boldsymbol{M}}_{s,a}$. Then we know that

$$\hat{\boldsymbol{M}}_{s,a} = \frac{1}{N_{traj}(s,a)}\sum_n \hat{\mathbb{P}}_{n,1}(\cdot \mid s,a)\hat{\mathbb{P}}_{n,2}(\cdot \mid s,a)^T$$

and we have

$$\|\hat{\boldsymbol{M}}_{s,a} - \boldsymbol{M}_{s,a}\| < \sqrt{\frac{32}{N_{traj}(s,a)}(2S\log(12)+\log(\frac{2}{\delta}))} + \frac{48G}{d_{min}(s,a)}\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}}$$

Remark 11. Note that since all trajectories are generated independently of each other and the process that generates them is identical, $\mathbb{P}(k_n = j \cap \mathcal{B}_n)$ is the same for all n. A similar observation holds for many conditional/unconditional probabilities and conditional/unconditional expectations in this proof, and will not be stated again.

Assume the lemma for now. The proof is delayed to after the proof of the theorem. We will combine this lemma with Lemma 3 from Chen and Poor [2022]. In the context of their lemma, $p^{(j)} = \mathbb{P}(k_n = j \mid \mathcal{B}_n)$, $\mathbf{y}^{(j)} = \mathbb{P}_j(\cdot \mid s, a)$. Now, we can use the first term on the right-hand side of the bound in Lemma 3 of Chen and Poor [2022] to get that for any $1 \leq l \leq L$

$$\|\mathbb{P}_{j_l}(\cdot \mid s, a) - \mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbb{P}_{j_l}(\cdot \mid s, a)\|_2 \leq \sqrt{\frac{2K}{\min_l(\mathbb{P}(k_n = j_l \mid \mathcal{B}_n))}\|\hat{\mathbf{M}}_{s,a} - \mathbf{M}_{s,a}\|} \tag{A.2}$$

A.5.2.1 Lower Bounding $\mathbb{P}(k_n = j_l \mid \mathcal{B}_n)$

Note that

$$\mathbb{P}(k_n = j_l \mid \mathcal{B}_n) = \frac{\mathbb{P}(k_n = j_l)\mathbb{P}(\mathcal{B}_n \mid k_n = j_l)}{\mathbb{P}(\mathcal{B}_n)} \geq f_{j_l}\mathbb{P}(\mathcal{B}_n \mid k_n = j_l)$$

So, we need only lower bound $\mathbb{P}(\mathcal{B}_n \mid k_n = j_l)$, for which we will need a lemma. We will use the following crucial lemma several times in our proofs. This is where we use Yu [1994]'s blocking technique.

Lemma A.5.3. *Consider a function h on segments of a Markov chain with mixing time $t_{mix} = t_{mix}(\frac{1}{4})$ with $C = \sup h$. Consider the joint distribution χ over the product of the σ-algebras of n such segments, with marginals χ_i. Let the product distribution of the marginals χ_i be called Ξ. Then for $\lambda = (\frac{1}{4})^{\frac{1}{t_{mix}}}$ and for the minimum distance between consecutive segments being a_n, we have*

$$|\mathbb{E}_\chi h - \mathbb{E}_\Xi h| \leq 4C(n-1)\lambda^{a_n}$$

Proof. Remember that each of our Markov processes is mixing, so there exists $t_{mix,j} = t_{mix,j}(\frac{1}{4})$ and a stationary distribution d_j so that $TV(P_j^n(x, \cdot), d_j) < \frac{1}{4}$ for $n \geq t_{mix,j}$. Let $t_{mix} = \max_j t_{mix,j}$. Since the decay in total variation distance is multiplicative, $TV(P_j^n(x, \cdot), d_j) < (\frac{1}{4})^c$ for all j and $n \geq ct_{mix}$. This implies that

$$\max_j TV(P_j^n(x, \cdot), d_j) < \left(\frac{1}{4}\right)^{\frac{T_n}{4t_{mix}}-1} = 4\lambda^n$$

where $\lambda = (\frac{1}{4})^{\frac{1}{t_{mix}}}$

This means that we satisfy the definition of V-geometric ergodicity from Vidyasagar [2010], with V being the constant function with value 1, $\mu = 4$ and λ as above. That means that any of our processes is beta-mixing by (the proof of) Theorem 3.10 from the text and

$$\beta_n \leq \mu\lambda^n = 4\lambda^n$$

we employ an argument analogous to the setup and argument used to prove Lemma 4.1 of Yu [1994], merely with H_i's replaced by the segments of arbitrary length instead of a_n-sized blocks while T_i's stay at a_n sized blocks. Then, Q from Corollary 2.7 is the probability distribution of the segments here, Ω_i from Corollary 2.7 is the real vector space of the same dimension as the length of the i^{th} segment, Σ_i is the product Borel field on this vector space and m in the theorem is the number of segments n here (note that n is called μ_n in Lemma 4.1). $\tilde{Q}$ is the product distribution over the marginals of Q, as in the theorem. Note that $\beta(Q)$ from Corollary 2.7 used in the proof remains less than β_{a_n}. Now we can directly quote Corollary 2.7 to conclude that

$$|\mathbb{E}_\chi h - \mathbb{E}_\Xi h| \leq C(n-1)\beta_{a_n} \leq 4C(n-1)\lambda^{a_n}$$

□

Define

$$h = \mathbb{1}_{(c_{n,1}c_{n,2}=0)}$$

We are now ready to bound $P(\mathcal{B}_n \mid k_n = j) = P(c_{n,1}c_{n,2} = 0 \mid k_n = j)$. Consider the joint distribution over the segments Ω_1 and Ω_2 of a trajectory sampled from hidden label j. Call this χ and let its marginals on Ω_i be χ_i. Let the product distribution of its marginals be $\Xi := \chi_1 \times \chi_2$. Notice that then

$$\mathbb{E}_\Xi h = P(c_{n,1} = 0 \mid k_n = j)P(c_{n,2} = 0 \mid k_n = j)$$

by definition of Ξ. Also, clearly we have

$$\mathbb{E}_\chi h = P(c_{n,1}c_{n,2} = 0 \mid k_n = j)$$

Now, using Lemma A.5.3, we get that for $C = \sup h = 1$ and $n = 2$, we have the following

inequality.

$$|P(c_{n,1}c_{n,2}=0 \mid k_n=j) - P(c_{n,1}=0 \mid k_n=j)P(c_{n,2}=0 \mid k_n=j)| = |\mathbb{E}_\chi h - \mathbb{E}_\Xi h| \leq 4\lambda^{\frac{T_n}{4}} \tag{A.3}$$

Additionally, for $i=1,2$, if $d_{t,j}(s,a)$ is the distribution at time t, the following is obtained by the definition of mixing times.

$$\begin{aligned}
\mathbb{P}(c_{n,i}=0 \mid k_n=j) &\leq (1-d_{(2i-1)T,j}(s,a)) \\
&\leq (1-d_j(s,a)+TV(d_{(2i-1)T,j},\pi)) \\
&\leq (1-d_{min}(s,a)+4\lambda^{\frac{T_n}{4}}) \\
&\leq \left(1-\frac{d_{min}(s,a)}{2}\right)
\end{aligned} \tag{A.4}$$

where the last inequality holds for $T_n > 4t_{mix}\frac{\log(8/d_{min}(s,a))}{\log 4}$. This allows us to use inequality A.3 and

$$\begin{aligned}
P(c_{n,1}c_{n,2}=0 \mid k_n=j) &\leq 4\lambda^{\frac{T_n}{4}} + P(c_{n,1}=0 \mid k_n=j)P(c_{n,2}=0 \mid k_n=j) \\
&\leq 4\lambda^{\frac{T_n}{4}} + \left(1-\frac{d_{min}(s,a)}{2}\right)^2 \\
&\leq 1-d_{min}(s,a)+\frac{d_{min}(s,a)^2}{4}+4\lambda^{\frac{T_n}{4}} \\
&\leq 1-d_{min}(s,a)+\frac{d_{min}(s,a)}{4}+4\lambda^{\frac{T_n}{4}} \\
&\leq 1-\frac{d_{min}(s,a)}{2}
\end{aligned} \tag{A.5}$$

where the last inequality holds for $T_n > 4t_{mix}\frac{\log(16/d_{min}(s,a))}{\log 4}$. We conclude that for $T_n > 4t_{mix}\frac{\log(16/d_{min}(s,a))}{\log 4}$, and $j=j_l$ for some l,

$$P(\mathcal{B}_n \mid k_n=j) \geq \frac{d_{min}(s,a)}{2}$$

And so,

$$\min_l f_{j_l}(\mathbb{P}(k_n=j_l \mid \mathcal{B}_n)) \geq \min_l f_{j_l}(\mathbb{P}(k_n=j_l \cap \mathcal{B}_n))$$

$$\geq \min_l f_{j_l}(\mathbb{P}(\mathcal{B}_n \mid k_n = j)\mathbb{P}(k_n = j))$$
$$\geq \frac{f_{min} d_{min}(s,a)}{2}$$

We can thus conclude that for $T_n > 4t_{mix}\frac{\log(16/d_{min}(s,a))}{\log 4}$,

$$\|\mathbb{P}_{j_l}(\cdot \mid s,a) - \mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbb{P}_{j_l}(\cdot \mid s,a)\|_2 \leq \sqrt{\frac{4K}{f_{min} d_{min}(s,a)}\|\hat{\mathbf{M}}_{s,a} - \mathbf{M}_{s,a}\|} \tag{A.6}$$

A.5.2.2 Absorbing the extra terms into the exponent of $1/4$

Now remember from Lemma A.5.2 that

$$\|\hat{\mathbf{M}}_{s,a} - \mathbf{M}_{s,a}\| < \sqrt{\frac{32}{N_{traj}(s,a)}(2S\log(12) + \log(\frac{2}{\delta}))} + \frac{48G}{d_{min}(s,a)}\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}}$$

Notice that for $\frac{T_n}{8t_{mix}} > \frac{G\log(48G/d_{min}(s,a))}{\log 2} > \frac{\log(16/d_{min}(s,a))}{2\log 4}$, we have that

$$\begin{aligned}
\frac{48G}{d_{min}(s,a)}\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}} &= \frac{48G}{d_{min}(s,a)}\left(\frac{1}{4}\right)^{\frac{T_n}{16Gt_{mix}}}\left(\frac{1}{4}\right)^{\frac{T_n}{16Gt_{mix}}} \\
&= \frac{48G}{d_{min}(s,a)}\left(\frac{1}{2}\right)^{\frac{T_n}{8Gt_{mix}}}\left(\frac{1}{2}\right)^{\frac{T_n}{8Gt_{mix}}} \\
&\leq \left(\frac{1}{2}\right)^{\frac{T_n}{8Gt_{mix}}}
\end{aligned}$$

A.5.2.3 Bounding the concentration term

We finally need to bound $N_{traj}(s,a)$ from below to bound the first term in this sum. Note that $\mathbb{E}[N_{traj}(s,a)] \geq N_{sub}(1 - P(c_{n,1}c_{n,2} = 0)) \geq N_{sub}\frac{d_{min}(s,a)}{2}$ from equation A.5 above. Now, by Hoeffding's inequality, we have

$$\begin{aligned}
\mathbb{P}\left(N_{traj}(s,a) < N_{sub}\frac{d_{min}(s,a)}{4}\right) &= \mathbb{P}\left(N_{traj}(s,a) < N_{sub}\frac{d_{min}(s,a)}{2} - N_{sub}\frac{d_{min}(s,a)}{4}\right) \\
&\leq \mathbb{P}\left(N_{traj}(s,a) < \mathbb{E}[N_{traj}(s,a)] - N_{sub}\frac{d_{min}(s,a)}{4}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{P}\Bigg(\sum_{n \in \mathcal{N}_{sub}} \mathbb{1}_{c_{n,1}c_{n,2} \neq 0} \\
&\qquad < N_{sub}\mathbb{E}[\mathbb{1}_{c_{n,1}c_{n,2}\neq 0}] - N_{sub}\frac{d_{min}(s,a)}{4}\Bigg) \\
&\leq \exp\left(-\frac{d_{min}(s,a)^2 N_{sub}}{8}\right)
\end{aligned}
$$

This is less than δ for $N_{sub} \geq \frac{8}{d_{min}(s,a)^2}\log\left(\frac{1}{\delta}\right)$.

Combining this with equation A.6 and splitting the two δ, we have our result that

$$\|\mathbb{P}_j(\cdot \mid s,a) - \mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbb{P}_j(\cdot \mid s,a)\|_2$$

is bounded above by

$$\sqrt{\frac{4K}{f_{min}d_{min}(s,a)}\left(\sqrt{\frac{128}{N_{sub}\cdot d_{min}(s,a)}(2S\log(12)+\log(4/\delta))} + \left(\frac{1}{2}\right)^{\frac{T_n}{8Gt_{mix}}}\right)}$$

for $N_{sub} \geq \frac{8}{d_{min}(s,a)^2}\log\left(\frac{1}{\delta}\right)$ and $\frac{T_n}{8t_{mix}} > \frac{G\log(48G/d_{min}(s,a))}{\log 2}$. □

A.5.3 Proof of Lemma A.5.2

We recall Lemma A.5.2.

Lemma A.5.2. *Let $\mathcal{B}_n$ be the event given by $n \in \mathcal{N}_{traj}(s,a)$, which is the same as $c_{n,1}c_{n,2} \neq 0$ and let*

$$\boldsymbol{M}_{s,a} = \sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T$$

.

Call our estimator $\hat{\boldsymbol{M}}_{s,a}$. Then we know that

$$\hat{\boldsymbol{M}}_{s,a} = \frac{1}{N_{traj}(s,a)}\sum_{n}\hat{\mathbb{P}}_{n,1}(\cdot \mid s,a)\hat{\mathbb{P}}_{n,2}(\cdot \mid s,a)^T$$

and we have

$$\|\hat{\boldsymbol{M}}_{s,a} - \boldsymbol{M}_{s,a}\| < \sqrt{\frac{32}{N_{traj}(s,a)}(2S\log(12) + \log(\frac{2}{\delta}))} + \frac{48G}{d_{min}(s,a)}\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}}$$

Proof. We divide the proof into subsections. We first remind ourselves that the estimator $\hat{\mathbf{M}}_{s,a}$ is given by the matrix

$$\hat{\mathbf{M}}_{s,a} = \frac{1}{N_{traj}(s,a)} \sum_{n \in \mathcal{N}_{traj}(s,a)} \left(\hat{\mathbb{P}}_{n,1}(\cdot|s,a)\hat{\mathbb{P}}_{n,2}(\cdot|s,a)^T\right)$$

A.5.3.1 Estimating $\mathbb{E}[\hat{\mathbf{M}}_{s,a}]$

We will split the expectation into the desired term and the error coming from correlation between the two segments Ω_1 and Ω_2. Remember that for brevity of notation, let $c_{n,i} := N(n,i,s,a)$, $\mathbf{w}_{n,i} := N(n,i,s,a,\cdot)$. Call the estimate from each trajectory a random variable $\hat{\mathbf{M}}_{n,s,a}$, that is

$$\hat{\mathbf{M}}_{n,s,a} = \frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}}$$

Now

$$\hat{\mathbf{M}}_{s,a} = \sum_{n \in \mathcal{N}_{traj}(s,a)} \frac{1}{N_{traj}(s,a)} \hat{\mathbf{M}}_{n,s,a}$$

Remember that

$$\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) := \frac{\mathbf{w}_{n,i}}{c_{n,i}} \mathbb{1}_{c_{n,i} \neq 0}$$

Let k_n be the hidden label for trajectory n, as usual. Define the event $\mathcal{B}_n$ to be $n \in \mathcal{N}_{traj}(s,a)$, which is the same as $c_{n,1}c_{n,2} \neq 0$. We establish the following equality, essentially just defining the quantity $\mathrm{Mix}(j)$.

$$\begin{aligned}
\mathbb{E}[\hat{\mathbf{M}}_{n,s,a} \mid \mathcal{B}_n] &= \mathbb{E}\left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}} \middle| \mathcal{B}_n\right] \\
&= \sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\mathbb{E}\left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}} \middle| k_n = j, \mathcal{B}_n\right] \\
&= \sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T + \sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\,\mathrm{Mix}(j) \quad \text{(A.7)}
\end{aligned}$$

where $\text{Mix}(j) = \mathbb{E}\left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}}\middle| k_n = j, \mathcal{B}_n\right] - \mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T$. Notice that this has connotations of covariance. Now note the following chain of equations.

$$\begin{aligned}
\mathbb{E}[\hat{\mathbf{M}}_{s,a} \mid \mathcal{N}_{traj}(s,a)] &= \mathbb{E}\left[\sum_{n\in\mathcal{N}_{traj}(s,a)} \frac{1}{N_{traj}(s,a)}\hat{\mathbf{M}}_{n,s,a}\middle|\mathcal{N}_{traj}(s,a)\right] \\
&= \sum_{n\in\mathcal{N}_{traj}(s,a)} \frac{1}{N_{traj}(s,a)}\mathbb{E}\left[\hat{\mathbf{M}}_{n,s,a}\middle|\mathcal{N}_{traj}(s,a)\right] \\
&= \sum_{n\in\mathcal{N}_{traj}(s,a)} \frac{1}{N_{traj}(s,a)}\mathbb{E}\left[\hat{\mathbf{M}}_{n,s,a}\middle|\mathbb{1}_{\mathcal{B}_1}, \mathbb{1}_{\mathcal{B}_2}, \ldots \mathbb{1}_{\mathcal{B}_{N_{sub}}}\right] \\
&= \sum_{n\in\mathcal{N}_{traj}(s,a)} \frac{1}{N_{traj}(s,a)}\mathbb{E}\left[\hat{\mathbf{M}}_{n,s,a}\middle|\mathcal{B}_n\right] \\
&= \mathbb{E}\left[\hat{\mathbf{M}}_{n,s,a}\middle|\mathcal{B}_n\right] \\
&= \sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T + \sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\,\text{Mix}(j) \\
&= \mathbf{M}_{s,a} + \sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\,\text{Mix}(j) \qquad \text{(A.8)}
\end{aligned}$$

Here, the third equality is because the set $\mathcal{N}_{traj}(s,a)$ is exactly described by the indicators listed, and they generate the same σ-algebra, The fourth equality holds since all trajectories are independent and so conditioning on events in other trajectories doesn't affect the expectation of $\hat{\mathbf{M}}_{n,s,a}$. The fifth equality is because $\mathbb{E}[\hat{\mathbf{M}}_{n,s,a} \mid \mathcal{B}_n]$ is the same for all n as determined above (in fact, we have shown that it is a constant random variable).

A.5.3.2 Setup for the main bound

We have that

$$\mathbf{M}_{s,a} = \sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T$$

By equation A.8,

$$\begin{aligned}
\|\hat{\mathbf{M}}_{s,a} - \mathbf{M}_{s,a}\| &\leq \|\hat{\mathbf{M}}_{s,a} - \mathbb{E}[\hat{\mathbf{M}}_{s,a} \mid \mathcal{N}_{traj}(s,a)]\| + \sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\,\|\text{Mix}(j)\| \\
&\leq \|\hat{\mathbf{M}}_{s,a} - \mathbb{E}[\hat{\mathbf{M}}_{s,a} \mid \mathcal{N}_{traj}(s,a)]\| + \left(\sum_{j=1}^{K} \mathbb{P}(k_n = j \mid \mathcal{B}_n)\right) \max_j \|\text{Mix}(j)\|
\end{aligned}$$

$$= \|\hat{\mathbf{M}}_{s,a} - \mathbb{E}[\hat{\mathbf{M}}_{s,a} \mid \mathcal{N}_{traj}(s,a)]\| + \max_j \|\mathrm{Mix}(j)\|$$

The first term represents the error in concentration across trajectories and the second term represents the correlation between the two segments Ω_1 and Ω_2 in the same trajectory. We bound the first using a covering argument and use Bin Yu's work to bound the other.

A.5.3.3 Covering argument to bound $\hat{\mathbf{M}}_{s,a} - \mathbb{E}[\hat{\mathbf{M}}_{s,a}]$

We will need this conditional version of Hoeffding's inequality for this section. Note that this is not quite the Azuma-Hoeffding inequality with a constant filtration due to the conditional probability involved, as well as due to the conditional independence needed.

Lemma A.5.4. *Consider a σ-algebra $\mathcal{F}$ and let $A_i \leq B_i$ be random variables measurable over it. If random variables X_i are almost surely bounded in $[A_i, B_i]$ and are conditionally independent over some σ-algebra $\mathcal{F}$, then the following inequalities hold for $S_n = \sum_{i=1}^n X_i$*

$$\mathbb{P}\left(S_n - \mathbb{E}[S_n \mid \mathcal{F}] > \epsilon | \mathcal{F}\right) \leq \exp\left(-\frac{2\epsilon}{\sum_i (B_i - A_i)^2}\right)$$
$$\mathbb{P}\left(S_n - \mathbb{E}[S_n \mid \mathcal{F}] < -\epsilon | \mathcal{F}\right) \leq \exp\left(-\frac{2\epsilon^2}{\sum_i (B_i - A_i)^2}\right)$$

Proof. The proof is essentially a repeat of one of the standard proofs of Hoeffding's inequality. Note that we have the conditional Markov inequality $\mathbb{P}(X \geq a \mid \mathcal{F}) \leq \frac{1}{a}\mathbb{E}[X \geq a \mid \mathcal{F}]$, shown exactly the way Markov's inequality is shown. Now, we have the following chain of inequalities.

$$\begin{aligned}\mathbb{P}((S_n - \mathbb{E}[S_n \mid \mathcal{F}] > \epsilon | \mathcal{F}) &= e^{-s\epsilon}\mathbb{E}[e^{S_n - \mathbb{E}[S_n|\mathcal{F}]} \mid \mathcal{F}] \\ &= e^{-s\epsilon}\prod_{i=1}^n \mathbb{E}[e^{X_i - \mathbb{E}[X_i|\mathcal{F}]} \mid \mathcal{F}]\end{aligned}$$

We now show a conditional expectation version of Hoeffding's lemma by repeating the steps for a standard proof to show that $\mathbb{E}[e^{X-\mathbb{E}[X|\mathcal{F}]} \mid \mathcal{F}] \leq \frac{\lambda^2(B-A)^2}{8}$ for random variables $A \leq B$ measurable over $\mathcal{F}$ and $X \in [A, B]$ almost surely. Note that by convexity of $e^{\lambda x}$, we have the following for $x \in [A, B]$ at any value of A and B.

$$e^{\lambda x} \leq \frac{B-x}{B-A}e^{\lambda A} + \frac{x-A}{B-A}e^{\lambda B}$$

WLOG, we can replace X by $X - \mathbb{E}[X \mid \mathcal{F}]$ and assume $\mathbb{E}[X \mid \mathcal{F}] = 0$. In that case, we

note the following inequality, where we define for any fixed value of A and B the function $L(y) := \frac{yA}{B-A} + \log\left(1 + \frac{A-e^y B}{B-A}\right)$.

$$\begin{aligned}\mathbb{E}[e^{\lambda X} \mid \mathcal{F}] &\leq \frac{B - \mathbb{E}[X \mid \mathcal{F}]}{B-A} e^{\lambda A} + \frac{\mathbb{E}[X \mid \mathcal{F}] - A}{B-A} e^{\lambda B} \\ &= \frac{B}{B-A} e^{\lambda A} + \frac{-A}{B-A} e^{\lambda B} \\ &= e^{L(\lambda(B-A))}\end{aligned}$$

Basic computations involving Taylor's theorem from a standard proof of Hoeffding's inequality show that $L(y) \leq \frac{y^2}{8}$ for any value of A, B. This gives us the condition version of Hoeffding's lemma, $\mathbb{E}[e^{X - \mathbb{E}[X|\mathcal{F}]} \mid \mathcal{F}] \leq \frac{\lambda^2 (B-A)^2}{8}$. This allows us to establish the following chain of inequalities.

$$\begin{aligned}\mathbb{P}((S_n - \mathbb{E}[S_n \mid \mathcal{F}] > \epsilon | \mathcal{F}) &= e^{-s\epsilon} \prod_{i=1}^{n} \mathbb{E}[e^{X_i - \mathbb{E}[X_i|\mathcal{F}]} \mid \mathcal{F}] \\ &\leq \exp(-s\epsilon) \prod_{i=1}^{n} \exp\left(\frac{s^2 (B_i - A_i)^2}{8}\right) \\ &= \exp\left(-s\epsilon + \sum_{i=1}^{n} \frac{s^2 (B_i - A_i)^2}{8}\right)\end{aligned}$$

Since s is arbitrary, we can pick $s = \frac{4\epsilon}{\sum_i (B_i - A_i)^2}$ above to get an upper bound of $\exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{n} (B_i - A_i)^2}\right)$. The other inequality is proved analogously. □

We now show that the first term from the previous section concentrates. Pick $\mathbf{u}, \mathbf{v} \in \mathbb{S}^{S-1}$, that is they lie in the unit Euclidean norm sphere in $\mathbb{R}^S$. We need only bound this term when $N_{traj}(s, a) \neq 0$, as otherwise the lemma holds vacuously.

Note that

$$\hat{\mathbf{M}}_{s,a} - \mathbb{E}[\hat{\mathbf{M}}_{s,a} \mid \mathcal{N}_{traj}(s,a)] = \sum_{n \in \mathcal{N}_{traj}(s,a)} \frac{\hat{\mathbf{M}}_{n,s,a} - \mathbb{E}[\hat{\mathbf{M}}_{n,s,a} \mid \mathcal{N}_{traj}(s,a)]}{N_{traj}(s,a)}$$

Now we set up our covering argument. Consider a covering of $\mathbb{S}^{S-1}$ by balls of radius $\frac{1}{4}$. We will need at most 12^S such balls and if C is the set of their centers, then for any matrix X, the following

holds in regard to its norm.

$$\|X\| = \sup_{\mathbf{u},\mathbf{v}\in\mathbb{S}^{S-1}} |\mathbf{u}^T X \mathbf{v}| \leq 2 \max_{\mathbf{u},\mathbf{v}\in C} |\mathbf{u}^T X \mathbf{v}| \leq 2\|X\| \tag{A.9}$$

For any pair $\mathbf{u},\mathbf{v} \in C$, note that

$$\begin{aligned}
|\mathbf{u}^T \hat{\mathbf{M}}_{n,s,a} \mathbf{v}| &= \left| \mathbf{u}^T \frac{\mathbf{w}_{n,1}}{c_{n,1}} \right| \left| \frac{\mathbf{w}_{n,2}^T}{c_{n,2}} \mathbf{v} \right| \mathbb{1}_{c_{n,1} c_{n,2} \neq 0} \\
&\leq \|\mathbf{u}\|_2 \|\mathbf{v}\|_2 \left\| \frac{\mathbf{w}_{n,1}}{c_{n,1}} \right\|_2 \left\| \frac{\mathbf{w}_{n,2}}{c_{n,2}} \right\|_2 \\
&\leq \left\| \frac{\mathbf{w}_{n,1}}{c_{n,1}} \right\|_1 \left\| \frac{\mathbf{w}_{n,2}}{c_{n,2}} \right\|_1 \\
&\leq 1
\end{aligned}$$

and so $|\mathbf{u}^T \mathbb{E}[\hat{\mathbf{M}}_{n,s,a}] \mathbf{v}| \leq \mathbb{E}[|\mathbf{u}^T \hat{\mathbf{M}}_{n,s,a} \mathbf{v}|] \leq 1$. A little thought shows that the estimates $\hat{\mathbf{M}}_{n,s,a}$ are independent for $n \in \mathcal{N}_{traj}(s,a)$ when conditioned on the $\mathcal{N}_{traj}(s,a)$.

$$\mathbb{P}\left(\left| \sum_{n \in \mathcal{N}_{traj}(s,a)} \frac{1}{N_{traj}(s,a)} \mathbf{u}^T (\hat{\mathbf{M}}_{n,s,a} - \mathbb{E}[\hat{\mathbf{M}}_{n,s,a} \mid \mathcal{N}_{traj}(s,a)]) \mathbf{v} \right| > \frac{\epsilon}{4} \middle| \mathcal{N}_{traj}(s,a) \right) < 2e^{-\frac{\epsilon^2 N_{traj}(s,a)}{32}}$$

Doing this for all 12^{2S} pairs $\mathbf{u},\mathbf{v}$, we use inequality A.9 to get that the conditionalprobability given by

$$\mathbb{P}\left(\left\| \sum_{n \in \mathcal{N}_{traj}(s,a)} \frac{1}{N_{traj}(s,a)} \hat{\mathbf{M}}_{n,s,a} - \mathbb{E}[\hat{\mathbf{M}}_{n,s,a} \mid \mathcal{N}_{traj}(s,a)] \right\| > \frac{\epsilon}{2} \middle| \mathcal{N}_{traj}(s,a) \right)$$

is bounded above by the following expression.

$$\begin{aligned}
&\mathbb{P}\left(\exists \mathbf{u},\mathbf{v} \in C; \left| \sum_{n \in \mathcal{N}_{traj}(s,a)} \frac{1}{N_{traj}(s,a)} \mathbf{u}^T (\hat{\mathbf{M}}_{n,s,a} - \mathbb{E}[\hat{\mathbf{M}}_{n,s,a} \mid \mathcal{N}_{traj}(s,a)]) \mathbf{v} \right| > \frac{\epsilon}{4} \middle| \mathcal{N}_{traj}(s,a) \right) \\
&\leq \sum_{\mathbf{u},\mathbf{v} \in C} \mathbb{P}\left(\left| \sum_{n \in \mathcal{N}_{traj}(s,a)} \frac{1}{N_{traj}(s,a)} \mathbf{u}^T (\hat{\mathbf{M}}_{n,s,a} - \mathbb{E}[\hat{\mathbf{M}}_{n,s,a} \mid \mathcal{N}_{traj}(s,a)]) \mathbf{v} \right| > \frac{\epsilon}{4} \middle| \mathcal{N}_{traj}(s,a) \right)
\end{aligned}$$

$$< 2 * 12^{2S} * e^{-\frac{\epsilon^2 N_{traj}(s,a)}{32}}$$

This is less than δ for $N_{traj}(s,a) > \frac{32}{\epsilon^2}(2S\log(12) + \log(\frac{2}{\delta}))$. Since this holds for such values of $N_{traj}(s,a)$ irrespective of $\mathcal{N}_{traj}(s,a)$, we can conclude that for $N_{traj}(s,a) > \frac{32}{\epsilon^2}(2S\log(12) + \log(\frac{2}{\delta}))$, with probability universally greater than $1-\delta$,

$$\|\hat{\mathbf{M}}_{s,a} - \mathbf{M}_{s,a}\| < \frac{\epsilon}{2} + \max_j \|\text{Mix}(j)\|$$

Alternatively, this establishes that with probability greater than $1-\delta$, we have the following inequality involving the random variables $\hat{\mathbf{M}}_{s,a}$ and $N_{traj}(s,a)$.

$$\|\hat{\mathbf{M}}_{s,a} - \mathbf{M}_{s,a}\| < \sqrt{\frac{32}{N_{traj}(s,a)}(2S\log(12) + \log(\frac{2}{\delta}))} + \max_j \|\text{Mix}(j)\|$$

A.5.3.4 Bounding the mixing term

We now resolve the last remaining thread, which is that of bounding the mixing term. Let's fix a j for this section, since proving our upper bounds for arbitrary j is sufficient. Let the joint distribution of the observations under label j be χ. Let its marginal on the segment Ω_i be χ_i. Let the marginals on each of the G single-step sub-blocks be $\chi_{i,g}$. Denote the product distribution $\prod_g \chi_{i,g}$ by Q_i.

$$\begin{aligned}
\|\text{Mix}(j)\| &= \left\| \mathbb{E}\left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}} \middle| k_n = j, \mathcal{B}_n \right] - \mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T \right\| \\
&= \left\| \frac{1}{\mathbb{P}(\mathcal{B}_n)} \mathbb{E}_\chi \left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}} \mathbb{1}_{\mathcal{B}_n} \right] - \mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T \right\| \\
&\leq \frac{1}{\mathbb{P}(\mathcal{B}_n)} \left\| \mathbb{E}_\chi \left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}} \mathbb{1}_{\mathcal{B}_n} \right] - \mathbb{E}_{\chi_1}\left[\frac{\mathbf{w}_{n,1}}{c_{n,1}} \mathbb{1}_{c_{n,1}\neq 0} \right] \mathbb{E}_{\chi_2}\left[\frac{\mathbf{w}_{n,2}^T}{c_{n,2}} \mathbb{1}_{c_{n,2}\neq 0} \right] \right\| \\
&\quad + \frac{1}{\mathbb{P}(\mathcal{B}_n)} \Bigg\| \mathbb{E}_{\chi_1}\left[\frac{\mathbf{w}_{n,1}}{c_{n,1}} \mathbb{1}_{c_{n,1}\neq 0} \right] \mathbb{E}_{\chi_2}\left[\frac{\mathbf{w}_{n,2}^T}{c_{n,2}} \mathbb{1}_{c_{n,2}\neq 0} \right] \\
&\qquad\qquad - \mathbb{P}_{Q_1}(c_{n,1}\neq 0)\mathbb{P}_{Q_2}(c_{n,2}\neq 0)\mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T \Bigg\| \\
&\quad + \frac{1}{\mathbb{P}(\mathcal{B}_n)} \big\| (\mathbb{P}_{Q_1}(c_{n,1}\neq 0)\mathbb{P}_{Q_2}(c_{n,2}\neq 0) \\
&\qquad\qquad - \mathbb{P}(c_{n,1}\neq 0)\mathbb{P}(c_{n,2}\neq 0))\mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T \big\| \\
&\quad + \frac{1}{\mathbb{P}(\mathcal{B}_n)} \left\| \left(\mathbb{P}(c_{n,1}\neq 0)\mathbb{P}(c_{n,2}\neq 0) - \mathbb{P}(\mathcal{B}_n)\right) \mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T \right\| \\
&\leq \frac{1}{\mathbb{P}(\mathcal{B}_n)} \left\| \mathbb{E}_\chi \left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}} \mathbb{1}_{\mathcal{B}_n} \right] - \mathbb{E}_{\chi_1}\left[\frac{\mathbf{w}_{n,1}}{c_{n,1}} \mathbb{1}_{c_{n,1}\neq 0} \right] \mathbb{E}_{\chi_2}\left[\frac{\mathbf{w}_{n,2}^T}{c_{n,2}} \mathbb{1}_{c_{n,2}\neq 0} \right] \right\|
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{1}{\mathbb{P}(\mathcal{B}_n)} \Bigg\| \mathbb{E}_{\chi_1}\left[\frac{\mathbf{w}_{n,1}}{c_{n,1}} \mathbb{1}_{c_{n,1}\neq 0}\right] \mathbb{E}_{\chi_2}\left[\frac{\mathbf{w}_{n,2}^T}{c_{n,2}} \mathbb{1}_{c_{n,2}\neq 0}\right] \\
&\qquad - \mathbb{P}_{Q_1}(c_{n,1}\neq 0)\mathbb{P}_{Q_2}(c_{n,2}\neq 0)\mathbb{P}_j(\cdot \mid s,a)\mathbb{P}_j(\cdot \mid s,a)^T \Bigg\| \\
&+ \frac{1}{\mathbb{P}(\mathcal{B}_n)} \left|\mathbb{P}_{Q_1}(c_{n,1}\neq 0)\mathbb{P}_{Q_2}(c_{n,2}\neq 0) - \mathbb{P}_{\chi_1}(c_{n,1}\neq 0)\mathbb{P}_{\chi_2}(c_{n,2}\neq 0)\right| \\
&+ \frac{1}{\mathbb{P}(\mathcal{B}_n)} \left|\mathbb{P}_{\chi_1}(c_{n,1}\neq 0)\mathbb{P}_{\chi_2}(c_{n,2}\neq 0) - \mathbb{P}(\mathcal{B}_n)\right| \qquad \text{(A.10)}
\end{aligned}
$$

Here, in the last inequality, we used the fact that $\|\mathbb{P}_j(\cdot \mid s,a)\|_2 \leq \|\mathbb{P}_j(\cdot \mid s,a)\|_1 = 1$ and $\|\mathbf{a}\mathbf{a}^T\| \leq \|\mathbf{a}\|_2\|\mathbf{a}\|_2$. Also note that $\mathbb{P}_{\chi_i}(c_{n,i}\neq 0) = \mathbb{P}_{\chi}(c_{n,i}\neq 0) = \mathbb{P}(c_{n,i}\neq 0)$.

Intuitively, the first term represents mixing of the expectation across the two segments, the second term represents mixing of the expectations across the single-step sub-blocks inside segments, the third term represents mixing of the observation probabilities across the single-step sub-blocks inside segments, and the fourth term represents mixing of the observation probabilities across the two segments. In short, the first and fourth represent segment-level mixing while the second and third represent sub-block-level mixing.

Bounding the first term (segment-level mixing)

We will use Yu [1994]'s blocking technique again, invoking Lemma A.5.3. Pick an arbitrary $\mathbf{u},\mathbf{v} \in \mathbb{S}^{S-1}$. Recall that

$$
\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) := \frac{\mathbf{N}(n,i,s,a,\cdot)}{N(n,i,s,a)} \mathbb{1}_{N(n,i,s,a)\neq 0} = \frac{\mathbf{w}_{n,i}}{c_{n,i}} \mathbb{1}_{c_{n,i}\neq 0}
$$

Consider the real-valued random variable

$$
h_{\mathbf{u},\mathbf{v}} := \mathbf{u}^T \left(\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}} \mathbb{1}_{\mathcal{B}_n}\right) \mathbf{v}
$$

We have the following basic computations for expectations. Remember that $\mathbb{1}_{\mathcal{B}_n} = \mathbb{1}_{c_{n,1}c_{n,2}\neq 0} = \mathbb{1}_{c_{n,1}\neq 0}\mathbb{1}_{c_{n,2}\neq 0}$.

$$
\mathbb{E}_{\chi} h_{\mathbf{u},\mathbf{v}} = \mathbf{u}^T \left(\mathbb{E}_{\chi}\left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}} \mathbb{1}_{\mathcal{B}_n}\right]\right) \mathbf{v}
$$

and

$$
\mathbb{E}_{\chi_1\times\chi_2} h_{\mathbf{u},\mathbf{v}} = \mathbf{u}^T \left(\mathbb{E}_{\chi_1}\left[\frac{\mathbf{w}_{n,1}}{c_{n,1}} \mathbb{1}_{c_{n,1}\neq 0}\right] \mathbb{E}_{\chi_2}\left[\frac{\mathbf{w}_{n,2}^T}{c_{n,2}} \mathbb{1}_{c_{n,2}\neq 0}\right]\right) \mathbf{v}
$$

This allows us to establish the following relation.

$$\sup_{\mathbf{u},\mathbf{v}\in\mathbb{S}^{S-1}} |\mathbb{E}_{\chi} h_{\mathbf{u},\mathbf{v}} - \mathbb{E}_{\chi_1\times\chi_2} h_{\mathbf{u},\mathbf{v}}| = \left\| \mathbb{E}_{\chi}\left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}}\mathbb{1}_{\mathcal{B}_n}\right] - \mathbb{E}_{\chi_1}\left[\frac{\mathbf{w}_{n,1}}{c_{n,1}}\mathbb{1}_{c_{n,1}\neq 0}\right]\mathbb{E}_{\chi_2}\left[\frac{\mathbf{w}_{n,2}^T}{c_{n,2}}\mathbb{1}_{c_{n,2}\neq 0}\right]\right\|$$

Now, we want to use Lemma A.5.3. Note the following upper bound.

$$\begin{aligned} |h_{\mathbf{u},\mathbf{v}}| &\leq \|\mathbf{u}\|_2 \left\|\frac{\mathbf{w}_{n,1}}{c_{n,1}}\right\|_2 \left\|\frac{\mathbf{w}_{n,2}}{c_{n,2}}\right\|_2 \|\mathbf{v}\|_2 \\ &\leq \left\|\frac{\mathbf{w}_{n,1}}{c_{n,1}}\right\|_1 \left\|\frac{\mathbf{w}_{n,2}}{c_{n,2}}\right\|_1 \\ &= 1 \end{aligned}$$

So, we can use Lemma A.5.3 with $C = C_{\mathbf{u},\mathbf{v}} := \sup h_{\mathbf{u},\mathbf{v}}$ and $n = 2$ for any $\mathbf{u},\mathbf{v} \in \mathbb{S}^{S-1}$, giving us the following inequality.

$$|\mathbb{E}_{\chi} h_{\mathbf{u},\mathbf{v}} - \mathbb{E}_{\chi_1\times\chi_2} h_{\mathbf{u},\mathbf{v}}| \leq 4\lambda^{\frac{T_n}{4}} \tag{A.11}$$

Since inequality A.11 holds for any $\mathbf{u}, \mathbf{v} \in \mathbb{S}^{S-1}$, we can take the supremum over such $\mathbf{u}, \mathbf{v}$ to get the desired inequality below. We also recall that $\mathbb{P}(\mathcal{B}_n) \geq \frac{d_{min}s,a}{2}$ from equation A.5.

$$\begin{aligned} \frac{1}{\mathbb{P}(\mathcal{B}_n)}\left\|\mathbb{E}_{\chi}\left[\frac{\mathbf{w}_{n,1}\mathbf{w}_{n,2}^T}{c_{n,1}c_{n,2}}\mathbb{1}_{\mathcal{B}_n}\right] - \mathbb{E}_{\chi_1}\left[\frac{\mathbf{w}_{n,1}}{c_{n,1}}\mathbb{1}_{c_{n,1}\neq 0}\right]\mathbb{E}_{\chi_2}\left[\frac{\mathbf{w}_{n,2}^T}{c_{n,2}}\mathbb{1}_{c_{n,2}\neq 0}\right]\right\| &\leq \frac{1}{\mathbb{P}(\mathcal{B}_n)} 4\lambda^{\frac{T_n}{4}} \\ &\leq \frac{8\lambda^{\frac{T_n}{4}}}{d_{min}(s,a)} \end{aligned}$$

Bounding the second term (sub-block-level mixing)

Remember that the product distribution $\prod_g \chi_{i,g}$ is Q_i. First note that, since under Q_i, each observation is independent, we have the following expectation.

$$\begin{aligned} \mathbb{E}_{Q_i}\left[\frac{\mathbf{w}_{n,i}}{c_{n,i}}\mathbb{1}_{c_{n,i}\neq 0}\right] &= \mathbb{E}_{Q_i}\left[\frac{\mathbb{E}_{Q_i}[\mathbf{w}_{n,i} \mid c_{n,i}]}{c_{n,i}}\mathbb{1}_{c_{n,i}\neq 0}\right] \\ &= \mathbb{E}_{Q_i}\left[\frac{\mathbb{P}_j(\cdot \mid s,a)c_{n,i}}{c_{n,i}}\mathbb{1}_{c_{n,i}\neq 0}\right] \end{aligned}$$

$$= \mathbb{P}_j(\cdot \mid s, a)\mathbb{P}_{Q_i}(c_{n,i} \neq 0) \tag{A.12}$$

Remark 12. Note that this holds crucially because we are working with the product distribution Q_i over the single-step sub-blocks.

Also, let $h_{\mathbf{u}} = \mathbf{u}^T \frac{\mathbf{w}_{n,1}}{c_{n,1}} \mathbb{1}_{c_{n,1}}$ and let $g_{\mathbf{v}} = \frac{\mathbf{w}_{n,2}^T}{c_{n,2}} \mathbb{1}_{c_{n,2}} \mathbf{v}$. Then the second term is exactly given by the following expression.

$$\frac{1}{\mathbb{P}(\mathcal{B}_n)} \sup_{\mathbf{u},\mathbf{v} \in \mathbb{S}^{S-1}} |\mathbb{E}_{\chi_1}[h_{\mathbf{u}}]\mathbb{E}_{\chi_2}[g_{\mathbf{v}}] - \mathbb{E}_{Q_1}[h_{\mathbf{u}}]\mathbb{E}_{Q_2}[g_{\mathbf{v}}]|$$

Also note that both $|h_{\mathbf{u}}|$ and $|g_{\mathbf{v}}|$ are bounded by 1. We then have the following chain of inequalities.

$$\begin{aligned} |\mathbb{E}_{\chi_1}[h_{\mathbf{u}}]\mathbb{E}_{\chi_2}[g_{\mathbf{v}}] - \mathbb{E}_{Q_1}[h_{\mathbf{u}}]\mathbb{E}_{Q_2}[g_{\mathbf{v}}]| &\leq |\mathbb{E}_{\chi_1}[h_{\mathbf{u}}] - \mathbb{E}_{Q_1}[h_{\mathbf{u}}]|\,|\mathbb{E}_{\chi_2}[g_{\mathbf{v}}]| + |\mathbb{E}_{\chi_2}[g_{\mathbf{v}}] - \mathbb{E}_{Q_2}[g_{\mathbf{v}}]||\mathbb{E}_{Q_1}[h_{\mathbf{u}}]| \\ &\leq |\mathbb{E}_{\chi_1}[h_{\mathbf{u}}] - \mathbb{E}_{Q_1}[h_{\mathbf{u}}]| + |\mathbb{E}_{\chi_2}[g_{\mathbf{v}}] - \mathbb{E}_{Q_2}[g_{\mathbf{v}}]| \end{aligned}$$

Since the single step sub-blocks are separated by at least $\frac{T_n}{8G}$ timesteps, we can apply Lemma A.5.3 with $C = 1$ and $n = G$ to get bounds on both terms here, since $Q_i = \prod_g \chi_{i,g}$. Also remember that $\mathbb{P}(\mathcal{B}_n) \geq \frac{d_{min}(s,a)}{2}$ from equation A.5.

$$\begin{aligned} \frac{1}{\mathbb{P}(\mathcal{B}_n)} \sup_{\mathbf{u},\mathbf{v} \in \mathbb{S}^{S-1}} |\mathbb{E}_{\chi_1}[h_{\mathbf{u}}]\mathbb{E}_{\chi_2}[g_{\mathbf{v}}] - \mathbb{E}_{Q_1}[h_{\mathbf{u}}]\mathbb{E}_{Q_2}[g_{\mathbf{v}}]| &\leq \frac{1}{\mathbb{P}(\mathcal{B}_n)} \left(4G\lambda^{\frac{T_n}{8G}} + 4G\lambda^{\frac{T_n}{8G}}\right) \\ &\leq \frac{16G\lambda^{\frac{T_n}{8G}}}{d_{min}(s,a)} \end{aligned}$$

Bounding the third term (sub-block-level mixing)

Again, note that the third term is given by the following expression.

$$\frac{1}{\mathbb{P}(\mathcal{B}_n)} \left|\mathbb{E}_{Q_1}[\mathbb{1}_{c_{n,1}\neq 0}]\mathbb{E}_{Q_2}[\mathbb{1}_{c_{n,2}\neq 0}] - \mathbb{E}_{\chi_1}[\mathbb{1}_{c_{n,1}\neq 0}]\mathbb{E}_{\chi_2}[\mathbb{1}_{c_{n,2}\neq 0}]\right|$$

We can bound this above using the fact that $|ab - cd| \leq |b||a - c| + |c||b - d|$, to get the following

upper bound.

$$\mathbb{E}_{Q_2}[\mathbb{1}_{c_{n,2}\neq 0}]\left|\mathbb{E}_{Q_1}[\mathbb{1}_{c_{n,1}\neq 0}]-\mathbb{E}_{\chi_1}[\mathbb{1}_{c_{n,1}\neq 0}]\right|+\mathbb{E}_{\chi_1}[\mathbb{1}_{c_{n,1}\neq 0}]\left|\mathbb{E}_{Q_2}[\mathbb{1}_{c_{n,2}\neq 0}]-\mathbb{E}_{\chi_2}[\mathbb{1}_{c_{n,2}\neq 0}]\right|$$

This in turn is bounded above by the expression below.

$$|\mathbb{E}_{Q_1}[\mathbb{1}_{c_{n,1}\neq 0}]-\mathbb{E}_{\chi_1}[\mathbb{1}_{c_{n,1}\neq 0}]|+|\mathbb{E}_{Q_2}[\mathbb{1}_{c_{n,2}\neq 0}]-\mathbb{E}_{\chi_2}[\mathbb{1}_{c_{n,2}\neq 0}]|$$

Since indicator functions are bounded above by 1, we can apply Lemma A.5.3 as in the second term ($C=1$, $n=G$) to bound both the differences above. Skipping the routine details, we finally get the following inequality, analogous to the second term.

$$\frac{1}{\mathbb{P}(\mathcal{B}_n)}\left|\mathbb{E}_{Q_1}[\mathbb{1}_{c_{n,1}\neq 0}]\mathbb{E}_{Q_2}[\mathbb{1}_{c_{n,2}\neq 0}]-\mathbb{E}_{\chi_1}[\mathbb{1}_{c_{n,1}\neq 0}]\mathbb{E}_{\chi_2}[\mathbb{1}_{c_{n,2}\neq 0}]\right|\leq\frac{16G\lambda^{\frac{T_n}{8G}}}{d_{min}(s,a)}$$

Bounding the fourth term (segment-level mixing)

Now note that the fourth term is the same as the expression below.

$$\frac{1}{\mathbb{P}(\mathcal{B}_n)}|\mathbb{E}_{\chi_1}[\mathbb{1}_{c_{n,1}\neq 0}]\mathbb{E}_{\chi_2}[\mathbb{1}_{c_{n,2}\neq 0}]-\mathbb{E}_{\chi}[\mathbb{1}_{c_{n,1}\neq 0}\mathbb{1}_{c_{n,2}\neq 0}]|$$
$$=\frac{1}{\mathbb{P}(\mathcal{B}_n)}|\mathbb{E}_{\chi_1\times\chi_2}[\mathbb{1}_{c_{n,1}\neq 0}\mathbb{1}_{c_{n,2}\neq 0}]-\mathbb{E}_{\chi}[\mathbb{1}_{c_{n,1}\neq 0}\mathbb{1}_{c_{n,2}\neq 0}]|$$

We can now apply Lemma A.5.3 with $C=1$ and $n=2$. The segments are separated by T and $\mathbb{P}(\mathcal{B}_n)\geq\frac{d_{min}(s,a)}{2}$, giving us the following bound.

$$\frac{1}{\mathbb{P}(\mathcal{B}_n)}\left|\mathbb{P}_{\chi_1}(c_{n,1}\neq 0)\mathbb{P}_{\chi_2}(c_{n,2}\neq 0)-\mathbb{P}(\mathcal{B}_n)\right|\leq\frac{8\lambda^{\frac{T_n}{4}}}{d_{min}(s,a)}$$

Combining all these, we get that

$$\|\hat{\mathbf{M}}_{s,a}-\mathbf{M}_{s,a}\|<\sqrt{\frac{32G}{N_{traj}(s,a)}\left(2S\log(12)+\log\left(\frac{2}{\delta}\right)\right)}$$
$$+\frac{16}{d_{min}(s,a)}\left(\frac{1}{4}\right)^{\frac{T_n}{4t_{mix}}}+\frac{32G}{d_{min}(s,a)}\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}}$$

$$\leq \sqrt{\frac{32}{N_{traj}(s,a)}\left(2S\log(12)+\log\left(\frac{2}{\delta}\right)\right)}+\frac{48G}{d_{min}(s,a)}\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}} \tag{A.13}$$

as desired.

A.6 Proof of Theorem 3.4.3

Theorem 3.4.3 (Exact Clustering Guarantee)**.** *Pick any pair of trajectories n, m. Then for Freq_β so that it contains (s,a) with $d_{min}(s,a) \geq \Omega(\alpha)$, $T_n = \Omega(t_{mix}\log^4(1/\delta)/\alpha^3)$, with probability at least $1-\delta$,*

$$\left|\mathrm{dist}_1(m,n) - \|\Delta_{m,n}\|_2^2\right|$$

is

$$O\left(\sqrt{\frac{K\log(1/\delta)}{\alpha}}\left(\frac{t_{mix}}{T_n}\right)^{\frac{1}{3}}\right)+4\epsilon_{sub}(\delta/2)$$

This means that if we choose $\lambda = 1$, then if $\epsilon_{sub}(\delta) \leq \Delta^2/32$ and $T_n = \Omega\left(K^{3/2}t_{mix}\frac{\log^4(N_{clust}/(\alpha\delta))}{\Delta^6\alpha^3}\right)$, no distance estimate attains a value between $\Delta^2/4$ and $\Delta^2/2$. So, Algorithm 2 attains exact clustering using a threshold of say $\Delta^2/3$ with probability at least $1-\delta$.

Proof. Consider the testing of trajectories m and n. Recall that we defined

$$\mathrm{dist}_1(m,n) := \max_{(s,a)\in SA_\alpha}\left[\left(\left(\hat{\mathbb{P}}_{n,1}(\cdot \mid s,a) - \hat{\mathbb{P}}_{m,1}(\cdot \mid s,a)\right)^T \mathbf{V}_{s,a}\right)\left(\left(\hat{\mathbb{P}}_{n,2}(\cdot \mid s,a) - \hat{\mathbb{P}}_{m,2}(\cdot \mid s,a)\right)^T \mathbf{V}_{s,a}\right)^T\right]$$

Let k_m be the label of trajectory m and k_n the label of trajectory n. According to our assumptions, if $k_m \neq k_n$, then we have an s, a so that $d_{k_m}(s,a), d_{k_n}(s,a) \geq \alpha$ and $\|\mathbb{P}_{k_m}(\cdot \mid s,a) - \mathbb{P}_{k_n}(\cdot \mid s,a)\|_2 \geq \Delta$. We will make s, a implicit in our notation except in $\mathbb{P}_j(\cdot \mid s,a)$. Let $c_{n,i} := N(n,i,s,a)$, $\mathbf{w}_{n,i} := \mathbf{N}(n,i,s,a,\cdot)$. Recall that we have two nested partitions: (1) of the entire trajectory into the two Ω_i and (2) of each segment Ω_i into G blocks. Finally, define $\mathrm{dist}_{1,(s,a)}$ as below, suppressing m and n. Note that $\mathrm{dist}_1(m,n)$ is the maximum of $\mathrm{dist}_{1,(s,a)}$ over all $(s,a) \in \mathrm{Freq}_\beta$, for the given two trajectories m and n.

$$\mathrm{dist}_{1,(s,a)} := \left[\left((\hat{\mathbb{P}}_{n,1}(\cdot \mid s,a) - \hat{\mathbb{P}}_{m,1}(\cdot \mid s,a))^T\mathbf{V}_{s,a}\right)\left((\hat{\mathbb{P}}_{n,2}(\cdot \mid s,a) - \hat{\mathbb{P}}_{m,2}(\cdot \mid s,a))^T V_{s,a}\right)^T\right]$$

We want to show that this is close to $\|\Delta_{m,n}(s,a)\|_2^2$ for the (s,a) pairs that we search over, where

$$\Delta_{m,n}(s,a) = \mathbb{P}_{k_m}(\cdot \mid s,a) - \mathbb{P}_{k_n}(\cdot \mid s,a)$$

Assume the lemma below for now, we prove it in the next subsection.

Lemma A.6.1. *We claim that there is a universal constant C_1 so that for any (s,a) with $d_{min}(s,a) \geq \alpha/3$, with probability at least $1-\delta$,*

$$\left|\text{dist}_{1,(s,a)} - \|\Delta_{m,n}(s,a)\|_2^2\right| \leq C_1 \left(\sqrt{\frac{K+\log(1/\delta)}{G\alpha}}\right) + 4\epsilon_{sub}(\delta/2)$$

whenever $T_n \geq \Omega\left(Gt_{mix}\log(G/\delta)\log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$. Here, $\epsilon_{sub}(\delta)$ is the high probability bound on $\|\mathbb{P}_j(\cdot \mid s,a) - \boldsymbol{V}_{s,a}\boldsymbol{V}_{s,a}^T\mathbb{P}_j(\cdot \mid s,a)\|_2$ with $j = k_n, k_m$, from Theorem 3.4.2 (satisfied with probability $> 1-\delta$).

We now set $G = \left(\frac{T_n}{t_{mix}}\right)^{\frac{2}{3}}$. Then a sufficient condition on T_n to meet the conditions of the lemma is $T_n = \Omega(t_{mix}\log^4(1/\delta)/\alpha^3)$, under which, with probability at lest $1-\delta$, we have the following bound for (s,a) with $d_{min}(s,a) \geq \alpha/3$.

$$\left|\text{dist}_{1,(s,a)} - \|\Delta_{m,n}(s,a)\|_2^2\right| \leq O\left(\sqrt{\frac{K\log(1/\delta)}{\alpha}}\left(\frac{t_{mix}}{T_n}\right)^{\frac{1}{3}}\right) + 4\epsilon_{sub}(\delta/2) \tag{A.14}$$

It is now easy to see that the first term on the right-hand side is less than $\Delta^2/8$ when $T_n = \Omega\left(K^{3/2}t_{mix}\frac{\log^{3/2}(1/\delta)}{\Delta^6\alpha^{3/2}}\right)$ and $T_n = \Omega(t_{mix}\log^4(1/\delta)/\alpha^3)$. We can combine these to have the guarantee that the first term on the right-hand side is less $\Delta^2/8$ with probability at least $1-\delta$ when $T_n = \Omega\left(K^{3/2}t_{mix}\frac{\log^4(1/\delta)}{\Delta^6\alpha^3}\right)$.

Now note that if $\beta \geq \alpha/3$, then a separating state action pair always lies in Freq_β and thus, the maximum over the $\|\Delta_{m,n}(s,a)\|_2^2$ values corresponding to Freq_β is in fact either 0 if $k_m = k_n$ or larger than Δ^2 if $k_m \neq k_n$. So, if $\epsilon_{sub}(\delta/2) \leq \Delta^2/32$ and for each of the (s,a) pairs, the first term on the right-hand side of inequality A.14 is less than $\Delta^2/8$, then our distance estimate $\text{dist}_1(m,n)$ is on the right side of any threshold as long as $\Delta^2/4 \leq \tau \leq \Delta^2/2$. That is, the distance estimate is then less than the threshold if $k_m = k_n$, and larger than it if $k_m \neq k_n$.

Note that upon choosing an occurrence threshold of order α, we will have at most $O(1/\alpha)$ many (s,a) pairs in Freq_β to maximize $\text{dist}_{1,(s,a)}$ over to get $\text{dist}_1(m,n)$. By applying a union bound over

all (s, a) pairs in Freq_β and using the conclusion of the previous paragraph, we correctly determine if $k_m = k_n$ with probability $1 - \delta$ for $T_n = \Omega\left(K^{3/2} t_{mix} \frac{\log^4(1/(\alpha\delta))}{\Delta^6 \alpha^3}\right)$, as long as $\epsilon_{sub}(\delta/2) \leq \Delta^2/32$ and $\Delta^2/4 \leq \tau \leq \Delta^2/2$.

By applying a union bound over incorrectly deciding whether or not $k_m = k_n$ for any of the $N_{clust}(N_{clust} - 1)/2$ pairs, we get that we can recover the true clusters with probability at least $1 - \delta$ for $T_n = \Omega\left(K^{3/2} t_{mix} \frac{\log^4(N_{clust}/(\alpha\delta))}{\Delta^6 \alpha^3}\right)$, whenever $\epsilon_{sub} \leq \Delta^2/32$ and as long as $\epsilon_{sub}(\delta/2) \leq \Delta^2/32$ and $\Delta^2/4 \leq \tau \leq \Delta^2/2$. □

A.6.1 Proof of Lemma A.6.1

We recall the statement of the lemma.

Lemma A.6.1. *We claim that there is a universal constant C_1 so that for any (s, a) with $d_{min}(s, a) \geq \alpha/3$, with probability at least $1 - \delta$,*

$$\left|\text{dist}_{1,(s,a)} - \|\Delta_{m,n}(s, a)\|_2^2\right| \leq C_1 \left(\sqrt{\frac{K + \log(1/\delta)}{G\alpha}}\right) + 4\epsilon_{sub}(\delta/2)$$

whenever $T_n \geq \Omega\left(G t_{mix} \log(G/\delta) \log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$. Here, $\epsilon_{sub}(\delta)$ is the high probability bound on $\|\mathbb{P}_j(\cdot \mid s, a) - \boldsymbol{V}_{s,a}\boldsymbol{V}_{s,a}^T \mathbb{P}_j(\cdot \mid s, a)\|_2$ with $j = k_n, k_m$, from Theorem 3.4.2 (satisfied with probability $> 1 - \delta$).

Notation: We say $c_{n,i} = N(n, i, s, a)$ as in the statement of the lemma and $\mathbf{w}_{n,i} = \mathbf{N}(n, i, s, a, \cdot)$. Let the joint distribution of the observations over the pair of trajectories (m, n) be χ. This means that χ is the product of the joint distribution of the observations over the trajectory m and that of the observations over the trajectory n, since trajectories are generated independently. Let its marginals on the segments Ω_i be χ_i. Let the marginals on each of the G single-step sub-blocks along with their next states be $\chi_{i,g}$. Denote the product distribution $\prod_g \chi_{i,g}$ by Q_i. Let $\mathcal{G}(s, a)$ denote the two sets of indices where the state-action pair (s, a) is observed in trajectories n and m. For brevity, we will abbreviate $\mathcal{G}(s, a)$ to $\mathcal{G}$. Note that the sizes of these two sets are exactly $c_{n,i}$ and $c_{m,i}$ respectively.

We first prove some preliminary lemmas.

A.6.1.1 Decomposition of $|\text{dist}_{1,(s,a)} - \|\Delta_{m,n}(s, a)\|_2^2|$

Lemma A.6.2. *We claim that for each fixed value of $\mathcal{G}(s, a)$ (abbreviated to $\mathcal{G}$), with probability at least $1 - \delta$, the following bound holds.*

$$\left|\text{dist}_{1,(s,a)} - \|\Delta_{m,n}(s,a)\|_2^2\right| \leq \sum_{i=1}^{2} 2\left\|\boldsymbol{\Delta}_i - \mathbb{E}_{Q_i}[\boldsymbol{\Delta}_i \mid \mathcal{G}]\right\|_2 + 4\epsilon_{sub}(\delta) + 4\left(\max_i \mathbb{1}_{c_{n,i}=0} + \max_i \mathbb{1}_{c_{m,i}=0}\right) \quad \text{(A.15)}$$

Here $c_{n,i} = N(n,i,s,a)$, $\epsilon_{sub}(\delta)$ is the high probability bound on $\|\mathbb{P}_{j_l}(\cdot \mid s,a) - \boldsymbol{V}_{s,a}\boldsymbol{V}_{s,a}^T\mathbb{P}_{j_l}(\cdot \mid s,a)\|_2$ from Theorem 3.4.2 (satisfied with probability $> 1-\delta$), and

$$\boldsymbol{\Delta}_i^T = (\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \hat{\mathbb{P}}_{m,i}(\cdot \mid s,a))^T\boldsymbol{V}_{s,a}$$

Remark 13. In the inequality,

- The first term is a concentration-type term, which will be broken into an "independent concentration" error and a mixing error to account for the low but non-zero dependence across blocks.
- The second term accounts for subspace estimation error.
- The third term accounts for actually observing s, a in our blocks.

Proof. We first establish a simple inequality.

$$\begin{aligned}
&|\,\text{dist}_{1,(s,a)} - \mathbb{E}_{Q_1}[\boldsymbol{\Delta}_1^T \mid \mathcal{G}]\mathbb{E}_{Q_2}[\boldsymbol{\Delta}_2 \mid \mathcal{G}]| \\
&\quad = |\boldsymbol{\Delta}_1^T\boldsymbol{\Delta}_2 - \mathbb{E}_{Q_1}[\boldsymbol{\Delta}_1^T \mid \mathcal{G}]\mathbb{E}_{Q_2}[\boldsymbol{\Delta}_2| \mid \mathcal{G}] \\
&\quad \leq |(\boldsymbol{\Delta}_1^T - \mathbb{E}_{Q_1}[\boldsymbol{\Delta}_1^T \mid \mathcal{G}])\mathbb{E}_{Q_2}[\boldsymbol{\Delta}_2 \mid \mathcal{G}]| + |\boldsymbol{\Delta}_1^T(\boldsymbol{\Delta}_2 - \mathbb{E}_{Q_2}[\boldsymbol{\Delta}_2 \mid \mathcal{G}])| \\
&\quad \leq \|\boldsymbol{\Delta}_1 - \mathbb{E}_{Q_1}[\boldsymbol{\Delta}_1 \mid \mathcal{G}]\|_2 \|\mathbb{E}_{Q_2}[\boldsymbol{\Delta}_2 \mid \mathcal{G}]\|_2 + \|\boldsymbol{\Delta}_1\|_2 \|\boldsymbol{\Delta}_2 - \mathbb{E}_{Q_2}[\boldsymbol{\Delta}_2 \mid \mathcal{G}]\|_2 \\
&\quad \leq 2\|\boldsymbol{\Delta}_1 - \mathbb{E}_{Q_1}[\boldsymbol{\Delta}_1 \mid \mathcal{G}]\|_2 + 2\|\boldsymbol{\Delta}_2 - \mathbb{E}_{Q_2}[\boldsymbol{\Delta}_2 \mid \mathcal{G}]\|_2 \qquad \text{(A.16)}
\end{aligned}$$

Remark 14. Notice that because of this inequality, the double estimator does not impact any theoretical guarantees for exact clustering w.h.p, which is the form of the guarantees in both Kong et al. [2020] and Chen and Poor [2022]. However, we find that using a double estimator allows for better performance in real life. This makes sense because while exact clustering doesn't need a double estimator, approximate clustering w.h.p. does depend on the expectation of the distances across pairs of trajectories. This expectation is controlled by the covariance of $\boldsymbol{\Delta}_1$ and $\boldsymbol{\Delta}_2$.

We define the following quantity.

$$\textbf{diff}_i = \left(\mathbb{1}_{c_{n,i}\neq 0}\mathbb{P}_{k_m}(\cdot \mid s,a) - \mathbb{1}_{c_{m,i}\neq 0}\mathbb{P}_{k_n}(\cdot \mid s,a)\right)$$

Note that $\|\mathbf{diff}_i\|_2 \leq 2$. Note the following expectation, which uses the dieas from equation A.12.

$$\begin{aligned}
\mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}] &= \mathbb{E}_{Q_i}\left[\mathbf{V}_{s,a}^T(\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \hat{\mathbb{P}}_{m,i}(\cdot \mid s,a))\right] \\
&= \mathbf{V}_{s,a}^T\left(\mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}] - \mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{m,i}(\cdot \mid s,a) \mid \mathcal{G}]\right) \\
&= \mathbf{V}_{s,a}^T\left(\mathbb{E}_{Q_i}\left[\frac{\mathbf{w}_{n,i}}{c_{n,i}}\mathbb{1}_{c_{n,i}\neq 0} \mid \mathcal{G}\right] - \mathbb{E}_{Q_i}\left[\frac{\mathbf{w}_{m,i}}{c_{m,i}}\mathbb{1}_{c_{m,i}\neq 0} \mid \mathcal{G}\right]\right) \\
&= \mathbf{V}_{s,a}^T\left(\frac{\mathbb{E}_{Q_i}[\mathbf{w}_{n,i} \mid \mathcal{G}]}{c_{n,i}}\mathbb{1}_{c_{n,i}\neq 0} - \frac{\mathbb{E}_{Q_i}[\mathbf{w}_{m,i} \mid \mathcal{G}]}{c_{m,i}}\mathbb{1}_{c_{m,i}\neq 0}\right) \\
&= \mathbf{V}_{s,a}^T\left(\frac{\mathbb{P}_{k_n}(\cdot \mid s,a)c_{n,i}}{c_{n,i}}\mathbb{1}_{c_{n,i}\neq 0} - \frac{\mathbb{P}_{k_m}(\cdot \mid s,a)c_{m,i}}{c_{m,i}}\mathbb{1}_{c_{m,i}\neq 0}\right) \\
&= \mathbf{V}_{s,a}^T\left(\mathbb{P}_{k_n}(\cdot \mid s,a)\mathbb{1}_{c_{n,i}\neq 0} - \mathbb{P}_{k_m}(\cdot \mid s,a)\mathbb{1}_{c_{m,i}\neq 0}\right) \\
&= \mathbf{V}_{s,a}^T\mathbf{diff}_i
\end{aligned}$$

We recall the following definition before proceeding to show the main inequality.

$$\Delta_{m,n}(s,a) = \mathbb{P}_{k_m}(\cdot \mid s,a) - \mathbb{P}_{k_n}(\cdot \mid s,a)$$

$$\begin{aligned}
&\left|\mathbb{E}_{Q_1}[\mathbf{\Delta}_1^T \mid \mathcal{G}]\mathbb{E}_{Q_2}[\mathbf{\Delta}_2 \mid \mathcal{G}] - \|\Delta_{m,n}(s,a)\|_2^2\right| \\
&\quad= \left|\mathbf{diff}_1^T\mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbf{diff}_2 - \mathbf{diff}_1^T\mathbf{diff}_2\right| + \left|\mathbf{diff}_1^T\mathbf{diff}_2 - \|\Delta_{m,n}(s,a)\|_2^2\right| \\
&\quad\leq \|\mathbf{diff}_1\|_2 \left\|\mathbf{diff}_2 - \mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbf{diff}_2\right\|_2 + \|\mathbf{diff}_1 - \Delta_{m,n}(s,a)\|_2 \|\mathbf{diff}_2\|_2 \\
&\qquad+ \|\mathbf{diff}_1\|_2 \|\mathbf{diff}_2 - \Delta_{m,n}(s,a)\|_2 \\
&\quad\leq \|\mathbf{diff}_1\|_1 \left\|\mathbf{diff}_2 - \mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbf{diff}_2\right\|_2 + \|\mathbf{diff}_1 - \Delta_{m,n}(s,a)\|_2 \|\mathbf{diff}_2\|_1 \\
&\qquad+ \|\mathbf{diff}_1\|_1 \|\mathbf{diff}_2 - \Delta_{m,n}(s,a)\|_2 \\
&\quad\leq 2\left\|\mathbf{diff}_2 - \mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbf{diff}_2\right\|_2 + 2\|\mathbf{diff}_1 - \Delta_{m,n}(s,a)\|_2 \\
&\qquad+ 2\|\mathbf{diff}_2 - \Delta_{m,n}(s,a)\|_2 \\
&\quad\leq 2\left\|\mathbb{P}_{k_m}(\cdot \mid s,a) - \mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbb{P}_{k_m}(\cdot \mid s,a)\right\|_2 \\
&\qquad+ 2\left\|\mathbb{P}_{k_n}(\cdot \mid s,a) - \mathbf{V}_{s,a}\mathbf{V}_{s,a}^T\mathbb{P}_{k_n}(\cdot \mid s,a)\right\|_2 \\
&\qquad+ 2\left\|\mathbb{1}_{c_{m,1}=0}\mathbb{P}_{k_m}(\cdot \mid s,a) - \mathbb{1}_{c_{n,1}=0}\mathbb{P}_{k_n}(\cdot \mid s,a)\right\|_2 \\
&\qquad+ 2\left\|\mathbb{1}_{c_{m,2}=0}\mathbb{P}_{k_m}(\cdot \mid s,a) - \mathbb{1}_{c_{n,2}=0}\mathbb{P}_{k_n}(\cdot \mid s,a)\right\|_2 \\
&\quad\leq 4\epsilon_{sub}(\delta) + 2\big(\mathbb{1}_{c_{m,1}=0}\|\mathbb{P}_{k_m}(\cdot \mid s,a)\|_2 + \mathbb{1}_{c_{n,1}=0}\|\mathbb{P}_{k_n}(\cdot \mid s,a)\|_2\big) \\
&\qquad+ 2\big(\mathbb{1}_{c_{m,2}=0}\|\mathbb{P}_{k_m}(\cdot \mid s,a)\|_2 + \mathbb{1}_{c_{n,2}=0}\|\mathbb{P}_{k_n}(\cdot \mid s,a)\|_2\big) \\
&\quad\leq 4\epsilon_{sub}(\delta) + 4\left(\max_i \mathbb{1}_{c_{n,i}=0} + \max_i \mathbb{1}_{c_{m,i}=0}\right)
\end{aligned}$$

Combining this with inequality A.16, we have the following final bound.

$$\left|\text{dist}_{1,(s,a)} - \|\Delta_{m,n}(s,a)\|_2^2\right| \leq \sum_{i=1}^{2} 2\left\|\mathbf{\Delta}_i - \mathbb{E}_{Q_i}[\mathbf{\Delta}_i \mid \mathcal{G}]\right\|_2 + 4\epsilon_{sub}(\delta) + 4\left(\max_i \mathbb{1}_{c_{n,i}=0} + \max_i \mathbb{1}_{c_{m,i}=0}\right) \quad \text{(A.17)}$$

where we remind the reader that $c_{n,i} = N(n,i,s,a)$ and recall the definition of Δ_i.

$$\mathbf{\Delta}_i^T = (\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \hat{\mathbb{P}}_{m,i}(\cdot \mid s,a))^T \mathbf{V}_{s,a}$$

□

A.6.1.2 Bounding the concentration-type term

We bound the first term in the decomposition lemma (Lemma A.6.2) with high probability.

Lemma A.6.3. *With probability at least $1-\delta$, when $T_n \geq \Omega\left(Gt_{mix}\log\left(\frac{G}{\delta}\log(1/\alpha)\right)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$, we have the following bound.*

$$\|\Delta_i - \mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}])\|_2 \leq O\left(\sqrt{\frac{K+\log(1/\delta)}{G\alpha}}\right)$$

Proof. Recall that the joint distribution of the observations over the pair of trajectories (m,n) is χ. Its marginals on the segments Ω_i are χ_i. The marginals on each of the G single-step sub-blocks is $\chi_{i,g}$. The product distribution $\prod_g \chi_{i,g}$ is Q_i. Recall that $\mathcal{G}(n,s,a)$ denotes the two sets of indices where (s,a) is observed in trajectory n and m respectively, and the sets have sizes $c_{n,i}$ and $c_{m,i}$ respectively.

Let $\mathbf{w}_{n,i,g}$ be the one hot vector of the next state if the (i,g) sub-block witnesses (s,a), and the zero vector otherwise. Let $c_{n,i,g}$ be the indicator of (s,a) in the (i,g) sub-block. Then $\mathbf{w}_{n,i} = \sum_g \mathbf{w}_{n,i,g}$ and $c_{n,i} = \sum_g c_{n,i,g}$.

1. Covering argument for the product distribution

Pick a unit vector $\mathbf{u} \in \mathcal{R}^K$ and consider the following inequality. Remember that we abbreviate $\mathcal{G}(n,s,a)$ to $\mathcal{G}$.

$$|\mathbf{u}^T(\Delta_i - \mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}])| \leq |\mathbf{u}^T\mathbf{V}_{s,a}(\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}])| + |\mathbf{u}^T\mathbf{V}_{s,a}(\hat{\mathbb{P}}_{m,i}(\cdot \mid s,a) - \mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{m,i}(\cdot \mid s,a) \mid \mathcal{G}])|$$

We work with the term for trajectory n, WLOG. Any bounds thus obtained will also apply to trajectory m. Notice the following equation.

$$|\mathbf{u}^T\mathbf{V}_{s,a}^T(\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}])| \\ = \left| \frac{1}{c_{n,i}} \sum_{g \in \mathcal{G}(n,s,a)} \left(\mathbf{u}^T\mathbf{V}_{s,a}^T\mathbf{w}_{n,i,g} - \mathbb{E}_{Q_i}[\mathbf{u}^T\mathbf{V}_{s,a}^T\mathbf{w}_{n,i,g} \mid \mathcal{G}]\right)\right|$$

Note that $|\mathbf{u}^T\mathbf{V}_{s,a}^T\mathbf{w}_{n,i,g}| \leq \|\mathbf{u}\|_2\|\mathbf{V}_{s,a}^T\mathbf{w}_{n,i,g}\|_2 \leq 1$. Note that conditioned on the set of (s,a) observations in trajectory n, the next states are independent under the product distribution Q_i (but not under χ_i, of course). Now, using the conditional version of Hoeffding's inequality from Lemma A.5.4, we get the following bound.

$$\mathbb{P}_{Q_i}\left(\left|\frac{1}{c_{n,i}} \sum_{g \in \mathcal{G}(n,s,a)} \left(\mathbf{u}^T\mathbf{V}_{s,a}^T\mathbf{w}_{n,i,g} - \mathbb{E}_{Q_i}[\mathbf{u}^T\mathbf{V}_{s,a}^T\mathbf{w}_{n,i,g} \mid \mathcal{G}]\right)\right| > \frac{\epsilon}{8}\middle| \mathcal{G}\right) \leq 2e^{-\frac{\epsilon^2 c_{n,i}}{32}}$$

Note that if $X \leq Y + Z$, then $\mathbb{P}(X > \frac{\epsilon}{4}) \leq \mathbb{P}(Y > \frac{\epsilon}{8}) + \mathbb{P}(Z > \frac{\epsilon}{8})$ by a union bound. We apply this to the inequalities above with $X = |\mathbf{u}^T(\Delta_i - \mathbb{E}_{Q_i}[\Delta_i])|$ to get the following concentration inequality.

$$\mathbb{P}_{Q_i}\left(|\mathbf{u}^T(\Delta_i - \mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}])| > \frac{\epsilon}{4} \mid \mathcal{G}\right) \leq 2e^{-\frac{\epsilon^2 c_{n,i}}{32}} + 2e^{-\frac{\epsilon^2 c_{n,i}}{32}} = 4e^{-\frac{\epsilon^2 c_{n,i}}{32}}$$

Consider a covering of $\mathbb{S}^{K-1}$ by balls of radius $1/4$. We will need at most 12^K such balls. Call the set of their centers C. We know that for any vector $\mathbf{v}$, the following holds.

$$\sup_{\|\mathbf{u}\|_2 \leq 1} \mathbf{u}^T\mathbf{v} = \|\mathbf{v}\|_2 \leq 2 \sup_{\mathbf{u} \in C} \mathbf{u}^T\mathbf{v}$$

We use this to arrive at the concentration inequality below.

$$\mathbb{P}_{Q_i}\left(\|\Delta_i - \mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}])\|_2 > \frac{\epsilon}{2} \mid \mathcal{G}\right) \leq \mathbb{P}_{Q_i}\left(\exists \mathbf{u} \in C; |\mathbf{u}^T(\Delta_i - \mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}])| > \frac{\epsilon}{4} \mid \mathcal{G}\right) \\ \leq \sum_{\mathbf{u} \in C} \mathbb{P}_{Q_i}\left(|\mathbf{u}^T(\Delta_i - \mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}])| > \frac{\epsilon}{4} \mid \mathcal{G}\right)$$

$$< 4 * 12^K * e^{-\frac{\epsilon^2 c_{n,i}}{32}}$$

3. Accounting for non-independence (mixing error)

We know that we can bound the difference in the probability of any event E between χ_i and Q_i by applying Lemma A.5.3 to the function $h = \mathbb{1}_E$ with $n = G$ and $C = 1$ as we have before, giving us the following inequality.

$$\begin{aligned}\mathbb{P}_{\chi_i}\left(\|\Delta_i - \mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}])\|_2 > \frac{\epsilon}{2}\right) &\leq \mathbb{P}_{Q_i}\left(\|\Delta_i - \mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}])\|_2 > \frac{\epsilon}{2}\right) + \frac{\delta}{2} + 4G\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}} \\ &\leq 4 * 12^K * e^{-\frac{\epsilon^2 G d_{min}(s,a)}{128}} + \frac{\delta}{2} + 4G\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}}\end{aligned}$$

We know that both terms are less than $\frac{\delta}{4}$ when $T_n \geq \Omega\left(Gt_{mix}\log\left(\frac{G}{\delta}\right)\right)$ and $G \geq \Omega\left(\frac{K+\log(1/\delta)}{\epsilon^2\alpha}\right)$, since $d_{min}(s,a) \geq \alpha/3$. We thus have the following bound with probability at least $1-\delta$, when $T_n \geq \Omega\left(Gt_{mix}\log\left(\frac{G}{\delta}\right)\log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$.

$$\|\Delta_i - \mathbb{E}_{Q_i}[\Delta_i \mid \mathcal{G}])\|_2 \leq O\left(\sqrt{\frac{K+\log(1/\delta)}{G\alpha}}\right)$$

□

A.6.1.3 Bounding the probability of not observing s, a

We bound the third term in the decomposition lemma (Lemma A.6.2) with high probability. We first need an auxiliary lemma for this.

Lemma A.6.4. *For $T_n \geq \Omega\left(Gt_{mix}\log(1/\alpha)\right)$, we have the following bound.*

$$\mathbb{P}(c_{n,i} = 0) \leq \left(1 - \frac{d_{min}(s,a)}{2}\right)^G + 4G\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}}$$

Remark 15. Again, we can think of this sum as a bound on the probability of not observing s, a in the blocks if they were independent (the first term) versus a mixing error between blocks to account for their non-independence (the second term).

Proof. Recall that the joint distribution of the observations over the pair of trajectories (m, n) is χ.

Its marginals on the segments Ω_i are χ_i. The marginals on each of the G single-step sub-blocks is $\chi_{i,g}$. The product distribution $\prod_g \chi_{i,g}$ is Q_i. Recall that $\mathcal{G}(n, s, a)$ denotes the two sets of indices where (s, a) is observed in trajectory n and m respectively, and the sets have sizes $c_{n,i}$ and $c_{m,i}$ respectively.

Remember that $\mathbf{w}_{n,i,g}$ is the one hot vector of the next state if the (i, g) sub-block witnesses (s, a), and the zero vector otherwise, and that $c_{n,i,g}$ is the indicator of (s, a) in the (i, g) sub-block. Also recall that then $\mathbf{w}_{n,i} = \sum_g \mathbf{w}_{n,i,g}$ and $c_{n,i} = \sum_g c_{n,i,g}$.

Define $h := \prod_{g=1}^{G}(1 - c_{n,i,g})$. Under any distribution Q over these sub-blocks, $\mathbb{E}_Q h$ is the probability of not observing s, a in any of them. Let $d_{i,g,n}$ be the distribution of state-action pairs at the first observation of sub-block (i, g). Let $d_{k_n}(\cdot, \cdot)$ be the stationary distribution under label k_n for state-action pairs. We use Lemma A.5.3 with h as above, $C = 1$, $n = G$ and $a_n = \frac{T_n}{8G}$ to note the following chain of inequalities.

$$
\begin{aligned}
\mathbb{P}(c_{n,i} = 0) &= \mathbb{E}_{\chi_i} h \\
&\leq \mathbb{E}_{Q_i} h + |\mathbb{E}_{Q_i} h - \mathbb{E}_{\chi_i} h| \\
&\leq \left(\prod_{g=1}^{G} \mathbb{E}_{Q_i}(1 - c_{n,i,g}) \right) + 4G\lambda^{\frac{T_n}{8G}} \\
&\leq \left(\prod_{g=1}^{G} (1 - d_{k_n}(s, a) + TV(d_{i,g,n}, d_{k_n}) \right) + 4G\lambda^{\frac{T_n}{8G}} \\
&\leq \left(\prod_{g=1}^{G} (1 - d_{k_n}(s, a) + 4\lambda^{\frac{T_n}{8G}}) \right) + 4G\lambda^{\frac{T_n}{8G}} \\
&= \left(1 - d_{k_n}(s, a) + 4\lambda^{\frac{T_n}{8G}} \right)^G + 4G\lambda^{\frac{T_n}{8G}} \\
&\leq \left(1 - \frac{d_{k_n}(s, a)}{2} \right)^G + 4G\lambda^{\frac{T_n}{8G}} \\
&\leq \left(1 - \frac{d_{min}(s, a)}{2} \right)^G + 4G\lambda^{\frac{T_n}{8G}}
\end{aligned}
$$

where the inequality in the second to last line holds for $T_n \geq \Omega\left(G t_{mix} \log(1/\alpha)\right) \geq \Omega\left(G t_{mix} \log(1/d_{min}(s, a))\right)$.

□

From the above lemma, the following corollary immediately follows by getting conditions to bound each term on the right hand side by $\delta/2$, upon also noting that $-\log(1-x) \geq x$, so $\log\left(\frac{1}{1-\alpha/2}\right) \geq \alpha/2$.

Corollary A.6.5. *For $T_n \geq \Omega\left(Gt_{mix}\log(G/\delta)\log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha}\right)$, we have with probability at least $1-\delta$ that*

$$4\left(\max_i \mathbb{1}_{c_{n,i}=0} + \max_i \mathbb{1}_{c_{m,i}=0}\right) = 0$$

A.6.1.4 Combining the bounds

We finally combine these lemmas to prove Lemma A.6.1 – the lemma that this section was dedicated to. The conditions of the lemmas combine to ask that $T_n \geq \Omega\left(Gt_{mix}\log(G/\delta)\log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$.

Proof of Lemma A.6.1. Combining the decomposition from Lemma A.6.2 with the bounds in Lemma A.6.3 and Corollary A.6.5, we conclude using union bounds on the low probability events that we are excluding that there is a universal constant C_1 so that with probability at least $1-\delta$,

$$\left|\text{dist}_{1,(s,a)} - \|\Delta_{m,n}(s,a)\|_2^2\right| \leq C_1\left(\sqrt{\frac{K+\log(1/\delta)}{G\alpha}}\right) + 4\epsilon_{sub}(\delta/2)$$

whenever $T_n \geq \Omega\left(Gt_{mix}\log(G/\delta)\log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$.

□

□

A.7 Guarantees for one step of the EM Algorithm for mixtures of MDPs

Remember that the M-step is just the model estimation step, so Theorem 3.4.4 provides guarantees for that. We also have the following guarantees for the E-step of hard EM.

Theorem A.7.1. *Consider any (s,a) with $d_{min}(s,a) \geq \alpha/3$ where model estimation accuracy is ϵ with $\epsilon \leq \min(\Delta/4, \Delta^2 g_{min}/64)$ where g_{min} is the least non-zero value of $\mathbb{P}_k(s' \mid s,a)$ across k, s'. Using log-likelihood ratios of transitions of all such (s,a) pairs, we can classify any set of N new trajectories with probability $1-\delta$ if it has length $T_n = \Omega(t_{mix}\log^4(N/\delta)\log^3(1/f_{min})/\alpha^3\Delta^3)$.*

Remark 16. The dependence on g_{min} is unavoidable. For example, if the estimate for the models was only off at the value of k, s' attaining g_{min} and our estimate for g_{min} was $\hat{\mathbb{P}}_k(s' \mid s, a) = 0$, then no trajectory from label k witnessing s' will get correctly classified. This event will happen roughly with probability g_{min}, up to a mixing error, and g_{min} cannot be made less than some arbitrary δ chosen to bound the probability of all undesirable events.

Proof. We are inspired by the lower bound obtained in Lemma 1 of Wong and Shen [1995] for obtaining our sample complexity bounds. Consider a separating state-action pair s, a. We first establish Hellinger distance lower bounds between the distributions $\hat{\mathbb{P}}_k(\cdot \mid s, a)$ and $\hat{\mathbb{P}}_l(\cdot \mid s, a)$. Notice that

$$TV(\hat{\mathbb{P}}_k(\cdot \mid s, a), \mathbb{P}_k(\cdot \mid s, a)) = \frac{1}{2}\|\hat{\mathbb{P}}_k(\cdot \mid s, a) - \mathbb{P}_k(\cdot \mid s, a)\|_1 \leq \epsilon/2 \leq \Delta/4$$

The same holds for l as well. Combining the latter with $\|\mathbb{P}_k(\cdot \mid s, a) - \mathbb{P}_l(\cdot \mid s, a)\|_1 \geq \|\mathbb{P}_k(\cdot \mid s, a) - \mathbb{P}_l(\cdot \mid s, a)\|_2 \geq \Delta$ and using the inequality $H(P, Q) \geq TV(P, Q)/\sqrt{2}$, we get the following bound.

$$H(\mathbb{P}_k(\cdot \mid s, a), \hat{\mathbb{P}}_l(\cdot \mid s, a)) \geq \frac{1}{\sqrt{2}} TV(\hat{\mathbb{P}}_k(\cdot \mid s, a), \hat{\mathbb{P}}_l(\cdot \mid s, a)) \geq \frac{\Delta}{4\sqrt{2}}$$

We now recall notation from the previous section. Again, we modify notation slightly, in a natural way. Let χ_n be the joint distribution of observations recorded in trajectory n, with their marginals on each single-element sub-block being $\chi_{n,g}$. Let Q_n be the product distribution $Q_n = \prod_{n,g} \chi_{n,g}$. Let $\mathcal{G}(n, s, a)$ be the set of sub-blocks (n, g) in which (s, a) is observed in trajectory n. Let c_n be the size of this set. We have the following lemma.

Lemma A.7.2. *Let the random variables for the next states following each (s, a) observation given by $S_1, S_2, \ldots S_{c_n}$ and let the true label be $k_n = k$. Then for any $l \neq k$, consider the likelihood ratio over next state transitions from (s, a).*

$$LR_n(s, a) = \prod_{i=1}^{c_n} \frac{\hat{\mathbb{P}}_k(S_i \mid s, a)}{\hat{\mathbb{P}}_l(S_i \mid s, a)}$$

We claim that $LR_n(s, a) > 0$ with probability at least $1 - \delta$ for $T_n \geq \Omega\left(Gt_{mix} \log\left(\frac{G}{\delta}\right) \log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/f_{min}) \log(1/\delta)}{\alpha^2 \Delta^2}\right)$.

Just like in the proof of Theorem 3.4.3, now set $G = \left(\frac{T_n}{t_{mix}}\right)^{\frac{2}{3}}$. Then a sufficient condition on T_n to meet the conditions of the lemma is $T_n = \Omega(t_{mix} \log^4(1/\delta) \log^3(1/f_{min})/\alpha^3\Delta^3)$.

Now remember that upon choosing an occurrence threshold β of order α, we will have at most

$O(1/\alpha)$ many (s,a) pairs in Freq_β. By applying a union bound over all (s,a) pairs in Freq_β, we get that with probability $1-\delta$, we get that the sum of the log-likelihood ratios of next-state transitions starting in Freq_β between the true label's model estimate and any other label's model estimate is positive whenever $T_n = \Omega(t_{mix}\log^4(1/\delta)\log^3(1/f_{min})/\alpha^3\Delta^3)$.

We now take another union bound over the N new trajectories to get that we can exactly classify all of them with probability at least $1-\delta$ whenever $T_n \geq \Omega(t_{mix}\log^4(N/\delta)\log^3(1/f_{min})/\alpha^3\Delta^3)$.

A.7.1 Proof of Lemma A.7.2

We first perform a computation analogous to Lemma 1 in Wong and Shen [1995]. Let $D_1 = \mathbb{P}_k(\cdot \mid s,a)$, $D_2 = \mathbb{P}_l(\cdot \mid s,a)$, $\hat{D}_1 = \hat{\mathbb{P}}_k(\cdot \mid s,a)$, $\hat{D}_2 = \hat{\mathbb{P}}_l(\cdot \mid s,a)$. Fix $b > 0$. We use the conditional Markov inequality and the fact that conditioned on $\mathcal{G}(n,s,a)$ and under the product distribution $\hat{Q}_n$, the Hellinger distance between the next-state distributions at any (s,a) observation is $H(\hat{D}_1,\hat{D}_2)$, which satisfies $H(\hat{D}_1,\hat{D}_2) \geq \Delta/4\sqrt{2}$. This is crucially due to the independence and the fact that we are fixing $\mathcal{G}(n,s,a)$ by conditioning on it. As usual, abbreviate $\mathcal{G}(n,s,a)$ to $\mathcal{G}$ for brevity.

$$
\begin{aligned}
\mathbb{P}_{Q_n}(LR_n(s,a) \leq e^{c_n b/2} \mid \mathcal{G}) &= \mathbb{P}_{Q_n}\left(\prod_{i=1}^{c_n}\left(\frac{\hat{D}_2(S_i)}{\hat{D}_1(S_i)}\right)^{1/2} \geq e^{-c_n b/2}\middle|\mathcal{G}\right) \\
&\leq e^{c_n b/2}\left(\mathbb{E}_{Q_n}\left[\left(\frac{\hat{D}_2(S_i)}{\hat{D}_1(S_i)}\right)^{1/2}\middle|\mathcal{G}\right]\right)^{c_n} \\
&= e^{c_n b/2}\left(\mathbb{E}_{D_1}\left[\left(\frac{\hat{D}_2(S_i)}{\hat{D}_1(S_i)}\right)^{1/2}\right]\right)^{c_n} \\
&= e^{c_n b/2}\left(\mathbb{E}_{D_1}\left[\left(\frac{D_1(S_i)}{\hat{D}_1(S_i)}\right)^{1/2}\left(\frac{\hat{D}_2(S_i)}{D_1(S_i)}\right)^{1/2}\right]\right)^{c_n} \\
&\leq e^{c_n b/2}\left(\mathbb{E}_{D_1}\left[\left(1+\Delta^2/64\right)^{1/2}\left(\frac{\hat{D}_2(S_i)}{D_1(S_i)}\right)^{1/2}\right]\right)^{c_n} \\
&= e^{c_n b/2}\left(1+\Delta^2/64\right)^{c_n/2}\left(1-\frac{H(D_1,D_2)^2}{2}\right)^{c_n} \\
&\leq e^{c_n b/2}\left(1-\Delta^2/128\right)^{c_n/2} \\
&\leq e^{c_n b/2}e^{-c_n\Delta^2/128}
\end{aligned}
$$

Setting $b = \Delta^2/256$, we get that $\mathbb{P}_{Q_n}(LR_n(s,a) \leq e^{c_n\Delta^2/256} \mid \mathcal{G}) \leq e^{-c_n\Delta^2/256}$. Now by

following a very similar computation to that in point 2 in section A.6.1.2, we get that for $T_n \geq \Omega(Gt_{mix}\log(1/\alpha))$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$, $c_n \geq Gd_{min}(s,a)/4$ with probability at least $1-\delta/2$. That is, for such T_n and G,

$$\mathbb{P}_{Q_n}(LR_n(s,a) \leq e^{Gd_{min}(s,a)\Delta^2/512} \mid \mathcal{G}) \leq \mathbb{P}_{Q_n}(LR_n(s,a) \leq e^{c_n\Delta^2/128} \mid \mathcal{G}) \leq e^{-Gd_{min}(s,a)\Delta^2/512} + \frac{\delta}{2}$$

Since this holds for any value of $\mathcal{G} = \mathcal{G}(n,s,a)$, we can say that with probability at least $1-\delta$, for $T_n \geq \Omega(Gt_{mix}\log(1/\alpha))$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$, $c_n \geq Gd_{min}(s,a)/4$, we have the following bound.

$$\mathbb{P}_{Q_n}(LR_n(s,a) \leq e^{Gd_{min}(s,a)\Delta^2/512}) \leq e^{-Gd_{min}(s,a)\Delta^2/512} + \frac{\delta}{2}$$

After following a computation very similar to that in point 3 of section A.6.1.2, we get that for $T_n \geq \Omega\left(Gt_{mix}\log\left(\frac{G}{\delta}\right)\log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2\Delta^2}\right)$,

$$\mathbb{P}_{\chi}(LR_n(s,a) \leq e^{Gd_{min}(s,a)\Delta^2/512}) \leq \delta$$

Note that we want $e^{Gd_{min}(s,a)\Delta^2/512} \geq f_l/f_k$, in which case it suffices to ask $e^{Gd_{min}(s,a)\Delta^2/512} \geq 1/f_{min}$. Combining this with earlier conditions, for $G \geq \Omega\left(\frac{\log(1/\delta)\log(1/f_{min})}{\alpha^2\Delta^2}\right)$ and $T_n \geq \Omega\left(Gt_{mix}\log\left(\frac{G}{\delta}\right)\log(1/\alpha)\right)$,

$$\mathbb{P}_{\chi}\left(\frac{f_k}{f_l}LR_n(s,a) \leq 1\right) \leq \delta$$

□

A.8 Proof of Theorem 3.4.4

Theorem 3.4.4 (Model Estimation Guarantee)**.** *For any state action pair (s,a) with $d_{min}(s,a) \geq \alpha/3$, and for $GN_{clust} \geq \Omega\left(\frac{\log(1/\delta)}{f_{min}^2\alpha^2}\right)$ and $T_n \geq \Omega(Gt_{mix}\log(G/\delta))$, with probability greater than $1-\delta$,*

$$\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_1$$

is bounded above by

$$O\left(\left(\frac{t_{mix}}{T_n}\right)^{1/3}\sqrt{\frac{1}{N_{clust}f_{min}\alpha}(S+\log(\frac{1}{\delta}))}\right)$$

Proof. The proof is quite straightforward and employs the techniques used so far, especially those used in section A.6.1.2. Let k be the (now known) label that we're working with.

We modify previous notation a bit for this proof. For brevity of notation, we denote by $c_{n,g}$ the indicator variable for observing (s,a) in the g^{th} single-step sub-block of the trajectory n. Denote by $\mathbf{w}_{n,g}$ one-hot vector of the next state observed if the currect state-action pair is (s,a), and set it to the zero-vector otherwise. Note that $\sum_g c_{n,g} = N(n,s,a)$ and $\sum_g \mathbf{w}_{n,g} = \mathbf{N}(n,s,a,\cdot)$. We denote the set of indices (n,g) of all s,a observations that come from label k (across the GN_{clust} observations recorded) by $\mathcal{N}(s,a,k)$. Let the size of this set be $N(s,a,k)$. Note that $N(s,a,k) = \sum_{n\in\mathcal{C}_k} N(n,s,a) = \sum_{n,g} c_{n,g}$. Also note the following alternate expression for $\hat{\mathbb{P}}_k(\cdot \mid s,a)$.

$$\hat{\mathbb{P}}_k(\cdot \mid s,a) := \frac{\sum_{(n,g)\in\mathcal{N}(s,a,k)} \mathbf{w}_{n,g}}{\sum_{(n,g)\in\mathcal{N}(s,a,k)} c_{n,g}}\mathbb{1}_{N(s,a,k)\neq 0} = \frac{\sum_{(n,g)\in\mathcal{N}(s,a,k)} \mathbf{w}_{n,g}}{N(s,a,k)}\mathbb{1}_{N(s,a,k)\neq 0} \tag{A.18}$$

Let χ_n be the joint distribution of observations recorded in trajectory n, with their marginals on each single-element sub-block being $\chi_{n,g}$. Let χ be the joint distribution of all observations recorded across all trajectories. Since the trajectories are independent, we know that $\chi = \prod_n \chi_n$. Let Q_g be the joint distribution of the observations at the g^{th} sub-block. Note that this is also the marginal of the joint distribution χ on the g^{th} sub-block, and since the trajectories are independent, $Q_g = \prod_n \chi_{g,n}$. Finally, denote by Q the product distribution $\prod_g Q_g = \prod_g \prod_n \chi_{g,n}$. This would be the distribution if all observations recorded were independent (across sub-blocks).

1. Concentration under the product distribution

We have the following computation.

$$\begin{aligned}
\mathbb{E}_Q[\hat{\mathbb{P}}_k(\cdot \mid s,a) \mid \mathcal{N}(s,a,k)] &= \mathbb{E}_Q\left[\frac{\sum_{n\in\mathcal{N}_{clust}} \mathbf{w}_n}{N(s,a,k)}\mathbb{1}_{N(s,a,k)\neq 0}\middle| N(s,a,k)\right] \\
&= \mathbb{E}\left[\frac{\sum_{n\in\mathcal{N}(s,a,k)} \mathbf{w}_n}{N(s,a,k)}\middle| N(s,a,k)\right]\mathbb{1}_{N(s,a,k)\neq 0} \\
&= \frac{\sum_n \mathbb{E}_Q[\mathbf{w}_n \mid \mathcal{N}(s,a,k)]}{N(s,a,k)}\mathbb{1}_{N(s,a,k)\neq 0} \\
&= \frac{\sum_n \mathbb{P}_k(\cdot \mid s,a)c_n}{N(s,a,k)}\mathbb{1}_{N(s,a,k)\neq 0}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\mathbb{P}_k(\cdot \mid s,a)(\sum_n c_n)}{N(s,a,k)} \mathbb{1}_{N(s,a,k)\neq 0} \\
&= \frac{\mathbb{P}_k(\cdot \mid s,a)N(s,a,k)}{N(s,a,k)} \mathbb{1}_{N(s,a,k)\neq 0} \\
&= \mathbb{P}_k(\cdot \mid s,a)\mathbb{1}_{N(s,a,k)\neq 0}
\end{aligned}
$$

Now we set up our covering argument. Remember that $[-1,1]^S$ is the set of all vectors $\mathbf{u} \in \mathcal{R}^S$ with $\|u\|_\infty \leq 1$. Consider a covering of $[-1,1]^S$ by boxes of side length $\frac{1}{4}$ and centers lying in $[-1,1]^S$. We will need at most 12^S such boxes and if C is the set of their centers, then for any vector $\mathbf{v}$

$$
\|\mathbf{v}\|_1 = \sup_{\mathbf{u}\in[-1,1]^{S-1}} |\mathbf{u}^T\mathbf{v}| \leq 2 \max_{\mathbf{u}\in C} |\mathbf{u}^T\mathbf{v}| \leq 2\|\mathbf{v}\|_1
$$

Also, for any $\mathbf{u} \in C$, note that

$$
\begin{aligned}
|\mathbf{u}^T\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a)| &\leq \|\mathbf{u}\|_\infty \left\| \frac{\mathbf{w}_{n,1}}{c_{n,1}} \right\|_1 \\
&\leq \left\| \frac{\mathbf{w}_{n,1}}{c_{n,1}} \right\|_1 \\
&= 1
\end{aligned}
$$

and so $|\mathbf{u}^T\mathbb{E}_Q[\hat{\mathbb{P}}_k(\cdot \mid s,a) \mid \mathcal{N}(s,a,k)]| \leq \mathbb{E}[|\mathbf{u}^T\hat{\mathbb{P}}_k(\cdot \mid s,a)| \mid \mathcal{N}(s,a,k)] \leq 1$. Again, note that conditioned on the set of all (s,a) observations recorded, the next states $\mathbf{w}_{n,g}$ are all independent under the product distribution Q (but not under χ, of course). Recalling the expression for $\hat{\mathbb{P}}_k(\cdot \mid s,a)$ from equation A.18, this means that we can use the conditional version of Hoeffding's inequality, giving us the following bound.

$$
\mathbb{P}_Q\left(\left| \mathbf{u}^T(\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{E}_Q[\hat{\mathbb{P}}_k(\cdot \mid s,a) \mid \mathcal{N}(s,a,k)]) \right| > \frac{\epsilon}{4} \middle| \mathcal{N}(s,a,k) \right) < 2e^{-\frac{\epsilon^2 N(s,a,k)}{8}}
$$

Doing this for all 12^S vectors $\mathbf{u} \in C$, we get the following inequality.

$$
\mathbb{P}_Q\left(\left\| (\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_Q[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{N}(s,a,k)]) \right\|_1 > \frac{\epsilon}{2} \middle| \mathcal{N}(s,a,k) \right)
$$

is bounded above by

$$
\mathbb{P}_Q\left(\exists \mathbf{u} \in C; \left| \mathbf{u}^T(\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{E}_Q[\hat{\mathbb{P}}_k(\cdot \mid s,a) \mid \mathcal{N}(s,a,k)]) \right| > \frac{\epsilon}{4} \middle| \mathcal{N}(s,a,k) \right)
$$

$$\leq \sum_{\mathbf{u} \in C} \mathbb{P}_Q \left(\left| \mathbf{u}^T (\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{E}_Q[\hat{\mathbb{P}}_k(\cdot \mid s,a) \mid \mathcal{N}(s,a,k)]) \right| > \frac{\epsilon}{4} \middle| \mathcal{N}(s,a,k) \right)$$
$$< 12^S * e^{-\frac{\epsilon^2 N(s,a,k)}{8}}$$

2. Bounding $N(s,a,k)$ under the product distribution

Now note that $N(s,a,k) = \sum_{(n,g) \in \mathcal{N}_{clust} \times [G]} c_{n,g}$. So,

$$\mathbb{E}_Q[N(s,a,k)] = \sum_{(n,g) \in \mathcal{N}_{clust} \times [G]} \mathbb{E}_Q[c_{n,g}] = \sum_{(n,g) \in \mathcal{N}_{clust} \times [G]} \mathbb{P}_\chi(c_{n,g} \neq 0)$$

We can show the following inequality.

$$\mathbb{P}_\chi(c_{n,g} \neq 0) = \mathbb{P}_\chi(c_{n,g} \neq 0 \mid k_n = k) \mathbb{P}(k_n = k) \geq \frac{d_{min}(s,a)}{2} f_{min}$$

for $T_n \geq \Omega(G t_{mix} \log(1/\alpha))$, getting the last inequality by using a computation very similar to the one in equation A.4, along with the fact that $\mathbb{P}(k_n = k) = f_k$. So, $\mathbb{E}_Q[N(s,a,k)] \geq \frac{G N_{clust} f_{min} d_{min}(s,a)}{2}$.

$$\begin{aligned}
&\mathbb{P}_Q \left(N(s,a,k) < G N_{clust} \frac{f_{min} d_{min}(s,a)}{4} \right) \\
&= \mathbb{P}_Q \left(N(s,a,k) < G N_{clust} \frac{f_{min} d_{min}(s,a)}{2} - G N_{clust} \frac{f_{min} d_{min}(s,a)}{4} \right) \\
&\leq \mathbb{P}_Q \left(N(s,a,k) < \mathbb{E}[N(s,a,k)] - G N_{clust} \frac{f_{min} d_{min}(s,a)}{4} \right) \\
&= \mathbb{P}_Q \left(\sum_{(n,g) \in \mathcal{N}_{clust} \times [G]} c_{n,g} < \mathbb{E}[N(s,a,k)] - G N_{clust} \frac{f_{min} d_{min}(s,a)}{4} \right) \\
&\leq \exp \left(- \frac{f_{min}^2 d_{min}(s,a)^2 G N_{clust}}{8} \right)
\end{aligned}$$

This is less than $\delta/2$ for $G N_{clust} \geq \Omega \left(\frac{\log(1/\delta)}{f_{min}^2 \alpha^2} \right)$. So, with probability at least $1 - \delta/2$, for $G N_{clust} \geq \Omega \left(\frac{\log(1/\delta)}{f_{min}^2 \alpha^2} \right)$ and $T_n \geq \Omega(G t_{mix} \log(1/\alpha))$, we have the following bound.

$$\mathbb{P}_Q \left(\left\| (\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_Q[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{N}(s,a,k)]) \right\|_1 > \frac{\epsilon}{2} \right) \leq 12^S e^{-\frac{\epsilon^2 G N_{clust} f_{min} d_{min}(s,a)}{128}}$$

3. Mixing error to account for non-independence in the true joint distribution

Note that we can think of the combined dataset as a Markov chain over the tuple of n observations, with a joint distribution χ over observations. Its marginal over the g^{th} single-step sub-blocks is Q_g and $Q = \prod_g Q_g$. We now want to apply Lemma A.5.3, noting that the relevant function of this Markov chain is $\mathbb{1}_E$ where E is the event $\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_1 < \frac{\epsilon}{2}$. Clearly, in this case, n from the lemma is G and C from the lemma is 1. We use this to get the following bound.

$$\mathbb{P}_\chi\left(\left\|(\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_Q[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{N}(s,a,k)])\right\|_1 > \frac{\epsilon}{2}\right)$$

is bounded above by

$$\mathbb{P}_Q\left(\left\|(\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_Q[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{N}(s,a,k)])\right\|_1 > \frac{\epsilon}{2}\right) + 4G\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}}$$
$$\leq 12^S e^{-\frac{\epsilon^2 G N_{clust} f_{min} d_{min}(s,a)}{128}} + 4G\left(\frac{1}{4}\right)^{\frac{T_n}{8Gt_{mix}}}$$

Each term is less than $\delta/4$ for $GN_{clust} \geq \Omega\left(\frac{1}{\epsilon^2 f_{min}\alpha}(S + \log(\frac{1}{\delta})\right)$ and $T_n \geq \Omega(Gt_{mix}\log(G/\delta))$. So for such G, N_{clust}, T_n, with probability greater than $1-\delta$,

$$\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_1 < \epsilon$$

Alternatively, for $GN_{clust} \geq \Omega\left(\frac{\log(1/\delta)}{f_{min}^2\alpha^2}\right)$ and $T_n \geq \Omega(Gt_{mix}\log(G/\delta)\log(1/\alpha))$, with probability greater than $1-\delta$,

$$\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_1 \leq O\left(\sqrt{\frac{1}{GN_{clust}f_{min}\alpha}(S + \log(\frac{1}{\delta}))}\right)$$

Letting $G = \left(\frac{T_n}{t_{mix}}\right)^{2/3}$, for $\left(\frac{T_n}{t_{mix}}\right)^{2/3} N_{clust} \geq \Omega\left(\frac{\log(1/\delta)}{f_{min}^2\alpha^2}\right)$ and $T_n \geq \Omega(t_{mix}\log^4(1/\delta)\log^4(1/\alpha))$, with probability greater than $1-\delta$,

$$\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_1 \leq O\left(\left(\frac{t_{mix}}{T_n}\right)^{1/3}\sqrt{\frac{1}{N_{clust}f_{min}\alpha}(S + \log(\frac{1}{\delta}))}\right)$$

□

A.9 Proof of Theorem 3.4.5

We recall the theorem here.

Theorem 3.4.5 (Classification Guarantee)**.** *Let $\epsilon_{mod}(\delta)$ be a high probability bound on the model estimation error $\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_2$. Then there is a universal constant C_3 so that Algorithm 3 can identify the true labels for trajectories in $\mathcal{N}_{class}$ with probability at least $1-\delta$ for $T_n = \Omega\left(K^{3/2} t_{mix} \frac{\log^4(N_{class}/(\alpha\delta))}{\Delta^6 \alpha^3}\right)$, whenever $\epsilon_{mod}(\delta/2) \leq \frac{C_3 \Delta^4 f_{min} \alpha}{K}$ and $N_{clust} \geq \Omega\left(\frac{\log(1/\delta)}{f_{min}^2 \alpha^2}\right)$.*

Proof. The proof is very similar to the proof of theorem 3.4.3. Consider the testing of trajectory n. Recall that in algorithm 3, we defined

$$\text{dist}_1(n,k) := \max_{(s,a)\in SA_\alpha} \left[\left(\left(\hat{\mathbb{P}}_{n,1}(\cdot \mid s,a) - \hat{\mathbb{P}}_k(\cdot \mid s,a) \right)^T \tilde{\mathbf{V}}_{s,a} \right) \left(\left(\hat{\mathbb{P}}_{n,2}(\cdot \mid s,a) - \hat{\mathbb{P}}_k(\cdot \mid s,a) \right)^T \tilde{\mathbf{V}}_{s,a} \right)^T \right]$$

Let k_n the label of trajectory n. According to our assumptions, if $k_n \neq k$, then we have an s,a so that $d_{k_n}(s,a) \geq \alpha$ and $\|\mathbb{P}_{k_n}(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_2 \geq \Delta$. Again, we will make s,a implicit in our notation except in $\mathbb{P}_j(\cdot \mid s,a)$. Let $c_{n,i} := N(n,i,s,a)$, $\mathbf{w}_{n,i} := \mathbf{N}(n,i,s,a,\cdot)$. Recall that we have two nested partitions: (1) of the entire trajectory into the two Ω_i and (2) of each segment Ω_i into G blocks. Finally, define $\text{dist}_{1,(s,a)}$ as below, suppressing n and k. Note that $\text{dist}_1(n,k)$ is the maximum of $\text{dist}_{1,(s,a)}$ over all $(s,a) \in \text{Freq}_\beta$, for the given trajectory n and label k.

$$\text{dist}_{1,(s,a)} := \left[\left((\hat{\mathbb{P}}_{n,1}(\cdot \mid s,a) - \hat{\mathbb{P}}_k(\cdot \mid s,a))^T \tilde{\mathbf{V}}_{s,a} \right) \left((\hat{\mathbb{P}}_{n,2}(\cdot \mid s,a) - \hat{\mathbb{P}}_k(\cdot \mid s,a))^T \tilde{\mathbf{V}}_{s,a} \right)^T \right]$$

We want to show that this is close to $\|\Delta_{n,k}(s,a)\|_2^2$ for the (s,a) pairs that we search over, where

$$\Delta_{n,k}(s,a) = \mathbb{P}_{k_n}(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)$$

Recall that $\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_2 \leq \epsilon_{mod}(\delta)$ for any $1 \leq k \leq K$. Let $\mathbf{M}_{s,a}^{true} = \sum_{1\leq k\leq K} \hat{f}_{k,s,a} \mathbb{P}_k(\cdot \mid s,a) \mathbb{P}_k(\cdot \mid s,a)^T$. We use the fact that $\|aa^T - bb^T\| \leq (\|a\|_2 + \|b\|_2)\|a-b\|_2$ in the bound below.

$$\|\mathbf{M}_{s,a}^{true} - \tilde{\mathbf{M}}_{s,a}\| \leq \sum_{1\leq k\leq K} \hat{f}_{k,s,a} \|\mathbb{P}_k(\cdot \mid s,a)\mathbb{P}_k(\cdot \mid s,a)^T - \hat{\mathbb{P}}_k(\cdot \mid s,a)\hat{\mathbb{P}}_k(\cdot \mid s,a)^T\|$$

$$
\begin{aligned}
&\leq \sum_{1\leq k\leq K} \hat{f}_{k,s,a}(\|\hat{\mathbb{P}}_k(\cdot \mid s,a)\|_2 + \|\mathbb{P}_k(\cdot \mid s,a)\|_2)\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_2 \\
&\leq \sum_{1\leq k\leq K} 2\hat{f}_{k,s,a}\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_2 \\
&\leq 2\epsilon_{mod}(\delta)
\end{aligned}
$$

Also note that if we redefine $\mathcal{B}_n$ to be the event of observing (s,a) in a trajectory (instead of in both segments as in the notation in previous proofs), then $\hat{f}_{k,s,a} = \frac{\sum_n \mathbb{1}_{k_n=k}\mathbb{1}_{\mathcal{B}_n}}{\sum_n \mathbb{1}_{\mathcal{B}_n}} \geq \frac{\sum_n \mathbb{1}_{k_n=k}\mathbb{1}_{\mathcal{B}_n}}{N_{clust}}$. So, $\mathbb{E}[\hat{f}_{k,s,a}] \geq \mathbb{P}(k_n = k \cap \mathcal{B}_n) = \mathbb{P}(\mathcal{B}_n \mid k_n = k)\mathbb{P}(k_n = k) \geq f_{min}\mathbb{P}(\mathcal{B}_n \mid k_n = k)$. Using a computation very similar to the one leading up to inequality A.5, we note that $\mathbb{P}(\mathcal{B}_n \mid k_n = k) \geq d_{min}(s,a)/2$ for $T_n \geq \Omega(t_{mix}\log(1/\alpha))$. In that case, $\mathbb{E}[\hat{f}_{k,s,a}] \geq f_{min}d_{min}(s,a)/2 \geq f_{min}\alpha/2$. Additionally, using a standard concentration argument, $\hat{f}_{k,s,a} \geq \mathbb{E}[\hat{f}_{k,s,a}]/2 \geq f_{min}\alpha/4$ for $N_{clust} \geq \Omega\left(\frac{\log(1/\delta)}{f_{min}^2\alpha^2}\right) \geq \Omega\left(\frac{\log(1/\delta)}{\mathbb{E}[\hat{f}_{k,s,a}]^2}\right)$

We now apply Lemma 3 of Chen and Poor [2022], with $p^{(k)} = \hat{f}_{k,s,a}$, $\mathbf{y}^{(k)} = \mathbb{P}_k(\cdot \mid s,a)$, $\mathbf{M} = \mathbf{M}_{s,a}^{true}$ and $\mathbf{M}_* = \mathbf{M}_{s,a}$. We use the right-hand side of the bound in the lemma to get the bound below for all $1 \leq k \leq K$, which holds for a universal constant C_2 with probability at least $1-\delta$ whenever $N_{clust} \geq \Omega\left(\frac{\log(1/\delta)}{f_{min}^2\alpha^2}\right)$ and $T_n \geq \Omega(t_{mix}\log(1/\alpha))$.

$$
\|\mathbb{P}_k(\cdot \mid s,a) - \tilde{\mathbf{V}}_{s,a}\tilde{\mathbf{V}}_{s,a}^T\mathbb{P}_k(\cdot \mid s,a)\|_2 \leq \sqrt{\frac{2K\epsilon_{mod}(\delta)}{\hat{f}_{k,s,a}}} \leq C_2\sqrt{\frac{K\epsilon_{mod}(\delta)}{f_{min}\alpha}} \tag{A.19}
$$

Assume the lemma below for now, we prove it in the next subsection.

Lemma A.9.1. *We claim that there is a universal constant C_1 so that for any (s,a) with $d_{min}(s,a) \geq \alpha/3$, with probability at least $1-\delta$,*

$$
\left|\text{dist}_{1,(s,a)} - \|\Delta_{n,k}(s,a)\|_2^2\right| \leq O\left(\sqrt{\frac{K+\log(1/\delta)}{G\alpha}}\right) + 8C_2\sqrt{\frac{K\epsilon_{mod}(\delta/2)}{f_{min}\alpha}}
$$

whenever $T_n \geq \Omega\left(Gt_{mix}\log(G/\delta)\log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$. Here, $\epsilon_{mod}(\delta)$ is a high probability bound on $\|\mathbb{P}_k(\cdot \mid s,a) - \tilde{\boldsymbol{V}}_{s,a}\tilde{\boldsymbol{V}}_{s,a}^T\mathbb{P}_k(\cdot \mid s,a)\|_2$ for all $1 \leq k \leq K$ (which holds with probability at least $1-\delta$).

We now set $G = \left(\frac{T_n}{t_{mix}}\right)^{\frac{2}{3}}$. Then a sufficient condition on T_n to meet the conditions of the lemma is $T_n = \Omega(t_{mix}\log^4(1/\delta)/\alpha^3)$, under which, with probability at lest $1-\delta$, we have the following bound for (s,a) with $d_{min}(s,a) \geq \alpha/3$.

$$\left|\text{dist}_{1,(s,a)} - \|\Delta_{n,k}(s,a)\|_2^2\right| \leq O\left(\sqrt{\frac{K\log(1/\delta)}{\alpha}}\left(\frac{t_{mix}}{T_n}\right)^{\frac{1}{3}}\right) + 8C_2\sqrt{\frac{K\epsilon_{mod}(\delta/2)}{f_{min}\alpha}} \tag{A.20}$$

It is now easy to see that the first term on the right-hand side is less than $\Delta^2/8$ when $T_n = \Omega\left(K^{3/2}t_{mix}\frac{\log^{3/2}(1/\delta)}{\Delta^6\alpha^{3/2}}\right)$ and $T_n = \Omega(t_{mix}\log^4(1/\delta)/\alpha^3)$. We can combine these to have the guarantee that the first term on the right-hand side is less $\Delta^2/8$ with probability at least $1-\delta$ when $T_n = \Omega\left(K^{3/2}t_{mix}\frac{\log^4(1/\delta)}{\Delta^6\alpha^3}\right)$.

Now note that if $\beta \geq \alpha/3$, then a separating state action pair always lies in Freq_β and thus, the maximum over the $\|\Delta_{n,k}(s,a)\|_2^2$ values corresponding to Freq_β is in fact either 0 if $k = k_n$ or larger than Δ^2 if $k \neq k_n$. So, if $8C_2\sqrt{\frac{K\epsilon_{mod}(\delta/2)}{f_{min}\alpha}} \leq \Delta^2/32$ and for each of the (s,a) pairs, the first term on the right-hand side of inequality A.20 is less than $\Delta^2/8$, then our distance estimate $\text{dist}_1(n,k)$ is on the right side of $\Delta^2/3$. That is, the distance estimate is then less than $\Delta^2/4$ if $k = k_n$, and larger than it if $k \neq k_n$. As a consequence, the output of the $\arg\min$ in algorithm 3 is k_n in this situation.

Note that upon choosing an occurrence threshold of order α, we will have at most $O(1/\alpha)$ many (s,a) pairs in Freq_β to maximize $\text{dist}_{1,(s,a)}$ over to get $\text{dist}_1(n,k)$. By applying a union bound over all (s,a) pairs in Freq_β and using the conclusion of the previous paragraph, algorithm 3 correctly predicts the label k_n for trajectory n with probability $1-\delta$ whenever $T_n = \Omega\left(K^{3/2}t_{mix}\frac{\log^4(1/(\alpha\delta))}{\Delta^6\alpha^3}\right)$ and $8C_2\sqrt{\frac{K\epsilon_{mod}(\delta/2)}{f_{min}\alpha}} \leq \Delta^2/32$.

By applying a union bound over incorrectly predicting k_n for any of the $N_{class}(N_{class}-1)/2$ pairs, we get that algorithm 3 can recover the true labels with probability at least $1-\delta$ for $T_n = \Omega\left(K^{3/2}t_{mix}\frac{\log^4(N_{class}/(\alpha\delta))}{\Delta^6\alpha^3}\right)$, whenever $8C_2\sqrt{\frac{K\epsilon_{mod}(\delta/2)}{f_{min}\alpha}} \leq \Delta^2/32$.

Finally note that due to inequality A.19, we get that algorithm 3 can recover the true labels with probability at least $1-\delta$ for $T_n = \Omega\left(K^{3/2}t_{mix}\frac{\log^4(N_{class}/(\alpha\delta))}{\Delta^6\alpha^3}\right)$, whenever $\epsilon_{mod}(\delta/2) \leq \frac{C_3\Delta^4 f_{min}\alpha}{K}$.

□

A.9.1 Proof of Lemma A.9.1

We recall the lemma here.

Lemma A.9.1. *We claim that there is a universal constant C_1 so that for any (s,a) with $d_{min}(s,a) \geq$*

$\alpha/3$*, with probability at least* $1-\delta$*,*

$$\left|\mathrm{dist}_{1,(s,a)} - \|\Delta_{n,k}(s,a)\|_2^2\right| \leq O\left(\sqrt{\frac{K+\log(1/\delta)}{G\alpha}}\right) + 8C_2\sqrt{\frac{K\epsilon_{mod}(\delta/2)}{f_{min}\alpha}}$$

whenever $T_n \geq \Omega\left(Gt_{mix}\log(G/\delta)\log(1/\alpha)\right)$ *and* $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$*. Here,* $\epsilon_{mod}(\delta)$ *is a high probability bound on* $\|\mathbb{P}_k(\cdot \mid s,a) - \tilde{\boldsymbol{V}}_{s,a}\tilde{\boldsymbol{V}}_{s,a}^T\mathbb{P}_k(\cdot \mid s,a)\|_2$ *for all* $1 \leq k \leq K$ *(which holds with probability at least* $1-\delta$*).*

Proof. The proof of this lemma is very similar to the proof of Lemma A.6.1.

Notation: We say $c_{n,i} = N(n,i,s,a)$ as in the statement of the lemma and $\mathbf{w}_{n,i} = \mathbf{N}(n,i,s,a,\cdot)$. Let the joint distribution of the observations over trajectory n be χ. Let its marginals on the segments Ω_i be χ_i. Let the marginals on each of the G single-step sub-blocks along with their next states be $\chi_{i,g}$. Denote the product distribution $\prod_g \chi_{i,g}$ by Q_i. Let $\mathcal{G}(n,s,a)$ denote the set of indices where the state-action pair (s,a) is observed in trajectory n. For brevity, we will abbreviate $\mathcal{G}(n,s,a)$ to $\mathcal{G}$. Note that the size of this set is exactly $c_{n,i}$.

We first prove a preliminary lemma, similar to lemma A.6.2.

A.9.1.1 Decomposition of $|\,\mathrm{dist}_{1,(s,a)} - \|\Delta_{n,k}(s,a)\|_2^2\,|$

Lemma A.9.2. *We claim that for each fixed value of* $\mathcal{G}(s,a)$ *(abbreviated to* $\mathcal{G}$*), with probability at least* $1-\delta$*, the following bound holds.*

$$\left|\mathrm{dist}_{1,(s,a)} - \|\Delta_{n,k}(s,a)\|_2^2\right| \leq \sum_{i=1}^{2} 2\left\|\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}]\right\|_2 + 8C_2\sqrt{\frac{K\epsilon_{mod}(\delta)}{f_{min}\alpha}} + 4\left(\max_i \mathbb{1}_{c_{n,i}=0}\right) \quad \text{(A.21)}$$

Here $c_{n,i} = N(n,i,s,a)$ *and* $\epsilon_{mod}(\delta)$ *is a high probability bound on* $\|\mathbb{P}_k(\cdot \mid s,a) - \tilde{\boldsymbol{V}}_{s,a}\tilde{\boldsymbol{V}}_{s,a}^T\mathbb{P}_k(\cdot \mid s,a)\|_2$ *(satisfied with probability* $> 1-\delta$*).*

Remark 17. In the inequality,

- The first term is a concentration-type term, which will be broken into an "independent concentration" error and a mixing error to account for the low but non-zero dependence across blocks.
- The second term accounts for subspace estimation error.

- The third term accounts for actually observing s, a in our blocks.

Proof. Define the following quantities.

$$\begin{aligned}\boldsymbol{\Delta}_i^T &= (\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \hat{\mathbb{P}}_k(\cdot \mid s,a))^T \tilde{\mathbf{V}}_{s,a} \\ \bar{\boldsymbol{\Delta}}_i^T &= (\mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}] - \mathbb{P}_k(\cdot \mid s,a))^T \tilde{\mathbf{V}}_{s,a}\end{aligned}$$

We first establish a simple inequality, using the fact that $|a^T b - c^T d| \leq \|b\|_2\|a-c\|_2 + \|c\|_2\|b-d\|_2$

$$\begin{aligned}|\operatorname{dist}_{1,(s,a)} - \bar{\boldsymbol{\Delta}}_1^T \bar{\boldsymbol{\Delta}}_2| &= |\boldsymbol{\Delta}_1^T \boldsymbol{\Delta}_2 - \bar{\boldsymbol{\Delta}}_1^T \bar{\boldsymbol{\Delta}}_2| \\ &\leq \|\boldsymbol{\Delta}_1 - \bar{\boldsymbol{\Delta}}_1\|_2 \|\boldsymbol{\Delta}_2\|_2 + \|\bar{\boldsymbol{\Delta}}_1^T\|_2 \|\boldsymbol{\Delta}_2 - \bar{\boldsymbol{\Delta}}_2\|_2 \\ &\leq 2\|\boldsymbol{\Delta}_1 - \bar{\boldsymbol{\Delta}}_1\|_2 + 2\|\boldsymbol{\Delta}_2 - \bar{\boldsymbol{\Delta}}_2\|_2 \\ &\leq \sum_{i=1}^{2} 2\|\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}]\|_2 \\ &\quad + \sum_{i=1}^{2} 2\|\hat{\mathbb{P}}_k(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)\|_2 \\ &\leq \sum_{i=1}^{2} 2\|\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}]\|_2 \\ &\quad + 4\epsilon_{mod}(\delta) \qquad \text{(A.22)}\end{aligned}$$

Also note the following computation.

$$\begin{aligned}\mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}] &= \mathbb{E}_{Q_i}\left[\frac{\mathbf{w}_{n,i}}{c_{n,i}} \mathbb{1}_{c_{n,i} \neq 0} \mid \mathcal{G}\right] \\ &= \frac{\mathbb{E}_{Q_i}[\mathbf{w}_{n,i} \mid \mathcal{G}]}{c_{n,i}} \mathbb{1}_{c_{n,i} \neq 0} \\ &= \frac{\mathbb{P}_{k_n}(\cdot \mid s,a) c_{n,i}}{c_{n,i}} \mathbb{1}_{c_{n,i} \neq 0} \\ &= \mathbb{1}_{c_{n,i} \neq 0} \mathbb{P}_{k_n}(\cdot \mid s,a)\end{aligned}$$

We define the following quantity, overloading notation from Lemma A.6.2.

$$\mathbf{diff}_i = \mathbb{1}_{c_{n,i} \neq 0} \mathbb{P}_{k_n}(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)$$

Note that $\bar{\boldsymbol{\Delta}}_i = \mathbf{diff}_i^T \tilde{\mathbf{V}}_{s,a}$. We recall the following definition before proceeding to show the main

inequality.

$$\Delta_{n,k}(s,a) = \mathbb{P}_{k_n}(\cdot \mid s,a) - \mathbb{P}_k(\cdot \mid s,a)$$

$$\begin{aligned}
&\left|\bar{\mathbf{\Delta}}_1^T \bar{\mathbf{\Delta}}_2 - \|\Delta_{n,k}(s,a)\|_2^2\right| \\
&= \left|\mathbf{diff}_1^T \tilde{\mathbf{V}}_{s,a} \tilde{\mathbf{V}}_{s,a}^T \mathbf{diff}_2 - \mathbf{diff}_1^T \mathbf{diff}_2\right| + \left|\mathbf{diff}_1^T \mathbf{diff}_2 - \|\Delta_{n,k}(s,a)\|_2^2\right| \\
&\leq \|\mathbf{diff}_1\|_2 \left\|\mathbf{diff}_2 - \tilde{\mathbf{V}}_{s,a} \tilde{\mathbf{V}}_{s,a}^T \mathbf{diff}_2\right\|_2 + \|\mathbf{diff}_1 - \Delta_{n,k}(s,a)\|_2 \|\mathbf{diff}_2\|_2 \\
&\quad + \|\mathbf{diff}_1\|_2 \|\mathbf{diff}_2 - \Delta_{n,k}(s,a)\|_2 \\
&\leq \|\mathbf{diff}_1\|_1 \left\|\mathbf{diff}_2 - \tilde{\mathbf{V}}_{s,a} \tilde{\mathbf{V}}_{s,a}^T \mathbf{diff}_2\right\|_2 + \|\mathbf{diff}_1 - \Delta_{n,k}(s,a)\|_2 \|\mathbf{diff}_2\|_1 \\
&\quad + \|\mathbf{diff}_1\|_1 \|\mathbf{diff}_2 - \Delta_{n,k}(s,a)\|_2 \\
&\leq 2 \left\|\mathbf{diff}_2 - \tilde{\mathbf{V}}_{s,a} \tilde{\mathbf{V}}_{s,a}^T \mathbf{diff}_2\right\|_2 + 2 \|\mathbf{diff}_1 - \Delta_{n,k}(s,a)\|_2 + 2 \|\mathbf{diff}_2 - \Delta_{n,k}(s,a)\|_2 \\
&\leq 2 \left\|\mathbb{P}_{k_n}(\cdot \mid s,a) - \tilde{\mathbf{V}}_{s,a} \tilde{\mathbf{V}}_{s,a}^T \mathbb{P}_{k_n}(\cdot \mid s,a)\right\|_2 \\
&\quad + 2 \left\|\mathbb{P}_k(\cdot \mid s,a) - \tilde{\mathbf{V}}_{s,a} \tilde{\mathbf{V}}_{s,a}^T \mathbb{P}_k(\cdot \mid s,a)\right\|_2 \\
&\quad + 2\mathbb{1}_{c_{n,1}=0} \|\mathbb{P}_{k_n}(\cdot \mid s,a)\|_2 + 2\mathbb{1}_{c_{n,2}=0} \|\mathbb{P}_{k_n}(\cdot \mid s,a)\|_2 \\
&\leq 4C_2 \sqrt{\frac{K\epsilon_{mod}(\delta)}{f_{min}\alpha}} + 4\left(\max_i \mathbb{1}_{c_{n,i}=0}\right)
\end{aligned}$$

Notice that $4C_2\sqrt{\frac{K\epsilon_{mod}(\delta)}{f_{min}\alpha}} \geq 4\epsilon_{mod}(\delta)$ since $\epsilon_{mod}(\delta) \leq 2$, $C_2 \geq 2$, $K \geq 1$, $f_{min}, \alpha \leq 1$. Combining this and the computation above with inequality A.16, we have the following final bound.

$$\begin{aligned}
\left|\mathrm{dist}_{1,(s,a)} - \|\Delta_{n,k}(s,a)\|_2^2\right| \leq \sum_{i=1}^{2} 2 \left\|\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}]\right\|_2 \\
+ 8C_2 \sqrt{\frac{K\epsilon_{mod}(\delta)}{f_{min}\alpha}} + 4\left(\max_i \mathbb{1}_{c_{n,i}=0}\right) \quad \text{(A.23)}
\end{aligned}$$

where we remind the reader that $c_{n,i} = N(n,i,s,a)$. □

A.9.1.2 Bounding the concentration-type term

We bound the first term in the decomposition lemma (Lemma A.9.2) with high probability.

Lemma A.9.3. *With probability at least* $1-\delta$*, when* $T_n \geq \Omega\left(Gt_{mix}\log\left(\frac{G}{\delta}\log(1/\alpha)\right)\right)$ *and*

$G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$, *we have the following bound.*

$$\left\|\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) - \mathbb{E}_{Q_i}[\hat{\mathbb{P}}_{n,i}(\cdot \mid s,a) \mid \mathcal{G}]\right\|_2 \leq O\left(\sqrt{\frac{K+\log(1/\delta)}{G\alpha}}\right)$$

Proof. The proof of this lemma is verbatim the proof of Lemma A.6.3 after the first inequality. □

A.9.1.3 Combining the bounds

We reuse Corollary A.6.5 along with Lemma A.9.3 applied to Lemma A.9.2 to get the following bound with probability at least $1-\delta$,

$$\left|\text{dist}_{1,(s,a)} - \|\Delta_{n,k}(s,a)\|_2^2\right| \leq O\left(\sqrt{\frac{K+\log(1/\delta)}{G\alpha}}\right) + 8C_2\sqrt{\frac{K\epsilon_{mod}(\delta)}{f_{min}\alpha}}$$

whenever $T_n \geq \Omega\left(Gt_{mix}\log(G/\delta)\log(1/\alpha)\right)$ and $G \geq \Omega\left(\frac{\log(1/\delta)}{\alpha^2}\right)$.

□

APPENDIX B

Supplementary Material for Chapter 3

This appendix contains proofs and supplementary material for Chapter 4.

B.1 Experimental Details

Computing Infrastructure. All numerical experiments were run on a single desktop computer with an Intel i9-13900K CPU, 128 gigabytes of RAM, and an NVIDIA RTX 3090 graphics card.

Estimating Policy Values for Global Confounders. Due to computationally expensive operations needed to compute the exact policy value for confounders, we use estimates of the policy values instead. Namely, we get estimates $\hat{V}_1(s_0, u, \pi)$ for a policy π, and report $\sum_u P(u)\hat{V}_1(s_0, u, \pi)$. Computing the true values $V_1(s_0, u, \pi)$ is computationally far more expensive. The estimates $\hat{V}_1(s_0, u, \pi)$ are obtained using standard FQE applied to the standard, unconfounded MDP determined by confounder u.

B.2 Lower Bounds for Memoryless Confounders

We recall and prove Theorem 4.2.1.

Theorem 4.2.1 (Lower Bound for Memoryless Confounders)**.** *There exists a parameter ε that determines a pair of confounded MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ with i.i.d. (and thus memoryless) confounders along with stationary policies π_{b_1}, π_{b_2} and π_e, so that data collected from $\mathcal{M}_i$ using π_{b_i} has the same distribution for $i = 1, 2$, but the values under π_e differ by $|V_1^{\pi_e}(\mathcal{M}_1) - V_1^{\pi_e}(\mathcal{M}_2)| = 2\varepsilon H$. In particular, when $\varepsilon = \frac{1}{2} - \frac{1}{H^2}$, the values under π_e differ by $\Omega(H)$.*

Proof. Consider two confounded MDP environments $\mathcal{M}_1$ and $\mathcal{M}_2$.

Environments. In both environments:

- $\mathcal{S} = \{1, 2\}, \mathcal{U} = \{1, 2\}, \mathcal{A} = \{1, 2\}$, horizon H.

- $r(s=1)=1, r(s=2)=0.$

For confounders:

- $P_1(u=1)=\frac{1}{2}-\varepsilon, P_1(u=2)=\frac{1}{2}+\varepsilon.$
- $P_2(u=1)=\frac{1}{2}+\varepsilon, P_2(u=2)=\frac{1}{2}-\varepsilon.$

For full state transitions:

$$\begin{aligned}
&\mathbb{P}_1(s'=1 \mid s, u=1, a=1)=z, \mathbb{P}_1(s'=1 \mid s, u=2, a=1)=1-z \\
&\mathbb{P}_1(s'=1 \mid s, u=1, a=2)=z_1, \mathbb{P}_1(s'=1 \mid s, u=2, a=2)=z_2 \\
&\mathbb{P}_2(s'=1 \mid s, u=1, a=1)=z, \mathbb{P}_2(s'=1 \mid s, u=2, a=1)=1-z \\
&\mathbb{P}_2(s'=1 \mid s, u=1, a=2)=z_2, \mathbb{P}_2(s'=1 \mid s, u=2, a=2)=z_1
\end{aligned}$$

Next, consider two behavior policies π_{b_1} and π_{b_2}:

$$\begin{aligned}
\pi_{b_1}(a=1 \mid s, u=1)&=\frac{1}{2}+\varepsilon, \;\; \pi_{b_1}(a=1 \mid s, u=2)=\frac{1}{2}-\varepsilon \\
\pi_{b_2}(a=1 \mid s, u=1)&=\frac{1}{2}-\varepsilon, \;\; \pi_{b_2}(a=1 \mid s, u=2)=\frac{1}{2}+\varepsilon
\end{aligned}$$

And an evaluation policy π_e:

$$\pi_e(s)=1, \;\; \text{for } s=\{1,2\}.$$

Data Collection. Suppose we collect data using π_{b_1} in $\mathcal{M}_1$ and using π_{b_2} in $\mathcal{M}_2$. Notice that the sensitivity Γ is given by

$$\Gamma=\left(\frac{\frac{1}{2}+\varepsilon}{\frac{1}{2}-\varepsilon}\right)\left(\frac{\frac{1}{2}+\varepsilon^2}{\frac{1}{2}-\varepsilon^2}\right)$$

Observations. Note that in the limit, i.e. infinite data, the observed transition probabilities and policies are given by

$$\begin{aligned}
\hat{\mathbb{P}}_1(s', a \mid s)&=\sum_u P_1(u)\pi_{b_1}(a \mid s, u)\mathbb{P}_1(s' \mid s, u, a), \\
\pi_1(a \mid s)&=\sum_u P(u)\pi_1(a \mid s, u), \\
\hat{\mathbb{P}}_1(s' \mid s, a)&=\hat{\mathbb{P}}_1(s', a \mid s)/\pi_1(a \mid s).
\end{aligned}$$

One can then easily verify that for all s, a, s', the observed transition probabilities will be equal:

$$\hat{\mathbb{P}}_1(s', a \mid s) = \hat{\mathbb{P}}_2(s', a \mid s),$$

For example, $\hat{\mathbb{P}}_i(s' = 1, a = 1 \mid s) = x(1 - x)$ for $i = 1, 2$.

The state transition and the observed policy induced by the two policies in their corresponding environment are thus also equal:

$$\pi_1(a \mid s) = \pi_2(a \mid s),$$
$$\hat{\mathbb{P}}_1(s' \mid s, a) = \hat{\mathbb{P}}_2(s' \mid s, a).$$

That means, no algorithm can distinguish the two environments based on the given two datasets.

Value under the evaluation policy. Recall that at each step, we take action 1. Note that the true marginalized state transitions will be different, which are what a confounder-oblivious policy will interact with:

$$\mathbb{P}_1(s' = 1 \mid s, a = 1) = \sum_u P_1(u)\mathbb{P}_1(s' = 1 \mid s, u, a = 1) = \left(\frac{1}{2} + \varepsilon\right)(1 - z) + \left(\frac{1}{2} - \varepsilon\right) z$$
$$\mathbb{P}_2(s' = 1 \mid s, a = 1) = \sum_u P_2(u)\mathbb{P}_2(s' = 1 \mid s, u, a = 1) = \left(\frac{1}{2} - \varepsilon\right)(1 - z) + \left(\frac{1}{2} + \varepsilon\right) z$$

Note that $\mathbb{P}_i^{\pi_e}(s' = 1 \mid s) = \mathbb{P}_i(s' = 1 \mid s, a = 1)$. Since state transitions are independent of the initial state, this is the same as generating a state independently at each step based on the action taken. Then under the evaluation policy $\pi_e(a = 1 \mid s) = 1$, the state $s = 1$ is generated i.i.d. at each step with probability $p_i = \mathbb{P}_i(s' = 1 \mid s, a = 1)$ in $\mathcal{M}_i$, while $s = 2$ is generated with probability $1 - p_i$. So, the reward of a trajectory is distributed according to $Bin(H, p_i)$, having an expected value of $V_1^{\pi_e}(\mathcal{M}_i) = Hp_i = H\mathbb{P}_i(s' = 1 \mid s, a = 1)$.

Necessity of a Sensitivity Assumption Let $\varepsilon = \frac{1}{2} - \frac{1}{H^2}$, $z = 0$. We then have the following

$$V_1^{\pi_e}(\mathcal{M}_1) = H((1 - \frac{1}{H^2})^2 + 1/H^4) = O(H)$$
$$V_1^{\pi_e}(\mathcal{M}_2) = 2H \cdot \frac{1}{H^2}(1 - \frac{1}{H^2}) = O(\frac{1}{H}).$$

From this example, we see that without information about Γ, no algorithm can universally give meaningful lower bounds for the true value function. One can compute that in this example,

$\Gamma = \Theta(H^2)$.

Lower Bound on Value Estimation Under Sensitivity Let ε be small and let $z = 0$. We then have the following.

$$V_1^{\pi_e}(\mathcal{M}_1) = H(\frac{1}{2} - \varepsilon)$$
$$V_1^{\pi_e}(\mathcal{M}_2) = H(\frac{1}{2} + \varepsilon)$$

Note that $\Gamma = 1 + O(\varepsilon + \varepsilon^2) = 1 + O(\varepsilon)$ for small ε. Since any estimator will return the same value for both MDPs (because they are observationally indistinguishable under the behavior policy), any estimator will have a worst-case error of at least εH. Thus, there does not exist a consistent estimator whenever $\Gamma > 1$.

□

B.3 FQE and Confounded FQE

We describe the FQE and CFQE algorithms here, adapted for memoryless systems instead of merely stationary ones.

B.3.1 FQE Algorithm

Algorithm 15 FQE

input: evaluation policy π_e.
initialize: $\hat{f}_{H+1} \leftarrow 0$.
for $h = H, H-1, \ldots, 1$ **do**
    $\hat{f}_h(s,a) \leftarrow \mathbb{E}_{(s,a,s') \sim \mathcal{D}_{\pi_b}, h} \left[r_h(s,a) + \sum_{a'} \pi_{e,(h+1)}(a' \mid s') \hat{f}_{h+1}(s', a') \right], \forall s, a.$
end for
return: $\sum_a \pi_{e,1}(a \mid s) \hat{f}_1(s,a)$ for $\forall s$.

B.3.2 Confounded FQE Algorithm

Confounded FQE (CFQE), proposed by Bruns-Smith [2021], provides an estimate for a lower bound by taking the characteristics of the data into account. Given infinite samples, this will actually be a lower bound, unlike the case of FQE. In particular, CFQE obtains an estimate for a lower bound by sequentially searching over the worst behavior policy consistent with the observations.

Let $\hat{\pi}_{b,h}(a \mid s)$ and $\hat{\mathbb{P}}_h(s' \mid s, a)$ be empirical estimates from finite data $\mathcal{D}_{\pi_b,h}$. Let $\mathbb{P}_h^{\pi_b}(s' \mid s, a)$ be the limit of $\hat{\mathbb{P}}_h(s' \mid s, a)$ under infinite data. We then define the following uncertainty sets.

Definition B.3.1 (Valid Behavior Policy Set)**.** Under a memoryless confounder, for all s, a, s', define $\mathcal{B}_{sa,h}$ to be the set of all $\pi(a \mid s, \cdot)$ that satisfy Assumption 5 and the two equations below.

$$\sum_{u \in \mathcal{U}} P_h(u \mid s)\pi_{b,h}(a \mid s, u) = \pi_{b,h}(a \mid s)$$
$$\sum_{u \in \mathcal{U}} P_h(u \mid s)\pi_{b,h}(a \mid s, u)P(s' \mid s, u, a) = \pi_{b,h}(a \mid s)\mathbb{P}_h^{\pi_b}(s' \mid s, a).$$

Now we define the following quantity using the posteriors $P_h^{\pi_b}(u \mid s, a)$, a confounded analog to inverse propensity weights.

$$\begin{aligned} g_h(s, a, s') &:= \sum_u \left(\frac{P_h^{\pi_b}(u \mid s, a)\mathbb{P}_h(s' \mid s, a, u)}{\hat{\mathbb{P}}_h^{\pi_b}(s' \mid s, a)} \right) \frac{1}{\pi_{b,h}(a \mid s, u)} \\ &= \sum_u \left(\frac{P_h(u \mid s)\mathbb{P}_h(s' \mid s, a, u)}{\hat{\mathbb{P}}_h^{\pi_b}(s' \mid s, a)} \right) \frac{1}{\pi_{b,h}(a \mid s)} \end{aligned}$$

Theorem 1 and the discussion following that in Bruns-Smith [2021] shows that we can reflect the same uncertainty using the set $\tilde{\mathcal{B}}_{sa,h}$ of possible values of $g_h(s, a, \cdot)$.

$$\begin{aligned} \tilde{\mathcal{B}}_{sa,h} := \{ g_h(s, a, \cdot) \;\mid\; & \alpha_h(s, a) \leq \pi_{b,h}(a \mid s)g_h(s, a, s') \leq \beta_h(s, a), \\ & \sum_{s'} \pi_{b,h}(a \mid s)g_h(s, a, s')\mathbb{P}_h^{\pi_b}(s' \mid s, a) = 1 \} \end{aligned} \tag{B.1}$$

$\tilde{\mathcal{B}}_{sa,h}$ presents a reparameterization of the uncertainty that allows us to get rid of the explicit presence of the unknown variable u while optimizing over the uncertainty set. Let $\hat{\mathcal{B}}_{sa,h}$ and $\hat{\tilde{\mathcal{B}}}_{sa,h}$ be the version of these sets determined by the point estimates $\hat{\pi}_b$ and $\hat{\mathbb{P}}(s' \mid s, a)$ under finite data, instead of by their true values.

However, if a very poor estimate of $\hat{\pi}_b$ and $\hat{\mathbb{P}}_{\pi_b}(s' \mid s, a)$ is collected (due to low $N(s, a)$ and/or $N(s)$), the estimated lower bound will be a lower bound on the output of FQE but not on the true value. To get a lower bound on the true value with probability at least $1 - \delta$, we modify $\hat{\tilde{B}}_{sa,h}$ using error bounds $\text{err}_\pi(N(s))$ and $\text{err}_\mathbb{P}(N(s, a))$ obtained using the Hoeffding inequality to get the following set.

Algorithm 16 Confounded FQE (adapted from Bruns-Smith [2021])

input: evaluation policy π_e.
initialize: $\hat{f}_{H+1} \leftarrow 0$.
for $h = H, H-1, \ldots, 1$ **do**
    Compute

$$\hat{f}_h(s,a) := \min_{g_h(s,a,\cdot) \in \hat{\tilde{\mathcal{B}}}_{sa,h}} \mathbb{E}_{(s,a,s') \sim \mathcal{D}_{\pi_b,h}} \Big[\hat{\pi}_{b,h}(a \mid s) g_h(s,a,s') \left(r_h(s,a) + \sum_{a'} \pi_{e,h}(a' \mid s') \hat{f}_{h+1}(s',a') \right) \Big]$$

end for
return: $\sum_a \pi_e(a \mid s) \hat{f}_1(s,a)$ for $\forall s$.

$$\begin{aligned} \{g_h(s,a,\cdot) \mid\ & \alpha_h(s,a) \le \pi_{b,h}(a \mid s) g_h(s,a,s') \le \beta_h(s,a), \\ & \sum_{s'} \pi_{b,h}(a \mid s) g_h(s,a,s') \mathbb{P}_h^{\pi_b}(s' \mid s,a) = 1 \\ & |\pi_{b,h}(s,a) - \hat{\pi}_{b,h}(s,a)| \le \text{err}_\pi(N(s)), \\ & |\mathbb{P}_h^{\pi_b}(s' \mid s,a) - \hat{\mathbb{P}}_h(s' \mid s,a)| \le \text{err}_{\mathbb{P}}(N(s,a))\} \end{aligned}$$

Additionally, the observant reader will note that CFQE finds a different optimal g_h for each time step. That is, it finds H different functions $g_1(s,a,\cdot), ..., g_H(s,a,\cdot) \in \tilde{\mathcal{B}}_{sa}$. If the transition structures were stationary, this does not leverage the stationarity. In that case, it is advisable to use our model-based method and its projected gradient descent version, as discussed in Section 4.2.3.

B.4 FQE and CFQE Theoretical Results

B.4.1 Proof of FQE Error Bounds, Theorem 4.2.2

We recall the theorem below.

Theorem 4.2.2 (FQE Error)**.** *Suppose $\Gamma = 1 + \varepsilon$ in Assumption 5. Then in the limit of infinite samples, the point estimate $\hat{f}_1(s,a)$ of the Q-function produced by FQE has a worst-case error of $|V_1^{\pi_e}(s) - \sum_a \pi_{e,1}(a \mid s) \hat{f}_1(s,a)| = O(\varepsilon H^2)$ for small ε.*

Proof. In the limit of an infinite amount of data, at every step of FQE, the update evaluates $\hat{f}_h(s,a)$

using:

$$\begin{aligned}
\hat{f}_h(s,a) &= \arg\min_{f_h(s,a)} \mathbb{E}_{(s,a,s')\sim\mathcal{D}^h_{\pi_b}}[\text{loss}_{FQE}(f_h(s,a),s')] \\
&= \arg\min_{f_h(s,a)} \sum_{u,s'} \mathbb{P}^{\pi_b}(s',u \mid s,a)\text{loss}_{FQE}(f_h(s,a),s') \\
&= \arg\min_{f_h(s,a)} \sum_{u,s'} P_h^{\pi_b}(u \mid s,a)\mathbb{P}_h(s' \mid s,u,a)\text{loss}_{FQE}(f_h(s,a),s') \\
&= \arg\min_{f_h(s,a)} \sum_{u,s'} P_h(u \mid s)\frac{\pi_{b,h}(a \mid s,u)}{\pi_{b,h}(a \mid s)} \sum_{s'} \mathbb{P}_h(s' \mid s,u,a)\text{loss}_{FQE}(f_h(s,a),s')
\end{aligned}$$

where $P_h^{\pi_b}(u \mid s,a)$ is the posterior on u under π_b and

$$\text{loss}_{FQE}(f_h(s,a),s') = \left(f_h(s,a) - r(s,a) - \sum_{a'} \pi_{e,h+1}(a' \mid s')\hat{f}_{h+1}(s',a')\right)^2$$

$\hat{f}_h(s,a)$ is then given by the following expression.

$$\begin{aligned}
&\sum_{u,s'} P_h(u \mid s)\frac{\pi_{b,h}(a \mid s,u)}{\pi_{b,h}(a \mid s)}\mathbb{P}_h(s' \mid s,u,a)\left(r(s,a) + \sum_{a'} \pi_{e,h+1}(a' \mid s')\hat{f}_{h+1}(s',a')\right) \\
&= r(s,a) + \sum_{u,s'} P_h(u \mid s)\frac{\pi_{b,h}(a \mid s,u)}{\pi_{b,h}(a \mid s)}\mathbb{P}_h(s' \mid s,u,a)\sum_{a'} \pi_{e,h+1}(a' \mid s')\hat{f}_{h+1}(s',a')
\end{aligned}$$

True marginalized transition structure. Note that under any confounding-unaware policy π_e, the induced transition structure $\mathbb{P}_h^{\pi_e}(s' \mid s)$ is determined by the marginalized transition dynamics $\mathbb{P}_h(s' \mid s,a) := \sum_u P_h(u \mid s)\mathbb{P}_h(s' \mid s,a,u)$. This is clear from the computation below.

$$\begin{aligned}
\mathbb{P}_h^{\pi_e}(s' \mid s) &= \sum_{u,a} \pi_{e,h}(a \mid s)P_h(u \mid s)\mathbb{P}_h(s' \mid s,a,u) \\
&= \sum_a \pi_{e,h}(a \mid s)\left(\sum_u P_h(u \mid s)\mathbb{P}_h(s' \mid s,a,u)\right) \quad = \sum_a \pi_{e,h}(a \mid s)\mathbb{P}_h(s' \mid s,a)
\end{aligned}$$

Bounding $\hat{f}_h(s,a)$. By Assumption 5 and the computations above, we can bound $\hat{f}_h(s,a)$ by:

$$\hat{f}_h(s,a) \le r(s,a) + \frac{1}{\alpha_h(s,a)}\sum_{s'} \mathbb{P}_h(s' \mid s,a)\sum_{a'} \pi_{e,h+1}(a' \mid s')\hat{f}_{h+1}(s',a'),$$

$$\hat{f}_h(s,a) \geq r(s,a) + \frac{1}{\beta_h(s,a)} \sum_{s'} \mathbb{P}_h(s' \mid s,a) \sum_{a'} \pi_{e,h+1}(a' \mid s') \hat{f}_{h+1}(s',a').$$

The ultimate goal is to bound $V_1^{\pi_e}(s) - \sum_a \pi_{e,1}(a \mid s) \hat{f}_1(s,a)$, which is given by $\sum_a \pi_{e,1}(a \mid s) \left(Q_1^{\pi_e}(s,a) - \hat{f}_1(s,a) \right)$. So, we consider the error of $\hat{f}_h(s,a)$ at every step, given by $\text{err}_h(s,a) := Q_h^{\pi_e}(s,a) - \hat{f}_h(s,a)$. We will use the following relation.

$$\begin{aligned} Q_h^{\pi_e}(s,a) &= r(s,a) + \sum_{u,s'} P_h(u \mid s) \mathbb{P}_h(s' \mid s,a,u) V_{h+1}^{\pi_e}(s') \\ &= r(s,a) + \sum_{s'} \mathbb{P}_h(s' \mid s,a) V_{h+1}^{\pi_e}(s') \end{aligned} \tag{B.2}$$

At $h = H$, by definition

$$\hat{f}_H(s,a) = r(s,a) = Q_H^{\pi_e}(s,a).$$

Thus, we get that $\text{err}_H(s,a) = 0$ for all s, a. Let $\beta_{max} := \max_{s,a,h} \beta_h(s,a)$ and let $\alpha_{min} = \min_{s,a,h} \alpha_h(s,a)$.

For step $H-1$,

$$\begin{aligned} \text{err}_{H-1}(s,a) &\leq \sum_{s'} \mathbb{P}_{H-1}(s' \mid s,a) V_H^{\pi_e}(s') \\ &\quad - \frac{1}{\beta_H(s,a)} \sum_{s'} \mathbb{P}_{H-1}(s' \mid s,a) \sum_{a'} \pi_{e,H}(a' \mid s') \hat{f}_H(s',a') \\ &= (1 - \frac{1}{\beta_H(s,a)}) \sum_{s'} \mathbb{P}_{H-1}(s' \mid s,a) V_H^{\pi_e}(s') \\ &\leq \left(1 - \frac{1}{\beta_{max}}\right) \sum_{s'} \mathbb{P}_{H-1}(s' \mid s,a) \\ &\quad \cdot \left(1 - \frac{1}{\beta_{max}}\right) \end{aligned}$$

By induction, we will show that for all h, the following holds.

$$\text{err}_h(s,a) \leq H - h - \sum_{i=1}^{H-h} \frac{1}{\beta_{max}^i}$$

We know this for $h = H-1$. For the induction step, we show this for $h-1$ given the statement

for h using the following computation.

$$\begin{aligned}
\text{err}_{h-1} &\leq \sum_{s'} \mathbb{P}_{h-1}(s' \mid s,a) V_h^{\pi_e}(s') - \frac{1}{\beta_h(s,a)} \sum_{s'} \mathbb{P}_{h-1}(s' \mid s,a) \sum_{a'} \pi_{e,h}(a' \mid s') \hat{f}_h(s',a') \\
&\leq \sum_{s'} \mathbb{P}_{h-1}(s' \mid s,a) V_h^{\pi_e}(s') \\
&\quad + \frac{1}{\beta_h(s,a)} \sum_{s'} \mathbb{P}_{h-1}(s' \mid s,a) \sum_{a'} \pi_{e,h}(a' \mid s') (\text{err}_h(s,a) - Q_h^{\pi_e}(s,a)) \\
&= (1 - \frac{1}{\beta_h(s,a)}) \sum_{s'} \mathbb{P}_{h-1}(s' \mid s,a) V_h^{\pi_e}(s') + \frac{1}{\beta_h(s,a)} \text{err}_h(s,a) \\
&\leq \left(1 - \frac{1}{\beta_{max}}\right) \sum_{s'} \mathbb{P}_{h-1}(s' \mid s,a)(H-h+1) + + \frac{1}{\beta_h(s,a)} \text{err}_h(s,a) \\
&\leq \left(1 - \frac{1}{\beta_{max}}\right)(H-h+1) + \frac{1}{\beta_{max}} \left(H - h - \sum_{i=1}^{H-h} \frac{1}{\beta_{max}^i}\right) \\
&= H - h + 1 - \sum_{i=1}^{H-h+1} \frac{1}{\beta_{max}^i}
\end{aligned}$$

Thus, the result holds by induction, giving us the following final bound.

$$Q_1^{\pi_e}(s,a) - \hat{f}_1(s,a) \leq H - 1 - \sum_{i=1}^{H-1} \frac{1}{\beta_{max}^i} = H - \frac{1 - \frac{1}{\beta_{max}^H}}{1 - \frac{1}{\beta_{max}}}$$

Similarly, we have the lower bound below:

$$Q_1^{\pi_e}(s,a) - \hat{f}_1(s,a) \geq H - 1 - \sum_{i=1}^{H-1} \frac{1}{\alpha_{min}^i} = H - \frac{1 - \frac{1}{\alpha_{min}^H}}{1 - \frac{1}{\alpha_{min}}}$$

Recall that $\alpha_h(s,a) = \pi_{b,h}(a \mid s) + \frac{1}{\Gamma}(1 - \pi_{b,h}(a \mid s))$ and $\beta_h(s,a) = \Gamma + \pi_{b,h}(a \mid s)(1-\Gamma)$. So, $\alpha_h(s,a) \geq \frac{1}{\Gamma}$ and $\beta_h(s,a) \leq \Gamma$ for all s, a, h. In particular, $\alpha_{min} \geq \frac{1}{\Gamma} = \frac{1}{1+\varepsilon}$ and $\beta_{max} \leq \Gamma = 1 + \varepsilon$.

In particular, we have the following bound.

$$\frac{1 + \varepsilon H - (1+\varepsilon)^H}{\varepsilon} \leq V_1^{\pi_e}(s) - \sum_a \pi_{e,1}(a \mid s) \hat{f}_1(s,a) \leq \frac{\frac{1}{(1+\varepsilon)^H} - (1 - \varepsilon H)}{\varepsilon}$$

We know that we have the following bounds for small ε: $(1+\varepsilon)^H \geq 1 + \varepsilon H + O(\varepsilon H^2)$ and $\frac{1}{(1+\varepsilon)^H} \leq 1 - \varepsilon H + O(\varepsilon H^2)$, giving us the following bound for small ε.

$$|V_1^{\pi_e}(s) - \sum_a \pi_{e,1}(a \mid s)\hat{f}_1(s,a)| \leq O(\varepsilon H^2)$$

□

Remark 18. For any ε, the lower bound $\frac{1+\varepsilon H-(1+\varepsilon)^H}{\varepsilon} \leq -\frac{\varepsilon H^2}{2}$, and thus we need to be at least as conservative as subtracting $\frac{\epsilon H^2}{2}$ from the FQE estimate to get a lower bound, if not more. This remark will be used in Section 4.3.

We further remark in Section 4.2.3 that the bound in the theorem is data-oblivious, being only dependent on the confounding sensitivity model and horizon, and note that the other two methods below (CFQE and MB) both produce bounds at least as tight as this one.

B.4.2 Proof of CFQE Error Bounds, Theorem 4.2.3

We recall the theorem below.

Theorem 4.2.3 (CFQE Error)**.** *Suppose $\Gamma = 1 + \varepsilon$ in Assumption 5. Then the worst-case error for the lower bound $\hat{f}_1(s,a)$ generated by CFQE in the infinite-sample case is $|V_1^{\pi_e}(s) - \sum_a \pi_{e,1}(a \mid s)\hat{f}_1(s,a)| = O(\varepsilon H^2)$ for any range of ε.*

Proof. In the limit of infinite data, the true value of g_h always lies in the set $\tilde{B}_{sa,h}$ by the sensitivity assumption. So, CFQE trivially gives a lower bound on the true value function in the limit of infinite data. We now give bounds on its error below.

We define the error term at each step by $\text{err}_h(s,a) := \max_{s,a} Q_h^{\pi_e}(s,a) - \hat{f}_h(s,a)$, where here f is generated by CFQE. We claim that

$$\text{err}_h(s,a) = (H-h) - \frac{\alpha_{min}}{\beta_{max}} - \cdots - \frac{\alpha_{min}^{H-h}}{\beta_{max}^{H-h}}. \tag{B.3}$$

Then, the following bound follows for any ε.

$$\begin{aligned} V_1^{\pi_e}(s) - \sum_a \pi_e(a \mid s)\hat{f}_1(s,a) &\leq H - 1 - \frac{\alpha_{min}}{\beta_{max}} - \cdots - \frac{\alpha_{min}^{H-1}}{\beta_{max}^{H-1}} \\ &\leq H - \sum_{i=0}^{H-1} \frac{1}{(1+\varepsilon)^{2i}} \\ &\leq 2\varepsilon H^2 \end{aligned}$$

This completes the proof since by induction, $\hat{f}_h(s,a) \leq Q_(s,a)$ for all h, and so we already have

the lower bound $V_1^{\pi_e}(s) - \sum_a \pi_e(a \mid s)\hat{f}_1(s,a) \geq 0$. Thus, it remains to prove B.3.

At step H of CFQE, we have

$$\hat{f}_H(s,a) = r(s,a).$$

Then as in the previous proof, the error at step H is given by $\text{err}_H(s,a) = 0$.

At step $h+1$, suppose $\text{err}_{h+1}(s,a) = (H-h-1) - \frac{\alpha_{min}}{\beta_{max}} - \cdots - \frac{\alpha_{min}^{H-h-1}}{\beta_{max}^{H-h-1}}$. Then for step h, we have the following chain of inequalities for $\text{err}_h(s,a) = Q_h^{\pi_e}(s,a) - \hat{f}_h(s,a)$.

$$\begin{aligned}
&\sum_{s'} \mathbb{P}_h(s' \mid s,a) V_{h+1}^{\pi_e}(s') \\
&\qquad - \sum_{u,s'} \mathbb{P}^{\pi_b}(s', u \mid s,a) \pi_{b,h}(a \mid s) g_h(s,a,s') \sum_{a'} \pi_{e,h+1}(a' \mid s') \hat{f}_{h+1}(s',a') \\
&= \sum_{s'} \mathbb{P}_h(s' \mid s,a) V_{h+1}^{\pi_e}(s') \\
&\qquad - \sum_{u,s'} P_h^{\pi_b}(u \mid s,a) \mathbb{P}_h(s' \mid s,u,a) \pi_{b,h}(a \mid s) g_h(s,a,s') \sum_{a'} \pi_{e,h+1}(a' \mid s') \hat{f}_{h+1}(s',a') \\
&= \sum_{s'} \mathbb{P}_h(s' \mid s,a) V_{h+1}^{\pi_e}(s') \\
&\qquad - \sum_{u,s'} P_h(u \mid s) \frac{\pi_{b,h}(a \mid s,u)}{\pi_{b,h}(a \mid s)} \mathbb{P}_h(s' \mid s,u,a) \pi_{b,h}(a \mid s) g_h(s,a,s') \\
&\qquad\qquad \cdot \sum_{a'} \pi_{e,h+1}(a' \mid s') \hat{f}_{h+1}(s',a') \\
&\leq \sum_{s'} \mathbb{P}_h(s' \mid s,a) V_{h+1}^{\pi_e}(s') \\
&\qquad - \frac{\alpha_h(s,a)}{\beta_h(s,a)} \sum_{u,s'} P_h(u \mid s) \mathbb{P}_h(s' \mid s,u,a) \sum_{a'} \pi_{e,h+1}(a' \mid s') (\text{err}_{h+1} - Q_{h+1}^{\pi_e}(s,a)) \\
&= \left(1 - \frac{\alpha_h(s,a)}{\beta_h(s,a)}\right) \sum_{s'} \mathbb{P}_h(s' \mid s,a) V_{h+1}^{\pi_e}(s') + \frac{\alpha_h(s,a)}{\beta_h(s,a)} \text{err}_{h+1} \\
&\leq \left(1 - \frac{\alpha_h(s,a)}{\beta_h(s,a)}\right)(H-h) + \frac{\alpha_h(s,a)}{\beta_h(s,a)} \left(H - h - 1 - \frac{\alpha_{min}}{\beta_{max}} - \cdots - \frac{\alpha_{min}^{H-h-1}}{\beta_{max}^{H-h-1}}\right) \\
&\leq \left(1 - \frac{\alpha_{min}}{\beta_{max}}\right)(H-h) + \frac{\alpha_{min}}{\beta_{max}} \left(H - h - 1 - \frac{\alpha_{min}}{\beta_{max}} - \cdots - \frac{\alpha_{min}^{H-h-1}}{\beta_{max}^{H-h-1}}\right) \\
&= H - h - \frac{\alpha_{min}}{\beta_{max}} - \cdots - \frac{\alpha_{min}^{H-h}}{\beta_{max}^{H-h}}.
\end{aligned}$$

The first expression comes from using equation B.2 as well as explicitly computing the $\arg\min$ involved in CFQE in the limit of infinite data, analogous to the proof of Theorem 4.2.2 above. In the

first inequality, we use the facts that $\frac{\pi_{b,h}(a|s,u)}{\pi_{b,h}(a|s)} \geq \frac{1}{\beta_h(s,a)}$ and $\pi_{b,h}(a \mid s) g_h(s,a,s') \leq \alpha_h(s,a)$. In the equality after that, we use the definition of $\text{err}_h(s,a)$. In the second inequality, we use equation B.2.

Thus, $\text{err}_h(s,a) = H - h - \frac{\alpha_{min}}{\beta_{max}} - \cdots - \frac{\alpha_{min}^{H-h}}{\beta_{max}^{H-h}}$ and (B.3) is proved. □

B.5 Model-Based Method

B.5.1 General Memoryless Version

Notice that the model-based method leverages the fact that the *marginalized transition dynamics* are stationary. In particular, we only need $\mathbb{P}_h(s' \mid s,a,u)$ and $P_h(u \mid s)$ to be stationary, since this makes the marginalized transition structure stationary. In that light, we discuss here the version of the method where π_b and π_e are non-stationary.

Consider the observed transition structure at timestep h, given by $\hat{\mathbb{P}}_h^{\pi_b}$, denote by $\hat{\pi}_{b,h}$ the observed behavior policy and by $\hat{\alpha}_h(s,a)$ and $\hat{\beta}_h(s,a)$ the versions of $\alpha_h(s,a)$ and $\beta_h(s,a)$ computed using $\hat{\pi}_{b,h}$. Let π_e also be non-stationary. For the model to improve over CFQE, we still need $P_h(u \mid s)$ to be the same for all timesteps h, so that the marginalized transition structure is stationary.

Define $\mathcal{G}_h := \{\mathbb{P} : \hat{\alpha}_h(s,a)\hat{\mathbb{P}}_h^{\pi_b}(s' \mid s,a) \leq \mathbb{P}(s' \mid s,a) \leq \hat{\beta}_h(s,a)\hat{\mathbb{P}}_h^{\pi_b}(s' \mid s,a),\ \forall s,a,s'\}$

Note that in the limit of infinite data, the true marginalized transition structure satisfies the following relation for each h.

$$\alpha_h(s,a)\mathbb{P}_h^{\pi_b}(s' \mid s,a) \leq \mathbb{P}(s' \mid s,a) \leq \beta_h(s,a)\mathbb{P}_h^{\pi_b}(s' \mid s,a),\ \forall s,a,s'$$

So, in the limit of infinite data, the true marginalized structure lies in $\cap_h \mathcal{G}_h$. We then define this to be $\mathcal{G} := \cap_h \mathcal{G}_h$ even in the finite sample case.

With this as our $\mathcal{G}$, we have the same program for obtaining a model-based lower bound on the value function.

$$\begin{aligned} &\min_{V_1(s_0),V_2,\ldots,V_H,V_{H+1}=0,\mathbb{P}} V_1(s_0) \\ &\text{s.t. } \mathbb{P} \in \mathcal{G},\ \sum_{s'} \mathbb{P}(s' \mid s,a) = 1\ \ \forall s,a. \\ &V_h(s) = \pi_{e,h}(\cdot \mid s)^T (R_s + \mathbb{P}_s V_{h+1}(\cdot))\ \ \forall h \in \{1,\ldots,H\}, s \end{aligned} \tag{B.4}$$

Remark 19. Note that assuming stationarity of π_b allows us to use data across timesteps to estimate a universal $\hat{\mathbb{P}}^{\pi_b}$, which helps with finite samples in practice.

We present our proofs below for stationary π_b and π_e for clarity, noting that they can easily be modified for general memoryless π_b and π_e in a similar vein as the proofs for CFQE.

B.5.2 Confidence Interval for State Transitions

We can use the following lemma to modify our definition of the set $\mathcal{G}$ to use confidence intervals instead of point estimates. We show that both methods converge to the lower bound obtained with infinite data. However, using Hoeffding confidence intervals to modify $\mathcal{G}$ ensures that for any amount of data, the output of the model-based method is a true lower bound on the value function. In the version that uses point estimates of π_b and $\mathbb{P}^{\pi_b}$, we only get estimates of a lower bound with finite data.

Let $N(s)$ and $N(s,a)$ be the counts of s and (s,a) in the data.

Lemma B.5.1 (Confidence Interval for State Transitions)**.** *For* $\Delta_\pi := \sqrt{\frac{1}{2N^*(s)}\log(\frac{2SA}{\delta_1})}$, $\Delta_\mathbb{P} := \sqrt{\frac{1}{2N^*(s,a)}\log(\frac{2S^2A}{\delta_2})}$, *bounds* $\alpha_{\delta_1}(s,a) := 1/\Gamma - (1-1/\Gamma)(\hat{\pi}_b(a|s) + \Delta_\pi)$ *and* $\beta_{\delta_1}(s,a) := \Gamma + (1-\Gamma)(\hat{\pi}_b(a|s) + \Delta_\pi)$, *and* $N^*(s) = -\text{logmeanexp}(\{-N(s_1), ...\})$, $\mathbb{P}(s'|s,a)$ *falls between* $\alpha_{\delta_1}(s,a)(\hat{\mathbb{P}}^{\pi_b}(s'|s,a) - \Delta_\mathbb{P})$ *and* $\beta_{\delta_1}(s,a)(\hat{\mathbb{P}}^{\pi_b}(s'|s,a) + \Delta_\mathbb{P})$ *with probability at least* $1 - \delta_1 - \delta_2$.

Proof. We attempt to use the data collected by π_b to construct a confidence interval for $\hat{\mathbb{P}}(s' \mid s,a)$ that also takes into account estimation error in the bounds $\alpha_{\delta_1}(s,a)$ and $\beta_{\delta_1}(s,a)$. We consider below empirical estimation for $\mathbb{P}(s' \mid s,a)$ and $\pi_b(a|s)$ using data collected by π_b:

$$\hat{\mathbb{P}}^{\pi_b}(s' \mid s,a) = \frac{N(s,a,s')}{N(s,a)}, \quad \hat{\pi}_b(a|s) = \frac{N(s,a)}{N(s)}$$

where $N(s,a,s') := \sum_{i=1}^n \mathbb{1}_{\{s_i=s, a_i=a, s_i'=s'\}}$, $N(s,a) := \sum_{i=1}^n \mathbb{1}_{\{s_i=s,a_i=a\}}$, and $N(s) := \sum_{i=1}^n \mathbb{1}_{\{s_i=s\}}$.

Note that $\mathbb{P}(s' \mid s,a) = \sum_u P(u \mid s)\mathbb{P}(s' \mid s,u,a)$, while $\mathbb{P}^{\pi_b}(s' \mid s,a) = \sum_u \mathbb{P}^{\pi_b}(u \mid s,a)\mathbb{P}(s' \mid s,u,a)$. In π_b, u and a are dependent.

We also have:

$$\begin{aligned}\mathbb{P}^{\pi_b}(s' \mid s,a) = \sum_u \mathbb{P}^{\pi_b}(s',u \mid s,a) &= \sum_u \mathbb{P}^{\pi_b}(u \mid s,a)\mathbb{P}(s' \mid s,u,a) \\ &= \sum_u P(u \mid s)\frac{\pi_b(a \mid s,u)}{\pi_b(a \mid s)}\mathbb{P}(s' \mid s,u,a).\end{aligned}$$

By Assumption 5,

$$\frac{1}{\beta(s,a)}\mathbb{P}(s' \mid s,a) \leq \mathbb{P}^{\pi_b}(s' \mid s,a) \leq \frac{1}{\alpha(s,a)}\mathbb{P}(s' \mid s,a) \tag{B.5}$$

$$\alpha(s,a)\mathbb{P}^{\pi_b}(s' \mid s,a) \leq \mathbb{P}(s' \mid s,a) \leq \beta(s,a)\mathbb{P}^{\pi_b}(s' \mid s,a). \tag{B.6}$$

We claim that by Hoeffding's inequality and the union bound, with probability at least $1-\delta_1-\delta_2$,

$$|\hat{\pi}_b(a \mid s) - \pi_b(s \mid a)| \leq \sqrt{\frac{1}{2N^*(s)}\log\left(\frac{2SA}{\delta_1}\right)} = \Delta_\pi$$

$$\left|\hat{\mathbb{P}}^{\pi_b}(s' \mid s,a) - \mathbb{P}^{\pi_b}(s' \mid s,a)\right| \leq \sqrt{\frac{1}{2N^*(s,a)}\log\left(\frac{2S^2A}{\delta_2}\right)} = \Delta_{\mathbb{P}} \tag{B.7}$$

where $N^*(s) = -\operatorname{logmeanexp}(\{-N(s_1), ...\})$ and $N^*(s,a) = -\operatorname{logmeanexp}(\{-N(s_1,a_1), ...\})$.

We illustrate this by showing the result for $\hat{\mathbb{P}}^{\pi_b}(s' \mid s,a)$, and the other case follows analogously.

$$\begin{aligned}
\mathbb{P}(\exists s', s, a \;\; s.t. \;\; |\hat{\mathbb{P}}^{\pi_b}(s' \mid s,a) - \mathbb{P}^{\pi_b}(s' \mid s,a)| \leq \epsilon) &\leq \sum_{s',s,a} \mathbb{P}(|\hat{\mathbb{P}}^{\pi_b}(s' \mid s,a) - \mathbb{P}^{\pi_b}(s' \mid s,a)| \leq \epsilon) \\
&\leq \sum_{s',s,a} 2\exp\{-2\epsilon^2 N(s,a)\} \\
&= S\sum_{s,a} 2\exp\{-2\epsilon^2 N(s,a)\} \\
&\leq 2S^2A\exp\{-2\epsilon^2 N^*(s,a)\} = \delta
\end{aligned}$$

for some N^* that satisfies the last inequality above. Various choices for N^* exist. Perhaps the most obvious choice is the $\min$ function, though it can be shown that $-\operatorname{logmeanexp}(-x)$ is optimal, as:

$$x^* \;\; s.t. \;\; \sum_n e^{X_n} = ne^{x^*} \iff e^{x^*} = \frac{1}{n}\sum_n e^{X_n} \iff \operatorname{logmeanexp}(X_1, ..., X_n) = x^*$$

The $\operatorname{logmeanexp}$ function returns a value between the maximum and the mean, and in our case, we use it to obtain a soft approximation to the minimum that provides a less conservative bound

than using the minimum of counts over all states (or states and actions).

Combining our inequalities B.7 and B.6 with the definitions of $\alpha(s,a)$ and $\beta(s,a)$, we have our result. □

B.5.3 Solving (4.2) Gives Better Lower Bound than Confounded FQE, Proof of Theorem 4.2.4

Recall Theorem 4.2.4 below.

Theorem 4.2.4 (Error for the Model-Based Method)**.** *Suppose $\Gamma = 1 + \varepsilon$ in Assumption 5. Then the value estimation from solving (4.2) with infinite data, denoted by $\tilde{V}_1$, provides a lower bound no looser than CFQE and satisfies that $|V_1^{\pi_e}(s_0) - \tilde{V}_1(s_0)| = O(\varepsilon H^2)$ for any range of ε.*

We consider the infinite sample setting, which means:

$$\mathcal{G} = \{\mathbb{P} : \alpha(s,a) \leq \frac{\mathbb{P}(s' \mid s,a)}{\mathbb{P}^{\pi_b}(s' \mid s,a)} \leq \beta(s,a), \text{ for } \forall s,a,s'\}$$

The key to the proof is the observation that we can always get a valid g_h from a valid $\mathbb{P} \in \mathcal{G}$ by setting $g_h(s,a,s') := \frac{\mathbb{P}(s'|s,a)}{\mathbb{P}^{\pi_b}(s'|s,a)\pi_b(a|s)}$, which formalizes the intuition that the uncertainty set $\mathcal{G}$ for $\mathbb{P}$ is tighter. Since we are in the stationary case, we drop all unnecessary h in subscripts.

Proof. We denote the solution of (4.2) in the infinite-sample setting by $\tilde{V}_1, \ldots, \tilde{V}_H, \tilde{V}_{H+1}, \tilde{\mathbb{P}}$. We will show that $\tilde{V}_1$ gives a lower bound on the true value function that is larger than the lower bound given by CFQE. That is, if the iterates of CFQE are $\hat{f}_h(s,a)$, then $\sum_a \pi_e(a \mid s)\hat{f}_1(s,a) \leq \tilde{V}_1(s) \leq V_1^{\pi_e}(s)$. Combining this with Theorem 4.2.3 gives us the whole theorem.

First note that in the infinite data setting, the marginalized transition kernel lies in $\mathcal{G}$, so the optimization problem minimizes V_1 over values of $\mathbb{P}$ that include the true marginalized transition structure. Thus, we trivially get that $\tilde{V}_1(s) \leq V_1^{\pi_e}(s)$.

We now prove that $V_h(s) \geq \sum_a \pi_e(a \mid s)\hat{f}_h(s,a)$ holds for all h by induction. Note that the argument below also works for the finite-sample case by merely replacing every quantity associated with π_b (such as $\mathbb{P}^{\pi_b}$) by its finite sample version.

For $h = H+1$:

$$\tilde{V}_{H+1}(s) = 0 \geq 0 = \sum_a \pi_e(a \mid s)\hat{f}_{H+1}(s,a)$$

Suppose we have $\tilde{V}_{h+1}(s) \geq \sum_a \pi_e(a \mid s)\hat{f}_{h+1}(s,a)$. Then for step h:

$$\tilde{V}_h(s) = \sum_a \pi_e(a \mid s) \left[R(s,a) + \sum_{s'} \tilde{\mathbb{P}}(s' \mid s,a)\tilde{V}_{h+1}(s') \right]$$

$$\begin{aligned}
\hat{f}_h(s,a) &= \min_{g \in \tilde{\mathcal{B}}_{sa}} \left(\sum_{u,s'} \mathbb{P}^{\pi_b}(s',u \mid s,a) CFQE(\hat{f}_{h+1}, g) \right) \\
&\leq \sum_{u,s'} \mathbb{P}^{\pi_b}(s',u \mid s,a) \frac{\tilde{\mathbb{P}}(s' \mid s,a)}{\mathbb{P}^{\pi_b}(s' \mid s,a)} \left[R(s,a) + \sum_{a'} \pi_e(a' \mid s')\hat{f}_{h+1}(s',a') \right] \\
&= \sum_{s'} \tilde{\mathbb{P}}(s' \mid s,a) \left[R(s,a) + \sum_{a'} \pi_e(a' \mid s')\hat{f}_{h+1}(s',a') \right] \leq R(s,a) + \sum_{s'} \tilde{\mathbb{P}}(s' \mid s,a)\tilde{V}_{h+1}(s').
\end{aligned}$$

where

$$CFQE(\hat{f}_{h+1}, g) := \left(\sum_{s'} \pi_b(a \mid s) g(s,a,s') \left[R(s,a) + \sum_{a'} \pi_e(a' \mid s')\hat{f}_{h+1}(s',a') \right] \right)$$

The first inequality in above is achieved by setting $g(s,a,s') = \frac{\tilde{\mathbb{P}}(s'|s,a)}{\mathbb{P}^{\pi_b}(s'|s,a)\pi_b(a|s)}$. It's easy to check that by this choice, $g(s,a,\cdot) \in \tilde{\mathcal{B}}_{sa}$ by (B.6). The second inequality is by the induction hypothesis. Thus, we have $\tilde{V}_h(s) \geq \sum_a \pi_e(a \mid s)\hat{f}_h(s,a)$.

By induction, $\tilde{V}_1(s) \geq \sum_a \pi_e(a \mid s)\hat{f}_1(s,a)$, which means the lower bound provided by (4.2) is always no worse than confounded FQE (Alg. 5).

□

B.5.4 Worst-Case Error for the Model-Based Method, An Independent Alternative Proof

In this section, we give an alternative proof of the fact that the output of (4.2) satisfies $|V_1^{\pi_e}(s) - \tilde{V}_1| = O(\varepsilon H^2)$ for $\Gamma = 1 + \varepsilon$ without comparing to CFQE. Again, recall that we consider the infinite sample setting, which means the following.

$$\mathcal{G} = \{\mathbb{P} : \alpha(s,a) \leq \frac{\mathbb{P}(s' \mid s,a)}{\mathbb{P}^{\pi_b}(s' \mid s,a)} \leq \beta(s,a), \text{ for } \forall s,a,s'\}$$

Proof. By definition, we know $V_H^{\pi_e}(s) = \tilde{V}_H(s)$ for all s. We define $\delta_h = \max_s |V_h^{\pi_e}(s) - \tilde{V}_h(s)|$.

Note that $\delta_H = 0$. Next, consider $|V_h^{\pi_e}(s) - \tilde{V}_h(s)|$:

$$\begin{aligned}
\delta_h &:= \max_s |V_h^{\pi_e}(s) - \tilde{V}_h(s)| \\
&= \max_s |\sum_a \pi_e(a \mid s) \sum_{s'} \mathbb{P}(s' \mid s, a) V_{h+1}(s') - \sum_a \pi_e(a \mid s) \sum_{s'} \tilde{\mathbb{P}}(s' \mid s, a) \tilde{V}_{h+1}(s')| \\
&\leq \max_s |\sum_a \pi_e(a \mid s) \sum_{s'} \mathbb{P}(s' \mid s, a) V_{h+1}(s') - \sum_a \pi_e(a \mid s) \sum_{s'} \tilde{\mathbb{P}}(s' \mid s, a) V_{h+1}(s')| \\
&\quad + \max_s |\sum_a \pi_e(a \mid s) \sum_{s'} \tilde{\mathbb{P}}(s' \mid s, a) V_{h+1}(s') - \sum_a \pi_e(a \mid s) \sum_{s'} \tilde{\mathbb{P}}(s' \mid s, a) \tilde{V}_{h+1}(s')| \\
&= \max_s |\sum_a \pi_e(a \mid s) \sum_{s'} \mathbb{P}(s' \mid s, a) V_{h+1}(s') - \sum_a \pi_e(a \mid s) \sum_{s'} \tilde{\mathbb{P}}(s' \mid s, a) V_{h+1}(s')| \\
&\quad + \delta_{h+1} \\
&\leq (\beta_{\text{max}} - \alpha_{\text{min}})(H - h) + \delta_{h+1},
\end{aligned}$$

where

$$\beta_{\text{max}} := \max_{s,a} \Gamma + \pi_b(a \mid s)(1 - \Gamma) \leq 1 + \varepsilon$$

and

$$\alpha_{\text{min}} := \min_{\pi_b(a|s)} \frac{\varepsilon}{1+\varepsilon} \pi_b(a \mid s) + \frac{1}{1+\varepsilon} \geq \frac{1}{1+\varepsilon}$$

It is easy to check $\beta_{\text{max}} - \alpha_{\text{min}} = \varepsilon + \frac{\varepsilon}{1+\varepsilon} = O(\varepsilon)$ (ignoring higher order terms of ε). So, we get that $\delta_h \leq O(\varepsilon(H - h)) + \delta_{h+1}$ from $h = 1, \ldots, H$. So, we have that

$$\delta_1 \leq O(\varepsilon H^2)$$

□

B.5.5 Consistency of the Model-Based Method

We first prove this extremely elementary and useful geometric lemma.

Lemma B.5.2. *If a function $f : X \to \mathbb{R}$ on a Hausdorff metric space X is continuous (resp. Lipschitz), then $f_{\min} : Comp(X) \to \mathbb{R}$ given by $f_{\min}(K) := \inf_{x \in K} f(x)$ is also continuous (resp. Lipschitz) in the Hausdorff metric on the space $Comp(X)$ of compact subsets of X. The same holds for $f_{\max}(K) := \sup_{x \in K} f(x)$.*

Proof. We prove this for α-Lipschitz f and $f_{\min}$, the other cases are similar. Consider compact sets K_1 and K_2, so that the infima are attained at $x_i \in K_i$. This means that $f_{\min}(K_i) = f(x_i)$. Since K_j are closed, we have points u_j that attain the closest distance from x_i to K_j. Combining these, we

know that

$$d(x_i, K_j) \leq d_{Haus}(K_i, K_j)$$

and

$$d(x_i, u_j) = d(x_i, K_j) := inf_{u \in K_j} d(x_i, u)$$

Using the Lipschitzness of f,

$$|f(x_i) - f(u_j)| \leq \alpha d(x_i, u_j) \leq \alpha d_{Haus}(K_i, K_j)$$

Also, $f(u_j) \geq f(x_j)$ by definition of x_j, since $u_j \in K_j$ and x_j minimizes f over K_j. So,

$$f(x_i) \geq f(u_j) - \alpha d_{Haus}(K_i, K_j) \geq f(x_j) - \alpha d_{Haus}(K_i, K_j)$$

This holds for $(i, j) = (1, 2), (2, 1)$, so we get that

$$|f_{\min}(K_i) - f_{\min}(K_j)| = |f(x_i) - f(x_j)| \leq \alpha d_{Haus}(K_i, K_j)$$

□

We use Lemma B.5.2 along with the fact that the objective function is Lipschitz. We will prove it for the version of the Model-Based method incorporating Hoeffding-based bounds (which are incorporated to give finite sample guarantees). The proof for the version with point estimates of the relevant quantities is in fact easier and subsumed by this by setting $\Delta_\pi = \Delta_{\mathbb{P}} = 0$. We first need the lemma below, which will we later combine with Lemma B.5.2.

Lemma B.5.3. *Let the feasible region given by the values of $P_{\pi_b}, \alpha(s, a)$ and $\beta(s, a)$ in the limit of infinite data be F. Let the feasible region obtained using our finite sample estimates in Lemma B.5.1 be $\hat{F}$. Then there is a constant K depending on Γ so that*

$$d_{Haus}(F, \hat{F}) \leq 2S^2A\Gamma \left(|\mathbb{P}^{\pi_b}(s' \mid s, a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s, a)| + |\pi_b(s \mid a) - \hat{\pi_b}(s \mid a)| + \Delta_{\mathbb{P}} + \Delta_\pi \right)$$

Notice that this also applies to the case of replacing the Hoeffding-based intervals by the point estimates, since that merely involves replacing Δ_π and/or $\Delta_{\mathbb{P}}$ by 0.

Proof. Notice that the condition

$$\sum_{s'} \mathbb{P}(s' \mid s, a) = 1$$

is identical across both sets, so the difference is only induced by the infinite-sample $\mathcal{G}_\infty$ and the

finite sample $\mathcal{G}$. That is, for $\mathbb{P} \in \mathcal{G}_\infty$, we have the following

$$\alpha(s,a)\mathbb{P}^{\pi_b}(s'|s,a) \leq \mathbb{P}(s' \mid s,a) \leq \beta(s,a)\mathbb{P}^{\pi_b}(s'|s,a)$$

Let's call the interval above $I_{s,a}$. For $\mathbb{P} \in \mathcal{G}$, we instead have

$$\alpha_{\delta_1}(s,a)(\hat{\mathbb{P}}^{\pi_b}(s'|s,a) - \Delta_{\mathbb{P}}) \leq \mathbb{P}(s' \mid s,a) \leq \beta_{\delta_1}(s,a)(\hat{\mathbb{P}}^{\pi_b}(s'|s,a) + \Delta_{\mathbb{P}})$$

We can check that using the inequalities above, the following hold for $\hat{w} \in \hat{F}$:

- If $\hat{w}_{s',s,a} < \alpha(s,a)\mathbb{P}^{\pi_b}(s' \mid s,a)$, then

$$\begin{aligned}
&d(\hat{w}_{s',s,a}, I_{s,a}) \\
&\leq \alpha(s,a)|\mathbb{P}^{\pi_b}(s' \mid s,a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s,a)| + \alpha(s,a)\Delta_{\mathbb{P}} \\
&\qquad + (\hat{\mathbb{P}}^{\pi_b}(s' \mid s,a) + \Delta_{\mathbb{P}})|\alpha(s,a) - \alpha_{\delta_1}(S,a)| \\
&\leq |\mathbb{P}^{\pi_b}(s' \mid s,a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s,a)| + \Delta_{\mathbb{P}} + 2\left(1 - \frac{1}{\Gamma}\right)(|\pi_b(s \mid a) - \hat{\pi}_b(s \mid a)| + \Delta_\pi) \\
&\leq |\mathbb{P}^{\pi_b}(s' \mid s,a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s,a)| + K_1|\pi_b(s \mid a) - \hat{\pi}_b(s \mid a)| + \Delta_{\mathbb{P}} + K_1\Delta_\pi
\end{aligned}$$

  where $K_1 = 2\left(1 - \frac{1}{\Gamma}\right)$.

- If $\hat{w}_{s',s,a} > \beta(s,a)\mathbb{P}^{\pi_b}(s' \mid s,a)$ then we get terms using β, so that we have

$$d(\hat{w}_{s',s,a}, I_{s,a}) \leq K_3|\mathbb{P}^{\pi_b}(s' \mid s,a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s,a)| + K_2|\pi_b(s \mid a) - \hat{\pi}_b(s \mid a)| + K_3\Delta_{\mathbb{P}} + K_2\Delta_\pi$$

  with $K_2 = 2(\Gamma - 1)$ and $K_3 = \Gamma$

- In the third case, $\hat{w}_{s',s,a} \in I_{s,a}$, so $d(\hat{w}_{s',s,a}, I_{s,a}) = 0$

Combining these and noting that $2\Gamma \geq K_1, K_2, K_3$, we have that

$$d(\hat{w}_{s',s,a}, I_{s,a}) \leq 2\Gamma(|\mathbb{P}^{\pi_b}(s' \mid s,a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s,a)| + |\pi_b(s \mid a) - \hat{\pi}_b(s \mid a)| + \Delta_{\mathbb{P}} + \Delta_\pi)$$

This means that by the triangle inequality, for any matrix/vector norm on $\mathbb{R}^{S^2A}$,

$$\begin{aligned}
d(\hat{w}, F) = d(\hat{w}, \prod_{s',s,a} I_{s,a}) &\leq S^2A \max_{s',s,a} d(\hat{w}_{s',s,a}, I_{s,a}) \\
&\leq 2S^2A\Gamma\left(|\mathbb{P}^{\pi_b}(s' \mid s,a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s,a)| + |\pi_b(s \mid a) - \hat{\pi}_b(s \mid a)| + \Delta_{\mathbb{P}} + \Delta_\pi\right)
\end{aligned}$$

Since $\hat{w} \in \hat{F}$ is arbitrary,

$$d_{Haus}(F, \hat{F}) \leq 2S^2A\Gamma \left(|\mathbb{P}^{\pi_b}(s' \mid s, a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s, a)| + |\pi_b(s \mid a) - \hat{\pi}_b(s \mid a)| + \Delta_{\mathbb{P}} + \Delta_{\pi} \right)$$

□

We finally recall and prove our consistency result below.

Theorem 4.2.5 (Consistent Estimation of the Lower Bound)**.** *The estimated lower bound from the model-based method is strongly consistent for the lower bound $\tilde{V}_1$, where $\tilde{V}_1$ is the lower bound estimate of the value function from solving* (4.2) *with infinite data. That is, $\hat{V}_1 \overset{a.s.}{\to} \tilde{V}_1$.*

Proof. To remind the reader of the precise sense in which "limit of infinite data" is used here, we mean that the behavior policy is exploratory, so that every s, a has a non-zero probability of occurring in the trajectory. In particular $N(s), N(s, a) \to \infty$ as we observe infinitely many trajectories.

We know that our objective function is a polynomial in the entries of $w = \mathbb{P}(\cdot \mid \cdot, \cdot)$. Since the entries of w lie in $[0, 1]$, the domain of our multivariate polynomial is compact and it is thus Lipschitz, since it is C^1. Let its Lipschitz constant be α. Call the minimum in the infinite data case $\tilde{V}_1$ and the one in the finite sample case $\hat{V}_1$. Combining Lemma B.5.3 with Lemma B.5.2, we get that

$$\begin{aligned} |\tilde{V}_1 - \hat{V}_1| &\leq \alpha d_{Haus}(F, \hat{F}) \\ &\leq 2\alpha S^2A\Gamma \left(|\mathbb{P}^{\pi_b}(s' \mid s, a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s, a)| + |\pi_b(s \mid a) - \hat{\pi}_b(s \mid a)| + \Delta_{\mathbb{P}} + \Delta_{\pi} \right) \end{aligned}$$

Note that as $N(s), N(s, a) \to \infty$, $|\mathbb{P}^{\pi_b}(s' \mid s, a) - \hat{\mathbb{P}}^{\pi_b}(s' \mid s, a)|, |\pi_b(s \mid a) - \hat{\pi}_b(s \mid a)| \to 0$ almost surely, and $\Delta_{\mathbb{P}}, \Delta_{\pi} \to 0$. This implies that as $N(s), N(s, a) \to \infty$, $|\tilde{V}_1 - \hat{V}_1| \to 0$ almost surely.

□

B.6 Variations of The Model-Based Method

B.6.1 Relaxation of (4.2)

Recall that in (4.2), we solved a non-convex optimization problem with $H \cdot |\mathcal{S}| + 1$ Bellman backup constraints. If one were to not require the $\mathbb{P}(s'|s,a)$ to stay constant at every step, one could sequentially solve $H \cdot |\mathcal{S}| + 1$ convex programs to obtain a lower bound that is looser than one obtained by (4.2). $\mathcal{G}_h$ is as defined in Appendix B.5. Computationally, to compute policy values for each starting state, confounded FQE (Alg. 5) solves $(H+1) \cdot |\mathcal{S}| \cdot |\mathcal{A}|$ linear programs, while Alg. 17 below solves $(H+1) \cdot |\mathcal{S}|$ convex programs.

Algorithm 17 Relaxation of Model-Based Method

1: **input:** evaluation policy π_e, starting state s_0.
2: **initialize:** $V_{H+1} \leftarrow 0$.
3: **for** $h = H, H-1, \ldots, 1$ **do**
4:

$$\begin{aligned} V_h(s) &:= \min_{\mathbb{P}_h \in \mathcal{G}_h} \sum_a \pi_{e,h}(a \mid s) \left[R(s,a) + \sum_{s'} \mathbb{P}_h(s' \mid s,a) V_{h+1}(s') \right] \\ &= \min_{\mathbb{P}_h \in \mathcal{G}_h} \pi_{e,h}(\cdot \mid s)^T (R_s + \mathbb{P}_{s,h} V_{h+1}(\cdot)). \end{aligned}$$

5: **end for**
6: **return** $V_1(s_0)$

Notice that this is similar to confounded FQE (Alg. 5) in that it optimizes over $\mathbb{P}_h(s'|s,a)$ at *each step*, instead of requiring it to stay constant for all $h = 1, \ldots H$. Consider the bijection $g_h(s,a,s') \leftrightarrow \frac{\mathbb{P}_h(s'|s,a)}{\hat{\mathbb{P}}_h^{\pi_b}(s'|s,a)\hat{\pi}_{b,h}(a|s)}$ between the uncertainty sets $\prod_h \tilde{B}_{sa,h}$ and $\prod_h \mathcal{G}_h$ for $g_1, \ldots g_H$ and $\mathbb{P}_1, \ldots, \mathbb{P}_H$ respectively. It is easy to check using the definitions of the sets that this is truly a bijection. We can see using this bijection and with an argument similar to the proof of Theorem 4.2.4, that the value estimates from this relaxation and CFQE are equal at each step. By the remark made in the proof of Theorem 4.2.4, this also holds for the finite sample versions.

B.6.2 Projected Gradient Descent

In a similar vein to Algorithm 4.1 in Kallus and Zhou [2020], we provide a method to efficiently compute the lower bound with projected gradient descent.

Given an estimate of $\mathbb{P}$, the corresponding estimate of $V_1(s_0)$ can be obtained by iteratively

performing $H+1$ Bellman backups, each of which is dependent on $\mathbb{P}$ itself. Each Bellman backup is obtained by translations and matrix multiplications of $\mathbb{P}$. As such, $V_1(s_0)$ is differentiable with respect to $\mathbb{P}$, and the gradient $\nabla_{\mathbb{P}} V_1(s_0)$ can be easily obtained with modern autograd tools.

Algorithm 18 Projected Gradient Descent for Model-Based Lower Bound

1: **input:** evaluation policy π_e, empirical estimate of $\mathbb{P}$, decaying learning rate η_t, starting state s_0.
2: **initialize:** $V_{H+1} \leftarrow 0$.
3: **for** $t = 1, ..., N$ **do**
4: **for** $h = H, H-1, \ldots, 1$ **do**
5:

$$\begin{aligned} V_h(s) &:= \sum_a \pi_e(a \mid s) \left[R(s,a) + \sum_{s'} \mathbb{P}(s' \mid s, a) V_{h+1}(s') \right] \\ &= \pi_e(\cdot \mid s)^T (R_s + \mathbb{P}_s V_{h+1}(\cdot)). \end{aligned}$$

6: **end for**
7: $\mathbb{P} \leftarrow \text{Proj}_{\mathcal{G}}(\mathbb{P} - \eta_t \nabla_{\mathbb{P}} V_1(s_0))$
8: **end for**
9: **return** the lowest $V_1(s_0)$ encountered.

B.7 FQE Does Not Work for Confounders with Memory

We recall Theorem 4.2.6 below.

Theorem 4.2.6 (Lower Bound for Confounders with Memory)**.** *There exists an MDP $\mathcal{M}$ having confounders with memory, a stationary unconfounded behavior policy π_b with sensitivity $\Gamma = 1$, a stationary evaluation policy π_e with $\frac{\pi_e(a|s)}{\pi_b(a|s)} \leq 2 \; \forall s, a$, and a state s_1, so that $V_1^{\pi_e}(s_1) = \Omega(H)$ while the output of FQE for π_e is $O(\log H)$, even with infinite data.*

Proof. We demonstrate that there exists a confounded MDP with non-memoryless confounders and a behavior policy π_e where even under the limit of infinite data, if the estimate obtained using FQE is $\hat{f}_1(s,a)$ and the true value function is $V_1^{\pi_e}(s)$, then $V_1^{\pi_e}(s) - \sum_a \pi_e(a \mid s) \hat{f}_1(s,a) = O(H)$.

Environment:

- Consider $S = \{s_1, s_2\}$, $A = \{a_1, a_2\}$, $U = \{u_0, u_{a_1}\}$, horizon H.
- Rewards: $r(s = s_1, a_1) = 1$, otherwise 0 reward.

- Starting state: Let the starting state be s_1.

Confounder distribution: The confounder's distribution starts at u_{a_1} and is induced by confounder transitions with memory. Specifically, consider the following confounder transitions.

- If $u = u_{a_1}$ and the current action is a_1, stay in u_{a_1}.
- In all other cases, transition to u_0.

State transitions: $\mathbb{P}(s_1 \mid s, a_1, u_{a_1}) = 1$ for any s, and for all other s, a, u, we have that $\mathbb{P}(s_1 \mid s, a, u) = 1/H$ and $\mathbb{P}(s_2 \mid s, a, u) = 1 - 1/H$

Behavior policy: Let $\pi_b(a \mid s, u) = \frac{1}{2}$ for any s, a, u.

Evaluation policy: Let $\pi_e(a_1 \mid s) = 1$.

Policy values: Notice that in the evaluation policy, we are always in u_{a_1} and always take action a_1, so we are always in state s_1. Thus the reward at each step is 1 and $V_1^{\pi_e}(s_1) = H$.

FQE Output: First note that to iterate through FQE for π_e, we need only compute $\hat{f}_h(s, a_1)$ for all s, h. Notice that under the behaviour policy, at timestep h, $\mathbb{P}_{\pi_b,h}(u_{a_1}) = \frac{1}{2^{h-1}}$ and $\mathbb{P}_{\pi_b,h}(u_0) = 1 - \frac{1}{2^{h-1}}$. We start with $\hat{f}_{H+1}(s, a) := 0$ and the update rule is given by

$$\begin{aligned}
\hat{f}_h(s,a) &= \mathbb{E}_{(s,a,s') \in \mathcal{D}_{\pi_b,h}}[r(s,a) + \sum_{a'} \pi_e(a' \mid s') \hat{f}_{h+1}(s', a')] \\
&= \mathbb{E}_{(s,a,s') \in \mathcal{D}_{\pi_b,h}}[r(s,a) + \hat{f}_{h+1}(s', a_1)] \\
&= r(s,a) + \sum_{s',u} \mathbb{P}_{\pi_b,h}(s', u \mid s, a) \hat{f}_{h+1}(s', a_1) \\
&= r(s,a) + \sum_{s',u} \mathbb{P}(s' \mid s, a, u) \mathbb{P}_{\pi_b,h}(u \mid s, a) \hat{f}_{h+1}(s', a_1)
\end{aligned}$$

Note that for $u = u_0, u_{a_1}$

$$\mathbb{P}_{\pi_b,h}(u \mid s, a) = \frac{\mathbb{P}_{\pi_b,h}(s, a \mid u) \mathbb{P}_{\pi_b,h}(u)}{\mathbb{P}_{\pi_b,h}(s, a \mid u_0) \mathbb{P}_{\pi_b,h}(u_0) + \mathbb{P}_{\pi_b,h}(s, a \mid u_{a_1}) \mathbb{P}_{\pi_b,h}(u_{a_1})}$$

For $s = s_2$, $\mathbb{P}(s_2, a \mid u_{a_1}) = 0$, so $\mathbb{P}(u_{a_1} \mid s_2, a) = 0$. On the other hand, for s_1, a_1, we have the following.

$$\mathbb{P}_{\pi_b,h}(u_{a_1} \mid s_1, a_1) = \frac{\frac{1}{2^{h-1}}}{\frac{1}{2H}\left(1 - \frac{1}{2^{h-1}}\right) + \frac{1}{2^{h-1}}} \leq \min\left(1, \frac{2H}{2^{h-1}}\right)$$

Thus, $\mathbb{P}_{\pi_b,h}(u_0 \mid s_1, a_1) \geq 1 - \frac{2H}{2^{h-1}}$

Thus, for s_1, a_1, the update rule is given by

$$\begin{aligned}\hat{f}_h(s_1,a_1) &= 1 + \frac{1}{H}\mathbb{P}_{\pi_b,h}(u_0 \mid s_1,a_1)\hat{f}_{h+1}(s_1,a_1) + \left(1-\frac{1}{H}\right)\mathbb{P}_{\pi_b,h}(u_0 \mid s_1,a_1)\hat{f}_{h+1}(s_2,a_1) \\ &\quad + \mathbb{P}_{\pi_b,h}(u_{a_1} \mid s_1,a_1)\hat{f}_{h+1}(s_1,a_1) \\ &\leq 1 + \left(\frac{1}{H} + \min\left(1, \frac{2H}{2^{h-1}}\right)\right)\hat{f}_{h+1}(s_1,a_1) + \left(1-\frac{1}{H}\right)\hat{f}_{h+1}(s_2,a_1)\end{aligned}$$

For s_2, it is given by

$$\begin{aligned}\hat{f}_h(s_2,a_1) &= 1 + \frac{1}{H}\mathbb{P}_{\pi_b,h}(u_0 \mid s_1,a_1)\hat{f}_{h+1}(s_1,a_1) + \left(1-\frac{1}{H}\right)\mathbb{P}_{\pi_b,h}(u_0 \mid s_1,a_1)\hat{f}_{h+1}(s_2,a_1) \\ &\quad + \mathbb{P}_{\pi_b,h}(u_{a_1} \mid s_1,a_1)\hat{f}_{h+1}(s_1,a_1) \\ &= \frac{1}{H}\hat{f}_{h+1}(s_1,a_1) + \left(1-\frac{1}{H}\right)\hat{f}_{h+1}(s_2,a_1)\end{aligned}$$

We can use these to perform a straightforward but tedious calculation and inductively verify that for $h \geq 2\log(H)+6$, $\hat{f}_h(s_1,a_1) \leq 1 + \frac{2H-2h}{H}$ and $\hat{f}_h(s_2,a_1) \leq \frac{2H-2h}{H}$. Induction starts at $h = H$ and works backwards. For $h \leq 2\log(H)+6$, we use the simple upper bounds on the FQE recursion.

$$\hat{f}_h(s_1,a_1) \leq 1 + \max(\hat{f}_{h+1}(s_1,a_1), \hat{f}_{h+1}(s_2,a_1))$$

$$\hat{f}_h(s_2,a_1) \leq \max(\hat{f}_{h+1}(s_1,a_1), \hat{f}_{h+1}(s_2,a_1))$$

In particular,

$$\max(\hat{f}_h(s_1,a_1), \hat{f}_h(s_2,a_1)) \leq 1 + \max(\hat{f}_{h+1}(s_1,a_1), \hat{f}_{h+1}(s_2,a_1))$$

This gives us the following relation.

$$\hat{f}_1(s_1,a_1) \leq \max(\hat{f}_1(s_1,a_1), \hat{f}_1(s_2,a_1)) \leq (2\log H+6)+1+\frac{2(H-(2\log H+6))}{H} \leq 2\log H+9$$

In particular, FQE gives an underestimate of the value and its estimation error is

$$V_1^{\pi_e}(s_1) - \sum_a \pi_e(a \mid s)\hat{f}_1(s_1,a) = O(H)$$

□

B.8 Proof of Consistency for Clustering OPE, Theorem 4.2.7

We first rephrase the end-to-end clustering guarantee from Kausik et al. [2022] in our context.

Theorem. *Under Assumptions 4, 6, and 7, there are constants H_0, N_0 depending polynomially on $\frac{1}{\alpha}, \Delta, \frac{1}{\min_u P(u)}, \log(1/\delta)$, so that for $n \geq U^2 S N_0 \log(1/\delta)$ trajectories of length $H \geq H_0 t_{mix} \log(n)$, we recover all clusters of trajectories exactly with probability at least $1-\delta$.*

We now recall Theorem 4.2.7.

Theorem 4.2.7 (Sample Complexity for OPE under Global Confounding)**.** *Under Assumptions 4, 6, 7, 8, there are constants H_0, N_0 depending polynomially on $\frac{1}{\alpha}, \Delta, \frac{1}{\min_u P(u)}, \log(1/\delta)$, so that for n trajectories of length $H \geq H_0 t_{mix} \log(n)$, we have that $|\hat{V}_1(s_0;\pi_e) - V_1(s_0;\pi_e)| < \epsilon$ with probability at least $1-\delta$ if $n \geq \Omega(\max(n_1, n_2, n_3, n_4))$, where*

$$n_1 := U^2 S N_0 \log(1/\delta), n_2 := \frac{\log(U/\delta)}{\min(\epsilon^2/H^2, \min_u P(u)^2)}$$
$$n_3 := \frac{H^2 \tau_a \tau_s S A \log(U/\delta)}{\epsilon^2}, n_4 := \frac{\tau_a H}{d_m}$$

As discussed in Section 4.2.5, we prove a more general version of this, in the form of the theorem below. Assume that we instantiate Algorithm 7 with an OPE estimator that requires an assumption $A(b)$ parameterized by a vector b and has sample complexity $N_2(\delta, \epsilon, b)$.

Theorem. *Under Assumptions 4, 6, 7, and $A(b)$, there are constants H_0, N_0 depending polynomially on $\frac{1}{\alpha}, \Delta, \frac{1}{\min_u P(u)}, \log(1/\delta)$, so that for n trajectories of length $H \geq H_0 t_{mix} \log(n)$, we have that $|\hat{V}_1(s_0;\pi_e) - V_1(s_0;\pi_e)| < \epsilon$ with probability at least $1-\delta$ if*

$$n \geq \Omega\left(\max\left(U^2 S N_0 \log(1/\delta), \frac{\log(U/\delta)}{\min(\epsilon^2/H^2, \min_u P(u)^2)}, N_2(\delta/U, \epsilon, b)\right)\right).$$

Proof. Note that $V_1(s_0;\pi_e) = \mathbb{E}_u[V_1(s_0;u,\pi_e)] = \sum_u P(u) V_1(s_0;u,\pi_e)$. Using the clustering guarantee from Kausik et al. [2022] (rephrased above), we know that for the same H_0 and N_0 as in the clustering guarantee, given $n \geq N(\delta) = U^2 S N_0 \log(1/\delta)$ trajectories of length $H \geq H_0 t_{mix} \log(n)$, we recover clusters $C_1, ..., C_U$ consisting of trajectories with the same confounders with probability at least $1-\delta$. Recall that H_0 is not explicitly dependent on S, A and t_{mix}, but could depend on the model.

We only identify the confounder labels in each trajectory up to permutation upon obtaining exact clustering, but for any permutation $\sigma \in S_U$, $\sum_{u=1}^{K} P(u) V_1(s_0; C_u, \pi_e) = \sum_{u=1}^{U} P(\sigma(u)) V_1(s_0; C_{\sigma(u)}, \pi_e)$. That is, the result of the sum is independent of the order of its terms $P(u)\hat{V}_1(s_0; C_u, \pi_e)$. So, we assume WLOG that we recover the true cluster labels.

Upon obtaining the confounder labels u_n in each trajectory, we can estimate $P(u)$ with $\hat{P}(u) := \frac{1}{N_{traj}} \sum_n \mathbf{1}(u_n = u)$ via label proportions. By a simple application of Hoeffding's inequality, there is another function $N_1(\delta, \alpha)$ so that for $n \geq N_1(\delta/U, \alpha)$, the weights satisfy $|\hat{P}(u) - P(u)| \leq \alpha$ for all u with probability at least $1 - \delta$.

We use $|ab - cd| \leq |b||a - c| + |c||b - d|$ to conclude that for $n \geq N_1(\delta/U, \epsilon/2H)$, we have the following bound with probability at least $1 - \delta$.

$$|V_1(s_0; \pi_e) - \hat{V}_1(s_0; \pi_e)| \leq \frac{\epsilon}{2H} \max_u \hat{V}_1(s_0; C_u, \pi_e) + \max_u (P(u)|\Delta(u)|) \leq \frac{\epsilon}{2} + \max_u |\Delta(u)| \quad \text{(B.8)}$$

where $\Delta(u) := V_1(s_0; C_u, \pi_e) - \hat{V}_1(s_0; C_u, \pi_e)$.

So, whenever we have exact clustering, there is a function $N_2(\delta, \epsilon, b)$ so that $|\Delta(u)| < \epsilon$ for all u outside of a set of probability δ whenever $\sum_n \mathbb{1}(u_n = u) \geq N_2(\delta/U, \epsilon, b)$. By Hoeffding's inequality from above, $\sum_n \mathbb{1}(u_n = u) \geq n(P(u) - \alpha) \geq nP(u)/2$ for $\alpha \leq \min_u P(u)/2$.

So, for $n \geq \max\left(N\left(\frac{\delta}{3}\right), N_1\left(\frac{\delta}{3U}, \min\left(\frac{\epsilon}{2H}, \frac{\min_u P(u)}{2}\right)\right), \frac{2}{\min_u P(u)} N_2\left(\frac{\delta}{3U}, \frac{\epsilon}{2}, b\right)\right)$, we get that $|V_1(s_0; \pi_e) - \hat{V}_1(s_0; \pi_e)| \leq \epsilon$

Note that $N(\delta/3) = U^2 S N_0 \log(3/\delta)$ and $N_1\left(\frac{\delta}{3U}, \min\left(\frac{\epsilon}{2H}, \frac{\min_u P(u)}{2}\right)\right) = \frac{2\log(3U/\delta)}{\min(\epsilon^2, \min_u P(u)^2)}$. This gives us our final bound.

□

B.9 The Necessity of the Horizon Being $O(t_{mix})$

We showed in Section 4.2.6 that under Assumptions 4, 6, 7 and 8, Algorithm 7 provides a point estimate of the policy's value with provable sample complexity guarantees. The only additional requirement was that $H \geq H_0 t_{mix} \log n$. We claim that the t_{mix} dependence is not an artifact of the clustering method used. In fact, the theorem below shows that if $H \leq \tilde{O}(t_{mix})$, clustering and value estimation can be arbitrarily bad even when t_{mix} is small. It essentially produces an example with logarithmically small t_{mix} where the confounders cannot be identified for $H \leq \tilde{O}(t_{mix})$. We prove it in Appendix B.9. We state Theorem B.9.1 below.

Theorem B.9.1 (Necessity of $H \geq \Omega(t_{mix})$)**.** *There exist globally confounded MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ and a behavior policy π_b with induced mixing time $t_{mix} = O(\log S)$ so that for $H \leq \tilde{O}(t_{mix})$, trajectories from confounders in both MDPs have the same distribution. Furthermore, there exists a stationary evaluation policy π_e and a starting state s so that $|V_1^{\pi_e}(s, \mathcal{M}_1) - V_1^{\pi_e}(s, \mathcal{M}_2)| = \Omega(H)$.*

Proof. We construct two MDPs which satisfy all our assumptions, but have the same distribution

over a horizon less than t_{mix} and thus cannot be distinguished. We will also note that given the reward structure, under a different starting distribution, the MDPs will have value functions differing by $O(H)$.

The intuition is that the state space is an n-dimensional Boolean hypercube with an extra rewarding state s_r, thought of as a "twin" to $(1, 1, \dots 1)$. If one identifies s_r to $(1, 1, \dots 1)$, then $a = 1$ pushes states to have more ones while $a = 2$ pushes states to have more zeros, and the actions taken with probability $1/2$ combine to produce a lazy random walk on the Boolean hypercube. Depending on which MDP one is in, s_r and $(1, 1, \dots 1)$ have proportional transition dynamics, with different levels of "traffic." Controlling this "traffic" allows us to control the rewards of a different evaluation policy in the MDPs, because we choose all states besides s_r to have 0 reward.

Environments:

- Consider $S = \{0, 1\}^n \cup \{s_r\}$, $A = \{1, 2\}$, $U = \{1, 2\}$, horizon H.
- Rewards: $r(s = s_r, a) = 1$ for any action a, otherwise 0 reward.
- Starting state: Let the starting state be $(0, 0 \dots 0)$.
- Confounders: $\mathbb{P}(u = 1) = \mathbb{P}(u = 2) = \frac{1}{2}$.

Transitions: We describe the transition structure below. Pick a parameter $p_{i,j} \in [0, 1]$ for MDP $\mathcal{M}_i$ and confounder $u = j$, whose role will be clear below. For both MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ and both confounders $u = 1, 2$, consider the following transition structure.

- Under $a = 1$: Consider $s \neq s_r$, and let it have $k > 1$ zeros. Pick one of the zeros with probability $\frac{1}{n}$ each and change it to a 1, doing nothing and staying in s with probability $\frac{n-k}{n}$. If s has exactly 1 zero, then for MDP $\mathcal{M}_i$ and confounder $u = j$, let s transition to s_r with probability $\frac{p_{i,j}}{n}$, to $(1, 1, \dots 1)$ with probability $\frac{1-p_{i,j}}{n}$ and stay at s with probability $1 - \frac{1}{n}$. Fix $p_{2,1} = p_{2,2} = \frac{1}{2}$. If $s = s_r$, then in $\mathcal{M}_i$ and confounder u_j, move to $(1, 1, \dots 1)$ with probability $1 - p_{i,j}$, staying with probability $p_{i,j}$. If $s = (1, 1, \dots 1)$, then in $\mathcal{M}_i$ and confounder u_j, move from to s_r with probability $p_{i,j}$, staying with probability $1 - p_{i,j}$.
- Under $a = 2$: Consider $s \neq s_r$, and let it have $k > 0$ zeros. Pick one of the ones with probability $\frac{1}{n}$ each and change it to a zero, doing nothing and staying in s with probability $\frac{k}{n}$. If $s = s_r, (1, 1, \dots 1)$, then let it transition to a state with a single zero with probability $\frac{1}{n}$.

Behavior policies: In both MDPs, choose the same policy $\pi(a \mid s) = \frac{1}{2}$ for all a, s. One can check that the occupancies of s_r and $(1, 1, \dots 1)$ are only non-zero together and always have the ratio $p_{i,j}/(1 - p_{i,j})$ in MDP $\mathcal{M}_i$ and confounder $u = j$. This will thus also hold in the stationary

distribution. Note that while in general, identifying states in a Markov chain does not create a Markov chain, this is true if two states always have the same ratio of occupancies. Additionally, since the occupancy ratios are fixed, for any MDP and confounder in our system, the TV distance between the distribution of the system and at any time t from the stationary distribution is the same if we identified s_r and $(1, 1, \dots 1)$. Thus, this system has the same mixing time as it would if we identified s_r and $(1, 1, \dots 1)$.

Notice that the transition structure of the induced Markov chains in both MDPs after identifying s_r and $(1, 1, \dots 1)$ is identical, and in fact it is the same as picking a bit in a state uniformly at random and flipping it with probability $1/2$, doing nothing otherwise. This is in fact the same as the lazy random walk on the Boolean hypercube in Levin and Peres [2017]. We thus know from Levin and Peres [2017] that both induced Markov chains have the same mixing time $t_{mix} = O(n \log n)$. Let k be a constant so that $t_{mix} \leq kn \log n$.

Observational indistinguishability: Consider $H \leq \frac{t_{mix}}{4k \log(t_{mix})} \leq \frac{n}{4}$. Since the MDPs have identical transition structures for $s \neq s_r$ with s having 2 or more zeros, and no state can have fewer than 2 zeros after less than $\frac{n}{4}$ bit flips starting from the starting state $(0, 0 \dots 0)$, trajectories generated under either MDP and either confounder have the same probability.

In particular, the confounders are observationally indistinguishable in either MDP and cannot be clustered even with infinite observations, even though transitions differ in $n + 2$ of the states with $\Delta > \max(|1 - 2p_{i,j}|, |p_{i,1} - p_{i,2}|) > 0$. Moreover, the MDPs themselves are observationally indistinguishable as well.

Evaluation policy: One can produce many examples of an evaluation policy π_e so that there is a state s with $V^{\pi_e}_{1,i}(s)$ very different across the two MDPs. Here we present a trivial one. Consider $\pi_e(a = 1 \mid s, u) = 1$ for all s, u.

Policy values: Let us say that we intend to find $V^{\pi_e}_{1,i}((1, 1, \dots 1))$. Notice that in the first step in confounder $u = j$ and MDP $\mathcal{M}_i$, the distribution of states will be $\mathbb{P}(s_r) = p_{i,j}$ and $\mathbb{P}((1, 1, \dots 1)) = 1 - p_{i,j}$ and stays that way for all future steps. This means that $V^{\pi_e}_{1,i}((1, 1, \dots 1)) = \left(\sum_{j=1}^{2} \frac{p_{i,j}}{2}\right)(H - 1)$ in MDP $\mathcal{M}_i$.

The difference in values is given by $|V^{\pi_e}_{1,1}((1, 1, \dots 1)) - V^{\pi_e}_{1,2}((1, 1, \dots 1))| = (H - 1)(p_{1,1} + p_{1,2} - p_{2,1} - p_{2,2})$. We arbitrarily instantiate our parameters to be say $p_{1,1} = 1 - \frac{1}{100}$, $p_{1,2} = 1 - \frac{2}{100}$, $p_{2,1} = -\frac{1}{100}$, $p_{2,2} = \frac{2}{100}$, to get that

$$|V^{\pi_e}_{1,1}((1, 1, \dots 1)) - V^{\pi_e}_{1,2}((1, 1, \dots 1))| = \frac{94}{100}(H - 1) = \Omega(H)$$

□

B.10 Policy Optimization under General and Memoryless Confounders

B.10.1 Bounds on Sub-optimality given Optimization Oracles

Here, we elaborate on the comment at the beginning of Section 4.2.7, where we claim that given error bounds on our value estimate $\hat{V}_1$ and an optimizer for $\hat{V}_1$, we can get suboptimality bounds for the output of the optimizer. Notice the slight change in notation below.

Lemma B.10.1. *Fix an arbitrary starting distribution d_0. If for any policy π, $|\hat{V}_1(\pi) - V_1(\pi)| \leq \epsilon$, then for $\hat{\pi}^* = \arg\max_\pi \hat{V}_1(\pi)$ and $\pi^* = \arg\max_\pi V_1(\pi)$, we have that $0 \leq V_1(\pi^*) - V_1(\hat{\pi^*}) \leq 2\epsilon$.*

Proof. Consider the following chain of inequalities.

$$\begin{aligned} & V_1(\pi^*) - V_1(\hat{\pi}^*) \\ &= V_1(\pi^*) - \hat{V}_1(\pi^*) + \hat{V}_1(\pi^*) - \hat{V}_1(\hat{\pi}^*) + \hat{V}_1(\hat{\pi}^*) - V_1(\hat{\pi}^*) \\ &\leq \epsilon + 0 + \epsilon \end{aligned}$$

Here, the first part of the last inequality holds by our assumption applied to $\pi = \pi^*$, while the second part holds by the definition of $\hat{\pi}^*$ as the optimal policy for $\hat{V}_1$. The third part holds by applying our assumption to $\pi = \pi^*$.

Finally, by the definition of π^* as the optimal policy for V_1, $V_1(\pi^*) - V_1(\hat{\pi^*}) \geq 0$. Combining these, we have our results. □

B.10.2 Gradient Ascent on the Lower Bound

Algorithm 19 Gradient Ascent on Differentiable Lower Bounds for Policy Improvement under Confounding

1: **input:** decaying learning rate η_t, π_θ.
2: **for** $t = 1, ..., N$ **do**
3: **run subroutine:** obtain differentiable lower bound $V_1(s_0; \pi_\theta)$ on π_θ via Alg. 6, Alg. 17, or Alg. 5
4: **update:** $\theta \leftarrow \theta + \eta_t \cdot \nabla_\theta V_1(s_0; \pi_\theta)$
5: **end for**
6: **return** π_θ

This enjoys the following elementary local convergence guarantees.

Lemma B.10.2. *If $\nabla_\theta V_1(s_0;\pi_\theta,\mathbb{P})$ and $\nabla_{\mathbb{P}} V_1(s_0;\pi_\theta,\mathbb{P})$ are Lipschitz, every local max-min is a gradient ascent/descent stable point.*

Lemma B.10.3. *If $V_1(s_0;\pi_\theta,\mathbb{P})$ is twice differentiable with a Lipschitz continuous gradient, its saddle points are a strict-saddle, and one waits for the inner minimization to converge in each iteration, in the limit of infinite trajectories the procedure converges to a local maxima of $V_1(s_0;\pi_\theta,\mathbb{P})$.*

The first result follows from Section 2 in Daskalakis and Panageas [2018], given the knowledge that $V_1(s_0;\pi_\theta,\mathbb{P})$, being constructed from translations and matrix multiplications, is smooth, and therefore so are its gradients. The second result follows from Lee et al. [2016].

B.11 Policy Optimization under Global Confounders

Algorithm 20 Clustering-Based Policy Gradient

1: **input:** Number of clusters U, clustering algorithm `cluster()`, offline policy gradient estimator `gradient()`, learning rate η, initial policy parameters θ_0.
2: **run subroutine:** Perform clustering on trajectories with clustering algorithm `cluster()`, obtain clusters $C_1, ..., C_K$.
3: Obtain cluster weight estimates $\hat{P}(u) := \frac{|C_u|}{N_{traj}}$.
4: **for** $t = 1, ..., T$: **do**
5: **run subroutine:** Use offline policy gradient estimator `gradient()` to estimate $Z_i(\theta_t) = \nabla_\theta V_1(s_0; u_i, \pi_{\theta_t})$ for each cluster C_i, obtaining $\hat{Z}_i(\theta_t)$.
6: Obtain gradient estimate of $Z(\theta_t) = \nabla_\theta V_1(s_0;\pi_{\theta_t})$ with $\hat{Z}(\theta_t) = \sum_{u=1}^{U} \hat{P}(u_i)\hat{Z}_i(\theta_t)$.
7: Update $\theta_{t+1} := \theta_t - \eta\hat{Z}(\theta_t)$.
8: **end for**
9: **return:** Output the final policy $\pi_{\theta_{T+1}}$.

We now recall Theorem 4.2.8 below. We remind the reader that like Theorem 4.2.7, the theorem below holds when $H \geq H_0 t_{mix} \log n$.

Theorem 4.2.8. *Let us have large enough $\beta > 1$ and $T = n^\beta$, for $n \geq \Omega\left(\max\left(U^2 S N_0 \log(1/\delta), \frac{\log(U/\delta)}{\min_u P(u)^2}\right)\right)$. Also let $H \geq H_0 t_{mix} \log n$, for H_0, N_0 as in Theorem 4.2.7. Then we have that $\frac{1}{T}\sum_{t=1}^{T} ||\nabla_\theta V_1(s_0;\pi_{\theta_t})||^2 = O(\max(\epsilon_{MSE}, \epsilon_{freq})$, where $\epsilon_{MSE} = \frac{H^4 \log(nU/\delta)}{n \min_u P(u)}$, and $\epsilon_{freq} = \frac{L^2 \log(U/\delta)}{n}$*

To prove this, we first provide a high-probability guarantee for the overall gradient estimate across all clusters analogous to that of Theorem 4.2.7 for OPE. This is proved in Section B.11.1.

Theorem B.11.1. *When Assumptions 4, 6, 7 and 8 are satisfied, there are constants H_0, N_0 depending polynomially on $\frac{1}{\alpha}, \Delta, \frac{1}{\min_u P(u)}, \log(1/\delta)$, so that for n trajectories of length $H \geq$*

$H_0 t_{mix} \log(n)$*, if we use the EOPPG offline policy gradient estimator from Kallus and Uehara [2020],*

$$n \geq \max\left(U^2 S N_0 \log(3/\delta), \frac{8\log(6U/\delta)}{\min\{\epsilon^2/L^2, \min_u P(u)^2\}}, \frac{C}{\min_u P(u)} \frac{H^4 \log(nU/\delta)}{\epsilon^2}\right)$$

then $||Z(\theta) - \hat{Z}(\theta)|| \leq \epsilon$ *with probability* $1 - \delta$ *for some constant* C*.*

The following result for the convergence of unconstrained gradient descent is effectively Theorem 11 in Kallus and Uehara [2020], combined with the bound in Theorem B.11.1. We repeat the proof in Section B.11.2 for completeness.

Theorem B.11.2. *Assume* $V_1(s_0; u, \pi_\theta)$ *and* $V_1(s_0; \pi_\theta)$ *are differentiable and* M*-smooth in* θ *for all* $u \in U$*, and the learning rate* $\eta < \frac{1}{4M}$*. Then, if the number of trajectories* n *satisfies the condition in Theorem B.11.1, the iterates* θ_t *from Algorithm 9 offer*

$$\frac{1}{T}\sum_{t=1}^{T} ||\nabla_\theta Z(\theta_t)||^2 = \frac{1}{T}\sum_{t=1}^{T} ||\nabla_\theta V_1(s_0; \pi_{\theta_t})||^2 \leq \frac{4}{\eta T}\left(V_1(s_0; \pi_{\theta^*}) - V_1(s_0; \pi_{\theta_1})\right) + 3\epsilon^2$$

The result of Theorem 4.2.8 then follows immediately from the two results above. The only additional observation needed is that since V_1 is Lipschitz, it is bounded in a compact domain and so the first term in Theorem B.11.2 is $O(1/n^\beta) \leq O(1/n)$.

Another Formulation of Policy Optimization Notice that the nature of the global confounder assumption permits another kind of policy optimization. One can optimize U different policies, one for each value of the confounder, with standard off-policy improvement methods. To deploy them, one will have to identify the confounder online, which is a nontrivial problem in itself. One avenue is to first deploy each of the U behavior policy components in any order for $O(t_{mix})$ time each, and then attempt to identify the confounder using the classification algorithm in Kausik et al. [2022]. If the classification algorithm successfully classifies trajectories generated in this way, we can achieve the optimal reward thereafter by deploying the optimal policy for the confounder in question.

B.11.1 Proof of Theorem B.11.1

Proof. Note that $\nabla_\theta V_1(s_0; \pi_\theta) = \mathbb{E}_u[\nabla_\theta V_1(s_0; u, \pi_\theta)] = \sum_u P(u) \nabla_\theta V_1(s_0; u, \pi_\theta)$.

Using the clustering guarantee from Kausik et al. [2022] rephrased in Section B.8, we know that there are numbers N_0 and H_0 so that given $n \geq U^2 S N_0 \log(1/\delta)$ trajectories of length $H \geq$

$H_0 t_{mix} \log(n)$, we recover clusters $C_1, ..., C_U$ consisting of trajectories with the same confounders with probability at least $1 - \delta$. Recall that N_0 and H_0 are not explicitly dependent on S, A and t_{mix}, but could depend on the model.

Write $Z(\theta) = \nabla_\theta V_1(s_0; \pi_\theta)$, $Z_i(\theta) = \nabla_\theta V_1(s_0; u_i, \pi_\theta)$ and $\hat{Z}_i(\theta)$ for the estimate of $Z_i(\theta)$ and $\hat{Z}(\theta) = \sum_{i=1}^{U} \hat{P}(u_i)\hat{Z}_i(\theta)$ for the estimate of $Z(\theta)$. We only identify the confounder labels in each trajectory up to permutation upon obtaining exact clustering, but as above we assume WLOG that we recover the true cluster labels.

Estimate $P(u)$ with $\hat{P}(u) := \frac{1}{N_{traj}} \sum_n \mathbf{1}(u_n = u)$ via label proportions. By a simple application of Hoeffding's inequality and the union bound, for $n \geq \frac{2\log(2U/\delta)}{\alpha^2}$, the weights satisfy $|\hat{P}(u) - P(u)| \leq \alpha$ with probability at least $1 - \delta$.

We can then bound

$$||Z(\theta) - \hat{Z}(\theta)|| = \left\| \sum_{i=1}^{U} \Big(P(u_i) Z_i(\theta) - \hat{P}(u_i)\hat{Z}_i(\theta) \Big) \right\| \tag{B.9}$$

$$= \sum_{i=1}^{U} \left\| P(u_i) Z_i(\theta) - \hat{P}(u_i)\hat{Z}_i(\theta) \right\| \tag{B.10}$$

$$\leq \sum_{i=1}^{U} ||Z_i(\theta)|| (P(u_i) - \hat{P}(u_i)) + \hat{P}(u_i)||Z_i(\theta) - \hat{Z}_i(\theta)|| \tag{B.11}$$

$$= \sum_{i=1}^{U} ||Z_i(\theta)|| (P(u_i) - \hat{P}(u_i)) + \sum_{i=1}^{U} \hat{P}(u_i)||Z_i(\theta) - \hat{Z}_i(\theta)|| \tag{B.12}$$

$$\leq \alpha \sum_{i=1}^{U} ||Z_i(\theta)|| + \sum_{i=1}^{U} \hat{P}(u_i)||Z_i(\theta) - \hat{Z}_i(\theta)|| \tag{B.13}$$

$$\leq \alpha \sum_{i=1}^{U} ||Z_i(\theta)|| + \sum_{i=1}^{U} 2P(u_i)||Z_i(\theta) - \hat{Z}_i(\theta)|| \tag{B.14}$$

where the second inequality holds with high probability and the last inequality holds for sufficiently small α. If all $||Z_i(\theta) - \hat{Z}_i(\theta)|| \leq \epsilon/4$ for some $\epsilon > 0$, then we would have $||Z(\theta) - \hat{Z}(\theta)|| \leq \sum_{i=1}^{U} 2P(u_i)||Z_i(\theta) - \hat{Z}_i(\theta)|| \leq \epsilon/2$.

It remains to bound the error of each $\hat{Z}_i$. Notice that the result of Theorem 7 in Kallus and Uehara [2020] is independent of the gradient update rule or the value of θ and only depends on the number of samples used to estimate $\hat{Z}^{EOPPG}$. So, it also holds for $\hat{Z}_i$ with n_i samples. Additionally, note that the proof of Theorem 12 in Kallus and Uehara [2020] only uses the supremum of the error over all possible values of θ and does not use any facts about the gradient update, it follows

verbatim for $\hat{Z}_i$ with n_i samples. In particular, with probability at least $1 - \delta/U$,

$$\|Z_i(\theta) - \hat{Z}_i(\theta)\|^2 \leq O\left(\frac{H^4 \log(TU/\delta)}{n_i}\right)$$

and so for $T = n^\beta$, we need $n_i \geq \Omega\left(\frac{H^4 \log(nU/\delta)}{\epsilon^2}\right)$ trajectories for $||Z_i(\theta) - \hat{Z}_i(\theta)|| \leq \epsilon$ to hold for all u_i with probability $1 - \delta$. To convert this into a bound for n, we use Hoeffding's inequality from above in a similar way to the previous proof to find $n_i = \sum_n \mathbb{1}(u_n = u) \geq n(P(u) - \alpha) \geq nP(u)/2$ for $\alpha \leq \min_u P(u)/2$. We therefore need $n \geq \Omega\left(\frac{1}{\min_u P(u)} \frac{H^4 \log(nU/\delta)}{\epsilon^2}\right)$ for the error of each Z_i to be bounded by ϵ with probability $1 - \delta$.

We then bound $\alpha \sum_{i=1}^{U} ||Z_i(\theta)|| \leq \epsilon/2$. Let L be a uniform bound over $\theta \in \Theta$ on the magnitude of the gradients $Z(\theta)$ (in the continuous case, this corresponds to a Lipschitz-type assumption on the value functions). It then suffices to require $\alpha \leq \frac{\epsilon}{2L}$.

Splitting the failure probability into $\delta/3$, requiring $\alpha \leq \min_u P(u)/2, \epsilon/2L$, and bounding the error of each Z_i by $\epsilon/4$, we get $||Z(\theta) - \hat{Z}(\theta)|| \leq \epsilon$ with probability $1 - \delta$ when

$$n \geq \Omega\left(\max\left(U^2 S N_0 \log(1/\delta), \frac{\log(U/\delta)}{\min\{\epsilon^2/L^2, \min_u P(u)^2\}}, \frac{1}{\min_u P(u)} \frac{H^4 \log(nU/\delta)}{\epsilon^2}\right)\right) \tag{B.15}$$

□

B.11.2 Proof of Theorem B.11.2

Proof. The result is largely analogous to Theorem 11 from Kallus and Uehara [2020], and in fact, we can transform our problem into theirs and follow their proof.

Assume $V_1(s_0; u, \pi_\theta)$ and $V_1(s_0; \pi_\theta)$ are differentiable and M-smooth in θ for all $u \in U$. Let $f(\theta) = -V_1(s_0; \pi_\theta)$, and $f_i(\theta) = -V_1(s_0; u_i, \pi_\theta)$ for each u_i. For simplicity, fix the learning rate for all time steps to be some $\eta < \frac{1}{4M}$. By M-smoothness,

$$f(\theta_{t+1}) \leq f(\theta_t) + \langle \nabla f(\theta_t, \theta_{t+1} - \theta_t \rangle + \frac{M}{2} ||\theta_{t+1} - \theta_t||^2.$$

Define $B_{it} = \hat{Z}_i(\theta) - Z_i(\theta)$ for confounder u_i, $B_t = \hat{Z}(\theta) - Z(\theta)$, $w_i = \hat{P}(u_i)$. Observe that

$$\theta_{t+1} = \theta_t - \eta \nabla f(\theta_t) - \eta B_t = \theta_t - \eta \sum_i \nabla w_i f(\theta_t) - \eta \sum_i w_i B_{it}.$$

Then, similarly to the proof in Kallus and Uehara [2020],

$$f(\theta_t) - f(\theta_{t+1}) \geq -\langle \nabla f(\theta_t), \theta_{t+1} - \theta_t \rangle - \frac{M}{2}||\theta_{t+1} - \theta_t||^2 \tag{B.16}$$

$$= \eta \langle \nabla f(\theta_t), \nabla f(\theta_t) - B_t \rangle - \frac{\eta^2 M}{2}||\nabla f(\theta_t) - B_t||^2 \tag{B.17}$$

$$= \eta ||\nabla f(\theta_t)||^2 + \eta \langle \nabla f(\theta_t), B_t \rangle - \frac{\eta^2 M}{2}||\nabla f(\theta_t) - B_t||^2 \tag{B.18}$$

$$\geq \eta ||\nabla f(\theta_t)||^2 - \eta |\langle \nabla f(\theta_t), B_t \rangle| - \frac{\eta^2 M}{2}||\nabla f(\theta_t) - B_t||^2 \tag{B.19}$$

$$\geq \eta ||\nabla f(\theta_t)||^2 - 0.5\eta(||\nabla f(\theta_t)||^2 + ||B_t||^2) - \eta^2 M||\nabla f(\theta_t) - B_t||^2 \tag{B.20}$$

$$\geq 0.25\eta ||\nabla f(\theta_t)||^2 - 0.5\eta||B_t||^2 - 0.25\eta||B_t||^2 \tag{B.21}$$

where the second-last inequality uses the parallelogram law and the last inequality uses the fact that $\eta < \frac{1}{4M}$. We then obtain

$$f(\theta_t) - f(\theta_{t+1}) + 0.75\eta||B_t||^2 \geq 0.25\eta||\nabla f(\theta_t)||^2.$$

Similarly, by a telescoping sum,

$$(f(\theta_1) - f(\theta^*))/T + \frac{0.75\eta}{T}\sum_t ||B_t||^2 \geq \frac{0.25\eta}{T}\sum_t ||\nabla f(\theta_t)||^2,$$

$$(V_1(s_0; \pi_{\theta^*}) - V_1(s_0; \pi_{\theta_1}))/T + \frac{0.75\eta}{T}\sum_t ||B_t||^2 \geq \frac{0.25\eta}{T}\sum_t ||\nabla f(\theta_t)||^2,$$

$$\frac{\eta}{T}\sum_t ||\nabla f(\theta_t)||^2 \leq \frac{4}{T}(V_1(s_0; \pi_{\theta^*}) - V_1(s_0; \pi_{\theta_1})) + \frac{3\eta}{T}\sum_t ||B_t||^2,$$

$$\frac{1}{T}\sum_t ||\nabla f(\theta_t)||^2 \leq \frac{4}{\eta T}(V_1(s_0; \pi_{\theta^*}) - V_1(s_0; \pi_{\theta_1})) + \frac{3}{T}\sum_t ||B_t||^2,$$

$$\frac{1}{T}\sum_t ||\nabla f(\theta_t)||^2 \leq \frac{4}{\eta T}(V_1(s_0; \pi_{\theta^*}) - V_1(s_0; \pi_{\theta_1})) + 3\max_t ||B_t||^2,$$

and finally by applying Theorem B.11.1 for an n that fulfills its conditions for some error threshold ϵ, we obtain

$$\frac{1}{T}\sum_t ||Z(\theta_t)||^2 = \frac{1}{T}\sum_t ||\nabla f(\theta_t)||^2 \leq \frac{4}{\eta T}(V_1(s_0; \pi_{\theta^*}) - V_1(s_0; \pi_{\theta_1})) + 3\epsilon^2.$$

□

APPENDIX C

Supplementary Material for Chapter 4

C.1 Lemmas and Discussion

C.1.1 Relation between V_w and V_g

Lemma 5.2.1 (Replacing w with g)**.** *For any history-dependent policy π that selects an action $a_h \sim \pi(\tau[h-1], s_h)$, $V_w(\mathcal{M}, \pi) = V_g(\mathcal{M}, \pi)$ holds for any $\mathcal{M}$.*

Proof. By a slight abuse of notation, the following chain of equalities holds. Here, (i) holds since $r_h(s_h, u_h, a_h)$ is a function of s_h, u_h, a_h. Equation (ii) holds since we have already conditioned on $\tau[h]$, which includes s_h, a_h. Equation (iii) holds by the definition of $g_h(\tau[h])$ as the conditional distribution of u_h given $\tau[h]$.

$$
\begin{aligned}
V_w(\mathcal{M}, \pi) &= \mathbb{E}_{\tau^u \sim \mathbb{P}^{w,\pi}} \left[\sum_{h \in \mathcal{H}_p} r_h(s_h, u_h, a_h) \right] \\
&= \sum_{h \in \mathcal{H}_p} \mathbb{E}_{\tau^u \sim \mathbb{P}^{w,\pi}} \left[r_h(s_h, u_h, a_h) \right] \\
&= \sum_{h \in \mathcal{H}_p} \mathbb{E}_{\tau^u[h] \sim \mathbb{P}^{w,\pi}} \left[r_h(s_h, u_h, a_h) \right] \\
&= \sum_{h \in \mathcal{H}_p} \mathbb{E}_{\tau[h] \sim \mathbb{P}^{\pi}} \left[\mathbb{E}_{\tau^u[h] \sim \mathbb{P}^{w,\pi}} \left[r_h(s_h, u_h, a_h) \right] \middle| \tau[h] \right] \\
&\overset{(i)}{=} \sum_{h \in \mathcal{H}_p} \mathbb{E}_{\tau[h] \sim \mathbb{P}^{\pi}} \left[\mathbb{E}_{u_h, s_h, a_h \sim \mathbb{P}^{w,\pi}} \left[r_h(s_h, u_h, a_h) \right] \middle| \tau[h] \right] \\
&\overset{(ii)}{=} \sum_{h \in \mathcal{H}_p} \mathbb{E}_{\tau[h] \sim \mathbb{P}^{\pi}} \left[\mathbb{E}_{u_h \sim \mathbb{P}^{w,\pi}} \left[r_h(s_h, u_h, a_h) \right] \middle| \tau[h] \right] \\
&\overset{(iii)}{=} \sum_{h \in \mathcal{H}_p} \mathbb{E}_{\tau[h] \sim \mathbb{P}^{\pi}} \left[\mathbb{E}_{u_h \sim g_h(\tau[h])} \left[r_h(s_h, u_h, a_h) \right] \right]
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{h \in \mathcal{H}_p} \mathbb{E}_{\tau \sim \mathbb{P}^\pi} \left[\mathbb{E}_{u_h \sim g_h(\tau[h])} \left[r_h(s_h, u_h, a_h) \right] \right] \\
&= \mathbb{E}_{\tau \sim \mathbb{P}^\pi} \left[\sum_{h \in \mathcal{H}_p} \mathbb{E}_{u_h \sim g_h(\tau[h])} \left[r_h(s_h, u_h, a_h) \right] \right] \\
&= V_g(\mathcal{M}, \pi)
\end{aligned}
$$

□

C.1.2 Ignoring Internal Reward-States is Bad for Alignment

We define traditional RL methods as those that output possibly time-dependent Markovian policies. In this section, we provide a toy example showing that there is a PORMDP with good sublinear regret guarantees where any time-dependent Markovian policy has value bounded away from the maximum value. This means that traditional RL methods will always incur linear regret. We hope that this illustrates that RL methods that ignore internal reward-states can be bad for alignment.

Lemma 5.2.2 (Markovian policies are not enough)**.** *There is a* PORMDP *where POR-UCRL and POR-UCBVI achieve* $poly(H, S, A)\sqrt{T}$ *regret, but any Markovian policy is at least* $\frac{1}{4}$*-suboptimal and so any method that outputs Markovian (possibly time-dependent) policies will lead to linear regret.*

Proof. Consider a PORMDP $\mathcal{M}$ in the setting of Chatterji et al. [2021] (see point (ii) below Definition 5.2.2) and set $\mathcal{S} = \{s_1, s_2\}$, $\mathcal{A} = \{a_1, a_2\}$ and $\mathbf{w} = 1 \in \mathcal{R}$. Let the transition matrix be $\mathbb{P}(s' \mid s, a) = \frac{1}{2}$ for all s, a, s'. Let the starting state always be s_1.

Consider the set $\mathcal{T}$ of all trajectories that have a_2 until s_2 appears, and then only have a_1. Choose $\phi(\tau) = \mathbb{1}(\tau \in \mathcal{T})$. The best non-Markovian policy $\pi_\star$ can follow this rule and achieve $\phi(\tau) = 1$ for all $\tau \sim \mathbb{P}^{\pi_\star}$. Thus, $\max_{\pi \in \Pi} V(\mathcal{M}, \pi) = 1$, where Π is given by all history-dependent policies.

On the other hand, consider a Markovian but potentially time-dependent policy π, where $\pi_h(a \mid s)$ denotes the probability of action a in state s at timestep h. If $\pi_1(a_2 \mid s_1) = 0$, then its value is zero. If $\pi_1(a_2 \mid s_1) > 0$, then conditioned on the event that s_2 appears first at timestep 2 (probability $1/2$), the expected total reward is at most $\pi_1(a_2 \mid s_1)(1 - \pi_2(a_2 \mid s_2))$. Conditioned on s_2 appearing first at timestep 3 (probability $1/4$, requiring s_1 at timestep 2), the expected total reward is at most $\pi_1(a_2 \mid s_1)\pi_2(a_2 \mid s_1)$. Conditioned on seeing s_2 first at timestep $h \geq 4$, the expected total reward is certainly at most 1. Using these crude inequalities, we can bound the expected reward of a

Markovian policy π by

$$\frac{\pi_1(a_2 \mid s_1)(1-\pi_2(a_2 \mid s_2))}{2} + \frac{\pi_1(a_2 \mid s_1)\pi_2(a_2 \mid s_1)}{4} + \sum_{h=4}^{H} \frac{1}{2^h}$$
$$\leq \frac{\pi_1(a_2 \mid s_1)(2-\pi_2(a_2 \mid s_2))}{4} + \frac{1}{4} \leq \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$$

This means that the value of any time-dependent Markovian policy is at most $\frac{3}{4}$ and so any time-dependent Markovian policy is at least $\frac{1}{4}$-suboptimal and incurs $\frac{T}{4}$ regret. □

Recall that we defined traditional RL algorithms as those that output (possibly time-dependent) Markovian policies. Clearly, any traditional RL algorithm in this sense will have at least $\frac{T}{4}$ regret, which is linear regret.

C.1.3 The Naive Reduction from Dueling to Cardinal PORRL Fails

Lemma 5.4.1 (Naive Reduction Lower Bound)**.** *Using any algorithm for cardinal PORRL with sublinear cardinal regret on $\overline{\mathcal{M}}$ with policy class $\Pi' := \Pi \times \Pi$ to get a sequence $(\pi_{1,1}, \pi_{2,1}), \ldots, (\pi_{1,T}, \pi_{2,T})$ leads to linear dueling regret for $\mathcal{M}$ whenever all policies π do not have the same value $V(\mathcal{M}, \pi)$.*

Proof. Define $\pi_\star := \arg\max_{\pi \in \Pi} V(\mathcal{M}, \pi)$ and let $\pi_{\min} := \arg\min_{\pi \in \Pi} V(\mathcal{M}, \pi)$. Then note that

$$\begin{aligned}
\max_{\pi, \pi' \in \Pi} V_D(\mathcal{M}, \pi, \pi') &= \max_{\pi, \pi' \in \Pi} V(\mathcal{M}, \pi) - V(\mathcal{M}, \pi') \\
&= \max_{\pi \in \Pi} V(\mathcal{M}, \pi) + \max_{\pi' \in \Pi} [-V(\mathcal{M}, \pi')] \\
&= \max_{\pi \in \Pi} V(\mathcal{M}, \pi) - \min_{\pi' \in \Pi} V(\mathcal{M}, \pi') \\
&= V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{min})
\end{aligned}$$

Under the naive reduction described in Section 5.4, a cardinal PORRL algorithm is used to maximize *dueling feedback*. If the algorithm has sublinear cardinal regret, then it will produce duels $(\pi_{1,t}, \pi_{2,t}), t = 1 \to T$, satisfying

$$\sum_{t=1}^{T} \max_{\pi, \pi' \in \Pi} V_D(\mathcal{M}, \pi, \pi') - V_D(\mathcal{M}, \pi_{1,t}, \pi_{2,t}) = o(T)$$

From above, this means that

$$\sum_{t=1}^{T} [V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{1,t})] + [V(\mathcal{M}, \pi_{2,t}) - V(\mathcal{M}, \pi_{min})] = o(T)$$

Now note that by definition of $\pi_\star$ and π_{min}, both terms are positive. This is the key point. We thus have

$$\sum_{t=1}^{T} V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{1,t}) = o(T)$$
$$\sum_{t=1}^{T} V(\mathcal{M}, \pi_{2,t}) - V(\mathcal{M}, \pi_{min}) = o(T)$$

This means that for dueling regret $\mathrm{Regret}_D(T)$, we have the following.

$$\begin{aligned}
\mathrm{Regret}_D(T) &= \sum_{t=1}^{T} [V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{1,t})] + [V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{2,t})] \\
&= \sum_{t=1}^{T} [V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{1,t})] + [V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{min})] \\
&\quad + \sum_{t=1}^{T} [V(\mathcal{M}, \pi_{min}) - V(\mathcal{M}, \pi_{2,t})] \\
&= o(T) + T\,[V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{min})] \\
&= \Theta(T)
\end{aligned}$$

Where the last line holds since all policies π do not have the same value $V(\mathcal{M}, \pi)$, and so $V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{min}) > 0$. □

C.2 Regret-to-PAC Conversion

When learning in MDPs, we can turn any guarantee on the regret into a corresponding PAC guarantee, the so-called "regret-to-PAC conversion" [Jin et al., 2018, Ménard et al., 2021, Wagenmaker et al., 2022, Tirinzoni et al., 2023]. Similarly, we want to convert guarantees on the cardinal and dueling regret (see Section 5.2) into corresponding PAC guarantees, which are more adherent to an offline setting. We provide distinct results for the cardinal and dueling regret below.

Lemma C.2.1 (Cardinal regret to PAC)**.** *For $T \in \mathcal{N}$ and $\delta \in [0, 1]$, let* ALG *be an algorithm for cardinal PORRL producing a sequence of policies $(\pi_t)_{t\in[T]}$ with cardinal regret bounded with*

probability at least $1-\delta$ *as*

$$\sum_{t=1}^{T} V(\mathcal{M},\pi_\star) - V(\mathcal{M},\pi_t) \leq R(T,\delta) \in \mathcal{R}.$$

Then, a policy $\widehat{\pi}_T \sim \pi_1,\dots,\pi_T$ *sampled uniformly satisfies with probability at least* $1-2\delta$

$$V(\mathcal{M},\pi_\star) - V(\mathcal{M},\widehat{\pi}_T) \leq \frac{R(T,\delta)}{T} + 8Bp\sqrt{\frac{\log(1/\delta)}{T}}.$$

Proof. We consider the sequence of random variables $Y_t = V(\mathcal{M},\pi_\star) - V(\mathcal{M},\pi_t)\ \forall t \in [T]$. Through the Hoeffding's inequality on Y_t and $|r_h| \leq B$ we have

$$\begin{aligned} V(\mathcal{M},\pi_\star) - V(\mathcal{M},\widehat{\pi}_T) &= \mathbb{E}[V(\mathcal{M},\pi_\star) - V(\mathcal{M},\pi_t)] \\ &\leq \frac{1}{T}\sum_{t=1}^{T}\Big(V(\mathcal{M},\pi_\star) - V(\mathcal{M},\pi_t)\Big) + 8Bp\sqrt{\frac{\log(1/\delta)}{T}} \end{aligned}$$

with probability at least $1-\delta$. Then, combining the latter inequality with the upper bound on the regret and a union bound, we get

$$V(\mathcal{M},\pi_\star) - V(\mathcal{M},\widehat{\pi}_T) \leq \frac{R(T,\delta)}{T} + 8Bp\sqrt{\frac{\log(1/\delta)}{T}}$$

with probability at least $1-2\delta$. □

The latter result implies a PAC guarantee of the form $\mathbb{P}(V(\mathcal{M},\pi_\star) - V(\mathcal{M},\widehat{\pi}_T) \geq \epsilon) \leq \delta$ for some $\epsilon > 0$ and $\delta \in [0,1]$ with a number of episodes of order $\widetilde{O}(1/\epsilon^2)$. An analogous result can be stated for the dueling setting.

Lemma C.2.2 (Dueling regret to PAC)**.** *For* $T \in \mathcal{N}$ *and* $\delta \in [0,1]$*, let* ALG *be an algorithm for dueling PORRL producing a sequence of policy pairs* $(\pi_{1,t},\pi_{2,t})_{t\in[T]}$ *with dueling regret bounded with probability at least* $1-\delta$ *as*

$$\sum_{t=1}^{T} V(\mathcal{M},\pi_\star) - \frac{V(\mathcal{M},\pi_{1,t}) + V(\mathcal{M},\pi_{2,t})}{2} \leq R_D(T,\delta) \in \mathcal{R}.$$

Then, a policy $\widehat{\pi}_T \sim \pi_1,\dots,\pi_T$ *sampled uniformly satisfies with probability at least* $1-4\delta$

$$V(\mathcal{M},\pi_\star) - V(\mathcal{M},\widehat{\pi}_T) \leq \frac{R_D(T,\delta)}{T} + 16Bp\sqrt{\frac{\log(1/\delta)}{T}}.$$

Proof. The proof proceeds as in the previous lemma by applying Hoeffding separately on the

sequences $Y_{1,t} = V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{1,t})$ and $Y_{2,t} = V(\mathcal{M}, \pi_\star) - V(\mathcal{M}, \pi_{2,t})$, then applying a union bound. □

C.3 Proofs for General Optimistic Algorithms for Cardinal PORRL

C.3.1 Generic Model-Based Optimism using Confidence-Sets

We present a template to get regret bounds for a *generic model-based optimistic algorithm using confidence sets*, which we will later instantiate into POR-UCRL and also use in our reduction from the dueling PORRL to optimistic algorithms for cardinal PORRL.

A generic algorithm using confidence sets is determined by confidence sets $\mathcal{C}_{\mathcal{M}}(\mathcal{D}, \delta)$ based on a dataset $\mathcal{D}$. Maintaining a running dataset $\mathcal{D}_t$, at each step t, we run π_t given by

$$\pi_t, \widetilde{\mathcal{M}}_t := \underset{\pi \in \Pi, \mathcal{M} \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)}{\arg\max} V(\mathbb{P}, f, \pi)$$

We obtain a trajectory $\tau_t \sim \mathbb{P}_\star^{\pi_t}$ and append it to $\mathcal{D}_t$ to get $\mathcal{D}_{t+1}$, recompute confidence sets $\mathcal{C}_{\mathcal{M}}(\mathcal{D}_{t+1}, \delta)$, and continue. This algorithm is formally presented in Appendix C.3.1 below.

Algorithm 21 Generic Confidence-Set Optimism

1: **Input** Known family of reward functions $\{\mathcal{R}_h\}_{h=1}^H$, known model class $\mathcal{M}$ induced by known probability transition kernel class $\mathcal{P}$ and known decoder-induced function class $\mathcal{F}$, confidence level δ.
2: **Initialize** dataset $\mathcal{D}_1 \leftarrow \{\}$ and $\mathcal{C}_{\mathcal{M}}(\mathcal{D}_1, \delta) \leftarrow \mathcal{M}$.
3: **for** $t = 1, ..., T$ **do**
4: **Compute** the optimistic history dependent policy,

$$\pi_t, \widetilde{\mathcal{M}}_t = \underset{\pi,\ \mathcal{M} \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)}{\arg\max} V(\mathcal{M}, \pi)$$

5: **Observe** trajectory $\tau_t = \{(s_h^t, a_h^t)\}_{h=1}^H$ and feedback $\{o_h\}_{h \in \mathcal{H}_p}$.
6: **Update** $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{\tau_t\}$ and compute new confidence set $\mathcal{C}_{\mathcal{M}}(\mathcal{D}_{t+1}, \delta)$.
7: **end for**

We now make the following assumption about our confidence sets. It essentially controls the effect of shrinking confidence sets for $\mathcal{P}$ and $\mathcal{F}$ on the value. Showing this assumption is the core

of proving regret bounds for any instantiation of this generic algorithm. We will see later that it is satisfied by the confidence sets for POR-UCRL.

Assumption 13 (Controlling Value Error due to Confidence Sets, Refined Version)**.** For a transition kernel $\mathbb{P}_\star$ and function $f_\star$, consider any sequence of policies π_t and datasets $\mathcal{D}_t$ that contain $\{\tau_i\}_{i=1}^t$ generated under $(\mathbb{P}_\star^{\pi_t}, f^\star)$. We require that $\mathcal{M}_\star \in \mathcal{C}_\mathcal{M}(\mathcal{D}_t, \delta)$ for all t with probability $1 - \delta/16$. We require that there exist problem dependent functions $C_P(\mathcal{M}, T, \delta)$ and $C_F(\mathcal{M}, T, \delta)$ so that for arbitrary sequences $(\mathbb{P}_t, f^t) \in \mathcal{C}_\mathcal{M}(\mathcal{D}_t, \delta)$, the following hold with probability $1 - \delta/2$ each.

$$\left|\sum_{t=1}^T V(\mathbb{P}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t)\right| = \widetilde{\mathcal{O}}(C_P(\mathcal{M}, T, \delta))$$

$$\left|\sum_{t=1}^T V(\mathbb{P}_\star, f^t, \pi_t) - V(\mathbb{P}_\star, f_\star, \pi_t)\right| = \widetilde{\mathcal{O}}(C_F(\mathcal{M}, T, \delta))$$

Theorem C.3.1 (Regret for Confidence-Set Optimism)**.** *Under Assumption 13, any generic optimistic algorithm using confidence sets $\mathcal{C}_\mathcal{M}(\mathcal{D}, \delta)$ satisfies the regret bound*

$$\mathrm{Regret}(T) = \widetilde{\mathcal{O}}\left(C_P(\mathcal{M}, T, \delta) + C_F(\mathcal{M}, T, \delta)\right)$$

Proof. Let $\widetilde{\mathcal{M}}_t$ be given by $\widetilde{\mathbb{P}}_t$ and $\widetilde{f}^t$. Note the following inequalities, where (i) holds with probability 1 by the optimistic definition of π_t.

$$\begin{aligned}
\mathrm{Regret}(T) &= \sum_{t=1}^T V(\mathbb{P}_\star, f^\star, \pi_\star) - V(\mathbb{P}_\star, f^\star, \pi_t) \\
&\overset{(i)}{\leq} \sum_{t=1}^T V(\widetilde{\mathbb{P}}_t, \widetilde{f}^t, \pi_t) - V(\mathbb{P}_\star, f^\star, \pi_t) \\
&\leq \sum_{t=1}^T \underbrace{V(\widetilde{\mathbb{P}}_t, \widetilde{f}^t, \pi_t) - V(\mathbb{P}_\star, \widetilde{f}^t, \pi_t)}_{(I)} + \underbrace{V(\mathbb{P}_\star, \widetilde{f}^t, \pi_t) - V(\mathbb{P}_\star, f^\star, \pi_t)}_{(II)}
\end{aligned}$$

We now apply Assumption 13 to bound (I) and (II). We can use the assumption since $\mathcal{D}_t$ contains trajectories $\{\tau_i\}_{i=1}^t$ generated by $\mathbb{P}_\star^{\pi_t}$, $\widetilde{f}^t \in \mathcal{C}_\mathcal{F}(\mathcal{D}_t, \delta) \subset \mathcal{F}$ and $\widetilde{\mathbb{P}}^t \in \mathcal{C}_\mathcal{F}(\mathcal{D}_t, \delta)$. This immediately gives us that with probability $1 - \delta$

$$\mathrm{Regret}(T) = \widetilde{\mathcal{O}}(C_F(\mathcal{M}, T, \delta) + C_P(\mathcal{M}, T, \delta))$$

as desired. □

C.3.2 Generic Model-Based Optimism using Bonuses

We present a template to get regret bounds for a *generic model-based optimistic algorithm using bonuses*, which we will later instantiate into POR-UCBVI and also use in our reduction from the dueling PORRL to optimistic algorithms for cardinal PORRL.

A generic optimistic algorithm using bonuses relies on bonuses $b_{\mathcal{P}}^{\mathcal{D}}(\mathbb{P}, \pi, \delta), b_{\mathcal{F}}^{\mathcal{D}}(\mathbb{P}, \pi, \delta)$ that depend on a policy π, transition kernel $\mathbb{P}$ and dataset $\mathcal{D}$. It also relies on estimates $\hat{\mathbb{P}}_{\mathcal{D}}$ and $\hat{f}_{\mathcal{D}}$ that depend on $\mathcal{D}$. Maintaining a running dataset $\mathcal{D}_t$, at each step t, we run $\pi_t := \arg\max_{\pi \in \Pi} \widetilde{V}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi)$, where $\widetilde{V}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi)$ is given by:

$$V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi) + b_{\mathcal{F}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi, \delta) + z(Bp) b_{\mathcal{P}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi, \delta)$$

where z is defined below. We obtain a trajectory $\tau_t \sim \mathbb{P}_\star^{\pi_t}$ and append it to $\mathcal{D}_t$ to get $\mathcal{D}_{t+1}$, compute new bonuses and estimates, and continue. This algorithm is formally presented in Appendix C.3.2.

Algorithm 22 Generic Bonus-Based Optimism

1: **Input** Known family of reward functions $\{\mathcal{R}_h\}_{h=1}^H$, method $\texttt{Est}(\mathcal{D})$ to estimate $\hat{\mathbb{P}}_{\mathcal{D}}$ and $\hat{f}_{\mathcal{D}}$ from dataset $\mathcal{D}$, bonus functions $b_{\mathcal{F}}^{\mathcal{D}}(\mathbb{P}, \pi, \delta)$ and $b_{\mathcal{P}}^{\mathcal{D}}(\mathbb{P}, \pi, \delta)$, confidence level δ

2: **Initialize** $\mathcal{D}_1 \leftarrow \{\}$, initialize $\hat{f}^{\mathcal{D}_1}, \hat{\mathbb{P}}_{\mathcal{D}_1}$ arbitrarily.

3: **for** $t = 1, ..., T$ **do**

4: **Compute** optimistic history dependent policy,

$$\pi_t = \arg\max_{\pi} V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi) + b_{\mathcal{F}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi, \delta) + z(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi, \delta))$$

5: **Observe** trajectory $\tau_t = \{(s_h^t, a_h^t)\}_{h=1}^H$ and feedback $\{o_h\}_{h \in \mathcal{H}_p}$.

6: **Compute** new estimates $\hat{f}^{\mathcal{D}_{t+1}}, \hat{\mathbb{P}}^{\mathcal{D}_{t+1}} \leftarrow \texttt{Est}(\mathcal{D}_{t+1})$ and compute new bonus functions $b_{\mathcal{F}}^{\mathcal{D}_{t+1}}(\hat{f}^{\mathcal{D}_{t+1}}, \cdot, \delta), b_{\mathcal{P}}^{\mathcal{D}_{t+1}}(\hat{\mathbb{P}}^{\mathcal{D}_{t+1}}, \cdot, \delta)$.

7: **end for**

We now make the following assumption about our bonuses. Showing this assumption is the core of proving regret bounds for any instantiation of this generic algorithm. We will see later that it is satisfied by the bonuses for POR-UCBVI.

Assumption 14 (Controlling Value Error via Bonuses)**.** For a transition kernel $\mathbb{P}_\star$ and function $f_\star$, consider any sequence of policies π_t and datasets $\mathcal{D}_t$ that contain $\{\tau_i\}_{i=1}^t$ generated under $(\mathbb{P}_\star^{\pi_t}, f^\star)$. We require that for sequences $\hat{\mathbb{P}}_{\mathcal{D}_t}$ and $\hat{f}_{\mathcal{D}_t}$ and any sequence $f^t \in \mathcal{F}$, the following hold.

- *Bounding effect of error in $\mathcal{F}$:* With probability $1 - \delta/32$, for any $\mathbb{P}$ and uniformly over all

policies π, $|V(\mathbb{P}, \hat{f}_{\mathcal{D}_t}, \pi) - V(\mathbb{P}, f^\star, \pi)| \le b_{\mathcal{F}}^{\mathcal{D}_t}(\mathbb{P}, \pi, \delta)$ and there is a function $C_F(\mathcal{M}, T, \delta)$ so that $\sum_{t=1}^T b_{\mathcal{F}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta) = \widetilde{\mathcal{O}}(C_F(\mathcal{M}, T, \delta))$ with probability $1 - \delta/32$

- *Bounding effect of error in $\mathcal{P}$:* For any function $\mu : \Gamma_H \to \mathcal{R}$ bounded by D, there is a function $z(D) \ge D$ so that the following holds uniformly over all policies π with probability $1 - \delta/32$.

$$\mathbb{E}_{\tau \sim (\hat{\mathbb{P}}_{\mathcal{D}_t})^\pi} \mu(\tau) - \mathbb{E}_{\tau \sim \mathbb{P}_\star^\pi} \mu(\tau) \le z(D) b_{\mathcal{P}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi, \delta)$$

  The statement also holds if we switch $\mathbb{P}_\star$ and $\hat{\mathbb{P}}_{\mathcal{D}_t}$. Additionally, the statement holds for a suitable D if we replace $\mathbb{E}_{\tau \sim \mathbb{P}^\pi} \mu(\tau)$ with $b_{\mathcal{P}}(\mathbb{P}, \pi, \delta)$ or $b_{\mathcal{F}}(\mathbb{P}, \pi, \delta)$.[1] Finally, there is a function $C_P(\mathcal{M}, T, \delta)$ so that $\sum_{t=1}^T b_{\mathcal{P}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta) = \widetilde{\mathcal{O}}(C_P(\mathcal{M}, T, \delta))$ with probability $1 - \delta/32$.

Theorem C.3.2 (Regret for Bonus-Based Optimism)**.** *Under Assumption 14, with $z_1(D) = z(D) + z(2D) + z(2z(D))$, any generic optimistic algorithm using bonuses satisfies*

$$\mathrm{Regret}(T) = \widetilde{\mathcal{O}}\left(C_F(\mathcal{M}, T, \delta) + z_1(Bp) C_P(\mathcal{M}, T, \delta)\right)$$

Proof. Note that we can use Assumption 14 since $\mathcal{D}_t$ contains trajectories $\{\tau_i\}_{i=1}^t$ generated by $\mathbb{P}_\star^{\pi_t}$, $\hat{f}_{\mathcal{D}_t} \in \mathcal{F}$ is computed using $\mathcal{D}_t$ and $\hat{\mathbb{P}}_{\mathcal{D}_t}$ is computed using $\mathcal{D}_t$. Also note that WLOG, $b_{\mathcal{P}}^{\mathcal{D}_t}(\mathbb{P}, \pi, \delta) \le 2$ always holds since we can otherwise clip it at 2 and our assumption will still hold. Similarly, WLOG $b_{\mathcal{F}}^{\mathcal{D}_t}(\mathbb{P}, \pi, \delta) \le 2z(Bp)$, otherwise we can clip it at 1 and our assumption will still hold. Now note the following inequalities.

$$\begin{aligned}
\mathrm{Regret}(T) &= \sum_{t=1}^T V(\mathbb{P}_\star, f^\star, \pi_\star) - V(\mathbb{P}_\star, f^\star, \pi_t) \\
&\overset{(i)}{\le} \sum_{t=1}^T V(\hat{\mathbb{P}}_{\mathcal{D}_t}, f^\star, \pi_\star) + z(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_\star, \delta)) - V(\mathbb{P}_\star, f^\star, \pi_t) \\
&\overset{(ii)}{\le} \sum_{t=1}^T V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi_\star) + b_{\mathcal{F}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_\star, \delta) + z(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_\star, \delta)) - V(\mathbb{P}_\star, f^\star, \pi_t) \\
&\overset{(iii)}{\le} \sum_{t=1}^T V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi_t) + b_{\mathcal{F}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_t, \delta) + z(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_t, \delta)) - V(\mathbb{P}_\star, f^\star, \pi_t) \\
&= \sum_{t=1}^T V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi_\star) - V(\mathbb{P}_\star, f^\star, \pi_t) + b_{\mathcal{F}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_t, \delta) + z(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_t, \delta)) \\
&= \sum_{t=1}^T V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi_\star) - V(\hat{\mathbb{P}}_{\mathcal{D}_t}, f^\star, \pi_\star) + V(\hat{\mathbb{P}}_{\mathcal{D}_t}, f^\star, \pi_\star) - V(\mathbb{P}_\star, f^\star, \pi_t)
\end{aligned}$$

[1] This would instantly hold with $D = Bp$ if $b_{\mathcal{F}}(\mathbb{P}, \pi, \delta) := \mathbb{E}_{\tau \sim \mathbb{P}^\pi} b_{\mathcal{F}}(\tau, \delta)$ for some trajectory level bonus $b_{\mathcal{F}}(\tau, \delta)$, and similarly for $\mathcal{P}$.

$$+ \sum_{t=1}^{T} b_{\mathcal{F}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_t, \delta) + z(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_t, \delta))$$

Here, inequality (i) holds with probability $1-\delta/16$ by the second point in Assumption 14. Inequality (ii) holds with probability $1-\delta/16$ by the first point in Assumption 14. Inequality (iii) holds with probability 1 by the optimistic definition of π_t. Continuing, we have

$$\begin{aligned}
\text{Regret}(T) &\overset{(iv)}{\leq} 2\sum_{t=1}^{T} b_{\mathcal{F}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_t, \delta) + z(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_t, \delta)) \\
&\overset{(v)}{\leq} 2\sum_{t=1}^{T} b_{\mathcal{F}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta) + 2z(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta)) + z(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta)) \\
&\quad + 2z(z(Bp))(b_{\mathcal{P}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta)) \\
&= \mathcal{O}\left(\sum_{t=1}^{T} b_{\mathcal{F}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta) + z_1(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta))\right)
\end{aligned}$$

Here, inequality (iv) holds with probability $1-\delta/8$ by a union bound over the first and the second point in Assumption 14. Finally, inequality (v) holds with probability $1-\delta/4$ by a union bound over four applications of the second point of Assumption 14. Finally, we use a union bound over both points of Assumption 14 to conclude that with probability $1-\delta/8$

$$\mathcal{O}\left(\sum_{t=1}^{T} b_{\mathcal{F}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta) + z_1(Bp)(b_{\mathcal{P}}^{\mathcal{D}_t}(\mathbb{P}_\star, \pi_t, \delta))\right) = \widetilde{\mathcal{O}}\left(C_F(\mathcal{M}, T, \delta) + z_1(Bp)C_P(\mathcal{M}, T, \delta)\right)$$

By taking a union bound over the events of all inequalities above, we have that with probability $1-\delta$

$$\text{Regret}(T) = \widetilde{\mathcal{O}}\left(C_F(\mathcal{M}, T, \delta) + z_1(Bp)C_P(\mathcal{M}, T, \delta)\right)$$

as desired. □

C.3.3 Generic Model-Free Optimism

Algorithm 23 Generic Model-Free Optimism

Input Known Bellman-complete class of Q-functions $\mathcal{Q}$, confidence level δ.
Initialize dataset $\mathcal{D}_1 \leftarrow \{\}$ and $\mathcal{C}_{\mathcal{Q}}(\mathcal{D}_1, \delta) \leftarrow \mathcal{Q}$.
for $t = 1, ..., T$ **do**
  $\tau[0] \leftarrow ()$
  for $h = 1, \dots H$ **do**
    Play $a_h^t, Q_h^t \leftarrow \arg\max_{a, Q \in \mathcal{C}_{\mathcal{Q}}(\mathcal{D}_t, \delta)} Q_h(\tau[h], a)$ and observe feedback o_h^t
  end for
  Update $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{\tau, (o_1^t, \dots o_H^t\}$
  Compute $\mathcal{C}_{\mathcal{Q}}(\mathcal{D}_{t+1}, \delta)$
end for

Note that the method for choosing actions a_h^t at time t induces a history dependent policy π_t, whose suboptimality is what we use to define regret. Regret is still given by

$$\text{Regret}(T) = \sum_{t=1}^{T} V(\mathcal{M}_\star, \pi_\star) - V(\mathcal{M}_\star, \pi_t)$$

We now make the following assumption about our confidence sets. Showing this assumption is the core of proving regret bounds for any instantiation of this generic algorithm. We know that this is satisfied by GOLF using the BE-dimension. We will show that in our case, it is also satisfied by a more refined notion known as the α-HABE dimension (the α-history aware Bellman eluder dimension).

Assumption 15. For a Q-function $Q^\star$ induced by model $\mathcal{M}_\star$, consider any sequence of policies π_t and datasets $\mathcal{D}_t$ that contain $\{\tau_i\}_{i=1}^t$ generated under $\mathcal{M}_\star$. We require that $Q^\star \in \mathcal{C}_{\mathcal{Q}}(\mathcal{D}_t, \delta)$ for all t with probability $1 - \delta/16$. We require that there exists a problem dependent function $C_Q(\mathcal{Q}, T, \delta)$, so that for arbitrary sequences $Q^t \in \mathcal{C}_{\mathcal{Q}}(\mathcal{D}_t, \delta)$, the following holds for all h with probability $1 - \delta/2$.

$$\sum_{j=1}^{t} |\mathbb{E}_{\mu_h(Q^t)}[Q_h^t - \mathcal{T}_h Q_h^{j+1}]| \leq C_Q(\mathcal{Q}, T, \delta)$$

Theorem C.3.3 (Regret for Generic Model-Free Optimism)**.** *If the confidence sets $\mathcal{C}_{\mathcal{Q}}(\mathcal{D}, \delta)$ used in Algorithm 23 satisfy Assumption 15, then the regret of Algorithm 23 is bounded by*

$$\text{Regret}(T) = O\left(H C_Q(\mathcal{Q}, T, \delta)\right)$$

Proof. Note that $V(\mathcal{M}_\star, \pi_\star) = \max_a Q_1^\star(s_1, a) \le \max_a Q_1^t(s_1, a)$ for all t, giving us the following result by the policy loss decomposition in Jiang et al. [2017].

$$\begin{aligned}
\text{Regret}(T) &= \sum_{t=1}^{T} V(\mathcal{M}_\star, \pi_\star) - V(\mathcal{M}_\star, \pi_t) \\
&\le \sum_{t=1}^{T} \max_a Q_1^t(s_1, a) - V(\mathcal{M}_\star, \pi_t) \\
&= \sum_{t=1}^{T} \sum_{h=1}^{H} \mathbb{E}_{\mu_h(Q^t)}[Q_h^t - \mathcal{T}_h Q_h^{j+1}] \\
&= \sum_{h=1}^{H} \sum_{t=1}^{T} \mathbb{E}_{\mu_h(Q^t)}[Q_h^t - \mathcal{T}_h Q_h^{j+1}] \\
&= O(HC_Q(\mathcal{Q}, T, \delta))
\end{aligned}$$

where the last line holds by Assumption 15. □

C.4 Details and Proofs for Cardinal POR-UCRL

We now instantiate Algorithm 21 using standard confidence sets to get POR-UCRL. We show that they satisfy Assumption 13 and get regret bounds for the algorithm. Note that our algorithm is **crucially different** from naively summarizing the history to define a modified state space, since we are separating the use of history summarization for getting confidence sets f from using only the current state while learning the Markovian transitions $\mathbb{P}$. In this case, it is a priori unclear if we can use ideas from optimism to prove guarantees with a favorable (non-exponential) dependence on the complexity of transitions.

Recall that given a dataset of the first t trajectory samples $\{\tau_i\}_{i=1}^t$ and an index $h \in [H]$, we consider the following least squares objective to estimate f:

$$\widehat{f}_h^{t+1} = \arg\min_{f_h \in \mathcal{F}_h} \sum_{i=1}^{t} \left(\sigma_h(f_h(\tau_i[h])) - o_h^i\right)^2$$

Simple least squares guarantees imply the lemma below.

Lemma C.4.1 (Concentration for $\sigma \circ f_h$)**.** *Define*

$$\mathrm{MSE}_{h,t}(f_h, f_h') := \sum_{i=1}^{t} \left(\sigma_h(f_h(\tau_i[h])) - \sigma_h(f_h'(\tau_i[h]))\right)^2$$

Also define $\bar{\beta}_{h,t}(\delta) = \eta_h^2 \log\left(\frac{N\left(\mathcal{F}_h, \frac{B}{T}, \|\cdot\|_\infty\right)}{\delta}\right) + \alpha_{h,t}$ *with* $\alpha_{h,t} := \frac{tB + t\eta_h \log\left(\frac{t}{\delta}\right)}{T}$*. Then* $f_h^\star$ *simultaneously satisfies* $\mathrm{MSE}_{h,t}(f_h^\star, \widehat{f}_h^{(t+1)}) \le \bar{\beta}_{h,t}\left(\frac{\delta}{2t^2 H}\right)$ *for all* h, t *with probability* $1 - \delta/32$*.*

Proof. We apply Lemma 6 in Chan et al. [2021] and the last statement in its proof to each h separately with the function class in the lemma set to $\{\sigma_h \circ f_h | f_h \in \mathcal{F}_h\}$, $P = 1$, $\mathbf{x}_{t,p} = \mathbf{x}_{t,1} = \tau_t[h]$ and misspecification $\epsilon = 0$ (decoupled from the Eluder dimension's ϵ). We also note that o_h^t are η_h-subgaussian samples with mean $\sigma_h(f_h(\tau[h]))$. This gives us that each of event indexed by h, t below holds with probability at least $1 - \frac{\delta}{2t^2 H}$.

$$\sum_{i=1}^{t} \left(\sigma_h(f_h(\tau_i[h])) - \sigma_h(f_h'(\tau_i[h]))\right)^2 \le \bar{\beta}_{h,t}\left(\frac{\delta}{2t^2 H}\right)$$

So, the events all simultaneously hold with probability at least $1 - \delta$ by a union bound. □

Recall the definition of our confidence sets below.

Confidence Sets for POR-UCRL. We instantiate the generic optimistic algorithm using confidence sets by defining $\mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta) := \mathcal{C}_{\mathcal{P}}^t(\delta) \times \mathcal{C}_{\mathcal{F}}^t(\delta)$ as our confidence sets below. We name the resulting algorithm POR-UCRL. We use the data from trajectories $\{\tau_i\}_{i=1}^t$ to build the confidence sets $\mathcal{C}_{\mathcal{F}}^{t+1}(\delta) = \prod_h \mathcal{C}_h^{t+1}(\delta)$ with $\mathcal{C}_h^{t+1}(\delta)$ defined below, where $\beta_{h,t}(\delta) := \bar{\beta}_{h,t}\left(\frac{\delta}{2t^2 H}\right)$.

$$\mathcal{C}_h^{t+1}(\delta) := \left\{ f_h \in \mathcal{F}_h \middle| \mathrm{MSE}_{h,t}(f_h^\star, \widehat{f}_h^{(t+1)}) \le \beta_{h,t}(\delta) \right\}$$

We also use the MLE estimate for $\mathbb{P}$ after t episodes to define $\hat{\mathbb{P}}^t(\cdot \mid s, a) := \frac{N_t(s,a,s')}{N_t(s,a)}$. Now for $\zeta(n, \delta) = 2\sqrt{\frac{S\log(2) + \log(n(n+1)SA/\delta)}{2n}}$, define $\mathcal{C}_{\mathbb{P}}^t(\delta)$ as below:

$$\left\{ \mathbb{P} \middle| \|\mathbb{P}(\cdot \mid s, a) - \hat{\mathbb{P}}_t(\cdot \mid s, a)\|_1 \le \zeta(N_t(s, a), \delta) \forall s, a \right\}$$

Confidence Sets for POR-UCRL in case $\mathbb{P}_\star$ is known. For known-model UCRL, the confidence sets $\mathcal{C}_{\mathcal{F}}^t(\delta)$ are still as above, but $\mathcal{C}_{\mathcal{P}}^t(\delta) := \{\mathbb{P}_\star\}$

For completeness, we repeat the algorithm POR-UCRL here, which is an instantiation of Algorithm 21, the generic optimistic algorithm using confidence sets.

Algorithm 24 POR-UCRL

1: **Input:** Known family of reward functions $\{\mathcal{R}_h\}_{h=1}^H$, known probability transition kernel class $\mathcal{P}$ and known decoder-induced function class $\mathcal{F}$, confidence level δ.
2: **Initialize** dataset $\mathcal{D}_1 \leftarrow \{\}$ and $\mathcal{C}_{\mathcal{F}}(\mathcal{D}_1, \delta) \leftarrow \prod_{h=1}^H \mathcal{F}_h, \mathcal{C}_{\mathcal{P}}(\mathcal{D}_1, \delta) \leftarrow \mathcal{P}$.
3: **for** $t = 1, ..., T$ **do**
4: **Compute** the optimistic history dependent policy,

$$\pi_t, \widetilde{f}_t, \widetilde{\mathcal{P}}_t = \underset{\pi,\ \mathcal{F} \in \mathcal{C}_{\mathcal{F}}(\mathcal{D}_t, \delta), \mathbb{P} \in \mathcal{P}(\mathcal{D}_t, \delta)}{\arg\max} V(\mathbb{P}, f, \pi)$$

5: **Collect** trajectory $\tau_t = \{(s_h^t, a_h^t)\}_{h=1}^H$ and feedback $\{o_h\}_{h \in \mathcal{H}_p}$ by sampling from $\mathbb{P}_\star^{\pi_t}$ with true decoder-induced function $f_\star$.
6: **Update** $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{\tau_t\}, \hat{\mathbb{P}}_{t+1}, \hat{f}_h^{t+1}$ for all h
7: **Compute** new confidence sets $\mathcal{C}_{\mathcal{F}}(\mathcal{D}_{t+1}, \delta) \leftarrow \prod_{h=1}^H \mathcal{C}_h^{t+1}(\delta)$ and $\mathcal{C}_{\mathcal{P}}(\mathcal{D}_{t+1}, \delta)$ where

$$\mathcal{C}_h^{t+1}(\delta) \leftarrow \left\{ f_h \in \mathcal{F}_h \middle| \mathrm{MSE}_{h,t}(f_h^\star, \widehat{f}_h^{(t+1)}) \le \beta_{h,t}(\delta) \right\}$$
$$\mathcal{C}_{\mathcal{P}}(\mathcal{D}_{t+1}, \delta) \leftarrow \left\{ \mathbb{P} \middle| \|\mathbb{P}(\cdot \mid s, a) - \hat{\mathbb{P}}_{t+1}(\cdot \mid s, a)\|_1 \le \zeta(N_{t+1}(s, a), \delta) \forall s, a \right\}$$

8: **end for**

We will now show our regret bound.

Theorem C.4.2 (POR-UCRL Regret)**.** *Under Assumption 9, the regret* $\mathrm{Regret}(T)$ *of POR-UCRL is bounded by the following with probability at least* $1 - \delta$

$$\widetilde{\mathcal{O}}\left(\left(pS\sqrt{HA} + \sum\nolimits_{h \in \mathcal{H}_p} \sqrt{d_{E,h} d_{C,h}} \right) \sqrt{T} \right)$$

where $d_{E,h} = \dim_E\left(\mathcal{F}_h, \frac{B}{T}\right)$ *and* $d_{C,h} = \log(\mathcal{N}(\mathcal{F}_h, 1/T, \|\cdot\|_\infty))$.

C.4.1 Showing that Assumption 13 is Satisfied

C.4.1.1 Bounding Reward Model Deviations

Lemma C.4.3 (Bounding Reward Model Deviations)**.** *Consider decoder-induced functions* $\{f_h\}_{h \in \mathcal{H}_p}$ *satisfying* $|f_h| \le B$ *that induce value functions* $V(\mathbb{P}, f, \pi)$*. For any sequence of policies*

π_t, if the confidence $\mathcal{C}^t_{\mathcal{F}}(\delta)$ is generated using data $\tau_i \sim \mathbb{P}^{\pi_i}_\star, i = 1 \to t$ and $\widetilde{f}^t \in \mathcal{C}^t_{\mathcal{F}}(\delta)$ is an arbitrary sequence of functions, then we have the following with probability $1 - \delta/4$.

$$\left| \sum_{t=1}^{T} V(\mathbb{P}_\star, \widetilde{f}^t, \pi_t) - V(\mathbb{P}_\star, f^\star, \pi_t) \right|$$

is bounded by

$$\mathcal{O}\left(Bp\sqrt{T\log(T/\delta)} + \sum_{h\in\mathcal{H}_p} B\kappa_{2,h} d_{E,h} + \sum_{h\in\mathcal{H}_p} \kappa_{2,h}\sqrt{d_{E,h}\beta_{h,T}(\delta)T} \right)$$

Proof.

$$\begin{aligned}
\sum_{t=1}^{T} V(\mathbb{P}_\star, \widetilde{f}^t, \pi_t) - V(\mathbb{P}_\star, f^\star, \pi_t) &= \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_t}_\star}\left[\sum_{h=1}^{H} \widetilde{f}^t_h(\tau[h])\right] - \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_t}_\star}\left[\sum_{h=1}^{H} f^\star_h(\tau[h])\right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_t}_\star}\left[\sum_{h\in\mathcal{H}_p} \widetilde{f}^t_h(\tau[h])\right] - \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_t}_\star}\left[\sum_{h\in\mathcal{H}_p} f^\star_h(\tau[h])\right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_t}_\star}\left[\sum_{h\in\mathcal{H}_p} \widetilde{f}^t_h(\tau[h]) - f^\star_h(\tau[h])\right] \\
&= \sum_{t=1}^{T} \left[\sum_{h\in\mathcal{H}_p} \widetilde{f}^t_h(\tau_t[h]) - f^\star_h(\tau_t[h]) + X_{1,t} + X_{2,t}\right] \\
&\overset{(ii)}{\leq} \sum_{t=1}^{T} \left[\sum_{h\in\mathcal{H}_p} \widetilde{f}^t_h(\tau_t[h]) - f^\star_h(\tau_t[h])\right] + \mathcal{O}\left(Bp\sqrt{T\log(T/\delta)}\right)
\end{aligned}$$

where

$$\begin{aligned}
X_{1,t} &:= \mathbb{E}_{\tau\sim\mathbb{P}^{\pi}}\left[\sum_{h\in\mathcal{H}_p} \widetilde{f}^t_h(\tau[h])\right] - \left[\sum_{h\in\mathcal{H}_p} \widetilde{f}^t_h(\tau_t[h])\right] \\
X_{2,t} &:= \left[\sum_{h\in\mathcal{H}_p} f^\star_h(\tau_t[h])\right] - \mathbb{E}_{\tau\sim\mathbb{P}^{\pi}}\left[\sum_{h\in\mathcal{H}_p} f^\star_h(\tau[h])\right]
\end{aligned}$$

Inequality (i) follows by the definition of π_t and $\widetilde{f}^t_h$ – that is, by optimism. Inequality (ii) holds with probability at least $1 - \delta$ since $X_{1,t}$ and $X_{2,t}$ are both martingales with respect to the filtration $\mathcal{G}_t$ given by the data of trajectories $\{\tau_s\}_{s=1}^{t-1}$. Also, $|X_{1,t}|, |X_{2,t}| \leq Bp$. We can thus apply the

Azuma-Hoeffding inequality twice to obtain inequality (ii).

Continuing, note the following.

$$
\begin{aligned}
&\sum_{t=1}^{T} V(\mathbb{P}_\star, \widetilde{f}^t, \pi_t) - V(\mathbb{P}_\star, f^\star, \pi_t) \\
&\leq \left[\sum_{h=1}^{H} f_h^t(\tau_t[h]) - f_h^\star(\tau_t[h])\right] + Bp\sqrt{T\log(T/\delta)} \\
&\leq \left[\sum_{h=1}^{H} \kappa_{2,h}\sigma_h(f_h^t(\tau_t[h])) - \kappa_{2,h}\sigma_h(f_h^\star(\tau_t[h]))\right] + Bp\sqrt{T\log(T/\delta)} \\
&\leq \kappa_{2,h}\sum_{t=1}^{T}\sum_{h\in\mathcal{H}_p} \underbrace{\max_{f_h, f_h'\in\mathcal{W}_h^t(\delta)} \sigma_h(f_h(\tau_t[h])) - \sigma_h(f_h'(\tau_t[h]))}_{=:\bar{\gamma}_{h,t}(\tau_t[h],\delta)} + Bp\sqrt{T\log(T/\delta)} \\
&= \kappa_{2,h}\sum_{h\in\mathcal{H}_p}\left[\sum_{t=1}^{T}\bar{\gamma}_{h,t}(\tau_t[h],\delta)\right] + Bp\sqrt{T\log(T/\delta)}
\end{aligned}
$$

The sum of these maximum uncertainty evaluations can be upper bounded using the Eluder dimension. The inequality below holds by applying Lemma 3 in [Chan et al., 2021] for each h separately, with the function class in the lemma set to $\{\sigma_h \circ f_h | f_h \in \mathcal{F}_h\}$, $P = 1$, $\mathbf{x}_{t,p} = \mathbf{x}_{t,1} = \tau_t[h]$ and misspecification $\epsilon = 0$ (decoupled from the Eluder dimension's ϵ). We also recall that o_h^t are η_h-subgaussian samples with mean $\sigma_h(f_h(\tau_t[h]))$. We obtain

$$
\sum_{t=1}^{T}\bar{\gamma}_{h,t}(\tau_t[h],\delta) \leq \mathcal{O}\left(Bd_{E,h} + \sqrt{d_{E,h}\beta_{h,T}(\delta)T}\right)
$$

Where $d_{E,h} = \dim_E\left(\mathcal{F}_h, \frac{B}{T}\right)$ is the Eluder dimension of $\mathcal{F}_h$ and $\beta_{h,T}(\delta) = \bar{\beta}_{h,t}\left(\frac{\delta}{2t^2H}\right)$. Therefore, we have our result.

$$
\sum_{t=1}^{T} V(\mathbb{P}_\star, \widetilde{f}^t, \pi_t) - V(\mathbb{P}_\star, f^\star, \pi_t)
$$

is bounded by

$$
\mathcal{O}\left(\sum_{h\in\mathcal{H}_p} B\kappa_{2,h}d_{E,h} + \sum_{h\in\mathcal{H}_p}\kappa_{2,h}\sqrt{d_{E,h}\beta_{h,T}(\delta)T} + Bp\sqrt{T\log(T/\delta)}\right)
$$

Note that this entire argument can be repeated with $f_\star$ and $\widetilde{f}^t$ switched, by the symmetry of the definition of $\bar{\gamma}_{h,t}(\tau_t[h], \delta)$ and the fact that the negative of a martingale is also a martingale. $\square$

C.4.1.2 Bounding Probability Model Deviations

Lemma C.4.4 (Bounding Probability Model Deviations)**.** *Consider an arbitrary sequence of functions $f^t \in \mathcal{F}$ satisfying $|f_h| \leq B$ that induce value functions $V(\mathbb{P}, f, \pi)$. For any sequence of policies π_t, if the confidence $\mathcal{C}^t_{\mathcal{P}}(\delta)$ is generated using data that includes $\tau_i \sim \mathbb{P}^{\pi_i}_\star, i = 1 \to t$ and $\widetilde{\mathbb{P}}_t \in \mathcal{C}^t_{\mathcal{P}}(\delta)$ is an arbitrary sequence of transition structures, then we have the following with probability $1 - \delta/4$.*

$$\left|\sum_{t=1}^{T} V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t)\right| \leq \mathcal{O}\left(c_\delta Bp\sqrt{SAHT} + c_\delta BpHSA + Bp\sqrt{pT\log(T/\delta)}\right)$$

where $c_\delta := 8\sqrt{S\log(2) + \log(HTSA/\delta)}$.

Proof. We first show the following.

Lemma C.4.5.

$$\sum_{t=1}^{T} V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t) \leq \sum_{t=1}^{T}\sum_{h=1}^{H} 2p\zeta(N_t(s^t_h, a^t_h), \delta) + \mathcal{O}\left(Bp\sqrt{pT\log(T/\delta)}\right)$$

Recall that we denote the Bellman operator by $\mathcal{T}^\pi$ where $\mathcal{T}^\pi f = \mathbb{E}_{a\sim\pi}\mathbb{P}f$. Momentarily define the following for $\tau = (\tau_{l-1}, s_l, \tau')$, where τ_{l-1} is an arbitrary trajectory of length $l - 1 \leq H$, and s_l is an arbitrary state.

$$\begin{aligned} V^t_{l,\mathbb{P}}(\tau_{l-1}, s_l) &:= \mathbb{E}_{\tau'\sim\mathbb{P}^{\pi_t}}\left[\sum_{h=l}^{H} \widetilde{f}^t_h(\tau[h])\right] \\ &= \mathbb{E}_{\tau'\sim\mathbb{P}^{\pi_t}}\left[\sum_{h\in\mathcal{H}_p, h\geq l} \widetilde{f}^t_h(\tau[h])\right] \end{aligned}$$

So in the definition above, the first $l - 1$ observations in τ come from τ_{l-1} while the rest are generated by the input $\mathbb{P}$ starting with state s_l. Note that $V(\mathbb{P}, f^t, \pi_t) = V^t_{1,\mathbb{P}}(\emptyset, s_1)$. Also note that by the Bellman equation, we have the following.

$$V^t_{l,\mathbb{P}}(\tau_t[l-1], s_l) = \mathbb{E}_{a\sim\pi_t}\left[f^t_l(\tau_t[l])\right] + \mathbb{E}_{a\sim\pi_t}\left[\mathbb{E}_{s'\sim\mathbb{P}(\cdot|s_l,a)} V^t_{l+1,\mathbb{P}}((\tau_t[l-1], s, a), s'))\right]$$

$$= \mathbb{E}_{a\sim\pi_t}\left[f_l^t(\tau_t[l])\right] + \mathbb{E}_{a\sim\pi_t}\left[\mathbb{P}(\cdot \mid s_l, a)^\top V^t_{l+1,\mathbb{P}}((\tau_t[l-1], s_l, a), \cdot)\right]$$

Now use τ_t to set $\tau_{l-1} := \tau_t[l-1]$ and define the following.

$$\Delta_l^t(s_l) := V^t_{l,\mathbb{P}_\star}(\tau_t[l-1], s_l) - V_{l,\widetilde{\mathbb{P}}_t}(\tau_t[l-1], s_l)$$

Note that

$$\Delta_1^t(s_1) = V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t) \tag{C.1}$$

The computation above then gives us the following.

$$\begin{aligned}
\Delta_l^t(s_l^t) &= \mathbb{E}_{a\sim\pi_t}\left[\widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a)^\top V^t_{l+1,\widetilde{\mathbb{P}}_t}((\tau_t[l-1], s_l^t, a), \cdot))\right] \\
&\quad - \mathbb{E}_{a\sim\pi_t}\left[\mathbb{P}_\star(\cdot \mid s_l^t, a)^\top V^t_{l+1,\mathbb{P}_\star}((\tau_t[l-1], s_l^t, a), \cdot))\right] \\
&= \widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t)^\top V^t_{l+1,\widetilde{\mathbb{P}}_t}(\tau_t[l], \cdot) - \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t)^\top V^t_{l+1,\mathbb{P}_\star}(\tau_t[l], \cdot) + Y_{l,t} + Z_{l,t}
\end{aligned}$$

where $Y_{l,t}$ and $Z_{l,t}$ are stochastic processes defined below.

$$\begin{aligned}
Y_{l,t} &:= \mathbb{P}_\star(\cdot \mid s_l^t, a)^\top V^t_{l+1,\mathbb{P}_\star}(\tau_t[l], \cdot) - \mathbb{E}_{a\sim\pi_t}\left[\mathbb{P}_\star(\cdot \mid s_l^t, a)^\top V^t_{l+1,\mathbb{P}_\star}((\tau_t[l-1], s_l^t, a), \cdot)\right] \\
Z_{l,t} &:= \mathbb{E}_{a\sim\pi_t}\left[\widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a)^\top V^t_{l+1,\widetilde{\mathbb{P}}_t}((\tau_t[l-1], s_l^t, a), \cdot)\right] - \widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a)^\top V^t_{l+1,\widetilde{\mathbb{P}}_t}(\tau_t[l], \cdot))
\end{aligned}$$

Consider the filtration $\mathcal{G}_{l,t}$ induced by the data of $\{\tau_s\}_{s=1}^{t-1} \cup \tau_t[l-1] \cup \{s_l^t\}$. Since $a_l^t \sim \pi_t$ and $(\tau_t[l-1], s_l^t, a_l^t) = \tau_t[l]$, we get that $\mathbb{E}[Y_{l,t}|\mathcal{G}_{l,t}] = \mathbb{E}[Z_{l,t}|\mathcal{G}_{l,t}] = 0$. So, one can see that both processes are martingales over $\mathcal{G}_{l,t}$. Also note that $|Y_{l,t}|, |Z_{l,t}| \le p$. We thus have that

$$\begin{aligned}
\Delta_l^t(s_l^t) &= \left[\widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) - \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t)\right] V^t_{l+1,\widetilde{\mathbb{P}}_t}(\tau_t[l], \cdot)) \\
&\quad + \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t)^\top \left[V^t_{l+1,\widetilde{\mathbb{P}}_t}(\tau_t[l], \cdot)) - V^t_{l+1,\mathbb{P}_\star}(\tau_t[l], \cdot)\right] + Y_{l,t} + Z_{l,t} \\
&= \left[\widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) - \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t)\right] V^t_{l+1,\widetilde{\mathbb{P}}_t}(\tau_t[l], \cdot)) + \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t)^\top \Delta^t_{l+1}(s') + Y_{l,t} + Z_{l,t} \\
&= \left[\widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) - \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t)\right] V^t_{l+1,\widetilde{\mathbb{P}}_t}(\tau_t[l], \cdot)) + \mathbb{E}_{s'\sim\mathbb{P}_\star(\cdot|s_l^t,a_l^t)}\left[\Delta^t_{l+1}(s')\right] + Y_{l,t} + Z_{l,t} \\
&= \left[\widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) - \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t)\right] V^t_{l+1,\widetilde{\mathbb{P}}_t}(\tau_t[l], \cdot)) + \Delta^t_{l+1}(s_{l+1}^t) + U_{l,t} + Y_{l,t} + Z_{l,t}
\end{aligned}$$

where

$$U_{l,t} := \mathbb{E}_{s'\sim\mathbb{P}_\star(\cdot|s_l^t,a_l^t)}\left[\Delta^t_{l+1}(s')\right] - \Delta^t_{l+1}(s_{l+1}^t)$$

Consider the filtration $\bar{\mathcal{G}}_{l,t}$ defined by the data of $\{\tau_s\}_{s=1}^{t-1} \cup \tau_t[l]$. Clearly, $U_{l,t}$ is a martingale over

$\bar{\mathcal{G}}_{l,t}$. Also note that $|U_{l,t}| \leq p$ To conclude, we have that

$$\Delta_l^t(s_l^t) - \Delta_{l+1}^t(s_{l+1}^t) = \left[\widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) - \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t)\right] V_{l+1,\widetilde{\mathbb{P}}_t}^t(\tau_t[l], \cdot)) + U_{l,t} + Y_{l,t} + Z_{l,t}$$

Using a telescoping sum over l for a fixed t and equation C.1, we get that for any t, the following holds.

$$\begin{aligned}
&V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t) \\
&= \Delta_1^t(s_1) \\
&= \sum_{l=1}^{H} \left[\widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) - \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t)\right] V_{l+1,\widetilde{\mathbb{P}}_t}^t(\tau_t[l], \cdot)) + U_{l,t} + Y_{l,t} + Z_{l,t}
\end{aligned}$$

$$\begin{aligned}
&V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t) \\
&\leq \sum_{l=1}^{H} Bp \left\| \widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) - \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t) \right\|_1 + U_{l,t} + Y_{l,t} + Z_{l,t} \\
&\leq \sum_{l=1}^{H} Bp \left\| \widetilde{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) - \hat{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) \right\|_1 \\
&\quad + Bp \left\| \mathbb{P}_\star(\cdot \mid s_l^t, a_l^t) - \hat{\mathbb{P}}_t(\cdot \mid s_l^t, a_l^t) \right\|_1 + U_{l,t} + Y_{l,t} + Z_{l,t}
\end{aligned} \tag{C.2}$$

Until equation C.2, all statements have held with probability 1 and did not use any facts about $\widetilde{\mathbb{P}}_t$. The last inequality also holds with probability 1 and uses the design of the confidence sets. Now, note the following well known concentration lemma. See, for example, Szepesvári [2023].

Lemma C.4.6. *For* $\zeta(n, \delta) = 8\sqrt{\frac{S\log(2) + \log(n(n+1)SA/\delta)}{2n}}$ *and*

$$\mathcal{C}_{\mathcal{P}}^t(\delta) = \left\{ \mathbb{P} \middle| \|\mathbb{P}(\cdot \mid s, a) - \hat{\mathbb{P}}_t(\cdot \mid s, a)\|_1 \leq \zeta(N_t(s, a), \delta) \forall s, a \right\}$$

the true model $\mathbb{P}^\star \in \mathcal{C}_{\mathcal{P}}^t(\delta)$ *for all* $t \geq 1$ *with probability at least* $1 - \delta/32$.

Applying the lemma twice and applying a union bound imply that the following holds with probability $1 - \delta/8$.

$$\sum_{t=1}^{T} V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t)$$

$$
\begin{aligned}
&\overset{(i)}{\leq} \sum_{l=1}^{H} 2Bp\zeta(N_t(s_l^t, a_l^t), \delta) + U_{l,t} + Y_{l,t} + Z_{l,t} \\
&= \sum_{t=1}^{T}\sum_{h=1}^{H} 2Bp\zeta(N_t(s_h^t, a_h^t), \delta) + \left[\sum_{t=1}^{T}\sum_{h\in\mathcal{H}_p} U_{h,t} + Y_{h,t} + Z_{h,t}\right] \\
&\overset{(ii)}{\leq} \sum_{t=1}^{T}\sum_{h=1}^{H} 2Bp\zeta(N_t(s_h^t, a_h^t), \delta) + \mathcal{O}\left(Bp\sqrt{pT\log(T/\delta)}\right)
\end{aligned}
$$

Note that inequality (i) is subtle since we could have used more data than that from $\tau_i, i = 1 \to t$ to construct $\mathcal{C}_{\mathcal{P}}^t$. The inequality still holds since $\zeta(n, \delta)$ is decreasing in n. Also, inequality (ii) holds by the Azuma-Hoeffding inequality.

Now note that the whole argument above can be repeated with $\mathbb{P}_\star$ and $\widetilde{\mathbb{P}}_t$ switched, since the negative of a martingale is also a martingale. So, we have that with probability $1 - \delta/4$

$$
\left|\sum_{t=1}^{T} V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t)\right| \leq \sum_{t=1}^{T}\sum_{h=1}^{H} 2Bp\zeta(N_t(s_h^t, a_h^t), \delta) + \mathcal{O}\left(Bp\sqrt{pT\log(T/\delta)}\right)
$$

Finally, we need the following easy lemma, proved in Szepesvári [2023].

Lemma C.4.7. *Let* $c_\delta := 8\sqrt{S\log(2) + \log(HTSA/\delta)}$. *Then the following holds almost surely.*

$$
\sum_{t=1}^{T}\sum_{h=1}^{H} 2Bp\zeta(N_t(s_h^t, a_h^t), \delta) \leq c_\delta Bp\sqrt{SAHT} + c_\delta BpHSA
$$

This establishes our claim. □

C.4.2 Putting It All Together

Theorem C.4.8 (POR-UCRL Regret)**.** *Under Assumption 9, the regret* $\mathrm{Regret}(T)$ *of POR-UCRL is bounded by the following with probability at least* $1 - \delta$

$$
\widetilde{\mathcal{O}}\left(\left(pS\sqrt{HA} + \sum\nolimits_{h\in\mathcal{H}_p}\sqrt{d_{E,h}d_{C,h}}\right)\sqrt{T}\right)
$$

where $d_{E,h} = \dim_E\left(\mathcal{F}_h, \frac{B}{T}\right)$ *and* $d_{C,h} = \log(\mathcal{N}(\mathcal{F}_h, 1/T, \|\cdot\|_\infty))$.

Proof. We can now combine Lemmas C.4.1and C.4.6 to conclude that $\mathcal{M}_\star \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$ for all t

with probability $1 - \delta/16$. We can now combine this observation with Lemmas C.4.3 and C.4.4 to observe that Assumption 13 is satisfied by POR-UCRL. By Theorem C.3.1, the following holds with probability $1 - \delta$.

$$\text{Regret}(T) = \mathcal{O}\left(c_\delta Bp\sqrt{SAHT} + c_\delta BpHSA + \sum_{h\in\mathcal{H}_p} B\kappa_{2,h} d_{E,h} + \sum_{h\in\mathcal{H}_p} \kappa_{2,h}\sqrt{d_{E,h}\beta_{h,T}(\delta)T}\right)$$

where $c_\delta = 8\sqrt{S\log(2) + \log(HTSA/\delta)}$, $d_{E,h} = \dim_E\left(\mathcal{F}_h, \frac{B}{T}\right)$ is the Eluder dimension of $\mathcal{F}_h$ and $\beta_{h,T}(\delta) = \beta_{h,t}\left(\frac{\delta}{2t^2H} = \widetilde{O}(B^2\eta_h^2 d_{C,h})\right)$. This is because all the terms dependent on p get absorbed by the first term in our expression below.

We further refine it by ignoring terms independent of T and using the fact that $\beta_{h,T}(\delta) = \widetilde{\mathcal{O}}(d_{C,h})$ to get that

$$\text{Regret}(T) = \widetilde{\mathcal{O}}\left(pS\sqrt{AHT} + \sum_{h\in\mathcal{H}_p} \sqrt{d_{E,h}d_{C,h}T}\right)$$

□

Analogously, we can provide a sample complexity result for POR-UCRL.

Corollary C.4.9 (POR-UCRL Sample complexity)**.** *Let $\epsilon > 0, \delta \in [0,1]$. Ignoring polynomial terms independent of ϵ, we can bound the sample complexity $N(\epsilon,\delta)$ of POR-UCRL as follows*

$$\widetilde{\mathcal{O}}\left(\frac{p^2HS^2A}{\epsilon^2} + \frac{p^2 d_E d_C}{\epsilon^2}\right)$$

where $d_E := \max_{h\in\mathcal{H}_p} d_{E,h}$, and $d_C := \max_{h\in\mathcal{H}_p} d_{C,h}$.

Proof. We invoke the regret-to-PAC conversion in Lemma C.2.1 with confidence $\delta' = \delta/2$ and we plug the regret bound in Theorem 5.3.1 to write

$$\epsilon = \widetilde{\mathcal{O}}\left(\left(BpS\sqrt{AH} + \sum_{h\in\mathcal{H}_p} \kappa_{2,h}\sqrt{d_{E,h}d_{C,h}} + Bp\sqrt{\log(1/\delta)}\right)\left(\frac{1}{\sqrt{T}}\right)\right)$$

from which we get the result by picking $N = T$ and the definition of d_E, d_C. □

Also note the following theorem and corresponding corollary.

Theorem C.4.10 (POR-UCRL Regret if $\mathbb{P}_\star$ is Known)**.** *If we know the transition matrix $\mathbb{P}_\star$ in POR-UCRL, then our regret is given by the following with probability $1-\delta$, ignoring polynomial terms independent of T.*

$$\mathrm{Regret}(T) = \widetilde{\mathcal{O}}\left(\left(Bp + \sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}d_{C,h}}\right)\sqrt{T}\right)$$

Proof. We can now use Lemmas C.4.3 and the fact that $\mathcal{C}^t_{\mathcal{P}}(\delta)$ is always a singleton to observe that Assumption 13 is satisfied by this version of POR-UCRL as well. By Theorem C.3.1, the following holds with probability $1-\delta$.

$$\mathrm{Regret}(T) = \widetilde{\mathcal{O}}\left(Bp\sqrt{T} + \sum_{h\in\mathcal{H}_p} B\kappa_{2,h}d_{E,h} + \sum_{h\in\mathcal{H}_p}\kappa_{2,h}\sqrt{d_{E,h}\beta_{h,T}(\delta)T}\right)$$

We further refine it by ignoring terms independent of T and using the fact that $\beta_{h,T}(\delta) = \widetilde{\mathcal{O}}(d_{C,h})$ to get that

$$\mathrm{Regret}(T) = \widetilde{\mathcal{O}}\left(\left(Bp + \sum_{h\in\mathcal{H}_p}\kappa_{2,h}\sqrt{d_{E,h}\beta_{h,T}(\delta)}\right)\sqrt{T}\right)$$

□

Corollary C.4.11 (POR-UCRL sample complexity if $\mathbb{P}_\star$ is Known)**.** *Let $\epsilon > 0, \delta \in [0,1]$. Ignoring polynomial terms independent of ϵ, we can bound the sample complexity $N(\epsilon,\delta)$ of POR-UCRL when $\mathbb{P}_\star$ is known as follows*

$$\widetilde{\mathcal{O}}\left(\frac{p^2 d_E d_C}{\epsilon^2}\right)$$

where $d_E := \max_{h\in\mathcal{H}_p} d_{E,h}$, and $d_{C,h} := \max_{h\in\mathcal{H}_p} d_{C,h}$.

Proof. The proof proceeds as in Corollary C.4.9 by plugging Theorem C.4.10 in Lemma C.2.1. □

C.5 Details and Proofs for Cardinal POR-UCBVI

We now describe how we instantiate POR-UCBVI from a generic optimistic algorithm using bonuses. Note that again, this is **crucially different** from naively summarizing the history to define a modified state space, since we are separating the use of history summarization for getting bonuses for f from using only the current state while getting bonuses for the Markovian transitions $\mathbb{P}$. Like

with confidence sets, it is a priori unclear if we can use ideas from optimism to prove guarantees with a favorable (non-exponential) dependence on the complexity of transitions. In particular, we will note that showing that the bonuses are optimistic would naively need a union bound over the doubly exponential ($A^{(SA)^H}$) set of history-dependent policies, which is a non-trivial challenge to overcome.

Given a dataset of the first t trajectory samples $\{\tau_i\}_{i=1}^t$ and an index $h \in [H]$, we consider the following:

Estimates for POR-UCBVI:

$$\widehat{f}_h^{t+1} = \arg\min_{f_h \in \mathcal{F}_h} \sum_{i=1}^{t} \left(\sigma(f_h(\tau_i[h])) - o_h^i\right)^2$$

We also use the MLE estimate for $\mathbb{P}$ after t episodes to define $\hat{\mathbb{P}}^t(\cdot \mid s,a) := \frac{N_t(s,a,s')}{N_t(s,a)}$. Now for $\zeta(n,\delta) = 2\sqrt{\frac{S\log(2)+\log(n(n+1)SA/\delta)}{2n}}$, define $\mathcal{C}_{\mathbb{P}_t}(\delta)$ as below:

$$\left\{\mathbb{P}\middle| \|\mathbb{P}(\cdot \mid s,a) - \hat{\mathbb{P}}_t(\cdot \mid s,a)\|_1 \le \zeta(N_t(s,a),\delta) \forall s,a\right\}$$

Recall the definition of our bonus below.

Bonuses for POR-UCBVI. Recall that simple least squares guarantees imply the lemma below.
Lemma C.4.1 (Concentration for $\sigma \circ f_h$)**.** *Define*

$$\mathrm{MSE}_{h,t}(f_h, f_h') := \sum_{i=1}^{t} \left(\sigma_h(f_h(\tau_i[h])) - \sigma_h(f_h'(\tau_i[h]))\right)^2$$

Also define $\bar{\beta}_{h,t}(\delta) = \eta_h^2 \log\left(\frac{N\left(\mathcal{F}_h, \frac{B}{T}, \|\cdot\|_\infty\right)}{\delta}\right) + \alpha_{h,t}$ *with* $\alpha_{h,t} := \frac{tB + t\eta_h \log\left(\frac{t}{\delta}\right)}{T}$. *Then* $f_h^\star$ *simultaneously satisfies* $\mathrm{MSE}_{h,t}(f_h^\star, \widehat{f}_h^{(t+1)}) \le \bar{\beta}_{h,t}\left(\frac{\delta}{2t^2H}\right)$ *for all* h,t *with probability* $1-\delta/32$.

We use the data from trajectories $\{\tau_i\}_{i=1}^t$ to build the confidence sets $\mathcal{C}_{\mathcal{F}}^{t+1}(\delta) = \prod_h \mathcal{C}_h^{t+1}(\delta)$ with $\mathcal{C}_h^{t+1}(\delta)$ defined below, where $\beta_{h,t}(\delta) := \bar{\beta}_{h,t}\left(\frac{\delta}{2t^2H}\right)$.

$$\mathcal{C}_h^{t+1}(\delta) := \left\{f_h \in \mathcal{F}_h \middle| \mathrm{MSE}_{h,t}(f_h^\star, \widehat{f}_h^{(t+1)}) \le \beta_{h,t}(\delta)\right\}$$

We first define a trajectory dependent bonus term below, with $\bar{\delta} := \frac{\delta}{HS^H A^H}$

$$\gamma_{h,t}(\tau[h], \delta) = \max_{f_h, f'_h \in \mathcal{C}_h^t(\bar{\delta})} f_h(\tau[h]) - f'_h(\tau[h])$$

Note that according to the definition of β, this does not create any exponential dependence in the confidence intervals used to define $\mathcal{C}_h^{t+1}$.

$$\begin{aligned}\beta_{h,t}\left(\frac{\delta}{16S^H A^H}\right) &\leq 64\left(\log(N(\mathcal{F}_h, \alpha, \|\cdot\|_\infty)) + B + \eta_h \log(1/\delta) + \eta_h^2 H \log(THSA/\delta)\right) \\ &= O(d_{C,h} + H)\end{aligned}$$

It follows by a union bound over all trajectory segments and all timesteps t that with probability at least $1 - \delta/16$ and for any trajectory τ and $t \geq 1, h \in \mathcal{H}_p$,

$$|f_h^\star(\tau[h]) - \hat{f}_h^t(\tau[h])| \leq \gamma_{h,t}(\tau[h], \delta) \tag{C.3}$$

Remark 20. In the case of many popular function classes $\mathcal{F}$, like the linear class $\mathcal{F}_H = \{\tau \mapsto \phi(\tau)^\top \mathbf{w} \mid \|\mathbf{w}\| \leq W\}$, we can compute $\gamma_{h,t}(\tau[h], \delta)$ quite easily. In this case $\gamma_{H,t}$ is given by

$$\sup_{\mathbf{w}, \mathbf{w}' \in W_t} \phi(\tau)^\top (\mathbf{w} - \mathbf{w}') = \|\phi(\tau)\|_{V_t} \sup_{\mathbf{w}, \mathbf{w}' \in W_t} \|w - w'\|_{V_t^{-1}}$$

for a suitable quadratic form V_t.

$\gamma_{h,t}(\tau[h], \delta)$ induces a trajectory-dependent bonus, given by

$$b_{\mathcal{F}}^t(\tau, \delta) := \sum_{h \in \mathcal{H}_p} \gamma_{h,t}(\tau[h], \delta)$$

This in turn induces a policy-level bonus (which depends on the transition kernel), given by:

$$b_{\mathcal{F}}^t(\mathbb{P}, \pi, \delta) := \mathbb{E}_{\tau \sim \mathbb{P}^\pi}\left[b_{\mathcal{F}}^t(\tau, \delta)\right] = \mathbb{E}_{\tau \sim \mathbb{P}^\pi}\left[\sum_{h \in \mathcal{H}_p} \gamma_{h,t}(\tau[h], \delta)\right]$$

Let us define a term $\xi^t(s, a, \delta)$ that will be used to define the probability bonus.

$$\xi^t(s, a, \delta) := \min\left(2, 4\sqrt{\frac{H \log(6HSA) + S \log(8t^2 H^2) + \log(32t^2 N_t(s,a)/\delta)}{2N_t(s,a)}}\right)$$

This induces a trajectory-dependent bonus, given by

$$b^t_{\mathcal{P}}(\tau, \delta) := \sum_{h=1}^{H-1} \xi^t(s_h, a_h, \delta)$$

This induces a policy-level bonus (which depends on the transition kernel), given by:

$$b^t_{\mathcal{P}}(\mathbb{P}, \pi, \delta) := \min\left(4, \mathbb{E}_{\tau\sim\mathbb{P}^\pi}\left[b^t_{\mathcal{F}}(\tau, \delta)\right]\right) = \min\left(4, \mathbb{E}_{\tau\sim\mathbb{P}^\pi}\left[\sum_{h=1}^{H-1} \xi^t(s_h, a_h, \delta)\right]\right)$$

Estimates and Bonuses in case $\mathbb{P}_\star$ is known. If $\mathbb{P}_\star$ is instead known, keep $\hat{f}^t$ and $b^t_{\mathcal{F}}(\mathbb{P}, \pi, \delta)$ the same as above, but set $\hat{\mathbb{P}}_t := \mathbb{P}_\star$ and $b^t_{\mathcal{P}}(\mathbb{P}, \pi, \delta) := 0$ for all t.

For completeness we state POR-UCBVI here, which is an instantiation of Algorithm 22, the generic optimistic algorithm using bonuses.

Algorithm 25 POR-UCBVI

1: **Input** Known family of reward functions $\{\mathcal{R}_h\}_{h=1}^H$, methods $\texttt{Est}(\mathcal{D})$ to estimate $\hat{\mathbb{P}}_t$ and $\hat{f}^t$ from dataset $\mathcal{D}$, confidence level δ

2: **Initialize** $\mathcal{D}_1 \leftarrow \{\}$, initialize $\hat{f}^{\mathcal{D}_1}, \hat{\mathbb{P}}_{\mathcal{D}_1}$ arbitrarily.

3: **for** $t = 1, ..., T$ **do**

4: **Compute** optimistic history dependent policy,

$$\pi_t = \arg\max_\pi V(\hat{\mathbb{P}}_t, \hat{f}^t, \pi) + b^t_{\mathcal{F}}(\hat{\mathbb{P}}_t, \pi, \delta) + z(Bp)(b^t_{\mathcal{P}}(\hat{\mathbb{P}}_t, \pi, \delta))$$

5: **Observe** trajectory $\tau_t = \{(s^t_h, a^t_h)\}_{h=1}^H$ and feedback $\{o_h\}_{h\in\mathcal{H}_p}$.

6: **Compute** new estimates $\hat{f}^{t+1}, \hat{\mathbb{P}}_{t+1} \leftarrow \texttt{Est}(\mathcal{D}_{t+1})$ and compute new bonus functions $b^{t+1}_{\mathcal{F}}(\hat{f}^{t+1}, \cdot, \delta), b^{t+1}_{\mathcal{P}}(\hat{\mathbb{P}}_{t+1}, \cdot, \delta)$.

7: **end for**

We will show the following regret bound.

Theorem C.5.1 (POR-UCBVI Regret)**.** *Under Assumption 9, POR-UCBVI satisfies Assumption 14 and its regret* $\mathrm{Regret}(T)$ *is bounded by the following with probability at least* $1 - \delta$*, ignoring polynomial terms independent of* T*.*

$$\widetilde{\mathcal{O}}\left(\left(pC(H, S, A) + \sum_{h\in\mathcal{H}_p} \sqrt{d_{E,h}(d_{C,h} + H)}\right)\sqrt{T}\right)$$

where $C(H,S,A) := H\sqrt{SA} + S\sqrt{HA}$

C.5.1 Showing that Assumption 14 is Satisfied

C.5.1.1 Bounding effect of error in $\mathcal{F}$

Lemma C.5.2 (Bounding $\hat{f}^t$ Value Error)**.** *Given any* $\mathbb{P}$*, with* $\hat{f}^t$ *computed using data from* $\{\tau_i\}_{i=1}^t \sim \mathbb{P}_\star^{\pi_i}$ *for any sequence of policies* π_i *using least squares, the following holds with probability* $1-\delta/16$ *uniformly over all* π*.*

$$|V(\mathbb{P}, \hat{f}^t, \pi) - V(\mathbb{P}, f^\star, \pi)| \le b_{\mathcal{F}}^t(\mathbb{P}, \pi, \delta)$$

Proof. Recall that with probability at least $1-\delta/16$, the following holds for any trajectory τ and any $t \ge 1, h \in \mathcal{H}_p$.

$$|f_h^\star(\tau[h]) - \hat{f}_h^t(\tau[h])| \le \gamma_{h,t}(\tau[h], \delta) \tag{C.4}$$

Now note the following inequalities, where (i) holds with probability $1-\delta/16$ uniformly over all policies due to inequality C.4 above.

$$\begin{aligned} V(\mathbb{P}, \hat{f}^t, \pi) - V(\mathbb{P}, f^\star, \pi) &= \mathbb{E}_{\tau\sim\mathbb{P}^\pi}\left[\sum_{h\in\mathcal{H}_p} \hat{f}_h^t(\tau[h]) - f_h^\star(\tau[h])\right] \\ &\overset{(i)}{\le} \mathbb{E}_{\tau\sim\mathbb{P}^\pi}\left[\sum_{h\in\mathcal{H}_p} \gamma_{h,t}(\tau[h], \delta)\right] \\ &= b_{\mathcal{F}}^t(\mathbb{P}, \pi, \delta) \end{aligned}$$

□

Lemma C.5.3 (Bounding Sum of $\mathcal{F}$ Bonuses)**.** *The following holds with probability* 1*.*

$$\sum_{t=1}^T b_{\mathcal{F}}^t(\mathbb{P}_\star, \pi_t, \delta) = \widetilde{\mathcal{O}}\left(\sum_{h\in\mathcal{H}_p} Bd_{E,h} + \sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}\beta_{h,T}(\bar{\delta})T} + Bp\sqrt{T\log(T/\delta)}\right)$$

Proof. First note the following inequality, which hold with probability $1-\delta/16$ by the Azuma-

Hoeffding inequality.

$$\sum_{t=1}^{T} b^t_{\mathcal{F}}(\mathbb{P}_\star, \pi_t, \delta) = \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_t}_\star}\left[\sum_{h\in\mathcal{H}_p} \gamma_{t,h}(\tau[h], \delta)\right]$$
$$\leq \sum_{t=1}^{T}\sum_{h\in\mathcal{H}_p} \gamma_{t,h}(\tau[h], \delta) + \mathcal{O}\left(Bp\sqrt{T\log(T/\delta)}\right)$$

Now apply Lemma 3 in [Chan et al., 2021] for each h separately, with the function class in the lemma set to $\{\sigma \circ f_h | f_h \in \mathcal{F}_h\}$, $P = 1$, $\mathbf{x}_{t,p} = \mathbf{x}_{t,1} = \tau_t[h]$ and misspecification $\epsilon = 0$ (decoupled from the Eluder dimension's ϵ). We also note that o^t_h are η-subgaussian samples with mean $\sigma(f_h(\tau[h]))$. We obtain

$$\sum_{t=1}^{T} \max_{f_h, f'_h \in \mathcal{C}^t_h(\delta)} \sigma(f_h(\tau[h])) - \sigma(f'_h(\tau[h])) \leq \mathcal{O}\left(Bd_{E,h} + \sqrt{d_{E,h}\beta_{h,T}(\bar{\delta})T}\right)$$

where $d_{E,h} = \dim_E\left(\mathcal{F}_h, \frac{B}{T}\right)$ is the Eluder dimension of $\mathcal{F}_h$ and $\beta_{h,T}(\bar{\delta}) = \bar{\beta}\left(\frac{\bar{\delta}}{2t^2H}\right)$. Since the Lipschitz constant of σ^{-1} is κ_2, we have that the following holds with probability 1.

$$\sum_{t=1}^{T} \gamma_{t,h}(\tau[h], \delta) \leq \mathcal{O}\left(B\kappa_2 d_{E,h} + \kappa_2\sqrt{d_{E,h}\beta_{h,T}(\bar{\delta})T}\right)$$

This implies that the following holds with probability $1 - \delta/16$.

$$\sum_{t=1}^{T} b^t_{\mathcal{F}}(\mathbb{P}_\star, \pi_t, \delta) \leq \sum_{t=1}^{T}\sum_{h\in\mathcal{H}_p} \gamma_{t,h}(\tau[h], \delta) + \mathcal{O}\left(Bp\sqrt{T\log(T/\delta)}\right)$$
$$= \widetilde{\mathcal{O}}\left(\sum_{h\in\mathcal{H}_p} B\kappa_2 d_{E,h} + \sum_{h\in\mathcal{H}_p} \kappa_2\sqrt{d_{E,h}\beta_{h,T}(\bar{\delta})T} + Bp\sqrt{T\log(T/\delta)}\right)$$

□

C.5.1.2 Bounding effect of error in $\mathcal{P}$

We now restate Lemma B.2 of Chatterji et al. [2021] in our notation.

Lemma C.5.4 (Change of Measure Inequality)**.** *For any function μ of trajectories bounded by D, if $\hat{\mathbb{P}}_t$ is computed from data that includes trajectories $\{\tau_i \sim \mathbb{P}^{\pi_i}_\star\}_{i=1}^t$ for any sequence of policies π_i,*

then the following holds uniformly over all policies π with probability $1-\delta/16$.

$$\mathbb{E}_{\tau\sim\mathbb{P}_\star^\pi}[\mu(\tau)] - \mathbb{E}_{\tau\sim\hat{\mathbb{P}}_t^\pi}[\mu(\tau)] \le 2D\sqrt{\log(D)}b_{\mathcal{P}}^t(\hat{\mathbb{P}}_t,\pi,\delta)$$

The same statement holds if we switch the roles of $\mathbb{P}$ and $\hat{\mathbb{P}}_t$ on both sides.

Proof. For the order of $\mathbb{P}$ and $\hat{\mathbb{P}}_t$ in the statement, the following follows from Lemma B.2 of Chatterji et al. [2021] with $\eta = D$ and $\epsilon = \frac{1}{t^2}$. We pull the additive $\log(D)$ in the square root outside to fit our assumption's phrasing.

$$\mathbb{E}_{\tau\sim\mathbb{P}_\star}[\mu(\tau)] - \mathbb{E}_{\tau\sim\hat{\mathbb{P}}_t}[\mu(\tau)] \le D\sqrt{\log(D)}b_{\mathcal{P}}^t(\hat{\mathbb{P}}_t,\pi,\delta) + \frac{1}{t^2} \le 2D\sqrt{\log(D)}b_{\mathcal{P}}^t(\hat{\mathbb{P}}_t,\pi,\delta)$$

The only subtlety is that more data than that from $\{\tau_i\}_{i=1}^t$ could have been used to compute $\hat{\mathbb{P}}_t$. The proof still follows since $c_{\mathcal{P}}(\hat{\mathbb{P}}_t,\pi,D)$ is decreasing in the counts $N_t(s,a)$.

Finally, if we switch $\mathbb{P}$ and $\hat{\mathbb{P}}_t$ on both sides, we can follow the proof of Lemma B.2 verbatim with $\mathbb{P}$ and $\hat{\mathbb{P}}_t$ switched everywhere, except for the martingale argument. There, instead of switching the two transition kernels, we negate the martingale to get our desired result. This exception is because we still need the expectation to be over the true transition kernel $\mathbb{P}_\star$ for the stochastic process defined to be a martingale. □

Lemma C.5.5 (Bounding $\hat{\mathcal{P}}_t$ Value Error)**.** *Consider any sequence of functions f^t that induce value functions $V(\mathbb{P}, f^t, \pi)$. For any sequence of policies π_t, if the estimates $\hat{\mathbb{P}}_t$, bonuses $b_{\mathcal{P}}$ and costs $c_{\mathcal{P}}$ are generated using data including that of $\tau_i \sim \mathbb{P}_\star^{\pi_i}, i = 1 \to t$, then the following holds uniformly over t and over all policies with probability $1-\delta/16$.*

$$V(\hat{\mathbb{P}}_t, f^t, \pi) - V(\mathbb{P}_\star, f^t, \pi) = Bp\sqrt{\log(Bp)}(b_{\mathcal{P}}^t(\mathbb{P}_\star,\pi,\delta))$$

The statement also holds if we switch $\hat{\mathbb{P}}_t$ and $\mathbb{P}_\star$.

Proof. Note the following two inequalities that immediately follow from Lemma C.5.4

$$V(\mathbb{P}_\star, f^t, \pi) - V(\hat{\mathbb{P}}_t, f^t, \pi) \le Bp\sqrt{\log(Bp)}(b_{\mathcal{P}}^t(\hat{\mathbb{P}}_t,\pi,\delta))$$
$$V(\hat{\mathbb{P}}_t, f^t, \pi) - V(\mathbb{P}_\star, f^t, \pi) \le Bp\sqrt{\log(Bp)}(b_{\mathcal{P}}^t(\mathbb{P}_\star,\pi,\delta))$$

Our result follows immediately. □

Lemma C.5.6 (Bounding Sum of $\mathcal{P}$ Bonuses)**.** *The following holds with probability $1-\delta/16$*

whenever the data used to compute $b^t_{\mathcal{P}}(\mathbb{P}, \pi, \delta)$ includes the data of trajectories $\tau_i, t = 1 \to t$.

$$\sum_{t=1}^{T} b^t_{\mathcal{P}}(\mathbb{P}_\star, \pi_t, \delta) = \widetilde{\mathcal{O}}\left(SA\bar{c}_\delta + \bar{c}_\delta\sqrt{HSAT}\right)$$

where

$$\bar{c}_\delta := 4\sqrt{\frac{H\log(6HSA) + S\log(8t^2H^2) + \log(32t^2N_T(s,a)/\delta)}{2}}$$

This means that for any s, a, $\xi^t(s, a, \delta) = 2$ until $N_t(s, a) \geq \frac{\bar{c}_\delta}{2}$

Proof. First note that by the definition of the bonus and the Azuma-Hoeffding inequality, we have the following.

$$\begin{aligned}
\sum_{t=1}^{T} b^t_{\mathcal{P}}(\mathbb{P}_\star, \pi_t, \delta) &\leq \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\mathbb{P}^\pi} b^t_{\mathcal{P}}(\tau, \delta) \\
&\leq \sum_{t=1}^{T} b^t_{\mathcal{P}}(\tau_t, \delta) + \mathcal{O}(4\sqrt{T\log(T/\delta)}) \\
&= \sum_{t=1}^{T}\sum_{h=1}^{H-1} \xi^t(s^t_h, a^t_h, \delta) + \mathcal{O}(4\sqrt{T\log(T/\delta)})
\end{aligned}$$

Now note that the first inequality holds even if more data beyond that of $\{\tau_i\}_{i=1}^t$ is used to compute $\xi^t(s, a, \delta)$, since $\xi^t(s, a, \delta)$ is decreasing in $N_t(s, a)$.

$$\begin{aligned}
\sum_{t=1}^{T}\sum_{h=1}^{H-1} \xi^t(s^t_h, a^t_h, \delta) &\leq SA\bar{c}_\delta + \sum_{t=1}^{T}\sum_{h=1}^{H-1} \frac{\bar{c}_\delta}{\sqrt{N_t(s^{(t)}_h, a^{(t)}_h)}} \\
&\leq SA\bar{c}_\delta + \bar{c}_\delta \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{l=1}^{N_T(s,a)} \frac{1}{\sqrt{l}} \\
&\leq SA\bar{c}_\delta + 2\bar{c}_\delta \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sqrt{N_T(s,a)} \\
&\leq SA\bar{c}_\delta + 2\bar{c}_\delta \sqrt{SA \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} N_T(s,a)} \\
&= \mathcal{O}\left(SA\bar{c}_\delta + \bar{c}_\delta\sqrt{SATH}\right)
\end{aligned}$$

This concludes our proof, since $1 = \widetilde{\mathcal{O}}(\sqrt{HSA})$ □

C.5.1.3 Putting everything together

Theorem C.5.7 (POR-UCBVI Regret)**.** *Under Assumption 9, POR-UCBVI satisfies Assumption 14 and its regret* $\mathrm{Regret}(T)$ *is bounded by the following with probability at least* $1-\delta$*, ignoring polynomial terms independent of* T*.*

$$\widetilde{\mathcal{O}}\left(\left(pC(H,S,A)+\sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}(d_{C,h}+H)}\right)\sqrt{T}\right)$$

where $C(H,S,A) := H\sqrt{SA}+S\sqrt{HA}$

Proof. Note that by Lemmas C.5.2, C.5.3, C.5.4, C.5.5, C.5.6, Assumption 14 is satisfied by POR-UCBVI. Using Theorem C.3.2 and Lemmas C.5.3 and C.5.6, we have the following.

$$\begin{aligned}\mathrm{Regret}(T) = \mathcal{O}\Big(&\sum_{h\in\mathcal{H}_p} B\kappa_{2,h}d_{E,h} + BpHSA\bar{c}_\delta + \sum_{h\in\mathcal{H}_p}\kappa_{2,h}\sqrt{d_{E,h}\beta_{h,T}(\bar{\delta})T}\\ &+ Bp\sqrt{T\log(T/\delta)} + \bar{c}_\delta BpH\sqrt{SAT}\Big)\end{aligned}$$

We further refine it grouping terms and ignoring terms independent of T, and also noting that $\bar{c}_\delta = \widetilde{\mathcal{O}}(\sqrt{H}+\sqrt{S})$ as well as $\beta_{h,T}(\bar{\delta}) = \mathcal{O}(d_{C,h}+H)$

$$\begin{aligned}\mathrm{Regret}(T) &= \widetilde{\mathcal{O}}\left(\left(\sum_{h\in\mathcal{H}_p}\kappa_2\sqrt{d_{E,h}(d_{C,h}+H)} + Bp(H\sqrt{SA}+S\sqrt{HA})\right)\sqrt{T}\right)\\ &= \mathcal{O}\left(\left(p(H\sqrt{SA}+S\sqrt{HA}) + \sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}(d_{C,h}+H)}\right)\sqrt{T}\right)\end{aligned}$$

□

From the latter we derive a sample complexity result as follows.

Corollary C.5.8 (POR-UCBVI Sample complexity)**.** *Let* $\epsilon>0, \delta\in[0,1]$*. Ignoring polynomial terms independent of* ϵ*, we can bound the sample complexity* $N(\epsilon,\delta)$ *of POR-UCBVI as follows*

$$\widetilde{\mathcal{O}}\left(\frac{p^2HSA\max(H,S)}{\epsilon^2} + \frac{p^2d_E\max(d_C,H)\log(1/\delta)}{\epsilon^2}\right)$$

where $d_E := \max_{h\in\mathcal{H}_p} d_{E,h}$*, and* $d_C := \max_{h\in\mathcal{H}_p} d_{C,h}$*.*

Proof. We invoke the regret-to-PAC conversion in Lemma C.2.1 with confidence $\delta' = \delta/2$ and we plug the regret bound in Theorem C.5.1 to write

$$\epsilon = \widetilde{\mathcal{O}}\left(\left(Bp(H\sqrt{SA} + S\sqrt{HA}) + \sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}\beta_{h,T}(\bar{\delta})} + Bp\sqrt{\log(1/\delta)}\right)\left(\frac{1}{\sqrt{T}}\right)\right)$$

from which we get the result by noting $N = pT$ and the definition of d_E, d_C. □

We also have the following theorem and corollary, in the same vein as Theorem C.4.10.

Theorem C.5.9 (Regret for POR-UCBVI if $\mathbb{P}_\star$ is Known)**.** *When $\mathbb{P}_\star$ is known, POR-UCBVI that sets $\hat{\mathbb{P}}_t := \mathbb{P}_\star$ and $b^t_{\mathcal{P}}(\mathbb{P}, \pi\delta) := 0$ for all $t \geq 1$ still satisfies Assumption 14 and its regret $\mathrm{Regret}(T)$ is bounded by the following with probability at least $1-\delta$, ignoring terms independent of T.*

$$\widetilde{\mathcal{O}}\left(\left(\sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}(d_{C,h}+H)}\right)\sqrt{T}\right)$$

where $d_{E,h} = \dim_E\left(\mathcal{F}_h, \frac{B}{T}\right)$.

Proof. Note that by Lemmas C.5.2 and C.5.3, Assumption 14 is satisfied by POR-UCBVI. Using Lemma C.5.3, we have the following.

$$\mathrm{Regret}(T) = \mathcal{O}\left(\sum_{h\in cH_p} B\kappa_2 d_{E,h} + \sum_{h\in\mathcal{H}_p}\kappa_2\sqrt{d_{E,h}\beta_{h,T}(\delta)T} + Bp\sqrt{T\log(T/\delta)}\right)$$

We further refine it grouping terms and ignoring terms independent of T, and also noting that $\bar{c}_\delta = \widetilde{\mathcal{O}}(\sqrt{H} + \sqrt{S})$

$$\begin{aligned}
\mathrm{Regret}(T) &= \widetilde{\mathcal{O}}\left(\left(\sum_{h\in\mathcal{H}_p}\kappa_2\sqrt{d_{E,h}\beta_{h,T}(\delta)} + Bp\right)\sqrt{T}\right) \\
&= \mathcal{O}\left(\left(Bp + \sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}\beta_{h,T}(\delta)}\right)\sqrt{T}\right) \\
&= \widetilde{\mathcal{O}}\left(\left(\sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}(d_{C,h}+H)}\right)\sqrt{T}\right)
\end{aligned}$$

□

Corollary C.5.10 (POR-UCBVI Sample complexity if $\mathbb{P}_\star$ is Known)**.** *Let $\epsilon > 0, \delta \in [0,1]$. Ignoring polynomial terms independent of ϵ, we can bound the sample complexity $N(\epsilon, \delta)$ of POR-UCBVI when $\mathbb{P}_\star$ is known as follows*

$$\widetilde{\mathcal{O}}\left(\frac{p^2 H d_E \max(d_C, H)}{\epsilon^2}\right)$$

where $d_E := \max_{h \in \mathcal{H}_p} d_{E,h}$, $\beta := \max_{h \in \mathcal{H}_p} \beta_{T,h}(\delta)$, and $d_C := \max_{h \in \mathcal{H}_p} d_{C,h}$.

Proof. The proof proceeds as in Corollary C.5.8 by plugging Theorem C.5.9 in Lemma C.2.1. □

C.6 Extension to General Function Approximation for Probability Transitions

Note that for simplicity of exposition and proofs, the confidence sets for $\mathcal{P}$ in POR-UCRL and the bonuses for $\mathcal{P}$ in POR-UCBVI are given for tabular observed states and actions s, a. The function approximation using $\mathcal{G}$ and $\mathcal{F}$ is relegated to the effect of latent states u. Using the work of Liu et al. [2022b], it is quite straightforward to extend and modify the proofs here to general function approximation for $\mathcal{P}$.

Let us assume that we have a set of parameters Θ and a mapping from $\theta \in \Theta$ to probability transition functions $\mathbb{P}_\theta \in \mathcal{P}$ so that the image of Θ under this mapping is all of $\mathcal{P}$. These functions $\mathbb{P}_\theta$ mapping from s, a to distributions over s' act on a featurization $\phi(s, a)$ of s, a. This is the most general function approximation setting for probability transition functions. Linear MDPs, tabular MDPs, factored MDPs, kernel linear MDPs, and many other function approximation settings are special cases of this.

For convenience of notation, we will occasionally drop the subscript θ. It is beneficial to remember that throughout this section, we are working under general function approximation nevertheless.

C.6.1 POR-UCRL

For POR-UCRL, we redefine the confidence sets to be the ones in Algorithm 2 (reward-free OMLE) in Liu et al. [2022b]. That is, we first recursively define $\mathcal{D}_{t+1} := \mathcal{D}_t \cup \{(\pi_t, \tau_t)\}$ instead of just appending τ_t. We also recursively define $\mathcal{C}_{\mathcal{P}}(\mathcal{D}_{t+1}, \delta)$ below. Note that we merely rephrased the

definition in Liu et al. [2022b] to use the negative log-likelihood instead of the log-likelihood.

$$\mathcal{C}_{\mathcal{P}}(\mathcal{D}_{t+1}, \delta) := \\ \mathcal{C}_{\mathcal{P}}(\mathcal{D}_t, \delta) \cap \left\{ \theta \in \Theta \middle| \sum_{(\pi,\tau)\in\mathcal{D}_{t+1}} -\log(\mathbb{P}_\theta^\pi(\tau)) \leq \min_\theta \sum_{(\pi,\tau)\in\mathcal{D}_{t+1}} -\log(\mathbb{P}_\theta^\pi(\tau)) + \beta \right\}$$

We now simply need to replace Lemma C.4.4 with a version involving function approximation. Recall the following.

$$\begin{aligned} V(\mathbb{P}, f^t, \pi_t) &:= \mathbb{E}_{\tau'\sim\mathbb{P}^{\pi_t}} \left[\sum_{h=1}^{H} \widetilde{f}_h^t(\tau[h]) \right] \\ &= \mathbb{E}_{\tau'\sim\mathbb{P}^{\pi_t}} \left[\sum_{h\in\mathcal{H}_p} \widetilde{f}_h^t(\tau[h]) \right] \end{aligned}$$

So, by a simple change of measure inequality and the fact that $|f_h^t| \leq B$ for all h, we have the following.

$$\left| V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t) \right| \leq Bpd_{TV}(\widetilde{\mathbb{P}}_t^{\pi_t}, \mathbb{P}_\star^{\pi_t}) \tag{C.5}$$

Where d_{TV} represents the TV distance taken over all trajectories of length H. This implies the following.

$$\left| \sum_{t=1}^{T} V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t) \right| \leq Bp \sum_{t=1}^{T} d_{TV}(\widetilde{\mathbb{P}}_t^{\pi_t}, \mathbb{P}_\star^{\pi_t})$$

Now, we have two options. We can either use the distributional eluder dimension directly, or we can use the strong SAIL condition of Liu et al. [2022b]. The former is more flexible and allows us to address all function approximation scenarios. The latter approach has very crisp guarantees and is still satisfied by most function approximation models.

C.6.1.1 Use the distributional Eluder dimension

Note that by proposition B.2 and step 2 of the proof of Theorem 3.2 in Liu et al. [2022b], we have the following upon setting $\Pi_{\text{exp}}(\pi) := \pi$ in their proofs. This way, we are not running any extra

"exploratory" policies at every step.

$$\sum_{k=1}^{t-1} d_{TV}^2(\widetilde{\mathbb{P}}_t^{\pi_k}, \mathbb{P}_\star^{\pi_k}) \leq \mathcal{O}(\beta) \tag{C.6}$$

Here, $\beta = c\log(\mathcal{N}_{br,\mathcal{P}}(1/T)) + c\log(T/\delta)$, where $\mathcal{N}_{br,\mathcal{P}}(\epsilon)$ is the ϵ-bracketing number of $\mathcal{P} = \{\mathbb{P}_\theta\}_{\theta\in\Theta}$ as in definition 2.2 of Liu et al. [2022b], and c is a universal constant.

Now, consider the functions ϕ_t defined on the space of policies by $\phi_t(\pi) := d_{TV}(\widetilde{\mathbb{P}}_t^{\pi}, \mathbb{P}_\star^{\pi})$. Define μ_t to be the Dirac-delta measures on π_t. Let the class of Dirac-delta measures over policies be $\mathcal{D}_\Pi$, and let the class of possible ϕ_t functions be Φ. Let $d_{E,\mathcal{P}}$ be the corresponding distributional Eluder dimension $\dim_{DE}(\Phi, \mathcal{D}_\Pi, \sqrt{1/T})$, as defined in Jin et al. [2021a]. Then by equation C.6 and the properties of the distributional Eluder dimension (Lemma 41 of Jin et al. 2021a), we have the following.

$$\sum_{t=1}^{T} d_{TV}(\widetilde{\mathbb{P}}_t^{\pi_t}, \mathbb{P}_\star^{\pi_t}) = \widetilde{\mathcal{O}}(\sqrt{d_{E,\mathcal{P}}\beta T})$$

We can further define $d_{C,\mathcal{P}} = \log(\mathcal{N}_{br,\mathcal{P}}(1/T))$ to be the bracketing dimension of $\mathcal{P}$. By equation C.5, we have the following.

$$\left|V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t)\right| \leq \widetilde{\mathcal{O}}(\sqrt{d_{E,\mathcal{P}}\beta T})$$

We can now follow the rest of the proof for the guarantee for POR-UCRL verbatim and establish the following bound on POR-UCRL regret.

Theorem C.6.1. *Consider the functions ϕ_t defined on the space of policies by $\phi_t(\pi) := d_{TV}(\widetilde{\mathbb{P}}_t^{\pi}, \mathbb{P}_\star^{\pi})$. Define μ_t to be the Dirac-delta measures on π_t. Let the class of Dirac-delta measures over policies be $\mathcal{D}_\Pi$, and let the class of possible ϕ_t functions be Φ. Let $d_{E,\mathcal{P}}$ be the corresponding distributional Eluder dimension $\dim_{DE}(\Phi, \mathcal{D}_\Pi, \sqrt{1/T})$. Define $d_{C,\mathcal{P}} = \log(\mathcal{N}_{br,\mathcal{P}}(1/T))$ to be the bracketing dimension of $\mathcal{P}$. Then, Then, we have the following bound for the regret of our modified POR-UCRL algorithm, with probability $1-\delta$.*

$$\text{Regret}(T) = \widetilde{\mathcal{O}}\left(\left(p\sqrt{d_{E,\mathcal{P}}d_{C,\mathcal{P}}} + \sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}d_{C,h}}\right)\sqrt{T}\right)$$

Future work can investigate the choice of the distribution class $\mathcal{D}_\Pi$ and make other technical tweaks to obtain crisper versions of the $d_{E,\mathcal{P}}$ defined here.

C.6.1.2 Use the strong SAIL condition

We make a minor modification to the POR-UCRL algorithm in the spirit of Algorithm 2 (reward-free OMLE) in Liu et al. [2022b] to use this condition. Instead of merely running π_t, we run all policies in an exploratory set $\Pi_{\exp}(\pi_t)$ of size $\Pi_{\exp}$ every time we collect trajectories. Recall that the strong SAIL condition of Liu et al. [2022b] is satisfied by well conditioned PSRs, factored MDPs, kernel linear MDPs, sparse linear bandits, etc. These automatically subsume tabular MDPs and linear MDPs.

Assume that our model class $\mathcal{P}$ satisfies the $(\sqrt{d_{S,\mathcal{P}}}, \kappa, C)$ strong SAIL condition. That is, we define $d_{S,\mathcal{P}} := d^2$ for the d in the original strong SAIL condition. Now by theorem 7.2 of Liu et al. [2022b], we can conclude that since $\widetilde{\mathbb{P}}_t \in \mathcal{C}_{\mathcal{P}}(\mathcal{D}_t, \delta)$, we have that

$$d_{TV}(\widetilde{\mathbb{P}}_t^{\pi_t}, \mathbb{P}_\star^{\pi_t}) \leq poly(H)\sqrt{d_{S,\mathcal{P}}}\left(\frac{C|\Pi_{\exp}|}{t} + \kappa\sqrt{\frac{\beta|\Pi_{\exp}|^2}{t}}\log^2(t/|\Pi_{\exp}|\right)$$

Here, $\beta = c\log(\mathcal{N}_{br,\mathcal{P}}(1/T)) + c\log(T/\delta)$, where $\mathcal{N}_{br,\mathcal{P}}(\epsilon)$ is the ϵ-bracketing number of $\mathcal{P} = \{\mathbb{P}_\theta\}_{\theta\in\Theta}$ and c is a universal constant. Adding these up across t and discarding logarithmic terms, we get the following.

$$Bp\sum_{t=1}^{T} d_{TV}(\widetilde{\mathbb{P}}_t^{\pi_t}, \mathbb{P}_\star^{\pi_t}) = \widetilde{\mathcal{O}}\left(poly(H)Bp\kappa\sqrt{d_{S,\mathcal{P}}\beta|\Pi_{\exp}|^2T}\right)$$

This means that by equation C.5, we have the following.

$$\left|\sum_{t=1}^{T} V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t)\right| = \widetilde{\mathcal{O}}\left(poly(H)Bp\kappa\sqrt{d_{S,\mathcal{P}}\beta|\Pi_{\exp}|^2T}\right)$$

Let us also define $d_{C,\mathcal{P}} = \log(\mathcal{N}_{br,\mathcal{P}}(1/T))$ to be the bracketing dimension of $\mathcal{P}$. We can now follow the rest of the proof for the guarantee for POR-UCRL verbatim and get the following bound on POR-UCRL regret, with the caveat that we are ignoring the contribution of the non-optimal exploratory policies. We have thus established the following theorem.

Theorem C.6.2. *Assume that our model class $\mathcal{P}$ satisfies the $(\sqrt{d_{S,\mathcal{P}}}, \kappa, C)$ strong SAIL condition. Define $d_{C,\mathcal{P}} = \log(\mathcal{N}_{br,\mathcal{P}}(1/T))$ to be the bracketing dimension of $\mathcal{P}$. Then, we have the following bound for the regret of our modified POR-UCRL algorithm, with probability $1-\delta$.*

$$\mathrm{Regret}(T) = \widetilde{\mathcal{O}}\left(\left(poly(H)p\kappa\sqrt{d_{S,\mathcal{P}}d_{C,\mathcal{P}}|\Pi_{\exp}|^2} + \sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}d_{C,h}}\right)\sqrt{T}\right)$$

C.6.2 POR-UCBVI

Again, we start with confidence sets that we will use to define the bonuses. That is, we define the confidence sets to be the ones in Algorithm 2 (reward-free OMLE) in Liu et al. [2022b]. That is, we first recursively define $\mathcal{D}_{t+1} := \mathcal{D}_t \cup \{(\pi_t, \tau_t)\}$ instead of just appending τ_t. We also recursively define $\mathcal{C}_{\mathcal{P}}(\mathcal{D}_{t+1}, \delta)$ below. Note that we merely rephrased the definition in Liu et al. [2022b] to use the negative log-likelihood instead of the log-likelihood.

$$\mathcal{C}_{\mathcal{P}}(\mathcal{D}_{t+1}, \delta) := \\ \mathcal{C}_{\mathcal{P}}(\mathcal{D}_t, \delta) \cap \left\{ \theta \in \Theta \middle| \sum_{(\pi,\tau)\in\mathcal{D}_{t+1}} -\log(\mathbb{P}_\theta^\pi(\tau)) \leq \min_\theta \sum_{(\pi,\tau)\in\mathcal{D}_{t+1}} -\log(\mathbb{P}_\theta^\pi(\tau)) + \beta \right\}$$

Defining bonuses is trickier than defining confidence sets. Like the case of POR-UCRL above, we have two options again – we can use the distributional Eluder dimension or the strong SAIL condition. **While it is possible to work out the details for the former, they are much more involved and do not cover significantly more useful examples than those covered by the strong SAIL condition. So, we will focus on the latter.** This is especially true when considering sample complexity, since the use of exploratory policies under the strong SAIL condition creates no complications in giving sample complexity guarantees, unlike it does for regret guarantees.

C.6.2.1 Using the strong SAIL condition

Again, we make a minor modification to the POR-UCBVI algorithm in the spirit of Algorithm 2 (reward-free OMLE) in Liu et al. [2022b] to use this condition. Instead of merely running π_t, we run all policies in an exploratory set $\Pi_{\text{exp}}(\pi_t)$ of size Π_{exp} every time we collect trajectories. Again, recall that the strong SAIL condition of Liu et al. [2022b] is satisfied by well conditioned PSRs, factored MDPs, kernel linear MDPs, sparse linear bandits, etc. These automatically subsume tabular MDPs and linear MDPs.

Let $\hat{\mathbb{P}}_t$ be the MLE model at time t. Assume that our model class $\mathcal{P}$ satisfies the $(\sqrt{d_{S,\mathcal{P}}}, \kappa, C)$ strong SAIL condition. That is, we define $d_{S,\mathcal{P}} := d^2$ for the d in the original strong SAIL condition. Now by theorem 7.2 of Liu et al. [2022b], we can conclude that since $\mathbb{P}_t \in \mathcal{C}_{\mathcal{P}}(\mathcal{D}_t, \delta)$, we have that the following holds with probability $1 - \delta$.

$$\max_\pi d_{TV}(\hat{\mathbb{P}}_t^\pi, \mathbb{P}_\star^\pi) \leq poly(H)\sqrt{d_{S,\mathcal{P}}} \left(\frac{C|\Pi_{\text{exp}}|}{t} + \kappa\sqrt{\frac{\beta|\Pi_{\text{exp}}|^2}{t}} \log^2(t/|\Pi_{\text{exp}}|) \right)$$

Here, $\beta = c\log(\mathcal{N}_{br,\mathcal{P}}(1/T)) + c\log(T/\delta)$, where $\mathcal{N}_{br,\mathcal{P}}(\epsilon)$ is the ϵ-bracketing number of $\mathcal{P} = \{\mathbb{P}_\theta\}_{\theta\in\Theta}$ and c is a universal constant. By a simple change of measure, for any policy π, we have the following.

$$\left|V(\hat{\mathbb{P}}_t, f^t, \pi) - V(\mathbb{P}_\star, f^t, \pi)\right| \leq Bpd_{TV}(\hat{\mathbb{P}}_t^\pi, \mathbb{P}_\star^\pi)$$

holds for any policy π with probability 1, the following holds simultaneously for all policies with probability $1-\delta$.

$$\left|V(\hat{\mathbb{P}}_t, f^t, \pi) - V(\mathbb{P}_\star, f^t, \pi)\right| \leq poly(H)Bp\sqrt{d_{S,\mathcal{P}}}\left(\frac{C|\Pi_{\exp}|}{t} + \kappa\sqrt{\frac{\beta|\Pi_{\exp}|^2}{t}}\log^2(t/|\Pi_{\exp}|\right)$$

Define the right hand side as the bonus $b_{\mathcal{P}}^t(\mathbb{P}, \pi, \delta)$. Adding these up across t and discarding logarithmic terms, we get the following.

$$\sum_{t=1}^{T} b_{\mathcal{P}}^t(\mathbb{P}, \pi, \delta) = \widetilde{\mathcal{O}}\left(poly(H)Bp\kappa\sqrt{d_{S,\mathcal{P}}\beta|\Pi_{\exp}|^2 T}\right)$$

This means that by equation C.5, we have the following.

$$\left|\sum_{t=1}^{T} V(\widetilde{\mathbb{P}}_t, f^t, \pi_t) - V(\mathbb{P}_\star, f^t, \pi_t)\right| = \widetilde{\mathcal{O}}\left(poly(H)Bp\kappa\sqrt{d_{S,\mathcal{P}}\beta|\Pi_{\exp}|^2 T}\right)$$

Let us also define $d_{C,\mathcal{P}} = \log(\mathcal{N}_{br,\mathcal{P}}(1/T))$ to be the bracketing dimension of $\mathcal{P}$. We can now follow the rest of the proof for the guarantee for POR-UCBVI verbatim and get a bound on POR-UCBVI regret, with the caveat that we are ignoring the contribution of the non-optimal exploratory policies. We have thus established the following theorem.

Theorem C.6.3. *Assume that our model class $\mathcal{P}$ satisfies the $(\sqrt{d_{S,\mathcal{P}}}, \kappa, C)$ strong SAIL condition. Define $d_{C,\mathcal{P}} = \log(\mathcal{N}_{br,\mathcal{P}}(1/T))$ to be the bracketing dimension of $\mathcal{P}$. Then, we have the following bound for the regret of our modified POR-UCBVI algorithm, with probability $1-\delta$.*

$$\mathrm{Regret}(T) = \widetilde{\mathcal{O}}\left(\left(poly(H)p\kappa\sqrt{d_{S,\mathcal{P}}d_{C,\mathcal{P}}|\Pi_{\exp}|^2} + \sum_{h\in\mathcal{H}_p}\sqrt{d_{E,h}d_{C,h}}\right)\sqrt{T}\right)$$

C.6.3 Dueling PORRL

We have the following immediate corollary of Theorem 5.4.2, Theorem C.6.1 and Lemma 5.4.3.

Corollary C.6.4 (Dueling Regret using modified POR-UCRL Confidence Sets)**.** *Consider the*

functions ϕ_t defined on the space of policies by $\phi_t(\pi) := d_{TV}(\widetilde{\mathbb{P}}_t^\pi, \mathbb{P}_\star^\pi)$. Define μ_t to be the Dirac-delta measures on π_t. Let the class of Dirac-delta measures over policies be $\mathcal{D}_\Pi$, and let the class of possible ϕ_t functions be Φ. Let $d_{E,\mathcal{P}}$ be the corresponding distributional Eluder dimension $\dim_{DE}(\Phi, \mathcal{D}_\Pi, \sqrt{1/T})$. Define $d_{C,\mathcal{P}} = \log(\mathcal{N}_{br,\mathcal{P}}(1/T))$ to be the bracketing dimension of $\mathcal{P}$. Then, the modified confidence sets for POR-UCRL described in section C.6.1.1 satisfy Assumption 10 and using them in Algorithm 10 leads to the following dueling regret bound with probability $1-\delta$.

$$\mathrm{Regret}_D(T) = \widetilde{\mathcal{O}}\left(\left(p\sqrt{d_{E,\mathcal{P}} d_{C,\mathcal{P}}} + \sum_{h\in\mathcal{H}_p} \sqrt{d_{E,h} d_{C,h}}\right)\sqrt{T}\right)$$

Similar corollaries can be immediately produced for the strong SAIL method, whenever $|\Pi_{\mathrm{exp}}| = 1$. The strong SAIL method doesn't quite work for dueling regret guarantees when $|\Pi_{\mathrm{exp}}| > 1$, so we must stick to the distributional eluder dimension in that case.

C.7 Details and Proofs for PORRL with GOLF

For completeness and establishing notation, we recall GOLF here.

Algorithm 26 GOLF

Input Known class of Bellman consistent Q-functions $\mathcal{Q}$, confidence level δ.
Initialize dataset $\mathcal{D}_1 \leftarrow \{\}$ and $\mathcal{C}_\mathcal{Q}(\mathcal{D}_1, \delta) \leftarrow \mathcal{Q}$.
for $t = 1, ..., T$ **do**
  $\tau[0] \leftarrow ()$
  for $h = 1, \dots H$ **do**
    Compute $a_h^t, Q_h^t \leftarrow \arg\max_{a, Q\in\mathcal{C}_\mathcal{Q}(\mathcal{D}_t,\delta)} Q(\tau[h], a)$
    Play a_h^t and observe feedback o_h^t
  end for
  Update $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{\tau, (o_1^t, \dots o_H^t\}$
  Compute

$$\mathcal{C}_\mathcal{Q}(\mathcal{D}_{t+1}, \delta) \leftarrow \left\{\mathcal{L}_{\mathcal{D}_t}(Q_h, Q_{h+1}) \le \inf_{g\in\mathcal{G}_h} \mathcal{L}_{\mathcal{D}_t}(g, Q_{h+1}) + \beta\right\}$$

end for

Theorem C.7.1 (Modified GOLF Regret)**.** *Let Assumption 9 hold, let $\mathcal{Q}$ be Bellman complete, and let $d_{\mathrm{HABE}} = \dim_{\mathrm{HABE}}(\mathcal{Q}, \alpha, \min(\alpha, \sqrt{1/T}))$. Choose hyperparameter $\beta = c\log(HT\mathcal{N}(\mathcal{Q} \cup$*

$\mathcal{G}, 1/T, \|\cdot\|_\infty))$ *for some universal constant* c *and the auxiliary function class* $\mathcal{G}$ *used in GOLF, and define* $d_{C,\mathcal{Q}} := \log(\mathcal{N}(\mathcal{Q} \cup \mathcal{G}, 1/T, \|\cdot\|_\infty))$. *Then, GOLF satisfies* $\mathrm{Regret}(T) = \mathcal{O}\left(pH\sqrt{d_{\mathrm{HABE}} d_{C,\mathcal{Q}} T}\right)$.

Proof. The meat of the theorem is in proving Lemma C.7.4. We $\beta = c\log(HT\mathcal{N}(\mathcal{Q} \cup \mathcal{G}, 1/T, \|\cdot\|_\infty))$ for some suitably large universal constant c, and use Theorem C.3.3 and Lemma C.7.4 to get that

$$\mathrm{Regret}(T) = \sum_{h=1}^{H}\left(T\omega + Bp\left(\frac{B^2p^2\beta}{\alpha^2}+1\right)\left(\sum_{l=1}^{h-1} d_l(\alpha)\right) + \min(d_h(\omega), T)Bp + 2Bp\sqrt{\beta d_h(\omega)T}\right)$$

where $d_h(\epsilon) := \dim_{DE}(\Phi_h, \mathcal{D}_h, \epsilon)$. Now set $\omega = \frac{Bp}{T}$ and use the fact that $d_h(\epsilon)$ increases with decreasing ϵ to get that

$$\begin{aligned}\mathrm{Regret}(T) &= \widetilde{\mathcal{O}}\left(pH\sqrt{d_{\mathrm{HABE}}\beta T}\right)\\ &= \widetilde{\mathcal{O}}\left(pH\sqrt{d_{\mathrm{HABE}} d_{C,\mathcal{Q}} T}\right)\end{aligned}$$

since $d_{\mathrm{HABE}} := \dim_{\mathrm{HABE}}(\mathcal{Q}, \min(\alpha, Bp/T)) := \max_h d_h(\min(\alpha, Bp/T))$. □

Corollary C.7.2 (GOLF Sample complexity)**.** *Let* $\epsilon > 0, \delta \in [0,1]$. *Ignoring polynomial terms independent of* ϵ*, we can bound the sample complexity* $N(\epsilon, \delta)$ *of GOLF as follows*

$$\widetilde{\mathcal{O}}\left(\frac{p^2H^2 d_{\mathrm{HABE}} d_{c,\mathcal{Q}}}{\epsilon^2}\right).$$

Proof. Again, we use Lemma C.2.1 and a quick computation shows our result. □

C.7.1 Comparing $\dim_{\mathrm{HABE}}$ and $\dim_{\mathcal{T}}$

It is easy to see that since the function class Φ_h is a subset of the class Ψ_h of all Bellman errors, $\dim_{\mathrm{HABE}} \le \max_h \dim_{DE}(\Psi_h, \mathcal{D}_{h,\mathcal{Q}}, \epsilon)$. Recall that the Bellman eluder dimension is a minimum over the RHS and another term that uses Dirac-δ distributions, but typically, the RHS is smaller. So, in many cases, $\dim_{\mathrm{HABE}} \le \dim_{\mathcal{T}}$. However, we don't have a universal inequality in either direction.

C.7.2 Computing dimensions for the combination lock

Proposition C.7.3 (Dimensions for the Combination Lock)**.** *Consider the combination lock problem with model class $\mathcal{M} = \mathcal{P} \times \mathcal{F}$ and induced Q-function class $\mathcal{Q}$.*

- *Under dense intermediate feedback with $\mathcal{H}_p = [H]$, $\dim_{\mathrm{HABE}}(\mathcal{Q}, \alpha) = A$ for all $\alpha < q$, while its BE dimension is at least $A^H - 2$. The eluder dimension for reward functions $\dim_E(\mathcal{F}_h, \frac{B}{T})$ is at least A^h for any $h \leq H$.*
- *For sparse intermediate feedback with $\mathcal{H}_p = \{H\}$ and any $\alpha > 0$, the α-HABE dimension, the BE dimension and the eluder dimension of $\mathcal{F}_H$ are all at least $A^H - 2$. Moreover, any algorithm in this setting will have regret $\Omega(\sqrt{A^H T})$.*

Proof. We separately resolve the cases of sparse and dense intermediate feedback.

C.7.2.1 Dense intermediate feedback, $\mathcal{H}_p = [H]$

Notice that we get a reward $Ber(q)$ at every step as long as we are on the correct sequence of actions $a_1^\star, \dots a_H^\star$, and as soon as we take a wrong action, we always get a reward of 0 subsequently. It is then easy to see that the induced function classes $\mathcal{Q}$ then are given by $\mathcal{Q} = \{(Q_1, \dots Q_H) \mid \exists a_1, \dots a_H \in \mathcal{A} \text{ s.t. } Q_h = (H - h + 1) q \mathbb{1}_{a_1, \dots a_h}\}$.

α-HABE dimension: It suffices to show the upper bound using $\mathcal{D}_{h, \mathcal{Q}(\alpha, h-1)}$, since the α-HABE dimension takes the minimum of distirbutional eluder dimensions over two distributions. For any $\alpha < q$, consider the function class

$$\mathcal{Q}(\alpha, h-1) = \left\{ Q \in \mathcal{Q}, \left| |\mathbb{E}_{\mu_l(Q)}[Q_l - \mathcal{T}_l Q_{l+1}]| \leq \alpha, \ \forall 1 \leq l \leq h-1 \right. \right\}$$

Now note that $\mathbb{E}_{\mu_l(Q)}[Q_l - \mathcal{T}_l Q_{l+1}] = q \mathbb{1}_{a_1, \dots a_l} - q \mathbb{1}_{a_1^\star, \dots a_l^\star}$. If this is smaller than α, then this is smaller than q and thus must be 0. So, $(a_1, \dots a_{h-1}) = (a_1^\star, \dots a_{h-1}^\star)$ for any $Q \in \mathcal{Q}(\alpha, h-1)$. This also means that any $\phi_h \in \Phi_h$, there is a $Q \in \mathcal{Q}(\alpha, h-1)$ so that

$$\phi_h = Q_h - \mathcal{T}_h Q_{h+1} = q \mathbb{1}_{a_1^\star, \dots a_{h-1}^\star, a_h} - q \mathbb{1}_{a_1^\star, \dots a_{h-1}^\star, a_h^\star}$$

Thus, the size of Φ_h is just A. More importantly, the set $\mathcal{D}_{h, \mathcal{Q}(\alpha, h-1)}$ of distributions $\mu_h(Q)$ induced by $Q \in \mathcal{Q}(\alpha, h-1)$ only include indicators of the form $\mathbb{1}_{a_1^\star, \dots a_{h-1}^\star, a}$ for actions a. Thus, the set of distributions $\mathcal{D}_{\mathcal{Q}(\alpha, h-1)}$ has size A. We know that the distributional eluder dimension $d = \dim_{DE}(\Phi_h, \mathcal{D}_{\mathcal{D}(\alpha, h-1)}, \min(\alpha, Bp/T))$ is bounded by the number of possible distributions $\left|\mathcal{D}_{\mathcal{Q}(\alpha, h-1)}\right|$. So, $d \leq A$.

BE dimension: The Bellman differences, from above, are $q \mathbb{1}_{a_1, \dots a_h} - q \mathbb{1}_{a_1^\star, \dots a_h^\star}$. This is an affine

transformation of a family of A^H indicator functions. The distributions $\mu_l(Q)$ over trajectories induced by Q include indicators $\mathbb{1}_{a'_1,\dots a'_l}$ of all trajectories of length l. Now for any sequence $\mu_1, \dots \mu_n, \mu_{n+1}$ of different indicator distributions not including $a^\star_1, \dots a^\star_l$, we consider the Bellman difference $g_{n+1} = q\mathbb{1}_{a_1,\dots a_h} - q\mathbb{1}_{a^\star_1,\dots a^\star_h}$ with action sequence given by μ_{n+1}. Note that $\mathbb{E}_{\mu_i} g_{n+1} = 0$ for all $i \leq n$ but $\mathbb{E}_{\mu_{n+1}} g_{n+1} = q$. This means that the longest possible sequence in the definition of the distributional eluder dimension has length $A^H - 2$. So, the BE dimension is at least $A^H - 2$.

Eluder dimension: The reward function class $\mathcal{F}_h$ is given by all functions of the form $q\mathbb{1}_{a_1,\dots a_h}$. This is a scaled version of a class of A^h indicator functions. Since it contains A^h indicator functions, its eluder dimension is at least A^h.

C.7.2.2 Sparse intermediate feedback, $\mathcal{H}_p = [H]$

Notice that we get a reward $Ber(q)$ at the *last* step if we took correct sequence of actions $a^\star_1, \dots a^\star_H$, and reward 0 otherwise. It is then easy to see that now, the induced function classes $\mathcal{Q}$ then are given by $\mathcal{Q} = \{(Q_1, \dots Q_H) \mid \exists a_1, \dots a_H \in \mathcal{A} \text{ s.t. } Q_h = q\mathbb{1}_{a_1,\dots a_h}\}$.

α-HABE dimension: This time, note that $\mathbb{E}_{\mu_h(Q)}[Q_l - \mathcal{T}_h Q_{h+1}] = 0$ for all $h \leq H-1$. So, the function class $\Phi_h = \{0\}$ for all $h \leq H-1$. Only for $h = H$ do we have that $\mathbb{E}_{\mu_H(Q)}[Q_H - \mathcal{T}_H Q_{H+1}] = q\mathbb{1}_{a_1,\dots a_H} - q\mathbb{1}_{a^\star_1,\dots a^\star_H}$. Also note that $\mathcal{Q}(\alpha, H-1) = \mathcal{Q}$ for all α since $\mathbb{E}_{\mu_h(Q)}[Q_l - \mathcal{T}_h Q_{h+1}] = 0$ for all $h \leq H-1$. So, this is merely the BE dimension of the problem. Now, the Bellman differences at timestep H are identical to those for the sparse feedback problem, and the distributions $\mathcal{D}_{\mathcal{Q}(\alpha,H-1)} = \mathcal{D}_{\mathcal{Q}}$ since we have established that $\mathcal{Q}(\alpha, H-1) = \mathcal{Q}$. This means that by the argument for BE dimension in the dense feedback case, we have that the distributional eluder dimension of Φ_H is at least $A^H - 2$, which is then also the α-HABE dimension of this problem.

BE dimension: From the argument for the α-HABE dimension in the sparse case, the BE dimension and the α-HABE dimension match in this case, and are both at least $A^H - 2$.

Eluder dimension: Again, the reward function class $\mathcal{F}_H$ is given by all functions of the form $q\mathbb{1}_{a_1,\dots a_H}$. This is a scaled version of a class of A^H indicator functions. Since it contains A^H indicator functions, its eluder dimension is at least A^H.

Also note that this example also produces a universal lower bound of $\sqrt{(A^H T)}$ under any algorithm. That is, no algorithm can improve over our bounds under sparse feedback. This lower bound is an immediate consequence of regret lower bounds for bandit algorithms by treating each sequence of actions taken as a different arm.

□

C.7.3 Proofs of Lemmas

Recall that $\mathcal{Q}(\alpha, h) = \{Q \in \mathcal{Q} \mid |\mathbb{E}_{\mu_l(Q)}[Q_l - \mathcal{T}_l Q_{l+1}]| \leq \alpha,\ \forall 1 \leq l \leq h\}$, that $\mu_h(Q)$ is the distribution induced on $\tau[h-1], a_h$ by π_Q and $\mathcal{D}_{h,\mathcal{Q}} := \{\mu_h(Q) \mid Q \in \mathcal{Q}\}$.

Lemma C.7.4. *Let* $d_h(\epsilon) := \dim_{DE}(\Phi_h, \mathcal{D}_{h,\mathcal{Q}(\alpha,h-1)}, \epsilon)$ *with*

$$\Phi_h := \left\{ Q_h - \mathcal{T}_h Q_{h+1} \middle| Q \in \mathcal{Q}(\alpha, h-1) \right\}$$

Then, we have that for $\beta = c\log(HT\mathcal{N}(\mathcal{Q} \cup \mathcal{G}, 1/T, \|\cdot\|_\infty))$, $\sum_{j=1}^t |\mathbb{E}_{\mu_h(Q^j)}[Q_h^j - \mathcal{T}_h Q_{h+1}^j]|$ *is bounded by*

$$t\omega + Bp\left(\frac{B^2p^2\beta}{\alpha^2} + 1\right)\left(\sum_{l=1}^{h-1} d_l(\alpha)\right) + \min(d_h(\omega), t)Bp + 2Bp\sqrt{\beta d_h(\omega) t}$$

Proof. We modify the proof of Lemma 41 in Jin et al. [2021a]. Pick arbitrary h and t and let Ψ_h be the function class given by

$$\begin{aligned}\Phi_h &:= \left\{ Q_h - \mathcal{T}_h Q_{h+1} \middle| Q \in \mathcal{Q}(\alpha, h) \right\} \\ &= \left\{ Q_h - \mathcal{T}_h Q_{h+1} \middle| (Q_1, \cdots Q_H) \in \mathcal{Q}, |\mathbb{E}_{\mu_l(Q)}[Q_l - \mathcal{T}_l Q_{l+1}]| \leq \alpha,\ \forall 1 \leq l \leq h-1 \right\}\end{aligned}$$

Also note that we have the function class Φ_h of timestep h Bellman errors induced by "historically α-accurate" functions - functions whose expected Bellman errors in previous timesteps are smaller than α. The distribution used for computing the expected Bellman errors for previous timesteps is $\mu_l(Q)$.

Now abbreviating $\psi_l^j := Q_l^j - \mathcal{T}_l Q_{l+1}^j$ gives a sequence $\psi_l^1, \ldots \psi_l^t$ of functions in Ψ_l for every $1 \leq l \leq h$. This must have a subsequence $\phi_l^1, \ldots \phi_l^{r_l}$ consisting of all the functions in the sequence that lie in Φ_l, for every $1 \leq l \leq h$. Also let $d_h(\epsilon) = \dim_{DE}(\Phi_h, \mathcal{D}_h, \epsilon)$ for any ϵ. Now note that

$$\begin{aligned}&\sum_{j=1}^t |\mathbb{E}_{\mu_h(Q^j)}[Q_h^j - \mathcal{T}_h Q_{h+1}^j]| \\ &= \sum_{j=1}^t |\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]| \\ &\overset{(i)}{=} \sum_{j=1}^t \left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \mathbb{1}\left(|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]| \leq \omega\right) \\ &\quad + \sum_{l=1}^{h-1}\sum_{j=1}^t \left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \mathbb{1}\left(|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]| > \omega, Q \in \mathcal{Q}(\alpha, l-1) \setminus \mathcal{Q}(\alpha, l)\right)\end{aligned}$$

$$
\begin{aligned}
&\quad + \sum_{j=1}^{t} \left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \mathbb{1}\left(\left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| > \omega, Q \in \mathcal{Q}(\alpha, h-1)\right) \\
&\leq \sum_{j=1}^{t} \left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \mathbb{1}\left(\left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \leq \omega\right) \\
&\quad + \sum_{l=1}^{h-1} \sum_{j=1}^{t} \left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \mathbb{1}\left(Q \in \mathcal{Q}(\alpha, l-1) \setminus \mathcal{Q}(\alpha, l)\right) \\
&\quad + \sum_{j=1}^{t} \left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \mathbb{1}\left(\left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| > \omega, Q \in \mathcal{Q}(\alpha, h-1)\right) \\
&\leq t\omega + \sum_{l=1}^{h-1} \sum_{j=1}^{t} \left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \mathbb{1}\left(\left|\mathbb{E}_{\mu_l(Q^j)}\psi_l^j\right| > \alpha, Q \in \mathcal{Q}(\alpha, l-1)\right) \\
&\quad \sum_{j=1}^{t} \left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \mathbb{1}\left(\left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| > \omega, Q \in \mathcal{Q}(\alpha, h-1)\right) \\
&\overset{(ii)}{\leq} t\omega + \sum_{l=1}^{h-1} \sum_{j=1}^{r_l} \left|\mathbb{E}_{\mu_h(Q^j)}[\phi_h^j]\right| \mathbb{1}\left(\left|\mathbb{E}_{\mu_l(Q^j)}[\phi_l^j]\right| > \alpha\right) + \sum_{j=1}^{r_h} \left|\mathbb{E}_{\mu_h(Q^j)}[\phi_h^j]\right| \mathbb{1}\left(\left|\mathbb{E}_{\mu_h(Q^j)}[\phi_h^j]\right| > \omega\right) \\
&\overset{(ii)}{\leq} t\omega + Bp\left(\frac{B^2p^2\beta}{\alpha^2} + 1\right)\left(\sum_{l=1}^{h-1} d_l(\alpha)\right) + \sum_{j=1}^{r_h} \left|\mathbb{E}_{\mu_h(Q^j)}[\phi_h^j]\right| \mathbb{1}\left(\left|\mathbb{E}_{\mu_h(Q^j)}[\phi_h^j]\right| > \omega\right)
\end{aligned}
$$

Here, (i) holds since one of three possibilities holds – either $\left|\mathbb{E}_{\mu_h(Q^j)}[\psi_h^j]\right| \leq \omega$, or $|\mathbb{E}_{\mu_h(Q^j)}\psi_l^j| > \omega$ and there is a least $l \leq h-1$ so that $Q \in \mathcal{Q}(\alpha, l-1)$ but $Q \notin \mathcal{Q}(\alpha, h-1)$, or $Q \in \mathcal{Q}(\alpha, h-1)$. (ii) holds since if $|\mathbb{E}_{\mu_k(Q^j)}\psi_k^j| \leq \alpha$ for all $k \leq l-1$, then $\psi_l^j = \phi_l^i$ for some i. Finally, (iii) holds by Proposition 43 of Jin et al. [2021a] since $\sum_{j=1}^{s-1} \mathbb{E}_{\mu_l(Q^j)}[(\phi_l^j)^2] \leq \beta$ by Lemma 39(a) of Jin et al. [2021a]. While our rewards are stochastic and theirs are not, we can repeat their arguments verbatim after noting that the martingale defined in the beginning of their proof continues to be a martingale even for stochastic rewards that have second moments.

Now arrange the sequence $\left|\mathbb{E}_{\mu_h(Q^j)}\phi_s\right|$ in order to get $e_1, \ldots e_{r_h}$. We can then write

$$
\sum_{j=1}^{t} \left|\mathbb{E}_{\mu_h(Q^j)}[Q_h^j - \mathcal{T}_h Q_{h+1}^j]\right| \leq t\omega + Bp\left(\frac{B^2p^2\beta}{\alpha^2} + 1\right)\left(\sum_{l=1}^{h-1} d_l(\alpha)\right) + \sum_{j=1}^{r_h} e_j \mathbb{1}(e_j > \omega)
$$

For any $e_j > \omega$, consider arbitrary γ such that $e_j > \gamma > \omega$. This means that by Proposition 43 of

Jin et al. [2021a] again,

$$j \leq \sum_{i=1}^{r_h} \mathbb{1}(e_i > \gamma) \leq \left(\frac{B^2p^2\beta}{\gamma^2} + 1\right) d_h(\omega)$$

This means that $\gamma \leq Bp\sqrt{\frac{\beta d_h(\omega)}{j - d_h(\omega)}}$ for any such γ. Since $e_j \leq Bp$, we get that $e_j \leq \min\left(Bp, Bp\sqrt{\frac{\beta d_h(\omega)}{j-d_h(\omega)}}\right)$. Finally, this means that

$$\begin{aligned}
&\sum_{j=1}^{t} \left|\mathbb{E}_{\mu_h(Q^j)}[Q_h^j - \mathcal{T}_h Q_{h+1}^j]\right| \\
&\leq t\omega + Bp\left(\frac{B^2p^2\beta}{\alpha^2} + 1\right)\left(\sum_{l=1}^{h-1} d_l(\alpha)\right) + \sum_{j=1}^{r_h} e_j \mathbb{1}(e_j > \omega) \\
&\leq t\omega + Bp\left(\frac{B^2p^2\beta}{\alpha^2} + 1\right)\left(\sum_{l=1}^{h-1} d_l(\alpha)\right) + \sum_{j=1}^{r_h} \min\left(Bp, Bp\sqrt{\frac{\beta d_h(\omega)}{j - d_h(\omega)}}\right) \\
&\leq t\omega + Bp\left(\frac{B^2p^2\beta}{\alpha^2} + 1\right)\left(\sum_{l=1}^{h-1} d_l(\alpha)\right) + \min(d_h, r_h)Bp + \sum_{j=1}^{r_h} Bp\sqrt{\frac{\beta d_h(\omega)}{j}} \\
&\leq t\omega + Bp\left(\frac{B^2p^2\beta}{\alpha^2} + 1\right)\left(\sum_{l=1}^{h-1} d_l(\alpha)\right) + \min(d_h(\omega), r_h)Bp + 2Bp\sqrt{\beta d_h(\omega) r_h} \\
&\leq t\omega + Bp\left(\frac{B^2p^2\beta}{\alpha^2} + 1\right)\left(\sum_{l=1}^{h-1} d_l(\alpha)\right) + \min(d_h(\omega), t)Bp + 2Bp\sqrt{\beta d_h(\omega) t}
\end{aligned}$$

as desired. □

C.8 Proofs for Dueling Feedback

C.8.1 Proof for Reduction to Confidence-Set Optimism

Theorem C.8.1 (Reduction from Dueling to Confidence-Set-Based Optimism)**.** *If the confidence sets $\mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$ satisfy Assumption 10, then the dueling regret $\mathrm{Regret}_D(T)$ of Algorithm 10 is given by*

$$\mathrm{Regret}_D(T) = \widetilde{\mathcal{O}}(C_P(\mathcal{M}, T, \delta) + C_F(\overline{\mathcal{M}}, T, \delta))$$

Remark 21. While the theorem states that we need Assumption 10 from the main paper, we actually use its slightly more refined version – Assumption 13. The less refined version was added to the

main paper for brevity.

Proof. For ease of notation, let us use the sets $\mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$ given by the pre-image of $\mathcal{C}_{\overline{\mathcal{M}}}(\mathcal{D}_t, \delta)$ under the map $\mathcal{M} \mapsto \overline{\mathcal{M}}$ from Section 5.4. We first recall that $\mathcal{M}_\star \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$ and so $\pi_\star \in \Pi_t$ for all t with probability $1 - \delta/16$. Recall that the value of a duel (π, π') under model $\overline{\mathcal{M}} \leftrightarrow$ is denoted by

$$V_D(\overline{\mathcal{M}}, \pi, \pi') := V(\mathcal{M}, \pi) - V(\mathcal{M}, \pi') = V(\mathbb{P}, f, \pi) - V(\mathbb{P}, g, \pi')$$

We overload notation and use the natural maps $(\mathbb{P}, f) \leftrightarrow \mathcal{M} \mapsto \overline{\mathcal{M}}$ to define

$$V_D(\mathcal{M}, \pi, \pi') := V_D(\overline{\mathcal{M}}, \pi, \pi')$$

For ease of notation, we will then work with $\mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$ in this proof until we can. Since $\pi_{i,t} \in \Pi_t$ for $i = 1, 2$, there exists some $\mathcal{M}_{i,t} \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)$ for $i = 1, 2$ so that $V_D(\mathcal{M}_{i,t}, \pi, \pi_{1,t}) \leq 0$ for all π. Note that dueling regret is given below. Inequality (i) is by definition of $\mathcal{M}_{i,t}$, since $V_D(\mathcal{M}_{i,t}, \pi_\star, \pi_{i,t}) \leq 0$ for $i = 1, 2$. Inequality (ii) holds by definition of $\pi_{1,t}, \pi_{2,t}$.

$$\begin{aligned}
\mathrm{Regret}_D(T) &= \sum_{t=1}^{T} \sum_{i=1}^{2} V_D(\mathcal{M}_\star, \pi_\star, \pi_{i,t}) \\
&= \sum_{t=1}^{T} \Big[\sum_{i=1}^{2} V_D(\mathcal{M}_\star, \pi_\star, \pi_{i,t}) - V_D(\mathcal{M}_{i,t}, \pi_\star, \pi_{i,t}) + \sum_{i=1}^{2} V_D(\mathcal{M}_{i,t}, \pi_\star, \pi_{i,t}) \Big] \\
&\overset{(i)}{\leq} \sum_{i=1}^{2} \sum_{t=1}^{T} \left[V_D(\mathcal{M}_\star, \pi_\star, \pi_{i,t}) - V_D(\mathcal{M}_{i,t}, \pi_\star, \pi_{i,t}) \right] \\
&\overset{(ii)}{\leq} \sum_{t=1}^{T} 2 \max_{\mathcal{M}, \mathcal{M}' \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)} \left[V_D(\mathcal{M}, \pi_{1,t}, \pi_{2,t}) - V_D(\mathcal{M}', \pi_{1,t}, \pi_{2,t}) \right]
\end{aligned}$$

Continuing, we have

$$\begin{aligned}
\mathrm{Regret}_D(T) &\leq \sum_{t=1}^{T} 2 \max_{\mathcal{M}, \mathcal{M}' \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)} \left[V_D(\mathcal{M}, \pi_{1,t}, \pi_{2,t}) - V_D(\mathcal{M}', \pi_{1,t}, \pi_{2,t}) \right] \\
&= 2 \sum_{t=1}^{T} \max_{\mathcal{M}, \mathcal{M}' \in \mathcal{C}_{\mathcal{M}}(\mathcal{D}_t, \delta)} \Big[V_D(\mathcal{M}, \pi_{1,t}, \pi_{2,t}) - V_D(\mathcal{M}_\star, \pi_{1,t}, \pi_{2,t}) + V_D(\mathcal{M}_\star, \pi_{1,t}, \pi_{2,t}) \\
&\qquad\qquad - V_D(\mathcal{M}', \pi_{1,t}, \pi_{2,t}) \Big]
\end{aligned}$$

$$
\begin{aligned}
&\leq 2\sum_{t=1}^{T} \max_{\mathcal{M},\mathcal{M}'\in\mathcal{C}_{\mathcal{M}}(\mathcal{D}_t,\delta)} [V_D(\mathcal{M},\pi_{1,t},\pi_{2,t}) - V_D(\mathcal{M}_\star,\pi_{1,t},\pi_{2,t})] + \\
&\qquad \max_{\mathcal{M},\mathcal{M}'\in\mathcal{C}_{\mathcal{M}}(\mathcal{D}_t,\delta)} [V_D(\mathcal{M}_\star,\pi_{1,t},\pi_{2,t}) - V_D(\mathcal{M}',\pi_{1,t},\pi_{2,t})] \\
&= 2\sum_{t=1}^{T} \Big[V_D(\widetilde{\mathcal{M}}_t,\pi_{1,t},\pi_{2,t}) - V_D(\mathcal{M}_\star,\pi_{1,t},\pi_{2,t})\Big] \\
&\qquad + \Big[V_D(\mathcal{M}_\star,\pi_{1,t},\pi_{2,t}) - V_D(\widetilde{\mathcal{M}}'_t,\pi_{1,t},\pi_{2,t})\Big]
\end{aligned}
$$

where $\widetilde{\mathcal{M}}_t$ and $\widetilde{\mathcal{M}}'_t$ are the respective maximisers. It suffices to analyse only one of the terms, as a consequence of the symmetry of Assumption 13.

We can now use the fact that $\mathcal{M}$ is described by $(\mathbb{P}, f)$ to analyse the first term, letting $\widetilde{\mathcal{M}}_t \leftrightarrow (\widetilde{\mathbb{P}}_t, \widetilde{f}^t)$.

$$
\begin{aligned}
&\sum_{t=1}^{T} \Big[V_D(\widetilde{\mathcal{M}}_t,\pi_{1,t},\pi_{2,t}) - V_D(\mathcal{M}_\star,\pi_{1,t},\pi_{2,t})\Big] \\
&= 2\sum_{t=1}^{T} \Big[V_D(\widetilde{\mathbb{P}}_t,\widetilde{f}^t,\pi_{1,t},\pi_{2,t}) - V_D(\mathbb{P}_\star,f^\star,\pi_{1,t},\pi_{2,t})\Big] \\
&\leq 2\sum_{t=1}^{T} \Big[V_D(\widetilde{\mathbb{P}}_t,\widetilde{f}^t,\pi_{1,t},\pi_{2,t}) - V_D(\mathbb{P}_\star,\widetilde{f}^t,\pi_{1,t},\pi_{2,t})\Big] \\
&\qquad + \Big[V_D(\mathbb{P}_\star,\widetilde{f}^t,\pi_{1,t},\pi_{2,t}) - V_D(\mathbb{P}_\star,f^\star,\pi_{1,t},\pi_{2,t})\Big] \\
&\leq 2\sum_{t=1}^{T} \Big[V_D(\widetilde{\mathbb{P}}_t,\widetilde{f}^t,\pi_{1,t},\pi_{2,t}) - V_D(\mathbb{P}_\star,\widetilde{f}^t,\pi_{1,t},\pi_{2,t})\Big] \\
&\qquad + \Big[V_D(\mathbb{P}_\star,\widetilde{f}^t,\pi_{1,t},\pi_{2,t}) - V_D(\mathbb{P}_\star,f^\star,\pi_{1,t},\pi_{2,t})\Big] \\
&\overset{(i)}{=} 2\sum_{t=1}^{T} \Big[V(\widetilde{\mathbb{P}}_t,\widetilde{f}^t,\pi_{1,t}) - V(\mathbb{P}_\star,\widetilde{f}^t,\pi_{1,t})\Big] - \Big[V(\widetilde{\mathbb{P}}_t,\widetilde{f}^t,\pi_{2,t}) - V(\mathbb{P}_\star,\widetilde{f}^t,\pi_{2,t})\Big] \\
&\qquad + \Big[V_D(\mathbb{P}_\star,\widetilde{f}^t,\pi_{1,t},\pi_{2,t}) - V_D(\mathbb{P}_\star,f^\star,\pi_{1,t},\pi_{2,t})\Big] \\
&\overset{(ii)}{=} 2\sum_{t=1}^{T} \Big[V(\widetilde{\mathbb{P}}_t,\widetilde{f}^t,\pi_{1,t}) - V(\mathbb{P}_\star,\widetilde{f}^t,\pi_{1,t})\Big] - \Big[V(\widetilde{\mathbb{P}}_t,\widetilde{f}^t,\pi_{2,t}) - V(\mathbb{P}_\star,\widetilde{f}^t,\pi_{2,t})\Big] \\
&\qquad + \Big[V(\mathbb{P}_\star\otimes\mathbb{P}_\star,\overline{f}^t,(\pi_{1,t},\pi_{2,t})) - V(\mathbb{P}_\star\otimes\mathbb{P}_\star,\overline{f}^\star,(\pi_{1,t},\pi_{2,t}))\Big]
\end{aligned}
$$

Where (i) holds by the definition of V_D and V, and (ii) holds in the product MDP $\overline{\mathcal{M}}_\star$ once we define $\overline{f}^t_h((\tau_1,\tau_2)[h]) := \widetilde{f}^t_h(\tau_1[h]) - \widetilde{f}^t_h(\tau_2[h])$ and recall that $\overline{\mathbb{P}}_\star = \mathbb{P}_\star \otimes \mathbb{P}_\star$. Now, we can

immediately apply Assumption 13 to the last line in two different ways. For the first two terms, we apply the first point in the assumption to each under cardinal feedback for MDP $\mathcal{M}_\star$, noting that the datasets $\mathcal{D}_t$ contain trajectories from $\pi_{1,t}$ as well as $\pi_{2,t}$. For the last term, we apply the second point in the assumption under cardinal feedback for the MDP $(\overline{\mathbb{P}}_\star, \overline{f}^\star)$.

This gives us that with probability $1-\delta$,

$$\mathrm{Regret}(T) = \widetilde{\mathcal{O}}(C_P(\mathcal{M}, T, \delta) + C_F(\overline{\mathcal{M}}, T, \delta))$$

□

We have the following lemma, which is an immediate consequence of

Lemma 5.4.3 (Relating $\mathcal{F}$ and $\overline{\mathcal{F}}$)**.** *For any function class $\mathcal{F}$, $\dim_E(\overline{\mathcal{F}}, \epsilon) \le 9\dim_E(\mathcal{F}, \epsilon/2)$.*

Proof. Let $\overline{d}_h = \dim_E(\overline{\mathcal{F}}_h, \epsilon)$. Pick the ϵ' so that there is a sequence of $\overline{d}_h$ pairs $\overline{\tau}_j, j = 1 \to \overline{d}_h$ of length h trajectories, where each one is ϵ'-independent of its predecessors. Note that $\overline{\tau}_j = (\tau_{1,j}, \tau_{2,j})$. We now inductively build a sequence i_j so that each $\tau_{i_j,j}$ is $\epsilon'/2$-independent of its predecessors.

Pick the first i_1 arbitrarily. Now assume that we have built the sequence until index $j = k$. Also, by definition of this sequence, there exist $\overline{f}_j, \overline{f}'_j$, we have $\sqrt{\sum_{j=1}^k (\overline{f}_j(\overline{\tau}_j) - \overline{f}'_j(\overline{\tau}_j))^2} \le \epsilon'$ but $|\overline{f}_{k+1}(\overline{\tau}_j) - \overline{f}'_{k+1}(\overline{\tau}_j)| \ge \epsilon'$. Since $a^2 + b^2 \le 2(a+b)^2$, we have that

$$\begin{aligned}\sqrt{\sum_{j=1}^k (f_j(\tau_{i_j,j}) - f'_j(\tau_{i_j,j}))^2} &\le \sqrt{\sum_{j=1}^k (f_j(\tau_{i_j,j}) - f'_j(\tau_{i_j,j}))^2 + (f_j(\tau_{3-i_j,j}) - f'_j(\tau_{3-i_j,j}))^2} \\ &\le \sqrt{\sum_{j=1}^k 2(\overline{f}_j(\overline{\tau}_j) - \overline{f}'_j(\overline{\tau}_j))^2} \le \sqrt{2}\epsilon'\end{aligned}$$

Additionally, since

$$|f_{k+1}(\tau_{1,k+1}) - f'_{k+1}(\tau_{1,k+1})| + |f_{k+1}(\tau_{2,k+1}) - f'_{k+1}(\tau_{2,k+1})| \ge |\overline{f}_{k+1}(\overline{\tau}_j) - \overline{f}'_{k+1}(\overline{\tau}_j)| \ge \epsilon'$$

. So, there is an i_{k+1} so that

$$|f_{k+1}(\tau_{i_{k+1},k+1}) - f'_{k+1}(\tau_{i_{k+1},k+1})| \ge \epsilon'/2$$

So, we have a sequence $x_j := \tau_{i_j,j}$ and a sequence of pairs of functions f_j, f'_j so that for any $1 \le k \le \overline{d}_h$, $\sum_{j=1}^k (f_j(x_j) - f'_j(x_j))^2 \le 2(\epsilon')^2$ but $|f_{k+1}(x_{k+1}) - f'_{k+1}(x_{k+1})| \ge \epsilon'/2$. This implies

the following. Inequality (i) holds by Proposition 43 of Jin et al. [2021a] upon setting $\beta = 2(\epsilon')^2$ and setting the proposition's ϵ to $\epsilon'/2$. Inequality (ii) holds since $\epsilon'/2 \geq \epsilon/2$.

$$\begin{aligned}
\overline{d}_h &= \sum_{j=1}^{\overline{d}_h} \mathbb{1}(|f_j(x_j) - f'_j(x_j)| \geq \epsilon'/2) \\
&\overset{(i)}{\leq} \left(\frac{2(\epsilon')^2}{(\epsilon'/2)^2} + 1 \right) \dim_E(\mathcal{F}_h, \epsilon/2) \\
&= 9 \dim_E(\mathcal{F}_h, \epsilon'/2) \\
&\leq 9 \dim_E(\mathcal{F}_h, \epsilon/2)
\end{aligned}$$

This establishes our claim. □

We have the following immediate corollary of Theorem 5.4.2, Theorem C.4.10 and Lemma 5.4.3.

Corollary 5.4.4 (Dueling Regret using POR-UCRL Confidence Sets)**.** *The confidence sets from POR-UCRL satisfy Assumption 10 and using them in Algorithm 10 leads to the following regret bound* $\mathrm{Regret}_D(T) = \widetilde{\mathcal{O}}\left(\left(pS\sqrt{HA} + \sum_{h \in \mathcal{H}_p} \sqrt{d_{E,h} d_{C,h}}\right)\sqrt{T}\right)$.

C.8.2 Reduction to Bonus-Based Optimism

We define the reduction using the algorithm below.

Algorithm 27 Reduction from Dueling to Cardinal Bonus-Based Optimism

Input Known reward function $\{r_h\}_{h=1}^H$, method $\texttt{Est}(\mathcal{D})$ to estimate $\hat{\mathbb{P}}_{\mathcal{D}}$ and $\overline{f}_{\mathcal{D}}$ from dataset $\mathcal{D}$, bonus functions $b_{\overline{\mathcal{F}}}^{\mathcal{D}}(\mathbb{P}, \pi, \delta)$ and $b_{\mathcal{P}}^{\mathcal{D}}(\mathbb{P}, \pi, \delta)$, confidence level δ.
Initialize dataset $\mathcal{D}_1 \leftarrow \{\}$
for $t = 1, ..., T$ **do**
  Compute good set Π_t {Valid $\pi_\star$ candidates}

$$\Pi_t := \Big\{\pi \in \Pi \Big| V_D((\hat{\mathbb{P}}_{\mathcal{D}_t}, \overline{f}_{\mathcal{D}_t}), \pi, \pi_1) + b_{\overline{\mathcal{F}}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, (\pi, \pi_1), \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi, \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_1, \delta) \geq 0,\ \forall \pi_1 \in \Pi\Big\}$$

  Pick $(\pi_{1,t}, \pi_{2,t})$ given by {Most uncertain duel}

$$\arg\max_{\pi, \pi' \in \Pi_t} b_{\overline{\mathcal{F}}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, (\pi, \pi'), \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi, \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_1, \delta)$$

  Collect trajectories $\tau_{t,i} = \big\{(s_{h,i}^t, a_{h,i}^t))\big\}_{h=1}^H$ along with feedback $\{o_h\}_{h \in \mathcal{H}_p}$ by sampling from $\mathbb{P}_\star^{\pi_{i,t}}$ for $i = 1, 2$.
  Append the data to $\mathcal{D}_t$ to get $\mathcal{D}_{t+1}$, update estimates and bonuses.
end for

Theorem C.8.2 (Reduction from Dueling to Bonus-Based Optimism)**.** *If the bonuses and estimates used in Algorithm 27 satisfy Assumption 13, then with probability* $1 - \delta$*, the dueling regret* $\mathrm{Regret}_D(T)$ *of Algorithm 27 is given by*

$$\mathrm{Regret}_D(T) = \widetilde{\mathcal{O}}(C_P(\mathcal{M}, T, \delta) + C_F(\overline{\mathcal{M}}, T, \delta))$$

Proof. Recall that the value of a duel (π, π') under model $\overline{\mathcal{M}} \leftrightarrow \mathcal{M} \leftrightarrow (\mathbb{P}, f)$ is denoted by

$$V_D(\overline{\mathcal{M}}, \pi, \pi') := V(\mathcal{M}, \pi) - V(\mathcal{M}, \pi') = V(\mathbb{P}, f, \pi) - V(\mathbb{P}, g, \pi')$$

We overload notation and use the natural bijection $\mathcal{M} \leftrightarrow \overline{\mathcal{M}}$ to define

$$V_D(\mathcal{M}, \pi, \pi') := V_D(\overline{\mathcal{M}}, \pi, \pi')$$

For ease of notation in the proof, we often work with an arbitrary pre-image $\hat{f}_{\mathcal{D}}$ of $\overline{f}_{\mathcal{D}}$ under the map $f \mapsto \overline{f}$. A careful read will confirm that this does not affect the correctness of any of the statements. First note that $\pi_\star \in \Pi_t$ for all T with probability $1 - \delta/16$ since the following hold

uniformly over all $\pi_1 \in \Pi$

$$\begin{aligned}
-V_D((\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}), \pi_\star, \pi_1) &= V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi_1) - V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi_\star) \\
&= \Big[V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi_1) - V(\mathbb{P}_\star, \hat{f}_{\mathcal{D}_t}, \pi_1)\Big] \\
&\quad - \Big[V(\mathbb{P}_\star, \hat{f}_{\mathcal{D}_t}, \pi_1) - V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}, \pi_1)\Big] \\
&\quad + V(\mathbb{P}_\star, f^\star, \pi_1) - V(\mathbb{P}_\star, f^\star, \pi_\star) \\
&\quad + V_D((\mathbb{P}_\star, \overline{f}_{\mathcal{D}_t}), \pi_1, \pi_\star) - V_D((\mathbb{P}_\star, f^\star), \pi_1, \pi_\star) \\
&\leq z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_\star, \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_1, \delta) \\
&\quad + 0 \\
&\quad + b_{\overline{\mathcal{F}}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, (\pi_\star, \pi_1), \delta) + \\
&= b_{\overline{\mathcal{F}}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, (\pi_\star, \pi_1), \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_\star, \delta) \\
&\quad + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_1, \delta)
\end{aligned}$$

where the inequality holds by Assumption 14 and the optimality of $\pi_\star$ in the true model. Note let $\hat{\mathcal{M}}_t$ be the model given by $\hat{\mathbb{P}}_{\mathcal{D}_t}, \hat{f}_{\mathcal{D}_t}$ and let $\overline{\mathcal{M}}_t$ be the corresponding model in $\overline{\mathcal{M}}$. We make the following abbreviation:

$$b_{\overline{\mathcal{M}}}(\overline{\mathcal{M}}_t, (\pi, \pi'), \delta) := b_{\overline{\mathcal{F}}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, (\pi, \pi'), \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi, \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi', \delta)$$

$$\begin{aligned}
\mathrm{Regret}_D(T) &= \sum_{t=1}^{T} \sum_{i=1}^{2} V_D(\mathcal{M}_\star, \pi_\star, \pi_{i,t}) \\
&= \sum_{t=1}^{T} \Big[\sum_{i=1}^{2} V_D(\mathcal{M}_\star, \pi_\star, \pi_{i,t}) - V_D(\hat{\mathcal{M}}_t, \pi_\star, \pi_{i,t}) + b_{\overline{\mathcal{M}}}(\overline{\mathcal{M}}_t, (\pi_\star, \pi_{i,t}), \delta) \\
&\qquad + \sum_{i=1}^{2} V_D(\hat{\mathcal{M}}_t, \pi_\star, \pi_{i,t}) - b_{\overline{\mathcal{M}}}(\overline{\mathcal{M}}_t, (\pi_\star, \pi_{i,t}), \delta) \Big] \\
&\overset{(i)}{\leq} \sum_{i=1}^{2} \sum_{t=1}^{T} \Big[V_D(\mathcal{M}_\star, \pi_\star, \pi_{i,t}) - V_D(\hat{\mathcal{M}}_t, \pi_\star, \pi_{i,t}) + b_{\overline{\mathcal{M}}}(\overline{\mathcal{M}}_t, (\pi_\star, \pi_{i,t}), \delta) \Big]
\end{aligned}$$

Inequality (i) holds since $V_D(\hat{\mathcal{M}}_t, \pi_\star, \pi_{i,t}) = -V_D(\hat{\mathcal{M}}_t, \pi_{i,t}, \pi_\star)$, and $\pi_{i,t} \in \Pi_t$ for $i = 1, 2$ implies that

$$V_D(\hat{\mathcal{M}}_t, \pi_{i,t}, \pi_\star) + b_{\overline{\mathcal{F}}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, (\pi_\star, \pi_1), \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_\star, \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_1, \delta) \geq 0$$

Now note that the following holds uniformly over all timesteps t with probability $1 - 3\delta/8$ for $i = 1, 2$ simultaneously using Assumption 14 multiple times and applying a union bound.

$$\begin{aligned}
V_D(\mathcal{M}_\star, \pi_\star, \pi_{i,t}) - V_D(\hat{\mathcal{M}}_t, \pi_\star, \pi_{i,t}) &= V_D((\mathbb{P}_\star, \overline{f}^\star), \pi_\star, \pi_{i,t}) - V_D((\hat{\mathbb{P}}_{\mathcal{D}_t}, \overline{f}_{\mathcal{D}_t}), \pi_\star, \pi_{i,t}) \\
&= V_D((\mathbb{P}_\star, \overline{f}^\star), \pi_\star, \pi_{i,t}) - V_D((\hat{\mathbb{P}}_{\mathcal{D}_t}, \overline{f}^\star), \pi_\star, \pi_{i,t}) \\
&\quad + V_D((\hat{\mathbb{P}}_{\mathcal{D}_t}, \overline{f}^\star), \pi_\star, \pi_{i,t}) - V_D((\hat{\mathbb{P}}_{\mathcal{D}_t}, \overline{f}_{\mathcal{D}_t}), \pi_\star, \pi_{i,t}) \\
&= V(\mathbb{P}_\star, \overline{f}^\star, \pi_\star) - V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \overline{f}^\star, \pi_\star) \\
&\quad + V(\hat{\mathbb{P}}_{\mathcal{D}_t}, \overline{f}^\star, \pi_{i,t}) - V(\mathbb{P}_\star, \overline{f}^\star, \pi_{i,t}) \\
&\quad + V_D((\hat{\mathbb{P}}_{\mathcal{D}_t}, \overline{f}^\star), \pi_\star, \pi_{i,t}) - V_D((\hat{\mathbb{P}}_{\mathcal{D}_t}, \overline{f}_{\mathcal{D}_t}), \pi_\star, \pi_{i,t}) \\
&= z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_\star, \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_{i,t}, \delta) \\
&\quad + b_{\overline{\mathcal{F}}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, (\pi_\star, \pi_{i,t}), \delta) \\
&= b_{\overline{\mathcal{M}}}(\overline{\mathcal{M}}_t, (\pi_\star, \pi_{i,t}), \delta)
\end{aligned}$$

So, with probability $1 - 3\delta/16$, we have that

$$\begin{aligned}
\mathrm{Regret}_D(T) &\leq \sum_{t=1}^{T} \sum_{i=1}^{2} b_{\overline{\mathcal{M}}}(\overline{\mathcal{M}}_t, (\pi_\star, \pi_{i,t}), \delta) \\
&\overset{(i)}{\leq} 2 \sum_{t=1}^{T} b_{\overline{\mathcal{M}}}(\overline{\mathcal{M}}_t, (\pi_{1,t}, \pi_{2,t}), \delta) \\
&= 2 \sum_{t=1}^{T} z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_{1,t}, \delta) + z(Bp) b_{\mathcal{P}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, \pi_{2,t}, \delta) + b_{\overline{\mathcal{F}}}(\hat{\mathbb{P}}_{\mathcal{D}_t}, (\pi_{1,t}, \pi_{2,t}), \delta) \\
&\overset{(ii)}{\leq} \widetilde{\mathcal{O}}\left(\sum_{t=1}^{T} (z_1(Bp) b_{\mathcal{P}}(\mathbb{P}_\star, \pi_{1,t}, \delta) + z_1(Bp) b_{\mathcal{P}}(\mathbb{P}_\star, \pi_{2,t}, \delta) + b_{\overline{\mathcal{F}}}(\mathbb{P}_\star, (\pi_{1,t}, \pi_{2,t}), \delta) \right)
\end{aligned}$$

where inequality (i) holds since $(\pi_{1,t}, \pi_{2,t}) = \arg\max_{\pi, \pi' \in \Pi_t} b_{\overline{\mathcal{M}}}(\overline{\mathcal{M}}_t, (\pi, \pi'), \delta)$ and inequality (ii) holds with probability $1 - 3\delta/8$ by 6 applications of the change of measure inequality in Assumption 14.

Now, we can use the fact that Assumption 14 is satisfied again to conclude that with probability $1 - \delta/32$.

$$\begin{aligned}
\sum_{t=1}^{T} \Big(& z_1(Bp) b_{\mathcal{P}}(\mathbb{P}_\star, \pi_{1,t}, \delta) + z_1(Bp) b_{\mathcal{P}}(\mathbb{P}_\star, \pi_{2,t}, \delta) \\
& + b_{\overline{\mathcal{F}}}(\mathbb{P}_\star, (\pi_{1,t}, \pi_{2,t}), \delta) \Big) = \widetilde{\mathcal{O}}(C_P(\mathcal{M}, T, \delta) + C_F(\overline{\mathcal{M}}, T, \delta))
\end{aligned}$$

Taking a union bound over all inequalities stated so far, we have the following with probability $1 - \delta$

$$\text{Regret}_D(T) = \widetilde{\mathcal{O}}(C_P(\mathcal{M}, T, \delta) + C_F(\overline{\mathcal{M}}, T, \delta))$$

as desired. □

Again, the following corollary is immediate from Theorem C.3.2, Theorem C.5.9 and Lemma 5.4.3.

Corollary C.8.3. *By using POR-UCBVI as the algorithm in the dueling reduction in Algorithm 27, we can get a bound on the dueling regret given by*

$$\text{Regret}_D(T) = \widetilde{\mathcal{O}}\left(\left(pC(H, S, A) + \sum_{h \in \mathcal{H}_p} \sqrt{\bar{d}_{E,h}(d_{C,h} + H)}\right)\sqrt{T}\right)$$

where $\bar{d}_{E,h} = \dim_E\left(\mathcal{F}_h, \frac{B}{2T}\right)$.

APPENDIX D

Supplementary Material for Chapter 5

D.1 Additional Lemmas and Discussion

D.1.1 Broader Impacts

Often, protected groups with private identities have temporal behavior correlated with their private identities. Latent state estimation methods have the potential to identify such private identities online, and must be used with care. Use of such methods should comply with the relevant privacy and data protection acts, and corporations with access to a large customer base as well as governments should be mindful of the impact of the use of latent state methods on their customers and citizens respectively.

D.1.2 Coherence with Equality is Necessary

Lemma D.1.1 (Coherence with Equality is Necessary)**.** *There exists a stateless decision process that is exchangeable and satisfies distributional coherence — for any two action sequences τ, τ' of lengths H and H' sharing the same actions $(a_h, \dots a_k)$ from index h to k, with $\mathcal{F}_H(\tau) = (Y_1, \dots Y_H)$ and $\mathcal{F}_{H'}(\tau') = (Y'_1, \dots Y'_{H'})$, we have $(Y_h, \dots Y_k) \sim (Y'_h, \dots Y'_k)$ — but is neither coherent nor a latent bandit.*

Proof. Consider an SDP with action set $\mathcal{A} = \{0, 1\}$ equipped with a random variable $\theta \in \{0, 1\}$ distributed as $Ber(1/2)$. For an action sequence of length H, define all rewards to be θ if H is even, and $1 - \theta$ if H is odd.

Exchangeability: Within any fixed H, the rewards $(Y_1, \dots, Y_H)$ are all identical (either all θ or all $1 - \theta$), so any permutation of action-reward pairs preserves the joint distribution.

Distributional coherence: Consider two action sequences τ, τ' of lengths H and H' sharing the same actions $(a_h, \dots a_k)$ from index h to k. The reward subsequence from τ is $(X, \dots, X)$ where

$X = \theta$ or $X = 1 - \theta$ depending on the parity of H, and similarly from τ' with the parity of H'. In either case, the subsequence is a vector of perfectly correlated $Ber(1/2)$ random variables, so $(Y_h, \dots Y_k) \sim (Y'_h, \dots Y'_k)$.

Not coherent: Consider τ of length 2 (even) and τ' of length 3 (odd) sharing any action at position 1. From τ, $Y_1 = \theta$; from τ', $Y_1 = 1 - \theta$. On the same sample point ω, $\theta(\omega) \neq 1 - \theta(\omega)$ whenever $\theta(\omega) \neq 1/2$, so the rewards are not equal as functions of ω.

Not a latent bandit: Suppose for contradiction that there exists $F : \Omega \to (\mathcal{A} \to \mathcal{P}(\mathbb{R}))$ satisfying the latent bandit conditions. Fix any action a.

Step 1 (law condition). For $H = 2$ (even), $Y_1 = \theta$, so $\mathcal{L}[\theta \mid F] = F(a)$. For $H = 1$ (odd), $Y_1 = 1 - \theta$, so $\mathcal{L}[1 - \theta \mid F] = F(a)$. Therefore $\mathcal{L}[\theta \mid F] = \mathcal{L}[1 - \theta \mid F]$ almost surely. Let $p(\omega) = P(\theta = 1 \mid F)(\omega)$. Then $\mathcal{L}[\theta \mid F]$ puts mass p on 1 and $1 - p$ on 0, while $\mathcal{L}[1 - \theta \mid F]$ puts mass $1 - p$ on 1 and p on 0. Equating these gives $p = 1 - p$, so $p = 1/2$ a.s., meaning $F(a) = Ber(1/2)$ a.s.

Step 2 (conditional independence). For $H = 2$, actions (a, a): $(Y_1, Y_2) = (\theta, \theta)$. The latent bandit condition requires Y_1, Y_2 to be conditionally independent given F. Since $F(a) = Ber(1/2)$ a.s., we have $P(Y_1 = 0, Y_2 = 0 \mid F) = F(a)(\{0\})^2 = 1/4$ a.s. But $Y_1 = Y_2 = \theta$, so $P(Y_1 = 0, Y_2 = 0 \mid F) = P(\theta = 0 \mid F) = 1/2$ a.s. This is a contradiction. □

D.1.3 Contexts, Latent State and Behavior Policy cannot be Dependent

Lemma 6.3.1 (Contexts, $\boldsymbol{\theta}$, and π_b cannot be dependent)**.** *For each of these conditions:*

1. *Contexts in a trajectory are dependent but do not depend on $\boldsymbol{\theta}$, and π_b also does not use $\boldsymbol{\theta}$,*
2. *Contexts are generated independently using $\boldsymbol{\theta}$, while π_b does not use $\boldsymbol{\theta}$,*
3. *Contexts are generated independently without using $\boldsymbol{\theta}$, while π_b uses $\boldsymbol{\theta}$,*

there exist two different linear latent contextual bandits with orthogonal latent subspaces satisfying the condition, and a behavior policy π_b so that the offline data distributions are indistinguishable and cover all (x, a) pairs with probability at least $1/4$. Since the latent subspaces are orthogonal, an action that gives the maximum reward on one latent bandit gives reward 0 on the other.

Proof. Let e_1, e_2 be the standard basis of $\mathcal{R}^2$. For all these examples, our two latent bandits will satisfy the following:

- **Latent contextual bandit** 1**:** Let $\boldsymbol{\theta}$ take values $e_1 + e_2$ and $-e_1 - e_2$ with probability $1/2$ each.

- **Latent contextual bandit** 2**:** Let $\boldsymbol{\theta}$ take values $e_1 - e_2$ and $e_2 - e_1$ with probability $1/2$ each.

D.1.3.1 Contexts cannot be dependent on each other

We consider two actions, so $\mathcal{A} = \{0, 1\}$ We have two contexts x, y so that $\phi(x, 0) = e_1$, $\phi(x, 1) = 2e_1$ while $\phi(y, 0) = e_2$, $\phi(y, 1) = 2e_2$. We design them to be dependent so that any trajectory either only sees x or only sees y. However, x and y are both seen with probability $1/2$. Pick π_b so that it takes each action with probability $1/2$. Now note that the mean reward of $x, 0$ in either latent bandit takes values ± 1 with probability $1/2$. So, the offline data distributions are also indistinguishable and every context-action pair is seen with probability at least $1/4$.

D.1.3.2 Contexts and latent state cannot be dependent

We consider two actions, so $\mathcal{A} = \{0, 1\}$ Again, consider two contexts x, y so that $\phi(x, 0) = e_1$, $\phi(x, 1) = 2e_1$ while $\phi(y, 0) = e_2$, $\phi(y, 1) = 2e_2$. Let us say that for either latent bandit and for any $\boldsymbol{\theta}$, the context distribution is a Dirac-δ over the context whose features have a positive dot product with $\boldsymbol{\theta}$. Then, note that either context is seen in both latent bandits with probability $1/2$. Again, let π_b take either action with probability $1/2$. So, the offline data distributions are indistinguishable and every context-action pair is seen with probability at least $1/4$.

D.1.3.3 Latent state and behavior policy cannot be dependent

We consider four actions, so that $\mathcal{A} = \{0, 1, 2, 3\}$. Let there be a single context x with $\phi(x, 0) = e_1, \phi(x, 1) = e_2, \phi(x, 2) = -e_1, \phi(x, 3) = -e_2$. For any latent state $\boldsymbol{\theta}$ in either bandit, let π_b be uniform over the actions that have a positive dot product with $\boldsymbol{\theta}$. It is then easy to see that the offline data distributions are indistinguishable and every context-action pair is seen with probability at least $1/4$. □

D.2 A de Finetti Theorem for decision processes

D.2.1 Discussion on Liu et al. [2023]'s smaller class of generalized bandits

We prove the de Finetti theorem for a very general formulation of decision processes. However, past work [Liu et al., 2023] has studied a simpler generalization of bandits, namely a stochastic process valued in $\mathcal{R}^{\mathcal{A}}$. A sample point in this space is a *sequence of functions* in $\mathcal{A} \to \mathcal{R}$, which rules out the possibility of adaptivity. In contrast, a sample point in our space is a *function of sequences*, which subsumes all sample points of their space, but allows for adaptivity.

Liu et al. [2023] are able to use the original de Finetti theorem on their random process directly, but work with a much more restrictive kind of decision process. We show that even when considering much more general stateless decision processes, latent bandits are the "right" objects produced by a de Finetti theorem for stateless decision processes.

D.2.2 Proof of the De Finetti Theorem for Stateless Decision Processes

A note on F, measurability and well-definedness in the definition of a latent bandit. Recall that F in the definition of a latent bandit needs to be a measurable map $\Omega \to (\mathcal{A} \to \mathcal{P}(\mathbb{R}))$. To define measurability for the output space of functions $(\mathcal{A} \to \mathcal{P}(\mathbb{R}))$, we endow $\mathcal{P}(\mathbb{R})$ with the topology of weak convergence, endow the space $\mathcal{P}(\mathbb{R})^{\mathcal{A}}$ of maps $\mathcal{A} \to \mathcal{P}(\mathbb{R})$ with the topology of point-wise convergence, and require F to be measurable w.r.t. the induced Borel σ-algebra. We also recall two abuses of notation in the definition of a latent bandit. First, we abuse notation to define the random measure $F(a) := (\omega \mapsto F(\omega)(a))$. Second, we also abuse notation to conflate $F(a)$ with the curried map $\kappa_a(\omega, B) := F(a)(\omega)(B)$, which is a map $\Omega \times \mathcal{B} \to [0, 1]$. This map κ_a will turn out to be a kernel by the construction of F. Equating $F(a)$ to a regular conditional distribution in the definition of a latent bandit *requires* that κ_a be a kernel.

We recall our de Finetti theorem for stateless decision processes here.

Theorem D.2.1 (De Finetti Theorem for Stateless Decision Processes)**.** *Every exchangeable and coherent stateless decision process is a latent bandit.*

Proof. Consider an exchangeable and coherent stateless decision process. We will establish that there is a latent bandit with the same reward distribution as this process for any sequence of actions.

For any sequence $\tau = (a_1, \dots a_H)$, denote by $(Y_{\tau,1}, \dots Y_{\tau,H}) := \mathcal{F}_H(a_1, \dots a_H)$. We intend to establish that there is a random measure-valued function F such that $(Y_{\tau,1}, \dots Y_{\tau,H})$ are independent given F and $\mathcal{L}[Y_{\tau,h} \mid F] = F(a_h)$ for all $h \leq H$ almost surely. Since conditional independence is a property of finite subsets of a set of random variables, it suffices to show this for finite H. The version for $H = \infty$ will immediately follow by coherence.

First recall that regular conditional distributions $\mathcal{L}[Y \mid F]$ are almost surely unique under if the σ-algebra of the output space of Y is countably generated. Since the Borel σ-algebra on $\mathbb{R}$ is countably generated, this is true for our case. We also recall for the rusty reader that conditioning on a random variable is the same as conditioning on the induced σ-algebra in the domain.

D.2.3 Constructing F

Fix any finite trajectory $\tau = (a_1, \dots a_H)$ and index h.

The trick: Consider the infinite sequence $\tau_\infty := (a_h, a_h, \dots)$. By coherence, $Y_{\tau,h} = Y_{\tau_\infty,h}$. Now τ_∞ is a sequence where exchanging any finite set of rewards preserves the reward distribution, since all actions are identical. Since $\mathbb{R}$ is locally compact, we can apply the usual de Finetti representation theorem (Theorem 12.26 in Klenke [2008]) to conclude that:

- The random measure $\Xi_{a_h} = \mathrm{wlim}_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \delta_{Y_{\tau_\infty,j}}$ is well defined. Here wlim is the weak limit of measures.
- The regular conditional distribution $\mathcal{L}[Y_{\tau_\infty,j} \mid \Xi_{a_h}] = \Xi_{a_h}$ for all j.

Since $Y_{\tau,h} = Y_{\tau_\infty,h}$, we conclude that the conditional distribution $\mathcal{L}[Y_{\tau,h} \mid \Xi_{a_h}] = \mathcal{L}[Y_{\tau_\infty,h} \mid \Xi_{a_h}] = \Xi_{a_h}$.

Constructing F: For every action, consider such a random measure Ξ_{a_h} and define a random measure-valued function $F : \Omega \to (\mathcal{A} \to \mathcal{P}(\mathbb{R}))$ in the following manner: for any $\omega \in \Omega$, define $F(\omega)(a) := \Xi_{a_h}(\omega)$. We know that for any a, Ξ_{a_h} is measurable w.r.t. the topology of weak convergence on $\mathcal{P}(\mathbb{R})$. It is now tedious but straightforward to verify that F is measurable w.r.t. the Borel σ-algebra generated by the topology of pointwise convergence on $\mathcal{P}(\mathbb{R})^{\mathcal{A}}$.

D.2.4 Establishing that $\mathcal{L}[Y_{\tau,h} \mid F] = F(a_h)$ almost surely for all h

Again, fix any finite trajectory $\tau = (a_1, \dots a_H)$ and index h. Recall that we abuse notation to denote by $F(a)$ the random-measure $\omega \mapsto F(\omega)(a)$. In particular, for any measurable set $B \subset \mathbb{R}$, we conflate $F(a)(\omega)(B) = F(\omega)(a)(B) = F(a)(\omega, B)$, where the last equality holds since F is a regular conditional distribution. Note that $F(a_h) = \Xi_{a_h}$ by the construction of F. Since $\Xi_{a_h} = F(a_h)$, we have $\mathcal{L}[Y_{\tau,h} \mid F(a_h)] = F(a_h)$. Thus, it suffices to show that $\mathcal{L}[Y_{\tau,h} \mid F] = \mathcal{L}[Y_{\tau,h} \mid F(a_h)]$.

Lemma D.2.2. $\mathcal{L}[Y_{\tau,h} \mid F] = \mathcal{L}[Y_{\tau,h} \mid F(a_h)]$ *almost surely.*

Proof. First note that $F(a_h)$ is measurable w.r.t. F. For showing this, view the set of maps $\mathcal{P}(\mathbb{R})^{\mathcal{A}}$ as the product set $\prod_{a'} \mathcal{P}(\mathbb{R})_{a'}$. Now merely note that $F^{-1}(E \times \prod_{a' \neq a} \mathcal{P}(\mathbb{R})_{a'}) = (F(a_h))^{-1}(E)$ for any measurable subset $E \subset \mathcal{P}(\mathbb{R})_a$. Hence, $F(a_h)$ is measurable w.r.t. F.

Let $\mathcal{B}$ be the Borel σ-algebra on $\mathbb{R}$. Now recall that the regular conditional distribution $\mathcal{L}[X \mid G]$ for a real-valued random variable X is the almost surely unique kernel $\kappa_{G,X} : \Omega \times \mathcal{B} \to \mathbb{R}$ such that:

- $\omega \to \kappa_{G,X}(\omega, B)$ is G-measurable for any set $B \in \mathcal{B}$
- $B \to \kappa_{G,X}(\omega, B)$ is a probability measure on $\mathbb{R}$ for any sample point $\omega \in \Omega$.

- For any measurable set $B \subset \mathbb{R}$ and any G-measurable set A,

$$\mathbb{E}[\mathbb{1}_B(Y)\mathbb{1}_A] = \int \mathbb{1}_B(Y(\omega))\mathbb{1}_A(\omega)d\mathbb{P}(\omega) = \int \kappa_{G,X}(\omega, B)\mathbb{1}_A(\omega)d\mathbb{P}(\omega)$$

We will show our claim using the definition and a.s. uniqueness of the regular conditional distribution in our case. Consider any F-measurable set $A \subset \Omega$ and Borel-measurable set $B \subset \mathbb{R}$, $B \in \mathcal{B}$. Denote by $\kappa_F := \mathcal{L}[Y_{\tau,h} \mid F]$ and by $\kappa_{a_h} := \mathcal{L}[Y_{\tau,h} \mid F(a_h)] = F(a)$. Note that by the coherence and exchangeability in section D.2.3,

$$\int \kappa_F(\omega, B)\mathbb{1}_A(\omega)d\mathbb{P}(\omega) = \mathbb{E}[\mathbb{1}_B(Y_{\tau,h})\mathbb{1}_A] = \mathbb{E}[\mathbb{1}_B(Y_{\tau_\infty,h})\mathbb{1}_A] = \mathbb{E}[\mathbb{1}_B(Y_{\tau_\infty,j})\mathbb{1}_A]$$

for all j. Averaging all these equations and taking a limit, we get that

$$\begin{aligned}\int \kappa_F(\omega, B)\mathbb{1}_A(\omega)d\mathbb{P}(\omega) = \mathbb{E}[\mathbb{1}_B(Y_{\tau,h})\mathbb{1}_A] &= \mathbb{E}[\mathbb{1}_B(Y_{\tau_\infty,h})\mathbb{1}_A] \\ &= \lim_{n\to\infty} \frac{1}{n}\sum_{j=1}^{n} \mathbb{E}[\mathbb{1}_B(Y_{\tau_\infty,j})\mathbb{1}_A] \\ &= \mathbb{E}\left[\left(\lim_{n\to\infty} \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}_B(Y_{\tau_\infty,j})\right)\mathbb{1}_A\right]\end{aligned}$$

where the last equality holds by the dominated convergence theorem, if the limit exists. To establish that the limit exists and compute it, we apply the usual de Finetti theorem – specificaly point (i) of remark 12.27 in Klenke [2008] with f set to the identity map. This gives us that $\lim_{n\to\infty} \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}_B(Y_{\tau_\infty,j}) = \mathbb{E}[\mathbb{1}_B(Y_{\tau_\infty,h}) \mid \Xi_{a_h}] = \Xi_{a_h}(B)$, where $\Xi_{a_h}(B)$ is the random variable $\omega \mapsto \Xi_{a_h}(\omega)(B)$. This in turn satisfies $\Xi_{a_h}(\omega)(B) = F(a_h)(\omega)(B) = \kappa_{a_h}(\omega, B)$. This establishes that for any F-measurable set A and Borel set B,

$$\int \kappa_F(\omega, B)\mathbb{1}_A(\omega)d\mathbb{P}(\omega) = \int \kappa_{a_h}(\omega, B)\mathbb{1}_A(\omega)d\mathbb{P}(\omega)$$

. In conclusion, κ_{a_h} satisfies:

- $F(a_h)(B) := \omega \to \kappa_{a_h}(\omega, B)$ is F-measurable for any set $B \in \mathcal{B}$ since $F(a_h)(B)$ is $F(a_h)$ measurable by definition of κ_{a_h} and $F(a_h)$ is F-measurable from above.
- $B \to \kappa_{a_h}(\omega, B)$ is a probability measure on $\mathbb{R}$ for any sample point $\omega \in \Omega$ by definition of κ_{a_h}

- For any measurable set $B \subset \mathbb{R}$ and any F-measurable set A, by the argument above,

$$\mathbb{E}[\mathbb{1}_B(Y_{\tau,h})\mathbb{1}_A] = \int \kappa_F(\omega, B)\mathbb{1}_A(\omega)d\mathbb{P}(\omega) = \int \kappa_{a_h}(\omega, B)\mathbb{1}_A(\omega)d\mathbb{P}(\omega)$$

By the definition as well as a.s. uniqueness of regular conditional distributions in our case, this establishes that $\mathcal{L}[Y_{\tau,h} \mid F] = \kappa_F = \kappa_{a_h} = \mathbb{E}[\mathbb{E}[Y_{\tau,h} \mid F(a_h)]$ almost surely.

□

D.2.5 Establishing conditional independence of rewards

This is the trickier to establish. It suffices to show that for any finite length h trajectory $\tau = (a_1, \dots a_H)$ and any tuple $(f_1, \dots f_H)$ of bounded measurable functions, $\mathbb{E}\left[\prod_{h=1}^H f_h(Y_{\tau,h}) \mid F\right] = \prod_{h=1}^H \mathbb{E}[f_h(Y_{\tau,h}) \mid F]$. Equivalently, it suffices to show that for any bounded F-measurable real-valued random variable U, the following holds.

$$\mathbb{E}\left[U \prod_{h=1}^{H} f_h(Y_{\tau,h})\right] = \mathbb{E}\left[U \prod_{h=1}^{H} \mathbb{E}[f_h(Y_{\tau,h}) \mid F]\right]$$

We will show this by induction on H. This clearly holds for $H = 1$ by section D.2.3 above. Now assume that this holds for $H = l - 1$.

Replacing the last term, $f_l(Y_{\tau,l})$, by an empirical average: Fix any trajectory $\tau = (a_1, \dots a_l)$. First consider the infinite trajectory $\tau' = (a_1, \dots a_{l-1}, a_l, a_l, a_l, \dots)$. This creates a random sequence of rewards $Y_{\tau',1}, Y_{\tau',2}, \dots$. Define τ'_j by switching indices l and $l + j$ in τ' and considering the first l actions. In particular, τ'_j gives the sequence of rewards $(Y_{\tau',1}, \dots Y_{\tau',l-1}, Y_{\tau',l+j})$.

First note that $\tau'_0 = \tau$. By coherence, $(Y_{\tau,1}, \dots Y_{\tau,l}) = (Y_{\tau',1}, \dots Y_{\tau',l})$. By exchangeability and coherence, $(Y_{\tau',1}, \dots Y_{\tau',l-1}, Y_{\tau',l}) \sim (Y_{\tau',1}, \dots Y_{\tau',l-1}, Y_{\tau',l+j})$. This means that for any $j \geq 0$,

$$\mathbb{E}\left[U \prod_{h=1}^{l} f_h(Y_{\tau,h})\right] = \mathbb{E}\left[U \left(\prod_{h=1}^{l-1} f_h(Y_{\tau',h})\right) f_l(Y_{\tau',l+j})\right]$$

We can then consider the average over all these equations for $j = 0 \to n - 1$ and get that for all $n \geq 1$,

$$\mathbb{E}\left[U \prod_{h=1}^{l} f_h(Y_{\tau,h})\right] = \mathbb{E}\left[U \left(\prod_{h=1}^{l-1} f_h(Y_{\tau',h})\right) \left(\frac{1}{n}\sum_{j=0}^{n-1} f_l(Y_{\tau',l+j})\right)\right]$$

Taking limits and using the dominated convergence theorem, we get that

$$\mathbb{E}\left[U\prod_{h=1}^{l} f_h(Y_{\tau,h})\right] = \mathbb{E}\left[U\left(\prod_{h=1}^{l-1} f_h(Y_{\tau',h})\right)\left(\lim_{n\to\infty}\frac{1}{n}\sum_{j=0}^{n-1} f_l(Y_{\tau',l+j})\right)\right] \tag{D.1}$$

if the limit on the right side exists.

Showing that the empirical average is the conditional expectation: Again, consider a different infinite trajectory $\tau_l = (a_l, a_l, \dots)$. Again, by coherence, $Y_{\tau_l,l+j} = Y_{\tau',l+j}$ for all $j \geq 0$. From the usual de Finetti theorem, specifically point (i) of remark 12.27 in Klenke [2008], we have that $\frac{1}{m}\sum_{j=1}^{m} f_l(Y_{\tau_l,j}) \to \mathbb{E}[f_l(Y_{\tau_l,l}) \mid F]$. We can then observe that by general properties of convergence, the following holds.

$$\begin{aligned}\mathbb{E}[f_l(Y_{\tau_l,l}) \mid F] &= \lim_{m\to\infty}\frac{1}{m}\sum_{j=1}^{m} f_l(Y_{\tau_l,j}) = \lim_{m\to\infty}\frac{1}{m-l}\sum_{j=l}^{m-1} f_l(Y_{\tau_l,j}) \\ &= \lim_{m\to\infty}\frac{1}{m-l}\sum_{j=l}^{m-1} f_l(Y_{\tau',j}) = \lim_{n\to\infty}\frac{1}{n}\sum_{j=0}^{n-1} f_l(Y_{\tau',l+j})\end{aligned}$$

We can combine this with equation D.1 to get that

$$\mathbb{E}\left[U\prod_{h=1}^{l} f_h(Y_{\tau,h})\right] = \mathbb{E}\left[U\left(\prod_{h=1}^{l-1} f_h(Y_{\tau',h})\right)\mathbb{E}[f_l(Y_{\tau_l,l}) \mid F]\right]$$

Since both U and $\mathbb{E}[f_l(Y_{\tau_l,l}) \mid F]$ are now F measurable, we can apply the induction hypothesis to τ' truncated at $l-1$ and conclude that

$$\mathbb{E}\left[U\prod_{h=1}^{l} f_h(Y_{\tau,h})\right] = \mathbb{E}\left[U\mathbb{E}[f_l(Y_{\tau_l,l}) \mid F]\prod_{h=1}^{l-1} f_h(Y_{\tau,h})\right] = \mathbb{E}\left[U\left(\prod_{h=1}^{l}\mathbb{E}[f_h(Y_{\tau',h}) \mid F]\right)\right]$$

Thus, the induction step holds and the claim holds for all finite H. We discussed at the beginning of the section how this implies conditional independence for $H = \infty$ as well. □

D.2.6 A de Finetti theorem for TACDPs

We consider contexts and define contextual decision processes in a manner agnostic to context transitions. That is, the process only carries the data of how rewards are generated from given sequences of contexts and actions, while the context transitions themselves may be generated by a different process.

Definition D.2.1. A *transition-agnostic contextual decision process* (TACDP) with action set $\mathcal{A}$ and context set $\mathcal{X}$ is a probability space $(\Omega, \mathcal{G}, \mathbb{P})$ equipped with a family of random maps $\mathcal{F}_H : \Omega \to ((\mathcal{X} \times \mathcal{A})^H \to \mathbb{R}^H)$ for $H \in \mathcal{N}$ as well as a map $\mathcal{F}_\infty : \Omega \to ((\mathcal{X} \times \mathcal{A})^{\mathcal{N}} \to \mathbb{R}^{\mathcal{N}})$. It is natural to isolate away context dynamics for processes like stochastic contextual bandits, in which actions do not affect the context transitions – in contrast to transition-aware definitions in Jiang et al. [2017].

Definition D.2.2. A TACDP is said to be *coherent* if for any $h \leq k \leq H, H' \in \mathcal{N} \cup \{\infty\}$ and for any two context-action sequences τ, τ' of lengths H and H' sharing the same context-action pairs $((x_h, a_h), \dots (x_k, a_k))$ from index h to k, with $\mathcal{F}_H(\tau) = (Y_1, \dots Y_H)$ and $\mathcal{F}_{H'}(\tau') = (Y'_1, \dots Y'_{H'})$, we have $(Y_h, \dots Y_k) = (Y'_h, \dots Y'_k)$, viewed as functions of Ω.

Definition D.2.3. A stateless decision process is said to be exchangeable if for any permutation $\pi : [H] \to [H]$ and $\mathcal{F}_H((x_1, a_1), \dots (x_H, a_H)) = (Y_1, \dots Y_H)$, we have $\mathcal{F}_h((x_{\pi(1)}, a_{\pi(1)}), \dots (x_{\pi(H)}, a_{\pi(H)})) \sim (Y_{\pi(1)}, \dots Y_{\pi(H)})$.

Definition D.2.4. A latent contextual bandit is a stateless decision process equipped with a random measure-valued function $F : \Omega \to ((\mathcal{X} \times \mathcal{A}) \to \mathcal{P}(\mathbb{R}))$ so that for any H and context-action sequence $((x_1, a_1), \dots (x_H, a_H))$, the rewards $(Y_1, \dots Y_H) := \mathcal{F}_H((x_1, a_1), \dots (x_H, a_H))$ are independent conditioned on F. Moreover, the conditional distribution $\mathcal{L}[Y_h \mid F] = F((x_h, a_h))$ for all $h \leq H$.[1]

Theorem D.2.3 (De Finetti Theorem for Stateless Decision Processes)**.** *Every exchangeable and coherent TACDP is a latent contextual bandit.*

Proof. The proof is verbatim the same as that for Theorem 6.8.1 after merely replacing $\mathcal{A}$ with $\mathcal{X} \times \mathcal{A}$ and a with (x, a). □

D.3 Proofs for SOLD

Recall that $\mu_\theta := \mathbb{E}[\boldsymbol{\theta}]$ and define $\mu_\beta := \mathbb{E}[\boldsymbol{\beta}] = \mathbf{U}_\star \mu_\theta$. Also recall that $\Lambda := \mathbb{E}[\boldsymbol{\theta}_n \boldsymbol{\theta}_n^\top]$. For a trajectory with index n, denote by $\boldsymbol{\beta}_n := \mathbf{U}_\star \boldsymbol{\theta}_n$. Denote by $X_n := [\phi(x_1, a_1), \dots \phi(x_H, a_H)]^\top$. Denote by $\eta_n := [\epsilon_{n,1}, \epsilon_{n,2} \dots \epsilon_{n,H}]^\top$ the vector of subgaussian noises with subgaussian parameter σ^2. Since rewards are bounded by R, we know that $\sigma^2 \leq R^2$. Denote by $X_{n,i}$ and $\eta_{n,i}$ the action matrix and noise vector corresponding to the trajectory halves $\tau_{n,i}$ for $i = 1, 2$. Since rewards are bounded by R, $\epsilon_{n,h}$ is subgaussian for all h and so $\eta_{n,i}$ is σ^2-subgaussian for $i = 1, 2$. Also note that

$$\hat{\boldsymbol{\beta}}_{n,i} = (\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top r_{n,i}$$

[1] We abuse notation twice here. First, we write $F((x_h, a_h)) := (\omega \mapsto F(\omega)((x_h, a_h)))$. Second, as the regular conditional distribution $\mathcal{L}[Y_h \mid F]$ is a kernel that maps from $\Omega \times \mathcal{B} \to \mathbb{R}$, we view $F((x_h, a_h))$ as its curried map $(\omega, B) \mapsto F((x_h, a_h))(\omega)(B)$. A discussion of issues like measurability and well-definedness is in Appendix D.2.2.

$$
\begin{aligned}
&= (\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top (X_{n,i}\boldsymbol{\beta}_n + \eta_{n,i}) \\
&= (I - \mu(\mu I + X_{n,i}^\top X_{n,i})^{-1})\boldsymbol{\beta}_n + (\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top \eta_{n,i} \\
&= (I - \mu(\mu I + X_{n,i}^\top X_{n,i})^{-1})\boldsymbol{\beta}_n + (\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top \eta_{n,i}
\end{aligned}
$$

Note that $\hat{\boldsymbol{\beta}}_{n,1}$ and $\hat{\boldsymbol{\beta}}_{n,2}$ are identically distributed. Now recall that

$$
\mathbf{M}_n = \frac{1}{2}(\hat{\boldsymbol{\beta}}_{n,1}\hat{\boldsymbol{\beta}}_{n,2}^\top + \hat{\boldsymbol{\beta}}_{n,2}\hat{\boldsymbol{\beta}}_{n,1}^\top)
$$

are i.i.d. random matrices. Denote by $\mathbf{D}_{n,i} := (I - \mu(\mu I + X_{n,i}^\top X_{n,i})^{-1})$ and denote by $u_i := (\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top \eta_{n,i}$.

Lemma D.3.1. *If the per-reward noise is σ^2-subgaussian, then the following inequalities hold*

$$
\begin{aligned}
\|\boldsymbol{M}_n\|_2 &\leq R^2\left(2 + \frac{H}{2\mu}\right) \\
\|\mathbb{E}[\boldsymbol{M}_n^2]\|_2 &\leq R^4 + \frac{\sigma^4(d_A + 1)}{8\mu^2}
\end{aligned}
$$

Proof. We prove various bounds and assemble them.

Bounding $\|(\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top\|_2$: Consider any X with SVD $X = U^\top \Sigma V$. This means that

$$
\begin{aligned}
\|(\mu I + X^\top X)^{-1} X^\top\|_2 &= V^\top(\Sigma^2 + \mu)^{-1}\Sigma U \\
&= \|V^\top(\Sigma^2 + \mu)^{-1}\Sigma\|_2 = \|(\Sigma^2 + \mu)^{-1}\Sigma\|_2 \\
&\leq \max_a \frac{a}{a^2 + \mu} \\
&= \frac{1}{2\sqrt{\mu}}
\end{aligned}
$$

We can now apply this to $X = X_{n,i}$ and conclude that $\|(\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top\|_2 \leq \frac{1}{2\sqrt{\mu}}$

u_i are independent, $\frac{\sigma^2}{4\mu}$-subgaussian and $\frac{\sigma\sqrt{H}}{2\sqrt{\mu}}$-bounded: We claim that u_i are independent, $\frac{\sigma^2}{4\mu}$-subgaussian and $\|u_i\|_2^2 \leq \frac{\sigma\sqrt{H}}{2\sqrt{\mu}}$. Recall that $\eta_{n,i}$ are σ^2-subgaussian vectors and $\|(\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top\|_2 \leq \frac{1}{2\sqrt{\mu}}$. So, we have that $u_i = (\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top \eta_{n,i}$ is $\frac{\sigma^2}{4\mu}$-subgaussian. Also recall that u_1 and u_2 are independent since both contexts and reward-noise are generated independently at each timestep. Finally, since $|\epsilon_{n,h}| \leq R$, we also have that $\|\eta_{n,i}\|_2^2 \leq R^2 H$, so $\|u_i\|_2^2 \leq \frac{R^2 H}{4\mu}$ since $\|(\mu I + X_{n,i}^\top X_{n,i})^{-1} X_{n,i}^\top\|_2 \leq \frac{1}{2\sqrt{\mu}}$.

Bounding $\|\mathbf{M}_n\|_2$: Note that $\|\mathbf{D}_{n,i}\|_2 \leq 1$, $\|\boldsymbol{\beta}_2\|_n \leq R$ and $\|u_i\|_2 \leq \frac{\sigma\sqrt{H}}{2\sqrt{\mu}}$. So, we have that

$$\|\mathbf{M}_n\|_2 \leq \left(R + \frac{R\sqrt{H}}{2\sqrt{\mu}}\right)^2 \leq R^2\left(2 + \frac{H}{2\mu}\right)$$

Bounding $\|\mathbb{E}[\mathbf{M}_n^2]\|_2$: We compute that

$$\begin{aligned}\mathbb{E}[\mathbf{M}_n^2] = \frac{1}{4}\Big(&\mathbb{E}[\mathbf{D}_{n,1}\boldsymbol{\beta}_n\boldsymbol{\beta}_n^\top\mathbf{D}_{n,1}]\mathbb{E}[\boldsymbol{\beta}_n^\top\mathbf{D}_{n,2}^2\boldsymbol{\beta}_n] + \mathbb{E}[\mathbf{D}_{n,2}\boldsymbol{\beta}_n\boldsymbol{\beta}_n^\top\mathbf{D}_{n,2}]\mathbb{E}[\boldsymbol{\beta}_n^\top\mathbf{D}_{n,1}^2\boldsymbol{\beta}_n] \\ &+ \mathbb{E}[(\boldsymbol{\beta}_n^\top\mathbf{D}_{n,2}\mathbf{D}_{n,1}\boldsymbol{\beta}_n)\mathbf{D}_{n,1}\boldsymbol{\beta}_n\boldsymbol{\beta}_n^\top\mathbf{D}_{n,2}]] + \mathbb{E}[(\boldsymbol{\beta}_n^\top\mathbf{D}_{n,1}\mathbf{D}_{n,2}\boldsymbol{\beta}_n)\mathbf{D}_{n,2}\boldsymbol{\beta}_n\boldsymbol{\beta}_n^\top\mathbf{D}_{n,1}]] \\ &\mathbb{E}[\|u_1\|_2^2]\mathbb{E}[u_2u_2^\top] + \mathbb{E}[\|u_2\|_2^2]\mathbb{E}[u_1u_1^\top] + \mathbb{E}[u_1u_2^\top u_1u_2^\top] + \mathbb{E}[u_2u_1^\top u_2u_1^\top]\Big)\end{aligned}$$

Now since $\|\mathbf{D}_{n,i}\|_2 \leq 1$ and $\|\beta_n\|_2 \leq R$, the norm of the first four terms is bounded by R^4. Now note that

$$\|\mathbb{E}[u_iu_i^\top]\|_2 = \max_{\mathbf{v},\|\mathbf{v}\|_2\leq 1}\mathbb{E}[\mathbf{v}^\top u_iu_i^\top\mathbf{v}] = \max_{\mathbf{v},\|\mathbf{v}\|_2\leq 1}\mathbb{E}[(u_i^\top\mathbf{v})^2] \leq \frac{\sigma^2}{2\mu}$$

$$\|\mathbb{E}[\|u_i\|_2^2]\|_2 = \mathrm{Tr}(\mathbb{E}[u_iu_i^\top]) \leq \|\mathbb{E}[u_iu_i^\top]\|_2 d_A \leq \frac{\sigma^2 d_A}{2\mu}$$

$$\|\mathbb{E}[u_2u_1^\top u_2u_1^\top]\|_2 = \|\mathbb{E}[u_1u_2^\top u_1u_2^\top]\|_2 = \max_{\mathbf{v},\|\mathbf{v}\|_2\leq 1}\mathbb{E}[\mathbf{v}^\top u_1u_2^\top u_1u_2^\top\mathbf{v}]$$

$$\max_{\mathbf{v},\|\mathbf{v}\|_2\leq 1}\mathbb{E}[\mathbf{v}^\top u_1u_2^\top u_1u_2^\top\mathbf{v}] = \max_{\mathbf{v},\|\mathbf{v}\|_2\leq 1}\mathrm{Tr}(\mathbf{v}^\top\mathbb{E}[u_1u_1^\top]\mathbb{E}[u_2u_2^\top]\mathbf{v}] = \|\mathbb{E}[u_1u_1^\top]\mathbb{E}[u_2u_2^\top]\|_2 \leq \frac{\sigma^4}{4\mu^2}$$

Combining all of these, we get that

$$\|\mathbb{E}[\mathbf{M}_n^2]\|_2 \leq R^4 + \frac{\sigma^4(d_A+1)}{8\mu^2}$$

□

Proposition D.3.2 (Confidence Bound for $\overline{\mathbf{M}}_N$)**.** *With probability at least $1-\delta/2$, we have that $\|\overline{\boldsymbol{M}}_N - \mathbb{E}[\boldsymbol{M}_1]\|_2 \leq \Delta_M$ with*

$$\begin{aligned}\Delta_M := \sqrt{2\left\|\sum_{n=1}^N \boldsymbol{M}_n^2\right\|_2 \frac{\log(4d_A/\delta)}{N}} &+ 2R^2\left(2+\frac{H}{2\mu}\right)\left(\frac{2\log(4d_A/\delta)}{N}\right)^{3/4} \\ &+ 4R^2\left(2+\frac{H}{2\mu}\right)\frac{\log(4d_A/\delta)}{3N}\end{aligned}$$

Proof. Now since $\|\mathbf{M}_n^2\|_2 \leq \|\mathbf{M}_n\|_2^2 \leq R^4\left(2+\frac{H}{4\mu}\right)^2$, we have that $\|\mathbf{M}_n^2 - \mathbb{E}[\mathbf{M}_1^2]\|_2 \leq 2R^4\left(2+\frac{H}{4\mu}\right)^2$ by the matrix Hoeffding bound, we have that with probability $1-\delta/2$,

$$\left\|\frac{1}{N}\sum_{n=1}^{N}\mathbf{M}_n^2 - \mathbb{E}[\mathbf{M}_1^2]\right\|_2 \leq 4R^4\left(2+\frac{H}{2\mu}\right)^2\sqrt{\frac{\log(d_A/\delta)}{N}}$$

Further, by the matrix Bernstein inequality, we have that with probability $1-\delta/2$,

$$\left\|\frac{1}{N}\sum_{n=1}^{N}\mathbf{M}_n - \mathbb{E}[\mathbf{M}_1]\right\|_2 \leq \sqrt{2\|\mathbb{E}[\mathbf{M}_1^2]\|_2\frac{\log(d_A/\delta)}{N}} + 4R^2\left(2+\frac{H}{2\mu}\right)\frac{\log(d_A/\delta)}{3N}$$

Combining the two results and using a union bound, we get that with probability $1-\delta/2$

$$\begin{aligned}\|\overline{\mathbf{M}}_N - \mathbb{E}[\mathbf{M}_1]\|_2 &= \left\|\frac{1}{N}\sum_{n=1}^{N}\mathbf{M}_n - \mathbb{E}[\mathbf{M}_1]\right\|_2 \\ &\leq \sqrt{2\left\|\sum_{n=1}^{N}\mathbf{M}_n^2\right\|_2\frac{\log(2d_A/\delta)}{N}} + 2R^2\left(2+\frac{H}{2\mu}\right)\left(\frac{2\log(2d_A/\delta)}{N}\right)^{3/4} \\ &\quad + 4R^2\left(2+\frac{H}{2\mu}\right)\frac{\log(d_A/\delta)}{3N}\end{aligned}$$

□

Proposition D.3.3 (Confidence Bound for $\overline{\mathbf{D}}_{N,i}$)**.** *With probability $1-\delta/4$, for $i=1,2$, we have that $\|\overline{\boldsymbol{D}}_{N,i} - \mathbb{E}[\boldsymbol{D}_{n,i}]\|_2 = \left\|\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{D}_{n,i} - \mathbb{E}[\boldsymbol{D}_{n,i}]\right\|_2 \leq \Delta_D$ with*

$$\Delta_D \leq \sqrt{\frac{8\log(4d_A/\delta)}{N}}$$

Proof. Since $\|\mathbf{D}_{n,i}\|_2 \leq 1$, this immediately follows by the matrix Hoeffding inequality. □

Lemma D.3.4 (Confidence Bound for $\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1}$)**.** *We have that with probability $1-\delta$*

$$\begin{aligned}\|\overline{\boldsymbol{D}}_{N,1}^{-1}\overline{\boldsymbol{M}}_N\overline{\boldsymbol{D}}_{N,2}^{-1} - \mathbb{E}[\boldsymbol{D}_{N,1}]^{-1}\mathbb{E}[\boldsymbol{M}_{N,1}]\mathbb{E}[\boldsymbol{D}_{N,2}]^{-1}\|_2 &\leq \left(\frac{B_D^3(2-B_D\Delta_D)}{(1-B_D\Delta_D)^2}\right)(R^2+\Delta_M)\Delta_D \\ &\quad + \left(\frac{B_D}{1-B_D\Delta_D}\right)^2\Delta_M\end{aligned}$$

where $B_D = \max_{i=1,2} \|\overline{\boldsymbol{D}}_{N,i}^{-1}\|_2$.

Proof. For brevity, just for this proof, we define $\mathbf{D}_i = \mathbb{E}[\overline{\mathbf{D}}_{N,i}]$ for $i = 1, 2$ and by $\mathbf{M} := \mathbb{E}[\mathbf{M}_1] = \mathbb{E}[\overline{\mathbf{M}}_N]$. By a union bound, Propositions D.3.2 and D.3.3 hold with probability at least $1 - \delta$. The statements in the rest of this proof thus hold with probability at least $1 - \delta$. Now note that

$$\begin{aligned}\|\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_{N,1}\overline{\mathbf{D}}_{N,2}^{-1} - \mathbf{D}_1^{-1}\mathbf{M}\mathbf{D}_2^{-1}\|_2 \leq{}& \|\overline{\mathbf{D}}_{N,1}^{-1}\|_2\|\overline{\mathbf{M}}_{N,1}\|_2\|\overline{\mathbf{D}}_{N,2}^{-1} - \mathbf{D}_2^{-1}\|_2 \\ &+ \|\overline{\mathbf{D}}_{N,1}^{-1} - \mathbf{D}_1^{-1}\|_2\|\overline{\mathbf{M}}_{N,1}\|_2\|\mathbf{D}^{-1}\|_2 \\ &+ \|\mathbf{D}_1^{-1}\|_2\|\overline{\mathbf{M}}_{N,1} - \mathbf{M}\|_2\|\mathbf{D}_2^{-1}\|_2\end{aligned}$$

Now note that by inequality (1.1) from Wei et al. [2005], we have that

$$\|\overline{\mathbf{D}}_{N,i}^{-1} - \mathbf{D}_i^{-1}\|_2 \leq \frac{B_D^2\Delta_D}{1 - B_D\Delta_D}$$

This means that

$$\|\mathbf{D}_i^{-1}\|_2 \leq \|\overline{\mathbf{D}}_{N,i}^{-1}\|_2 + \|\overline{\mathbf{D}}_{N,i}^{-1} - \mathbf{D}_i^{-1}\|_2 \leq \frac{B_D}{1 - B_D\Delta_D}$$

Also, since contexts in both trajectory halves have the same distribution and contexts are generated independently, $\mathbf{M} = \mathbb{E}[\overline{\mathbf{M}}_1] = \mathbf{D}_i\mathbb{E}[\beta_1\beta_1^\top]\mathbf{D}_i$. So, $\|\mathbf{M}\|_2 \leq \|\mathbf{D}_1\|_2\|\mathbf{D}_2\|_2R^2 \leq R^2$. This implies that

$$\|\overline{\mathbf{M}}_{N,1}\|_2 \leq R^2 + \|\overline{\mathbf{M}}_{N,1} - \mathbf{M}\|_2 \leq R^2 + \Delta_M$$

Combining all these with the bound above, we get that

$$\begin{aligned}\|\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1} - \mathbb{E}[\overline{\mathbf{D}}_{N,1}]^{-1}\mathbb{E}[\mathbf{M}_{n,1}]\mathbb{E}[\overline{\mathbf{D}}_{N,2}]^{-1}\|_2 \leq{}& \left(\frac{B_D^3(2 - B_D\Delta_D)}{(1 - B_D\Delta_D)^2}\right)(R^2 + \Delta_M)\Delta_D \\ &+ \left(\frac{B_D}{1 - B_D\Delta_D}\right)^2\Delta_M\end{aligned}$$

□

Proof. First note that $\mathbb{E}[\mathbf{D}_{N,1}]^{-1}\mathbb{E}[\mathbf{M}_{N,1}]\mathbb{E}[\mathbf{D}_{N,2}]^{-1} = \mathbf{U}_\star(\Lambda + \mu_\theta\mu_\theta^\top)\mathbf{U}_\star^\top$. Also recall that $\hat{\mathbf{U}}$ is given by the top-d_K eigenvectors for $\mathbb{E}[\mathbf{D}_{N,1}^{-1}\mathbf{M}_{N,1}\mathbf{D}_{N,2}^{-1}]$. This means that there is a $d_K \times d_K$ unitary matrix $\mathbf{W}$ such that $\mathbf{U}_\star\mathbf{W}$ forms the eigenvectors for $\mathbf{U}_\star(\Lambda + \mu_\theta\mu_\theta^\top)\mathbf{U}_\star^\top$. This means that by the Davis-Kahan theorem for statisticians Yu et al. [2014] and Lemma D.3.4, we have that with probability $1 - \delta$

$$\begin{aligned}\|\hat{\mathbf{U}}\hat{\mathbf{U}}^\top - \mathbf{U}_\star\mathbf{U}_\star^\top\|_2 = \|\hat{\mathbf{U}}\hat{\mathbf{U}}^\top - \mathbf{U}_\star\mathbf{W}\mathbf{W}^\top\mathbf{U}_\star^\top\|_2 &= \sqrt{2}(d_K - \|\hat{\mathbf{U}}^\top\mathbf{U}_\star\|_F^2) \\ &\leq \frac{2\sqrt{2d_K}}{\hat{\lambda}}\|\mathbb{E}[\mathbf{D}_{N,1}]^{-1}\mathbb{E}[\mathbf{M}_{N,1}]\mathbb{E}[\mathbf{D}_{N,2}]^{-1} - \mathbf{D}_{N,1}^{-1}\mathbf{M}_{N,1}\mathbf{D}_{N,2}^{-1}\|_2\end{aligned}$$

$$\leq \frac{2\sqrt{2d_K}}{\hat{\lambda}} \left(\left(\frac{B_D^3(2 - B_D\Delta_D)}{(1 - B_D\Delta_D)^2} \right) (R^2 + \Delta_M)\Delta_D + \left(\frac{B_D}{1 - B_D\Delta_D} \right)^2 \Delta_M \right)$$

We thus set Δ_{off} to be the value above.

Bounding Δ_{off}: Now note that for large enough N, $\|\mathbb{E}[\mathbf{D}_{n,1}]^{-1}\|_2\Delta_D = \sqrt{\frac{8\log(d_A/\delta)}{\lambda_A N}} \leq \frac{1}{2}$. In particular, by applying inequality (1.1) from Wei et al. [2005], we have that $B_D \leq 2\|\mathbb{E}[\mathbf{D}_{n,1}]^{-1}\|_2 = \frac{2}{\lambda_A}$. This already gives us the much simpler expression

$$\Delta_{\text{off}} \leq \frac{2\sqrt{2d_K}}{\hat{\lambda}} \left(\frac{64}{\lambda_A^3}(R^2 + \Delta_M)\Delta_D + \frac{16}{\lambda_A^2}\Delta_M \right)$$

Also note that by Lemma D.3.1 and the matrix Hoeffding inequality, we have that

$$\begin{aligned} \|\frac{1}{N}\sum_{n=1}^{N} \mathbf{M}_n^2\| &\leq \|\mathbb{E}[\mathbf{M}_1^2]\| + 4R^4 \left(2 + \frac{H}{2\mu}\right)^2 \sqrt{\frac{\log(d_A/\delta)}{N}} \\ &\leq R^4 + \frac{\sigma^4(d_A + 1)}{8\mu^2} + 4R^4 \left(2 + \frac{H}{2\mu}\right)^2 \sqrt{\frac{\log(d_A/\delta)}{N}} \end{aligned}$$

We combine this with Proposition D.3.2 to get that

$$\Delta_M = O\left(\left(R^2 + \frac{\sigma^2\sqrt{d_A}}{\mu} \right) \sqrt{\frac{\log(d_A/\delta)}{N}} \right)$$

Since $\sigma^2 \leq R^2$, we get that

$$\Delta_M = O\left(R^2 \sqrt{\frac{d_A \log(d_A/\delta)}{N}} \right) = O(R^2)$$

Also recall from Proposition D.3.3, we get that

$$\Delta_D = \sqrt{\frac{8\log(d_A/\delta)}{N}}$$

Also recall that λ_θ is the minimum eigenvalue of $\frac{1}{R^2}\Lambda$, and so the minimum eigenvalue of $\mathbf{U}_\star(\Lambda + \mu_\theta\mu_\theta^\top)\mathbf{U}_\star^\top$ is larger than $R^2\lambda_\theta$. We then conclude that $\hat{\lambda} \geq R^2\lambda_\theta - 2\Delta_M \geq \frac{\lambda_\theta}{2}$ for large enough N. Combining all these, we get that

$$\Delta_{\text{off}} = O\left(\frac{\sqrt{d_K}}{R^2\lambda_\theta} \left(\frac{R^2}{\lambda_A^3} + \frac{R^2}{\lambda_A^2}\sqrt{d_A} \right) \sqrt{\frac{\log(d_A/\delta)}{N}} \right)$$

$$= O\left(\frac{1}{\lambda_\theta \lambda_A^3}\sqrt{\frac{d_K d_A \log(d_A/\delta)}{N}}\right)$$

□

D.4 Proofs for LOCAL-UCB

Theorem D.4.1 (LOCAL-UCB Regret)**.** *Under Assumptions 11 and 12, if* $\alpha_{1,t} = R\sqrt{\mu} + CR\sqrt{d_K \log(2T/\delta)}$ *and* $\alpha_{2,t} = R\sqrt{\mu} + CR\sqrt{d_A \log(2T/\delta)}$ *for a universal constant* C*, then with probability at least* $1-\delta$ *over offline data and online rewards, LOCAL-UCB has regret* Reg_T *bounded by*

$$O\left(\min\left(Rd_A\sqrt{T}, Rd_K\sqrt{T}\left(1+\frac{1}{\lambda_\theta\lambda_A^3}\sqrt{\frac{d_A T}{d_K N}}\right)\right)\right).$$

Proof. Let the true latent state for the given trajectory be $\boldsymbol{\theta}_\star$, so that the reward parameter $\boldsymbol{\beta}_\star = \mathbf{U}_\star \boldsymbol{\theta}_\star$.

Showing that $\mathbf{U}_\star, \boldsymbol{\beta}_\star$ are in our confidence set: We check all our constraints:

- Note that with probability $1-\delta/3$, $\mathbf{U}_\star$ satisfies $\|\hat{\mathbf{U}}^\top \mathbf{U}_\star\|_F \geq \sqrt{d_K - \Delta_{\text{off}}^2/2}$.
- From the standard confidence ellipsoid bound for linear models applied to dimension d_A, with probability $1-\delta/3T$, $\|\hat{\boldsymbol{\beta}}_{2,t} - \boldsymbol{\beta}_\star\|_{\mathbf{V}_t^{-1}} \leq \alpha_{2,t}$ for all t.
- Note that $\mathbf{U}_\star \mathbf{U}_\star^\top \boldsymbol{\beta}_\star = \boldsymbol{\beta}_\star$.
- Finally, we apply the standard confidence ellipsoid bound for linear models to dimension d_K instead of d_A with model $r_t = (\phi(x_t, a_t)^\top \mathbf{U}_\star)(\mathbf{U}_\star^\top \beta_\star) + \epsilon_t$. This means that $\|\mathbf{U}_\star^\top \beta_\star - \mathbf{U}_\star^\top \hat{\beta}_{1,t}\|_{\mathbf{V}_{1,t}^{-1}} \leq \alpha_{1,t}$ where $\mathbf{V}_{1,t} = (\mathbf{I}_{d_K} + \sum_{s=1}^{t-1} \mathbf{U}_\star^\top \phi(x_t, a_t)\phi(x_t, a_t)^\top \mathbf{U}_\star)$. Note that since $\mathbf{U}_\star \mathbf{U}_\star^\top = \mathbf{I}_{d_K}$, we have that $\mathbf{V}_{1,t} = (\mathbf{U}_\star^\top \mathbf{V}_t \mathbf{U}_\star)$. So, $\|\mathbf{U}_\star^\top(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{1,t})\|_{(\mathbf{U}_\star^T \mathbf{V}_t \mathbf{U}_\star)^{-1}} \leq \alpha_{1,t}$ holds with probability at least $1-\delta/3T$ for all t.

So, by a union bound over all events, we get that for all actions a, the tuple $(a, \boldsymbol{\beta}_\star, \mathbf{U}_\star)$ satisfies our conditions with probability $1-\delta$.

Leveraging optimism in low dimension: From above and by the optimistic design of the algorithm, with probability $1-\delta$, $\phi(x_t, a_t)^\top \tilde{\boldsymbol{\beta}}_t$ is an upper bound on $\phi(x_t, a)^\top \boldsymbol{\beta}_\star$ for any action a.

Thus, we have the following regret decomposition with probability at least $1-\delta$:

$$
\begin{aligned}
\mathrm{Reg}_T &= \sum_{t=1}^{T} \phi(x_t, a_t^\star)^\top \boldsymbol{\beta}_\star - \phi(x_t, a_t)^\top \boldsymbol{\beta}_\star \\
&\leq \sum_{t=1}^{T} \phi(x_t, a_t)^\top \tilde{\boldsymbol{\beta}}_t - \phi(x_t, a_t)^\top \boldsymbol{\beta}_\star \\
&\overset{(i)}{=} \sum_{t=1}^{T} \phi(x_t, a_t)^\top \tilde{\mathbf{U}}_t \tilde{\mathbf{U}}_t^\top \tilde{\boldsymbol{\beta}}_t - \phi(x_t, a_t)^\top \mathbf{U}_\star \mathbf{U}_\star^\top \boldsymbol{\beta}_\star \\
&\leq \sum_{t=1}^{T} \phi(x_t, a_t)^\top \tilde{(\mathbf{U}_t \tilde{\mathbf{U}}_t^\top - \mathbf{U}_\star \mathbf{U}_\star^\top)} \tilde{\boldsymbol{\beta}}_t + \phi(x_t, a_t)^\top \mathbf{U}_\star \mathbf{U}_\star^\top (\boldsymbol{\beta}_t - \boldsymbol{\beta}_\star) \\
&\leq RT \|\mathbf{U}_t \tilde{\mathbf{U}}_t^\top - \mathbf{U}_\star \mathbf{U}_\star^\top\|_2 + \sum_{t=1}^{T} \phi(x_t, a_t)^\top \mathbf{U}_\star \mathbf{U}_\star^\top (\boldsymbol{\beta}_t - \boldsymbol{\beta}_\star) \\
&\leq RT\Delta_{\text{off}} + \sum_{t=1}^{T} \phi(x_t, a_t)^\top \mathbf{U}_\star \mathbf{U}_\star^\top (\boldsymbol{\beta}_t - \boldsymbol{\beta}_\star) \\
&\leq RT\Delta_{\text{off}} + \sum_{t=1}^{T} \|\phi(x_t, a_t)^\top \mathbf{U}_\star\|_{(\mathbf{U}_\star^\top \mathbf{V}_t \mathbf{U}_\star)^{-1}} \|\mathbf{U}_\star^\top (\boldsymbol{\beta}_t - \boldsymbol{\beta}_\star)\|_{(\mathbf{U}_\star^\top \mathbf{V}_t \mathbf{U}_\star)^{-1}} \\
&\leq RT\Delta_{\text{off}} + \sum_{t=1}^{T} \alpha_{2,t} \|\phi(x_t, a_t)^\top \mathbf{U}_\star\|_{(\mathbf{U}_\star^\top \mathbf{V}_t \mathbf{U}_\star)^{-1}} \\
&\overset{(ii)}{=} RT\Delta_{\text{off}} + \sum_{t=1}^{T} \alpha_{2,t} \|\phi(x_t, a_t)^\top \mathbf{U}_\star\|_{\mathbf{V}_{1,t}^{-1}} \\
&\overset{(iii)}{\leq} RT\Delta_{\text{off}} + O\left(d_K \sqrt{T \log(T/\delta)}\right) \\
&\overset{(iv)}{=} O\left(R d_K \sqrt{T \log(T/\delta)} + \frac{R\lambda_D^3}{\lambda_{\min}} \sqrt{\frac{T^2 d_K d_A \log(d_A/\delta)}{N}}\right) \\
&= O\left(R d_K \sqrt{T \log(T/\delta)} \left(1 + \frac{\lambda_D^3}{\lambda_{\min}} \sqrt{\frac{T d_A \log(d_A)}{d_K N}}\right)\right) \\
&= \tilde{O}\left(R d_K \sqrt{T} \left(1 + \sqrt{\frac{T d_A}{d_K N}}\right)\right)
\end{aligned}
$$

where (i) holds since $\tilde{\mathbf{U}}_t \tilde{\mathbf{U}}_t^\top \tilde{\boldsymbol{\beta}}_t = \tilde{\boldsymbol{\beta}}_t$ and $\mathbf{U}_\star \mathbf{U}_\star^\top \boldsymbol{\beta}_\star = \boldsymbol{\beta}_\star$ (ii) holds since $\mathbf{V}_{1,t}^{-1} = (\mathbf{U}_\star^\top \mathbf{V}_t \mathbf{U}_\star)^{-1}$, (iii) holds by the usual proof of LinUCB applied to dimension d_K, and (iv) holds by Theorem 6.3.2.

Leveraging optimism in high dimension: This is merely the proof of LinUCB. From above and by the optimistic design of the algorithm, with probability $1-\delta$, $\phi(x_t, a_t)^\top \tilde{\boldsymbol{\beta}}_t$ is an upper bound

on $\phi(x_t, a)^\top \boldsymbol{\beta}_\star$ for any action a. Thus, we have the following regret decomposition with probability at least $1 - \delta$:

$$\begin{aligned}
\mathrm{Reg}_T &= \sum_{t=1}^{T} \phi(x_t, a_t^\star)^\top \boldsymbol{\beta}_\star - \phi(x_t, a_t)^\top \tilde{\boldsymbol{\beta}}_\star \\
&\leq \sum_{t=1}^{T} \phi(x_t, a_t)^\top \tilde{\boldsymbol{\beta}}_t - \phi(x_t, a_t)^\top \boldsymbol{\beta}_\star \\
&= \sum_{t=1}^{T} \|\phi(x_t, a_t)^\top\|_{\mathbf{V}_t^{-1}} \|\tilde{\boldsymbol{\beta}}_t - \boldsymbol{\beta}_\star\|_{\mathbf{V}_t} \\
&= O(R d_A \sqrt{T \log(T/\delta)}) \\
&= \tilde{O}(R d_A \sqrt{T})
\end{aligned}$$

where the last line follows from the standard regret bound for LinUCB applied to dimension d_A.

Combining the two bounds, we have our result.

$$\begin{aligned}
\mathrm{Reg}_T &= O\left(\min\left(R d_A \sqrt{T \log(T/\delta)}, R d_K \sqrt{T \log(T/\delta)}\left(1 + \frac{\lambda_D^3}{\lambda_{\min}} \sqrt{\frac{T d_A \log(d_A)}{d_K N}}\right)\right)\right) \\
&= \tilde{O}\left(\min\left(R d_A \sqrt{T}, R d_K \sqrt{T}\left(1 + \sqrt{\frac{T d_A}{d_K N}}\right)\right)\right)
\end{aligned}$$

□

D.5 Proofs for ProBALL-UCB

D.5.1 Confidence bound for the low-dimensional reward parameter

Lemma D.5.1 (Confidence Bound for $\hat{\boldsymbol{\beta}}_{1,t}$)**.** *If for all timesteps t, $\phi(x_t, a_t)$ lies in the span of $\hat{\boldsymbol{U}}$, we have that for a universal constant C,*

$$\left\| \hat{\boldsymbol{U}}^\top \hat{\boldsymbol{\beta}}_{1,t} - \hat{\boldsymbol{U}}^\top \boldsymbol{\beta}_\star \right\|_{\boldsymbol{V}_{1,t}^{-1}} \leq R\sqrt{\mu} + CR\sqrt{d_A \log(t/\delta)}$$

Otherwise, we have that

$$\left\|\hat{\boldsymbol{U}}^\top \hat{\boldsymbol{\beta}}_{1,t} - \hat{\boldsymbol{U}}^\top \boldsymbol{\beta}_\star\right\|_{\boldsymbol{V}_{1,t}^{-1}} \leq R\sqrt{\mu} + R\kappa_t + \Delta_{\text{off}} CR\sqrt{d_A \log(t/\delta)}$$

where $\kappa_t = \|\hat{\boldsymbol{U}}^\top \boldsymbol{X}_t^\top \boldsymbol{X}_t\|_{(\hat{\boldsymbol{U}}^\top \boldsymbol{V}_t \hat{\boldsymbol{U}})^{-1}}$*, with notation* $\|\boldsymbol{A}\|_{\boldsymbol{C}} := \sqrt{\|\boldsymbol{A}^\top \boldsymbol{C}\boldsymbol{A}\|_2}$

Proof. For brevity, in this proof we will denote by $\mathbf{X}_t := [\phi(x_1, a_1), \dots \phi(x_{t-1}, a_{t-1})]^\top$, which is a $t \times d_A$ matrix. Recall that $\mathbf{V}_t = \mu \mathbf{I}_{d_A} + \mathbf{X}_t^\top \mathbf{X}_t$. We will also denote $\mathbf{V}_{1,t} = \hat{\mathbf{U}}^\top \mathbf{V}_t \hat{\mathbf{U}}$. Note that the vector of rewards is given by $\mathbf{X}_t \boldsymbol{\beta}_\star + \eta_t$, where η_t is a random vector of t independent R^2-subgaussian entries. So, $\boldsymbol{b}_t = \mathbf{X}_t^\top (\mathbf{X}_t \boldsymbol{\beta}_\star + \eta_t)$. Also define the notation $\Delta_U := \hat{\mathbf{U}}\hat{\mathbf{U}}^\top - \mathbf{U}_\star \mathbf{U}_\star^\top$.

If all actions lie in the span of $\hat{\mathbf{U}}$. Note that since all actions taken lie in the span of $\hat{\mathbf{U}}$, $\mathbf{X}_t = \mathbf{X}_t \hat{\mathbf{U}}\hat{\mathbf{U}}^\top$. *This is the key observation.* Now note that

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{1,t} &= \hat{\mathbf{U}} \mathbf{V}_{1,t}^{-1} \hat{\mathbf{U}}^\top \mathbf{X}_t^\top (\mathbf{X}_t \boldsymbol{\beta}_\star + \eta_t) \\
&= \hat{\mathbf{U}} \mathbf{V}_{1,t}^{-1} \hat{\mathbf{U}}^\top \mathbf{X}_t^\top \mathbf{X}_t \boldsymbol{\beta}_\star + \hat{\mathbf{U}} \mathbf{V}_{1,t}^{-1} \hat{\mathbf{U}}^\top \mathbf{X}_t^\top \eta_t \\
&= \hat{\mathbf{U}} \mathbf{V}_{1,t}^{-1} \hat{\mathbf{U}}^\top \mathbf{X}_t^\top \mathbf{X}_t \hat{\mathbf{U}}\hat{\mathbf{U}}^\top \boldsymbol{\beta}_\star + \hat{\mathbf{U}} \mathbf{V}_{1,t}^{-1} \hat{\mathbf{U}}^\top \mathbf{X}_t^\top \eta_t \\
&= \hat{\mathbf{U}} \mathbf{V}_{1,t}^{-1} (\mathbf{V}_{1,t} - \mu \mathbf{I}_{d_K}) \hat{\mathbf{U}}^\top \boldsymbol{\beta}_\star + \hat{\mathbf{U}} \mathbf{V}_{1,t}^{-1} \hat{\mathbf{U}}^\top \mathbf{X}_t^\top \eta_t \\
&= \hat{\mathbf{U}} \mathbf{U}^\top \boldsymbol{\beta}_\star - \mu \hat{\mathbf{U}} \mathbf{V}_{1,t}^{-1} \hat{\mathbf{U}}^\top \boldsymbol{\beta}_\star + \hat{\mathbf{U}} \mathbf{V}_{1,t}^{-1} \hat{\mathbf{U}}^\top \mathbf{X}_t^\top \eta_t
\end{aligned}$$

The rest of the proof is similar to Theorem 2 in Abbasi-Yadkori et al. [2011]. We first note that for any x, we have that

$$\begin{aligned}
x^\top (\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top \boldsymbol{\beta}_\star) &\leq \left| (x^\top \hat{\mathbf{U}}) \mathbf{V}_{1,t}^{-1} (-\mu \hat{\mathbf{U}}^\top \boldsymbol{\beta}_\star + \hat{\mathbf{U}}^\top \mathbf{X}_t^\top \eta_t) \right| \\
&\leq \|\hat{\mathbf{U}}^\top x\|_{\mathbf{V}_{1,t}^{-1}} \left\| (-\mu \hat{\mathbf{U}}^\top \boldsymbol{\beta}_\star + \hat{\mathbf{U}}^\top \mathbf{X}_t^\top \eta_t) \right\|_{\mathbf{V}_{1,t}^{-1}} \\
&\leq \|\hat{\mathbf{U}}^\top x\|_{\mathbf{V}_{1,t}^{-1}} \left(\mu \|\hat{\mathbf{U}}^\top \boldsymbol{\beta}_\star\|_{\mathbf{V}_{1,t}^{-1}} + \|\hat{\mathbf{U}}^\top \mathbf{X}_t^\top \eta_t\|_{\mathbf{V}_{1,t}^{-1}} \right)
\end{aligned}$$

Now note that $\mu \|\hat{\mathbf{U}}^\top \boldsymbol{\beta}_\star\|_{\mathbf{V}_{1,t}^{-1}} \leq \|\hat{\mathbf{U}}^\top \boldsymbol{\beta}_\star\|_2 \sqrt{\mu} \leq R\sqrt{\mu}$ and by the self normalized martingale concentration inequality from Abbasi-Yadkori et al. [2011] applied to d_K dimensional vectors $\hat{\mathbf{U}}^\top \mathbf{X}_t$, we have that with probability at least $1 - \delta$, $\|\hat{\mathbf{U}}^\top \mathbf{X}_t^\top \eta_t\|_{\mathbf{V}_{1,t}^{-1}} \leq CR\sqrt{d_A \log(t/\delta)}$ for some universal

constant C. So we have with probability at least $1-\delta$ that

$$= \left\|\hat{\mathbf{U}}(\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star)\right\|^2_{\mathbf{V}^{-1}_{1,t}} \leq \left\|\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\right\|_{\mathbf{V}^{-1}_{1,t}} \left(R\sqrt{\mu} + CR\sqrt{d_A \log(t/\delta)}\right)$$

This means that

$$\begin{aligned}\left\|\hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\right\|_{\mathbf{V}^{-1}_{1,t}} &= \left\|\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\right\|_{\mathbf{V}^{-1}_{1,t}} \\ &\leq R\sqrt{\mu} + CR\sqrt{d_A \log(t/\delta)}\end{aligned}$$

If all actions don't lie in the span of $\hat{\mathbf{U}}$. This time note that

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{1,t} &= \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top(\mathbf{X}_t\boldsymbol{\beta}_\star + \eta_t) \\ &= \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\boldsymbol{\beta}_\star + \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t \\ &= \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\hat{\mathbf{U}}_\star\mathbf{U}_\star^\top\boldsymbol{\beta}_\star + \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\mathbf{U}^\top\mathbf{X}_t^\top\eta_t \\ &= \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star + \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\Delta_U\boldsymbol{\beta}_\star + \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t \\ &= \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}(\mathbf{V}_{1,t} - \mu\mathbf{I}_{d_K})\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star + \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\Delta_U\boldsymbol{\beta}_\star + \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t \\ &= \hat{\mathbf{U}}\mathbf{U}^\top\boldsymbol{\beta}_\star - \mu\hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star + \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\Delta_U\boldsymbol{\beta}_\star + \hat{\mathbf{U}}\mathbf{V}^{-1}_{1,t}\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t\end{aligned}$$

The rest of the proof is similar to Theorem 2 in Abbasi-Yadkori et al. [2011]. We first note that for any x, we have that

$$\begin{aligned}x^\top(\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star) &\leq \left|(x^\top\hat{\mathbf{U}})\mathbf{V}^{-1}_{1,t}(-\mu\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star + \hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\Delta_U\boldsymbol{\beta}_\star + \hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t)\right| \\ &\leq \|\hat{\mathbf{U}}^\top x\|_{\mathbf{V}^{-1}_{1,t}} \left\|(-\mu\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star + \hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\Delta_U\boldsymbol{\beta}_\star + \hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t)\right\|_{\mathbf{V}^{-1}_{1,t}} \\ &\leq \|\hat{\mathbf{U}}^\top x\|_{\mathbf{V}^{-1}_{1,t}} \left(\mu\|\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\|_{\mathbf{V}^{-1}_{1,t}} + \|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\Delta_U\boldsymbol{\beta}_\star\|_{\mathbf{V}^{-1}_{1,t}} + \|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t\|_{\mathbf{V}^{-1}_{1,t}}\right) \\ &\overset{(i)}{\leq} \|\hat{\mathbf{U}}^\top x\|_{\mathbf{V}^{-1}_{1,t}} \left(\mu\|\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\|_{\mathbf{V}^{-1}_{1,t}} + \|\Delta_U\boldsymbol{\beta}_\star\|_2\|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\|_{\mathbf{V}^{-1}_{1,t}} + \|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t\|_{\mathbf{V}^{-1}_{1,t}}\right) \\ &\leq \|\hat{\mathbf{U}}^\top x\|_{\mathbf{V}^{-1}_{1,t}} \left(\mu\|\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\|_{\mathbf{V}^{-1}_{1,t}} + R\|\Delta_U\|_2\|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\|_{\mathbf{V}^{-1}_{1,t}} + \|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t\|_{\mathbf{V}^{-1}_{1,t}}\right) \\ &\leq \|\hat{\mathbf{U}}^\top x\|_{\mathbf{V}^{-1}_{1,t}} \left(\mu\|\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\|_{\mathbf{V}^{-1}_{1,t}} + R\Delta_{\text{off}}\|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\|_{\mathbf{V}^{-1}_{1,t}} + \|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t\|_{\mathbf{V}^{-1}_{1,t}}\right)\end{aligned}$$

Here, (i) holds because for any matrices $\mathbf{A}, \mathbf{C}$ and any vector $\mathbf{v}$, we have that $\|\mathbf{A}\mathbf{v}\|_{\mathbf{C}} = \sqrt{\mathbf{v}^\top\mathbf{A}^\top\mathbf{C}\mathbf{A}\mathbf{v}} \leq \|\mathbf{v}\|_2\sqrt{\|\mathbf{A}^\top\mathbf{C}\mathbf{A}\|_2} = \|\mathbf{v}\|_2\|\mathbf{A}\|_{\mathbf{C}}$, where we recall that $\|\mathbf{A}\|_{\mathbf{C}} := \sqrt{\|\mathbf{A}^\top\mathbf{C}\mathbf{A}\|_2}$.

Now note that $\mu\|\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\|_{\mathbf{V}_{1,t}^{-1}} \le \|\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\|_2\sqrt{\mu} \le R\sqrt{\mu}$ and by the self normalized martingale concentration inequality from Abbasi-Yadkori et al. [2011] applied to d_K dimensional vectors $\hat{\mathbf{U}}^\top\mathbf{X}_t$, we have that with probability at least $1-\delta$, $\|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\eta_t\|_{\mathbf{V}_{1,t}^{-1}} \le CR\sqrt{d_A\log(t/\delta)}$ for some universal constant C. Also recall that $\|\hat{\mathbf{U}}^\top\mathbf{X}_t^\top\mathbf{X}_t\|_{\mathbf{V}_{1,t}^{-1}} = \kappa_t$. So we have with probability at least $1-\delta$ that

$$\left\|\hat{\mathbf{U}}(\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star)\right\|^2_{\mathbf{V}_{1,t}^{-1}} \le \left\|\hat{\mathbf{U}}^\top(\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star)\right\|_{\mathbf{V}_{1,t}^{-1}} \left(R\sqrt{\mu} + R\Delta_{\text{off}}\kappa_t + CR\sqrt{d_A\log(t/\delta)}\right)$$

This means that

$$\begin{aligned}\left\|\hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star\right\|_{\mathbf{V}_{1,t}^{-1}} &= \left\|\hat{\mathbf{U}}(\hat{\boldsymbol{\beta}}_{1,t} - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star)\right\|_{\mathbf{V}_{1,t}^{-1}} \\ &\le R\sqrt{\mu} + R\Delta_{\text{off}}\kappa_t + CR\sqrt{d_A\log(t/\delta)}\end{aligned}$$

Note that $\kappa_t = \left\|\sum_{s=1}^{t-1}\hat{\mathbf{U}}^\top\phi(x_s,a_s)\phi(x_s,a_s)^\top\right\|_{\mathbf{V}_{1,t}^{-1}} = \frac{1}{\sqrt{\mu}}\left\|\sum_{s=1}^{t-1}\hat{\mathbf{U}}^\top\phi(x_s,a_s)\phi(x_s,a_s)^\top\right\|_2 = O(t)$, but this is a worst case bound. □

We also state a lemma, borrowed from the standard proof of LinUCB regret.

Lemma D.5.2. *For any sequence of actions and contexts x_t, a_t and $\boldsymbol{V}_t = \mu\boldsymbol{I} + \sum_{s=1}^{t-1}\phi(x_t,a_t)\phi(x_t,a_t)^\top$, we have that*

$$\sum_{t=1}^{T}\min\left(\|\phi(x_t,a_t)\|^2_{\boldsymbol{V}_t^{-1}}, 1\right) = O(\sqrt{d_A})$$

$$\sum_{t=1}^{T}\min\left(\|\hat{\boldsymbol{U}}^\top\phi(x_t,a_t)\|^2_{(\hat{\boldsymbol{U}}^\top\boldsymbol{V}_t\hat{\boldsymbol{U}})^{-1}}, 1\right) = O(\sqrt{d_K})$$

Proof. This follows immediately from Lemma 11 of Abbasi-Yadkori et al. [2011]. □

D.5.2 Proof of the theorem

Theorem D.5.3 (Regret for ProBALL-UCB)**.** *Let $\alpha_{1,t} = R\sqrt{\mu} + \tau' R\Delta_{\text{off}}\kappa_t + CR\sqrt{d_K\log(T/\delta)}$ and let $\alpha_{2,t} = R\sqrt{\mu} + CR\sqrt{d_A\log(T/\delta)}$. Let S be the first timestep when Algorithm 13 does not play Line 6 and let $S = T$ if no such timestep exists. For $\tau = \tau' = 1$ we have that*

$$\text{Reg}_T = \widetilde{O}\left(\min\left(\text{Reg}_{on,T}, \text{Reg}_{hyb,T}\right)\right).$$

where $\mathrm{Reg}_{on,T} = Rd_A\sqrt{T}$ *and* $\mathrm{Reg}_{hyb,T}$ *is defined as*

$$Rd_K\sqrt{T}\left(1+\frac{1}{\lambda_A^3\lambda_\theta}\left(\sqrt{\frac{d_AT}{d_KN}}+\sqrt{\frac{d_A}{SN}\sum_{t=1}^{S}\kappa_t^2}\right)\right).$$

In the worst case, $\kappa_t = O(t)$ *and so* $\frac{1}{S}\sum_{t=1}^{S}\kappa_t^2 = O(T^2)$*, but if all features* $\phi(x_t, a_t)$ *lie in the span of* $\hat{\boldsymbol{U}}$ *for* $t \le S$*, then* $\frac{1}{S}\sum_{t=1}^{S}\kappa_t^2 = O(T)$*.*

Proof. We will first show that our bonuses give optimistic estimates of the true reward whp and then leverage the optimism.

Showing optimism when $\tau\Delta_{\text{off}}\sqrt{T} \le d_A$**:** In this case, we are inside the "if" statement and are running a projected and modified version of LinUCB. In that case, for all timesteps the projected version is run, all features will lie in the span of $\hat{\mathbf{U}}$. This is because the maximization problem is given by

$$a_t = \arg\max_{a,\|\phi(x_t,a)\|_2\le 1} \phi^\top\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_{1,t} + \|\hat{\mathbf{U}}^\top\phi(x_t,a)\|_{(\hat{\mathbf{U}}^\top\mathbf{V}_{1,t}\hat{\mathbf{U}})^{-1}}$$

If our features are isotropic, then this is only maximized by a feature in the span of $\hat{\mathbf{U}}$. So, the conditions of the first bound in Lemma D.5.1 are fulfilled. We will denote $\mathbf{V}_{1,t} = \hat{\mathbf{U}}^\top\mathbf{V}_t\hat{\mathbf{U}}$. We have that with probability $1-\delta/2$, for all x, a, t, the following holds.

$$\begin{aligned}
|\phi(x,a)^\top\boldsymbol{\beta}_\star - \phi(x,a)^\top\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_{1,t}| &\le |\phi(x,a)^\top\hat{\mathbf{U}}(\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star - \hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_{1,t})| + |\phi(x,a)^\top(\mathbf{U}_\star\mathbf{U}_\star^\top - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top)\boldsymbol{\beta}_\star| \\
&\le \|\hat{\mathbf{U}}^\top\phi(x,a)\|_{\mathbf{V}_{1,t}^{-1}}\|\hat{\mathbf{U}}^\top\boldsymbol{\beta}_\star - \hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_{1,t}\|_{\mathbf{V}_{1,t}^{-1}} + R\|\mathbf{U}_\star\mathbf{U}_\star^\top - \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\|_2 \\
&\le \alpha_{1,t}\|\hat{\mathbf{U}}^\top\phi(x,a)\|_{\mathbf{V}_{1,t}^{-1}} + R\Delta_{\text{off}}
\end{aligned}$$

This shows that with probability at least $1-\delta$ and for all x, a, t,

$$\phi(x,a)^\top\boldsymbol{\beta}_\star \le \phi(x,a)^\top\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_{1,t} + \alpha_{1,t}\|\hat{\mathbf{U}}^\top\phi(x,a)\|_{\mathbf{V}_{1,t}^{-1}} + R\Delta_{\text{off}}$$

This implies that with probability at least $1-\delta$, for any action a,

$$\begin{aligned}
\phi(x_t,a)^\top\boldsymbol{\beta}_\star &\le \max_a \phi(x_t,a)^\top\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_{1,t} + \alpha_{1,t}\|\hat{\mathbf{U}}^\top\phi(x_t,a)\|_{\mathbf{V}_{1,t}^{-1}} + R\Delta_{\text{off}} \\
&= \phi(x_t,a_t)^\top\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\hat{\boldsymbol{\beta}}_{1,t} + \alpha_{1,t}\|\hat{\mathbf{U}}^\top\phi(x_t,a_t)\|_{\mathbf{V}_{1,t}^{-1}} + R\Delta_{\text{off}}
\end{aligned}$$

Leveraging optimism when $\tau\Delta_{\text{off}}\sqrt{T} \le d_A$**:** Consider the standard regret decomposition,

which holds with probability $1-\delta$:

$$
\begin{aligned}
\mathrm{Reg}_T &= \sum_{t=1}^{T} \phi(x_t, a_t^\star)^\top \boldsymbol{\beta}_\star - \phi(x_t, a_t)^\top \boldsymbol{\beta}_\star \\
&\leq \sum_{t=1}^{T} \phi(x_t, a_t)^\top \hat{\mathbf{U}}\hat{\mathbf{U}}^\top \hat{\boldsymbol{\beta}}_{1,t} - \phi(x_t, a_t)^\top \boldsymbol{\beta}_\star + \alpha_{1,t} \|\hat{\mathbf{U}}^\top \phi(x_t, a_t)\|_{\mathbf{V}_{1,t}^{-1}} + R\Delta_{\text{off}} \\
&\leq \sum_{t=1}^{T} 2\alpha_{1,t} \|\hat{\mathbf{U}}^\top \phi(x_t, a_t)\|_{\mathbf{V}_{1,t}^{-1}} + 2R\Delta_{\text{off}} \qquad \text{(D.2)} \\
&\overset{(i)}{\leq} \sum_{t=1}^{T} 2\alpha'_{1,t} \min\left(\|\hat{\mathbf{U}}^\top \phi(x_t, a_t)\|_{\mathbf{V}_{1,t}^{-1}}, 1\right) + 2R\Delta_{\text{off}} \\
&\leq \sum_{t=1}^{T} 2(R\sqrt{\mu} + C\sqrt{d_K \log(T/\delta)}) \min\left(\|\hat{\mathbf{U}}^\top \phi(x_t, a_t)\|_{\mathbf{V}_{1,t}^{-1}}, 1\right) + 2R\Delta_{\text{off}} \\
&\quad + \sum_{t=1}^{T} 2R\Delta_{\text{off}} \kappa_t \min\left(\|\hat{\mathbf{U}}^\top \phi(x_t, a_t)\|_{\mathbf{V}_{1,t}^{-1}}, 1\right) \\
&\overset{(ii)}{\leq} 2(R\sqrt{\mu} + C\sqrt{d_K \log(T/\delta)}) \sqrt{\sum_{t=1}^{T} \min\left(\|\hat{\mathbf{U}}^\top \phi(x_t, a_t)\|^2_{\mathbf{V}_{1,t}^{-1}}, 1\right)} + 2R\Delta_{\text{off}} T \\
&\quad + 2R\Delta_{\text{off}} \sqrt{\sum_{t=1}^{T} \kappa_t^2} \sqrt{\sum_{t=1}^{T} \min\left(\|\hat{\mathbf{U}}^\top \phi(x_t, a_t)\|^2_{\mathbf{V}_{1,t}^{-1}}, 1\right)} \\
&\overset{(iii)}{\leq} O(d_K \sqrt{T \log(T/\delta)}) + 2R\Delta_{\text{off}} T + 2R\Delta_{\text{off}} \sqrt{\frac{\sum_{t=1}^{T} \kappa_t^2}{T}} \sqrt{d_K T} \qquad \text{(D.3)} \\
&= O\left(Rd_K \sqrt{T \log(T/\delta)} \left(1 + \Delta_{\text{off}} \left(\frac{\sqrt{T}}{d_K} + \sqrt{\frac{1}{T d_K} \sum_{t=1}^{T} \kappa_t^2}\right)\right)\right)
\end{aligned}
$$

where (i) holds since $\phi(x_t, a_t^\star)^\top \boldsymbol{\beta}_\star - \phi(x_t, a_t)^\top \boldsymbol{\beta}_\star \leq 2R$, (ii) holds by the Cauchy Schwarz inequality and (iii) holds by Lemma D.5.2. So, we have that

$$
\begin{aligned}
\mathrm{Reg}_T &= O\left(Rd_K \sqrt{T \log(T/\delta)} \left(1 + \frac{1}{\lambda_A^3 \lambda_\theta} \sqrt{\frac{d_A \log(d_A)}{N d_K}} \left(\sqrt{T} + \sqrt{\frac{d_K}{T} \sum_{t=1}^{S} \kappa_t^2}\right)\right)\right) \\
&= \widetilde{O}\left(Rd_K \sqrt{T} \left(1 + \frac{1}{\lambda_A^3 \lambda_\theta} \left(\sqrt{\frac{d_A T}{N d_K}} + \sqrt{\frac{d_A}{TN} \sum_{t=1}^{T} \kappa_t^2}\right)\right)\right)
\end{aligned}
$$

$$= \widetilde{O}\left(Rd_K\sqrt{T}\left(1 + \frac{1}{\lambda_A^3\lambda_\theta}\left(\sqrt{\frac{d_A T}{Nd_K}} + \sqrt{\frac{d_A}{\min(S,T)N}\sum_{t=1}^{\min(S,T)}\kappa_t^2}\right)\right)\right)$$

where the last inequality holds since $T < S$ for the first timestep S satisfying $\Delta_{\text{off}}\left(\sqrt{S} + \sqrt{\frac{d_K}{S}\sum_{t=1}^{S}\kappa_t^2}\right) \geq d_A$. Additionally, since $\Delta_{\text{off}}\left(\sqrt{T} + \sqrt{\frac{d_K}{T}\sum_{t=1}^{T}\kappa_t^2}\right) \leq d_A$, we have using equation D.3 that

$$\text{Reg}_T = \widetilde{O}(d_A\sqrt{T})$$

as well. So, we have that when $\tau\Delta_{\text{off}}\sqrt{T} \leq d_A$,

$$\text{Reg}_T = \widetilde{O}\left(\min\left(d_A\sqrt{T}, Rd_K\sqrt{T}\left(1 + \frac{1}{\lambda_A^3\lambda_\theta}\left(\sqrt{\frac{d_A T}{Nd_K}} + \sqrt{\frac{d_A}{\min(S,T)N}\sum_{t=1}^{\min(S,T)}\kappa_t^2}\right)\right)\right)\right)$$

Bounding regret when $\tau\Delta_{\text{off}}\left(\sqrt{T} + \sqrt{\frac{d_K}{T}\sum_{t=1}^{T}\kappa_t^2}\right) \geq d_A$: In this regime, after the first timestep S satisfying $\Delta_{\text{off}}\left(\sqrt{S} + \sqrt{\frac{d_K}{S}\sum_{t=1}^{S}\kappa_t^2}\right) \geq d_A$, we run standard LinUCB with dimension d_A and incur $\widetilde{O}(d_A\sqrt{T})$ regret with probability $1-\delta$. Until timestep $S-1$, we run the projected and modified version of LinUCB and incur regret bounded by

$$\begin{aligned}\widetilde{O}(d_K\sqrt{S}) + 2R\Delta_{\text{off}}S + 2R\Delta_{\text{off}}\sqrt{\frac{\sum_{t=1}^{S}\kappa_t^2}{S}}\sqrt{d_K S} &= \widetilde{O}\left(d_A\sqrt{S} + d_K\sqrt{S}\right) \\ &= \widetilde{O}(d_A\sqrt{T} + d_K\sqrt{T}) \\ &= \widetilde{O}(d_A\sqrt{T})\end{aligned}$$

Combining these, we incur $O(d_A\sqrt{T})$ regret during the whole method.

Also, since $d_A\sqrt{T} \leq \Delta_{\text{off}}\left(\sqrt{TS} + \sqrt{\frac{d_K T}{S}\sum_{t=1}^{S}\kappa_t^2}\right)$, we get that our regret is also bounded by

$$\begin{aligned}\text{Reg}_T &= \widetilde{O}\left(\Delta_{\text{off}}\left(\sqrt{TS} + \sqrt{\frac{d_K T}{S}\sum_{t=1}^{S}\kappa_t^2}\right)\right) \\ &= \widetilde{O}(d_K\sqrt{T} + \Delta_{\text{off}}\left(\sqrt{TS} + \sqrt{\frac{d_K T}{S}\sum_{t=1}^{S}\kappa_t^2}\right))\end{aligned}$$

$$= \widetilde{O}\left(Rd_K\sqrt{T}\left(1+\frac{1}{\lambda_A^3\lambda_\theta}\left(\sqrt{\frac{d_AT}{Nd_K}}+\sqrt{\frac{d_A}{\min(S,T)N}\sum_{t=1}^{\min(S,T)}\kappa_t^2}\right)\right)\right)$$

So, our regret satisfies

$$\mathrm{Reg}_T = \widetilde{O}\left(\min\left(d_A\sqrt{T}, Rd_K\sqrt{T}\left(1+\frac{1}{\lambda_A^3\lambda_\theta}\left(\sqrt{\frac{d_AT}{Nd_K}}+\sqrt{\frac{d_A}{\min(S,T)N}\sum_{t=1}^{\min(S,T)}\kappa_t^2}\right)\right)\right)\right)$$

D.5.2.1 Understanding κ_t

Letting $\mathbf{X}_t = \hat{\mathbf{U}}^\top[\phi(x_1,a_1),\dots\phi(x_t,a_t)]^\top$, recall that $\kappa_t := \|\hat{\mathbf{U}}\mathbf{X}_t^\top\mathbf{X}_t\|_{(\mu\mathbf{I}+\mathbf{X}_t^\top\mathbf{X}_t)^{-1}}$. In the worst case, $\kappa_t \le \|\hat{\mathbf{U}}\mathbf{X}_t^\top\mathbf{X}_t\|_2 = O(t)$ since actions have norm 1. In that case, $\frac{1}{S}\sum_{t=1}^S \kappa_t = O(S^2)$. However, if $\mathbf{X}_t$ lies in the span of $\hat{\mathbf{U}}^\top$, then $\mathbf{X}_t\hat{\mathbf{U}}^\top\hat{\mathbf{U}} = \mathbf{X}_t$. This means that for $\mathbf{Y}_t := \mathbf{X}_t\hat{\mathbf{U}}^\top$,

$$\begin{aligned}
\kappa_t &= \sqrt{\hat{\mathbf{U}}\mathbf{X}_t^\top\mathbf{X}_t\hat{\mathbf{U}}^\top\hat{\mathbf{U}}(\mu\mathbf{I}+\mathbf{X}_t^\top\mathbf{X}_t)^{-1}\mathbf{U}^\top\hat{\mathbf{U}}\mathbf{X}_t^\top\mathbf{X}_t\hat{\mathbf{U}}^\top} \\
&= \sqrt{\mathbf{Y}_t^\top\mathbf{Y}_t(\mu I+\mathbf{Y}_t^\top\mathbf{Y}_t)^{-1}\mathbf{Y}_t^\top\mathbf{Y}_t} \\
&= \sqrt{(\mathbf{I}-\mu(\mu\mathbf{I}+\mathbf{Y}_t^\top\mathbf{Y}_t))\mathbf{Y}_t^\top\mathbf{Y}_t} \\
&= \widetilde{O}(\sqrt{t})
\end{aligned}$$

So, $\frac{1}{S}\sum_{t=1}^S \kappa_t = O(S) = O(T)$.

□

D.6 Lower Bounds

We formally state and prove the lower bound below. Much like how we generate families of reward parameters in lower bound proofs for purely online regret, we are now generating a family of tuples of latent bandits (for the offline data) and reward parameters represented in the latent bandit (for the online interaction).

Theorem D.6.1 (Regret Lower Bound)**.** *Let $d_A^2H \le 2T$, $d_A^2 \le N$, $d_K > 1$. Consider the action set $\mathcal{A} = \{a \mid \|a\|_2 \le 1\}$. For each regime, either $\frac{d_KT}{(d_A-d_K)N}$ being larger than 1, or between 1 and $1/2$, or less than $1/2$, there exists a family of tuples $(F, \boldsymbol{\beta})$, where F is a latent bandit with a rank d_K latent subspace and $\boldsymbol{\beta}$ is a reward parameter in its support, satisfying the following:*

(i) For any offline behavior policy π_b, all F have uniformly bounded λ_θ associated to the offline data.

(ii) For any offline behavior policy π_b and any learner, there is a $(F, \boldsymbol{\beta})$ such that the regret $\mathrm{Reg}(T, \boldsymbol{\beta})$ of the learner under offline data from π_b and F and online reward parameter $\boldsymbol{\beta}$ is bounded below by

$$\mathrm{Reg}(T, \boldsymbol{\beta}) \geq \Omega\left(\min\left(d_A\sqrt{T}, d_K\sqrt{T}\left(1 + \sqrt{\frac{d_A T}{d_K N}}\right)\right)\right)$$

Remark 22.

- We are stating a version with no contexts and only actions here for notational simplicity, the version for contextual bandits has the same proof verbatim. One just replaces $\mathcal{A}$ with $\{\phi(x, a) \mid x \in \mathcal{X}, a \in \mathcal{A}\}$.
- Note that the theorem statement is complicated to ensure that it is essentially the strongest version of the theorem possible. Condition (i) is needed to ensure that we aren't cheating by ensuring that the offline data itself obscures the correct subspace. Condition (ii) is the actual regret lower bound.

Proof. The proof is inspired by the proof of Theorem 24.2 in Lattimore and Szepesvári [2018], giving a regret lower bound for standard stochastic linear bandits with a unit ball action set. Without loss of generality, we can assume that $d_A \geq 1.01 d_K$, otherwise both terms in the minimum have the same order and the proof is complete. An astute reader will note that the regimes are separated based on whether $d_K\sqrt{T}\left(1 + \sqrt{\frac{(d_A - d_K)T}{d_K N}}\right) \leq d_A\sqrt{T}$. We will first address the difficult regime where $\frac{1}{2} \leq \sqrt{\frac{d_K T}{(d_A - d_K)N}} \leq 1$. Until stated otherwise in this proof, we will work in this regime and assume that $\frac{1}{2} \leq \sqrt{\frac{d_K T}{(d_A - d_K)N}} \leq 1$.

D.6.1 Setup

Consider $\Delta_{\text{in}} = \frac{1}{5\sqrt{3}}\sqrt{d_K/T}$ and $\Delta_{\text{out}} = \frac{1}{4\sqrt{3}}\sqrt{d_K/N}$. Let $\mathcal{B} := \{\pm\Delta_{\text{in}}\}^{d_K - 1} \times \{0\} \times \{\pm\Delta_{\text{out}}\}^{d_A - d_K}$ and let $\boldsymbol{\beta} \in \mathcal{B}$. We set the d_K^{th} coordinate to 0 for technical reasons. For any bandit instance $\boldsymbol{\beta}$, let the rewards have Gaussian noise with variance 1. Construct a family of latent bandit-online latent state pairs as follows. Define $F_{\boldsymbol{\beta}}$ to be a latent bandit with a uniform distribution over all $2^{d_K - 1}$ reward parameters obtained by negating any of the first $d_K - 1$ coordinates of $\boldsymbol{\beta}$. Notice that this latent bandit has $2^{d_K - 1}$ latent states sharing a d_K-dimensional subspace. Let us construct the family of pairs $(F_{\boldsymbol{\beta}}, \boldsymbol{\beta})$, where $F_{\boldsymbol{\beta}}$ is the latent bandit used to generate offline data and $\boldsymbol{\beta}$ is the reward parameter underlying the online trajectory.

Note that condition (i) is satisfied by merely computing the matrix $\mathbb{E}_{\boldsymbol{\beta}}[\boldsymbol{\beta}\boldsymbol{\beta}^\top]$ and noticing that eigenvalues are merely norms $\frac{d_K}{T\|\boldsymbol{\beta}\|_2}$ or $\frac{d_K}{N\|\boldsymbol{\beta}\|_2}$, up to a constant. Now note that $\|\boldsymbol{\beta}\|_2 =$

$\sqrt{\frac{d_K^2}{T} + \frac{d_K(d_A - d_K)}{N}}$ up to a constant, so λ_θ is bounded since $\frac{1}{2} \leq \frac{d_K T}{(d_A - d_K)N} \leq 1$

Notice that if $\boldsymbol{\beta}'$ is $\boldsymbol{\beta}$ with any of the first $d_K - 1$ coordinates negated, then $F_{\boldsymbol{\beta}'}$ is the same latent bandit as $F_{\boldsymbol{\beta}}$. Assume that a fixed behavior policy π_b is used to produce offline data, producing a known dataset of contexts and actions shared across all latent bandit instances. We will repeatedly use the fact that $d_K - 1 \geq d_K/2$ in this proof.

D.6.2 Proof Sketch and Intuition

Notice that we are working in the regime where N is significantly larger than T. That means that we are treating the first d_K coordinates as the main subspace and the rest of the coordinates as perturbations out of the subspace. This is represented in the notation Δ_{in} and Δ_{out}. In our regime, where $\sqrt{\frac{d_K T}{(d_A - d_K)N}} \leq 1$, Δ_{out} should be thought of as much smaller than Δ_{in}.

Eventually, we intend to lower bound the average regret over all pairs $(F_{\boldsymbol{\beta}}, \boldsymbol{\beta})$ ranging over the vertices of a hypercuboid $\mathcal{B}$ of reward parameters. This will allow us to claim that there exists one parameter $\boldsymbol{\beta} \in \mathcal{B}$ for which the regret is larger than this average. To lower bound this average, we first use change of measure inequalities, careful computation and clever design of $\mathcal{B}$ to get an intermediate lower bound, bounding the average regret over any pair of "adjacent" tuples $(F', \boldsymbol{\beta}')$ and $(F, \boldsymbol{\beta})$. These are pairs where the sign of only one coordinate is flipped from $\boldsymbol{\beta}$ to $\boldsymbol{\beta}'$ and F and F' at most differ in their "out of subspace" perturbation. We can then average over all such pairs to lower bound the regret averaged over all $\boldsymbol{\beta} \in \mathcal{B}$, as desired.

We will need two separate computations for the intermediate lower bound – one for when the pair of adjacent reward parameters corresponds to a coordinate $i < d_K$, and another for when it corresponds to a coordinate $i > d_K$. The proof ends with the averaging trick mentioned in the previous paragraph.

D.6.3 Regret Lower Bound Decomposition

For $i < d_K$, define $\tau_i = T \wedge \min\{t : \sum_{s=1}^t A_{si}^2 \geq T/d_K\}$. For $i \geq d_K$, define $\tau_i = T \wedge \min\{t : \sum_{s=1}^t A_{si}^2 \geq T/(d_A - d_K)\}$. For now, let us denote the regret under parameter $\boldsymbol{\beta}$ to be $\text{Reg}(T, \boldsymbol{\beta})$. Now note the following decomposition of the lower bound.

$$\text{Reg}(T, \boldsymbol{\beta}) = \mathbb{E}_{\boldsymbol{\beta}}\Big[\Delta_{\text{in}} \sum_{t=1}^{T} \sum_{i=1}^{d_K - 1} \left(\frac{1}{\sqrt{d_K - 1}} - A_{ti} \operatorname{sign}(\beta_i) \right)$$

$$
\begin{aligned}
&\quad + \Delta_{\text{out}} \sum_{t=1}^{T} \sum_{i=d_K+1}^{d_A} \left(\frac{1}{\sqrt{d_A - d_K}} - A_{ti} \operatorname{sign}(\beta_i) \right) \Big] \\
&\geq \mathbb{E}_{\boldsymbol{\beta}} \Big[\frac{\Delta_{\text{in}} \sqrt{d_K - 1}}{2} \sum_{t=1}^{T} \sum_{i=1}^{d_K - 1} \left(\frac{1}{\sqrt{d_K - 1}} - A_{ti} \operatorname{sign}(\beta_i) \right)^2 \\
&\quad + \frac{\Delta_{\text{out}} \sqrt{d_A - d_K}}{2} \sum_{t=1}^{T} \sum_{i=d_K+1}^{d_A} \left(\frac{1}{\sqrt{d_A - d_K}} - A_{ti} \operatorname{sign}(\beta_i) \right)^2 \Big] \\
&\geq \frac{\Delta_{\text{in}} \sqrt{d_K - 1}}{2} \sum_{i=1}^{d_K - 1} \mathbb{E}_{\boldsymbol{\beta}} \left[\sum_{t=1}^{\tau_i} \left(\frac{1}{\sqrt{d_K - 1}} - A_{ti} \operatorname{sign}(\beta_i) \right)^2 \right] \\
&\quad + \frac{\Delta_{\text{out}} \sqrt{d_A - d_K}}{2} \sum_{i=d_K+1}^{d_A} \mathbb{E}_{\boldsymbol{\beta}} \left[\sum_{t=1}^{\tau_i} \left(\frac{1}{\sqrt{d_A - d_K}} - A_{ti} \operatorname{sign}(\beta_i) \right)^2 \right]
\end{aligned}
$$

The first inequality holds by merely evaluating the square, simplifying and noting that $\|A_t\|_2^2 \leq 1$. The second inequality For $i < d_K$ and $x \in \{\pm 1\}$, define $U_i(x) := \sum_{t=1}^{\tau_i} \left(\frac{1}{\sqrt{d_K - 1}} - A_{ti} \operatorname{sign}(\beta_i) \right)^2$. For $i > d_K$ and $x \in \{\pm 1\}$, define $U_i(x) := \sum_{t=1}^{\tau_i} \left(\frac{1}{\sqrt{d_A - d_K}} - A_{ti} \operatorname{sign}(\beta_i) \right)^2$.

Fix i and let $\boldsymbol{\beta}'$ be such that $\beta'_j = \beta_j$ for $j \neq i$ and $\beta'_i = -\beta_i$. Let $\mathbb{P}$ and $\mathbb{P}'$ be the joint laws of the offline data and the bandit/learner interaction measure for $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ respectively. We will bound $\mathbb{E}_{\boldsymbol{\beta}}[U_i(1)] + \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)]$ in the following subsections, treating $i < d_K$ and $i > d_K$ separately. This will allow us to bound $\sum_{\boldsymbol{\beta} \in \mathcal{B}} \mathbb{E}_{\boldsymbol{\beta}}[U_i(\operatorname{sign}(\beta_i))]$ later and apply an averaging trick.

D.6.4 Bounding $\mathbb{E}_{\beta}[U_i(1)] + \mathbb{E}_{\beta'}[U_i(-1)]$ when $i < d_K$

Note that

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\beta}}[U_i(1)] &\overset{(i)}{\geq} \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] - \left(\frac{6T}{d_K} + 2 \right) \sqrt{\frac{1}{2} D(\mathbb{P}, \mathbb{P}')} \\
&\overset{(ii)}{\geq} \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] - \Delta_{\text{in}} \left(\frac{3T}{d_K} + 1 \right) \sqrt{\sum_{t=1}^{\tau_i} A_{ti}^2} \\
&\overset{(iii)}{\geq} \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] - \Delta_{\text{in}} \left(\frac{3T}{d_K} + 1 \right) \sqrt{\frac{T}{d_K} + 1} \\
&\overset{(iv)}{\geq} \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] - \frac{5\sqrt{3} \Delta_{\text{in}} T}{d_K} \sqrt{\frac{T}{d_K}}
\end{aligned} \tag{D.4}
$$

where in (i), we rely on the bound below and then use the TV distance change of measure inequality, followed by Pinsker's inequality. The bound below relies on the fact that $d_K - 1 \geq d_K/2$.

$$U_i(1) = \sum_{t=1}^{\tau_i} \left(\frac{1}{\sqrt{d_K - 1}} - A_{ti} \operatorname{sign}(\beta_i) \right)^2 \leq 2 \sum_{t=1}^{\tau_i} \frac{1}{d_K - 1} + 2 \sum_{t=1}^{\tau_i} A_{ti}^2$$
$$\leq \frac{4T}{d_K} + \frac{2T}{d_K} + 2 = \frac{6T}{d_K} + 2$$

For the bound above, we use the definition of τ_i. In (ii), we use the chain rule for KL divergence under a stopping time, and crucially note that the offline data distributions is identical in this case since $F_{\boldsymbol{\beta}} = F_{\boldsymbol{\beta}'}$. Inequality (iii) holds by the definition of τ_i, and inequality (iv) holds since $d_K \leq d_A \leq 2T$ since $d_A^2 H \leq 2T$.

So, we can conclude that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\beta}}[U_i(1)] + \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] &\geq \mathbb{E}_{\boldsymbol{\beta}'}[U_i(1) + U_i(-1)] - \frac{5\sqrt{3}\Delta_{\text{in}} T}{d_K} \sqrt{\frac{T}{d_K}} \\
&= 2\mathbb{E}_{\boldsymbol{\beta}'} \left[\frac{\tau_i}{d_K - 1} + \sum_{t=1}^{\tau_i} A_{ti^2} \right] - \frac{5\sqrt{3}\Delta_{\text{in}} T}{d_K} \sqrt{\frac{T}{d_K}} \\
&\geq \frac{2T}{d_K} - \frac{5\sqrt{3}\Delta_{\text{in}} T}{d_K} \sqrt{\frac{T}{d_K}} \\
&= \frac{T}{d_K}
\end{aligned} \tag{D.5}$$

D.6.5 Bounding $\mathbb{E}_{\beta}[U_i(1)] + \mathbb{E}_{\beta'}[U_i(-1)]$ when $i > d_K$

Note the following computation, where we let $A_{r,h}$ be the action chosen at step h of offline trajectories d, where $h = 1 \to H$ and $r = 1 \to N$.

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\beta}}[U_i(1)] &\overset{(i)}{\geq} \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] - \left(\frac{4T}{d_A - d_K} + 2 \right) \sqrt{\frac{1}{2} D(\mathbb{P}, \mathbb{P}')} \\
&\overset{(ii)}{\geq} \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] - \Delta_{\text{out}} \left(\frac{2T}{d_A - d_K} + 1 \right) \sqrt{\sum_{t=1}^{\tau_i} A_{ti}^2 + \frac{d_K}{48N} \mathbb{E}_{\pi_b}[\sum_{r=1}^{N} \sum_{h=1}^{H} A_{r,h,i}^2]} \\
&\overset{(iii)}{\geq} \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] - \Delta_{\text{out}} \left(\frac{2T}{d_A - d_K} + 1 \right) \sqrt{\frac{T}{d_A - d_K} + \frac{d_K H}{48}} \\
&\overset{(iv)}{\geq} \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] - \frac{5\sqrt{3}\Delta_{\text{out}} T}{d_A - d_K} \sqrt{\frac{T}{d_A - d_K}}
\end{aligned}$$

where again in (i), we rely on the bound below and use the TV distance change of measure inequality, followed by Pinsker's inequality.

$$U_i(1) = \sum_{t=1}^{\tau_i} \left(\frac{1}{\sqrt{d_A - d_K}} - A_{ti} \operatorname{sign}(\beta_i) \right)^2 \leq 2 \sum_{t=1}^{\tau_i} \frac{1}{d_A - d_K} + 2 \sum_{t=1}^{\tau_i} A_{ti}^2 \leq \frac{4T}{d_A - d_K} + 2$$

For the bound above, we use the definition of τ_i. In (ii), we use the chain rule for KL divergence under a stopping time and include the non-zero KL divergence coming from the offline term this time, which appears as the second term in the square root. Inequality (iii) holds by the definition of τ_i and the fact that $A_{ri}^2 \leq 1$. Inequality (iv) holds since $d_k(d_A - d_K)H \leq d_A^2 H \leq 2T$.

So, we can conclude that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\beta}}[U_i(1)] + \mathbb{E}_{\boldsymbol{\beta}'}[U_i(-1)] &\geq \mathbb{E}_{\boldsymbol{\beta}'}[U_i(1) + U_i(-1)] - \frac{4\sqrt{3}\Delta_{\text{out}} T}{d_A - d_K} \sqrt{\frac{T}{d_A - d_K}} \\
&= \mathbb{E}_{\boldsymbol{\beta}'} \left[\frac{\tau_i}{d_A - d_K} + \sum_{t=1}^{\tau_i} A_{ti^2} \right] - \frac{4\sqrt{3}\Delta_{\text{out}} T}{d_A - d_K} \sqrt{\frac{T}{d_A - d_K}} \\
&\geq \frac{2T}{d_A - d_K} - \frac{4\sqrt{3}\Delta_{\text{out}} T}{d_A - d_K} \sqrt{\frac{T}{d_A - d_K}} \\
&= \frac{2T}{d_A - d_K} - \frac{T}{d_A - d_K} \sqrt{\frac{d_K T}{(d_A - d_K)N}} \\
&\geq \frac{T}{d_A - d_K}
\end{aligned} \tag{D.6}$$

where crucially, the last inequality holds since $\sqrt{\frac{d_K T}{(d_A - d_K)N}} \leq 1$.

D.6.6 Lower bounding regret using an averaging trick

For $i \leq d_K$, define by $\mathcal{B}_{-i} := \{\pm\Delta_{\text{in}}\}^{d_K - 2} \times \{0\} \times \{\pm\Delta_{\text{out}}\}^{d_A - d_K}$, which is the slice of $\mathcal{B}$ where all coordinates but β_i vary. Similarly, for $i > d_K$, define the slice $\mathcal{B}_{-i} := \{\pm\Delta_{\text{in}}\}^{d_K - 1} \times \{0\} \times \{\pm\Delta_{\text{out}}\}^{d_A - d_K - 1}$. We will denote the tuple of coordinates of $\boldsymbol{\beta}$ other than i by β_{-i}. We thus get the following lower bound on regret, using inequalities D.4, D.5 and D.6.

$$\begin{aligned}
\sum_{\boldsymbol{\beta} \in \mathcal{B}} \operatorname{Reg}(T, \boldsymbol{\beta}) \geq \frac{\Delta_{\text{in}} \sqrt{d_K - 1}}{2} \sum_{i=1}^{d_K - 1} \sum_{\boldsymbol{\beta} \in \mathcal{B}} \mathbb{E}_{\boldsymbol{\beta}}[U_i(\operatorname{sign}(\beta_i))] \\
+ \frac{\Delta_{\text{out}} \sqrt{d_A - d_K}}{2} \sum_{i=d_K+1}^{d_A} \sum_{\boldsymbol{\beta} \in \mathcal{B}} \mathbb{E}_{\boldsymbol{\beta}}[U_i(\operatorname{sign}(\beta_i))]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\Delta_{\text{in}}\sqrt{d_K - 1}}{2} \sum_{i=1}^{d_K - 1} \sum_{\beta_{-i} \in \mathcal{B}_{-i}} \sum_{\beta_i \in \{\pm\Delta_{\text{in}}\}} \mathbb{E}_{\boldsymbol{\beta}}[U_i(\text{sign}(\beta_i))] \\
&\quad + \frac{\Delta_{\text{out}}\sqrt{d_A - d_K}}{2} \sum_{i=d_K+1}^{d_A} \sum_{\beta_{-i} \in \mathcal{B}_{-i}} \sum_{\beta_i \in \{\pm\Delta_{\text{out}}\}} \mathbb{E}_{\boldsymbol{\beta}}[U_i(\text{sign}(\beta_i))] \\
&\geq \frac{\Delta_{\text{in}}\sqrt{d_K - 1}}{2} \sum_{i=1}^{d_K - 1} \sum_{\beta_{-i} \in \mathcal{B}_{-i}} \frac{T}{d_K} + \frac{\Delta_{\text{out}}\sqrt{d_A - d_K}}{2} \sum_{i=d_K+1}^{d_A} \sum_{\beta_{-i} \in \mathcal{B}_{-i}} \frac{T}{d_A - d_K} \\
&\geq \frac{\Delta_{\text{in}}\sqrt{d_K}}{2\sqrt{2}} \sum_{\beta_{-i} \in \mathcal{B}_{-i}} \frac{T}{2} + \frac{\Delta_{\text{out}}\sqrt{d_A - d_K}}{2} \sum_{\beta_{-i} \in \mathcal{B}_{-i}} T \\
&= \frac{2^{d_A}}{80\sqrt{6}} d_K \sqrt{T} \left(1 + 5\sqrt{2\frac{(d_A - d_K)T}{d_K N}}\right)
\end{aligned}
$$

That means that there exists $\boldsymbol{\beta} \in \mathcal{B}$ so that

$$
\text{Reg}(T, \boldsymbol{\beta}) \geq \frac{1}{80\sqrt{6}} d_K \sqrt{T} \left(1 + \sqrt{\frac{(d_A - d_K)T}{d_K N}}\right)
$$

As desired.

D.6.7 The Other Two Regimes

In the regime $d_K T \geq (d_A - d_K)N$, one can simply use the 2^{d_A} bandit instances in the standard unit ball regret lower bound from Theorem 24.2 in Lattimore and Szepesvári [2018] with dimension d_A, and follow the proof essentially verbatim. The only difference is that we will be choosing pairs of tuples $(F', \boldsymbol{\beta}')$ and $(F, \boldsymbol{\beta})$ instead of just pairs of reward parameters $\boldsymbol{\beta}'$ and $\boldsymbol{\beta}$. One can choose any latent bandit with d_K reward parameters in its support, two of which are $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$, and set both F and F' to this. For this, it is convenient to choose the latent bandit to have a uniform distribution over 2_K^d reward parameters obtained by flipping signs of d_K chosen coordinates, since then one can easily compute that $\lambda_\theta = 1$. This will ensure that offline data distributions are identical and the KL divergence contribution from the offline data distribution is 0, allowing us to follow the proof of Theorem 24.2 in Lattimore and Szepesvári [2018] essentially verbatim. This establishes condition (ii), and we have also establis

Similarly, when $d_K T \ll (d_A - d_K)N$, we can use the standard lower bound from Theorem 24.2 in Lattimore and Szepesvári [2018] again, this time with dimension d_K. Fix F to be the latent bandit with a uniform distribution over all 2^{d_K} reward parameters $\mathcal{B} = \{\pm\Delta_{\text{in}}\}^{d_K} \times \{0\}^{d_A - d_K}$, and

consider the family $(F, \boldsymbol{\beta})$ of tuples with fixed F and $\boldsymbol{\beta}$ varying through $\mathcal{B}$. We can now follow the proof of Theorem 24.2 in Lattimore and Szepesvári [2018] verbatim. Again, the only difference is that we will be choosing pairs of tuples $(F, \boldsymbol{\beta}')$ and $(F, \boldsymbol{\beta})$ instead of just pairs of reward parameters $\boldsymbol{\beta}'$ and $\boldsymbol{\beta}$. And yet again, we can check that $\lambda_\theta = 1$ and condition (i) is thus satisfied. □

D.7 Additional Algorithms

We provide a version of SOLD that utilizes pseudoinverses. We use this within our experiments to avoid having to search for regularization parameters, and recommend that the user use this instead of Algorithm 11 when finding a suitable regularization parameter is a concern.

Algorithm 28 Subspace estimation from Offline Latent bandit Data (SOLD) – Pseudoinverse Version

Input: Dataset $\mathcal{D}_{\text{off}}$ of collected trajectories $\tau_n = ((x_{n,1}, a_{n,1}, r_{n,1}), ..., (x_{n,H}, a_{n,H}, r_{n,1}))$ under a behavior policy π_b, dimension of latent subspace d_K.
Divide each τ_n into odd and even steps, giving trajectory halves $\tau_{n,1}$ and $\tau_{n,2}$.
Estimate reward parameters $\hat{\boldsymbol{\beta}}_{n,i} \leftarrow \mathbf{V}_{n,i}^{\dagger} \boldsymbol{b}_{n,i}$, where $\mathbf{V}_{n,i} \leftarrow \sum_{(x,a,r)\in\tau_{n,i}} \phi(x,a)\phi(x,a)^\top$ and $\boldsymbol{b}_{n,i} \leftarrow \sum_{(x,a,r)\in\tau_{n,i}} \phi(x,a) r$ for $i = 1, 2$.
Compute $\mathbf{M}_n \leftarrow \frac{1}{2}(\hat{\boldsymbol{\beta}}_{n,1}\hat{\boldsymbol{\beta}}_{n,2}^\top + \hat{\boldsymbol{\beta}}_{n,2}\hat{\boldsymbol{\beta}}_{n,1}^\top)$ and compute $\overline{\mathbf{M}}_N \leftarrow \frac{1}{N}\sum_{n=1}^N \mathbf{M}_n$.
Compute $\mathbf{W}_{n,i}$, the eigenvectors of $\mathbf{V}_{n,i}$ corresponding to nonzero eigenvalues.
Compute $\overline{\mathbf{D}}_{N,i} \leftarrow \frac{1}{N}\sum_{n=1}^N (\mathbf{W}_{n,i}\mathbf{W}_{n,i}^\top)^\dagger$, $i = 1, 2$.
Obtain $\hat{\mathbf{U}}$, the top d_K eigenvectors of $\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1}$.
return Projection matrix $\hat{\mathbf{U}}\hat{\mathbf{U}}^\top$, Δ_{off} as in Theorem 6.3.2

We also provide a method of instantiating the ProBALL framework with linear Thompson sampling. Like ProBALL-UCB, ProBALL-TS operates within the estimated subspace until the online uncertainty is low enough. We therefore maintain two normal posterior distributions, one over the latent state parameter in the estimated subspace, and one over the high-dimensional reward parameter, and sample from them as such.

D.8 Experimental Details and Additional Experiments

D.8.1 Determining the Latent Rank from Offline Data

We note that as discussed in 6.3, we can use the eigenvalues of $\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1}$ to determine the rank of our subspace. We use the version of this arising from pseudo-inverses instead of regularization, just like in the MovieLens experiments. We demonstrate that we can indeed determine that the $d_K = 18$ by finding the significant eigenvalues of the pseudo-inverse version of $\overline{\mathbf{D}}_{N,1}^{-1}\overline{\mathbf{M}}_N\overline{\mathbf{D}}_{N,2}^{-1}$

Algorithm 29 Projection and Bonuses for Accelerating Latent bandit Thompson Sampling (ProBALL-TS)

Input: Projection matrix $\hat{\mathbf{U}}\hat{\mathbf{U}}^\top$, confidence bound Δ_{off}. Hyperparameters $\alpha_{1,t}, \alpha_{2,t}, \tau, \tau'$.
Initialize $\mathbf{V}_1 \leftarrow I$, $\boldsymbol{b}_1 \leftarrow 0$, $\mathbf{C}_t \leftarrow 0$
for $t = 1, \dots T$ **do**
  if $\Delta_{\text{off}}\tau\sqrt{t} + \Delta_{\text{off}}\tau'\sqrt{d_K \sum_{s=1}^{t} \kappa_s^2 / t} \leq d_A$ **then**
    Compute $\bar{\boldsymbol{\theta}}_{1,t} \leftarrow (\hat{\mathbf{U}}^\top \mathbf{V}_t \hat{\mathbf{U}})^{-1} \hat{\mathbf{U}}^\top \boldsymbol{b}_t$
    Sample $\hat{\boldsymbol{\theta}}_{1,t} \sim \mathcal{N}\left(\bar{\boldsymbol{\beta}}_{1,t}, \alpha_{1,t}^2 (\hat{\mathbf{U}}^\top \mathbf{V}_t \hat{\mathbf{U}})^{-1}\right)$
    Play $a_t \leftarrow \arg\max_a \phi(x_t, a)^\top \hat{\mathbf{U}} \hat{\boldsymbol{\theta}}_{1,t}$
  else
    Compute $\bar{\boldsymbol{\beta}}_{2,t} \leftarrow \mathbf{V}_t^{-1} \boldsymbol{b}_t$
    Sample $\hat{\boldsymbol{\beta}}_{2,t} \sim \mathcal{N}\left(\bar{\boldsymbol{\beta}}_{2,t}, \alpha_{2,t}^2 \mathbf{V}_t^{-1}\right)$
    Play $a_t \leftarrow \arg\max_a \phi(x_t, a)^\top \hat{\boldsymbol{\beta}}_{2,t}$
  end if
  Observe reward r_t and update $\boldsymbol{b}_{t+1} \leftarrow \boldsymbol{b}_t + \phi(x_t, a) r_t$, $\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t + \phi(x_t, a)\phi(x_t, a)^\top$
  Update $\mathbf{C}_{t+1} \leftarrow \mathbf{C}_t + \hat{\mathbf{U}}^\top \phi(x_t, a_t)\phi(x_t, a_t)^\top$, $\kappa_{t+1} \leftarrow \|\mathbf{C}_{t+1}\|_{(\hat{\mathbf{U}}^\top \mathbf{V}_{t+1} \hat{\mathbf{U}})^{-1}}$
end for

estimated from the offline dataset of 5000 samples. We show the plots and log plots of these eigenvalues. We also plot the eigenvalues of the completed ratings matrix for comparison. Notice that they match and both fall after 18 eigenvalues.

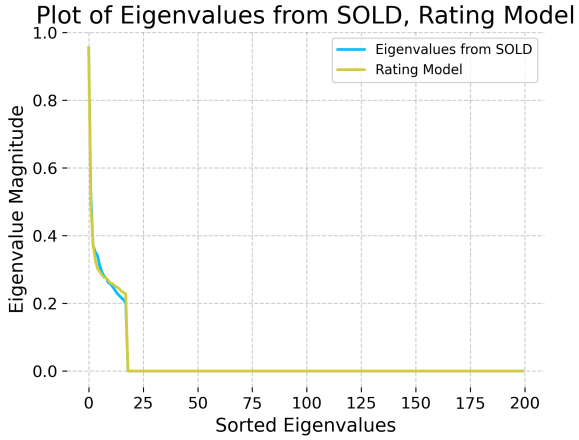

Figure D.1: Plot of eigenvalues of aforementioned matrix. Notice the drop after 18 eigenvalues.

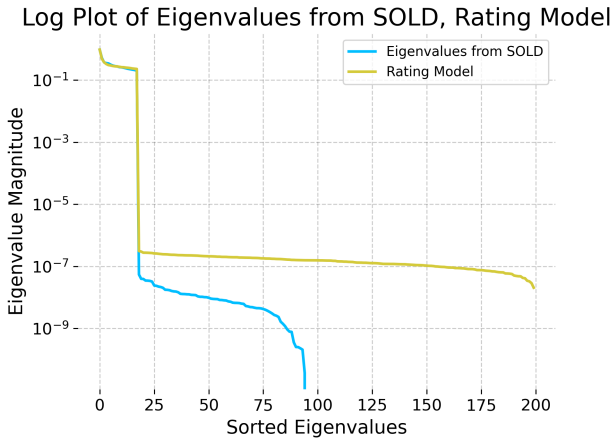

Figure D.2: Log-plot of eigenvalues of aforementioned matrix. Notice the drop after 18 eigenvalues.

D.8.2 Simulation Study

We generate $\mathbf{U}_*$ with $\mathbf{U}_{ij}$ i.i.d. Unif$(0, \frac{2.5}{d_K d_A})$. We simulate the hidden labels $\boldsymbol{\theta}_n \sim \mathcal{N}(0, d_K^{-1}\mathbf{I}_{d_K})$, generate feature vectors $\phi(x_{n,h}, a_{n,h}) \sim \mathcal{N}(0, \mathbf{I}_{d_A})$ normalized to unit norm, and sample noise $\epsilon_{n,h}$ i.i.d. $\mathcal{N}(0, 0.5^2)$. We use SOLD to estimate $\hat{\mathbf{U}}$ from the offline dataset $\mathcal{D}_{\text{off}}$, which consists of 5000 trajectories of length 20 each. In accordance with the confidence set determined by Li et al. [2010], we choose $\alpha_{1,t} = 0.33\sqrt{d_K \log(1 + 10T/d_K)}$ and $\alpha_{2,t} = 0.33\sqrt{d_A \log(1 + 10T/d_A)}$, and share the LinUCB and ProBALL-UCB hyperparameters by assigning $\alpha_t = \alpha_{2,t}$. [2]

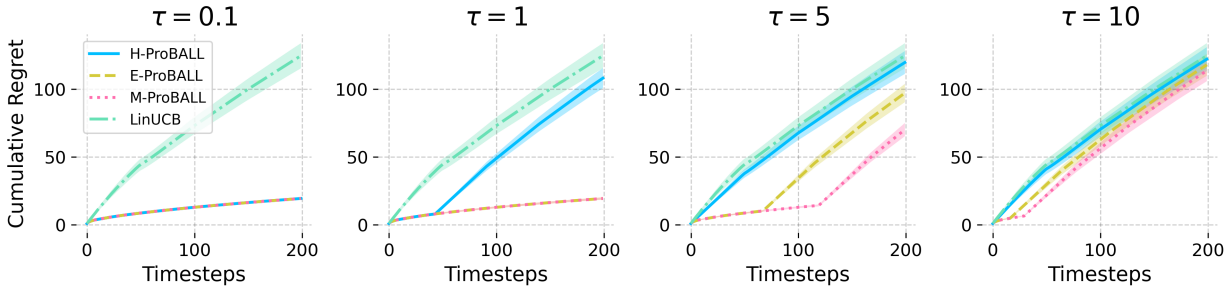

Figure D.3: Comparison of ProBALL-UCB with LinUCB, for different choices of τ and confidence bound constructions. All variants perform no worse than LinUCB, with martingale Bernstein performing the best. The shaded area depicts 1-standard error confidence intervals over 30 trials.

[2] All experiments were run on a single computer with an Intel i9-13900k CPU, 128GB of RAM, and a NVIDIA RTX 3090 GPU, in no more than an hour in total.

D.8.3 MovieLens

MovieLens [Harper and Konstan, 2015] is a large-scale movie recommendation dataset comprising 6040 users and 3883 movies, where each user may rate one or more movies. Like Hong et al. [2020], we filter the dataset to include only movies rated by at least 200 users and vice-versa. We factor the sparse rating matrix into user parameters $\boldsymbol{\beta}$ and movie features Φ using the probabilistic matrix factorization algorithm of Mnih and Salakhutdinov [2007a], using nuclear norm regularization so that the rank of $\boldsymbol{\beta}$ is $d_K = 18$. However, we consider a much higher dimensional problem than Hong et al. [2020] do – we let $d_A = 200$ so $\boldsymbol{\beta} \in \mathbb{R}^{1589 \times 200}$, $\Phi \in \mathbb{R}^{200 \times 1426}$. At each round for user i, the agent chooses between 20 movies of different genres with features $\Phi_{a_1}, ..., \Phi_{a_{20}}$, and has to recommend the best movie presented to it to maximize the user's rating of the movie. We generate rewards for recommending movie j to user i by $\boldsymbol{\beta}_i^T \Phi_j + \epsilon_{ij}, \epsilon_{ij}$ i.i.d. $\mathcal{N}(0, 0.5)$.

Our hyperparameters are chosen and varied just as in the simulation study. To reproduce the methods of Hong et al. [2020], we cluster the user features into d_K clusters using k-means, and provide mUCB and mmUCB with the mean vectors of each cluster as latent models. We initialize ProBALL-UCB with a subspace estimated with an unregularized variant of SOLD, that uses pseudo-inverses instead of inverses, because of difficulties in finding an appropriate regularization parameter for this large, noisy, and high-dimensional dataset. The subspace was estimated from 5000 trajectories of length 50 simulated from the reward model and the uniform behavior policy. Note that we assign Δ_{off} for ϵ in mmUCB, as this is their tolerance parameter for model misspecification.

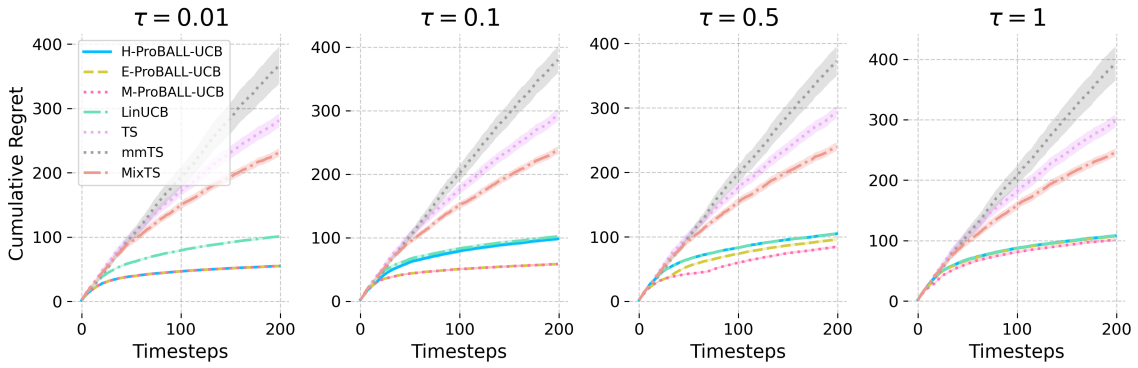

Figure D.4: Comparison of ProBALL-UCB with LinUCB and TS algorithms, for different choices of τ and confidence bound constructions. All variants perform no worse than LinUCB and outperform the TS algorithms, with martingale Bernstein performing the best. The shaded area depicts 1-standard error confidence intervals over 30 trials.

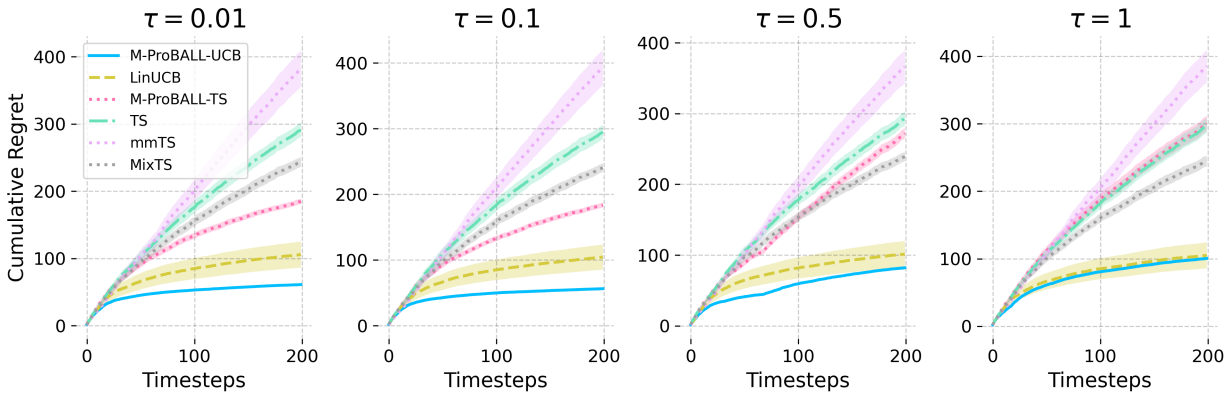

Figure D.5: Comparison of ProBALL-UCB and ProBALL-TS initialized with SOLD against LinUCB, TS, MixTS, and mmTS, for different choices of τ and confidence bound constructions. ProBALL-UCB outperforms LinUCB, and ProBALL-TS outperforms MixTS and mmTS. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh $\boldsymbol{\theta}$. The confidence intervals on regret thus account for the variation in frequentist regret for changing $\boldsymbol{\theta}$.

D.8.3.1 UCB Algorithms

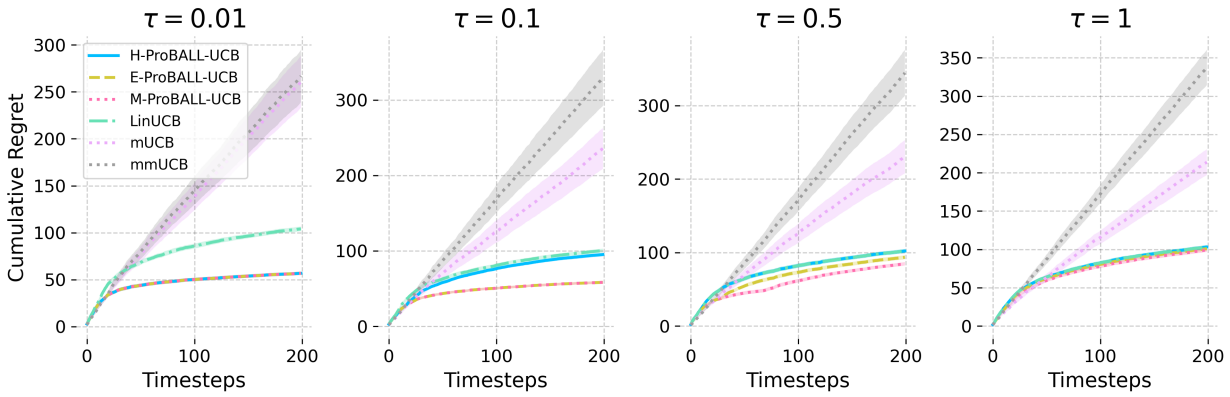

Figure D.6: Comparison of ProBALL-UCB initialized with SOLD against LinUCB, mUCB, and mmUCB, for different choices of τ and confidence bound constructions, in terms of regret. All variants of ProBALL-UCB perform no worse than LinUCB, and outperform mUCB and mmUCB. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ.

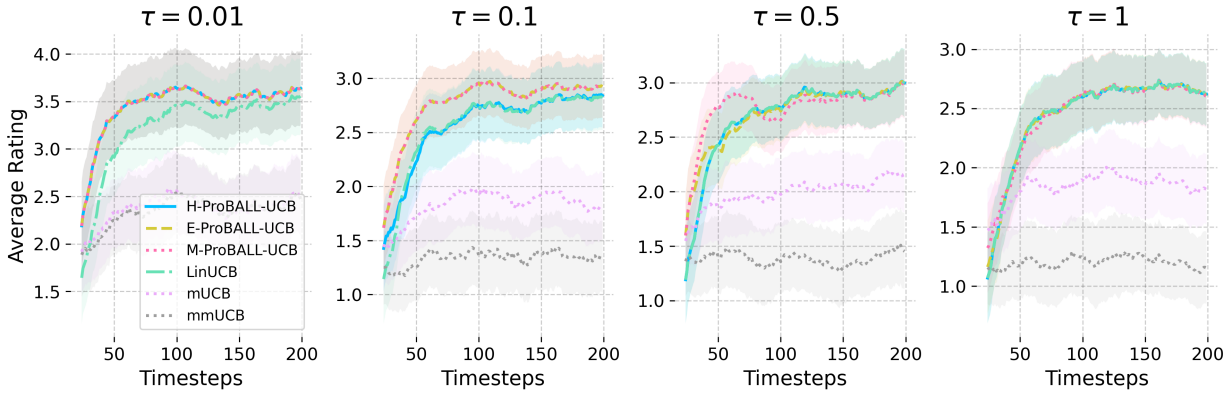

Figure D.7: Comparison of ProBALL-UCB initialized with SOLD against LinUCB, mUCB, and mmUCB, for different choices of τ and confidence bound constructions, in terms of rolling average rating over 25 timesteps. ProBALL-UCB performs no worse than LinUCB, and outperforms mUCB and mmUCB.

D.8.3.2 TS Algorithms

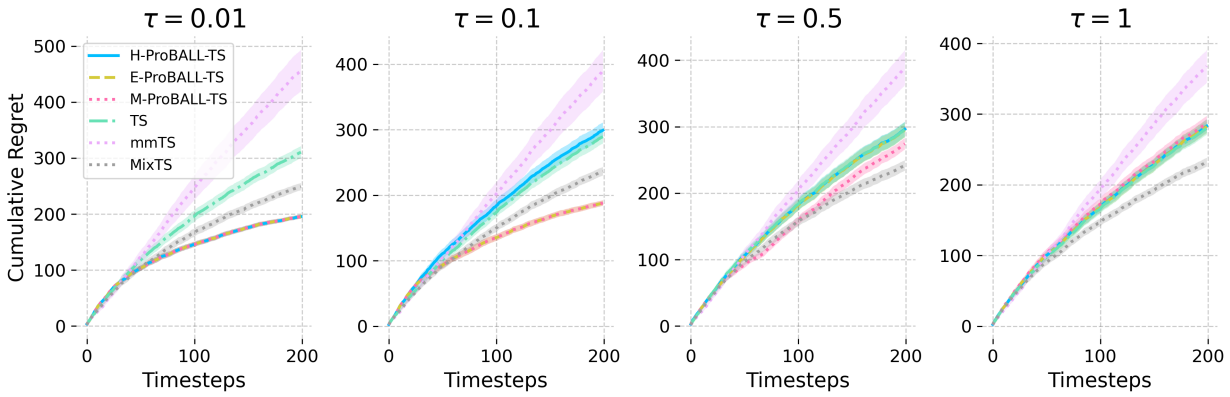

Figure D.8: Comparison of ProBALL-TS initialized with SOLD against TS, mmTS, and MixTS, for different choices of τ and confidence bound constructions, in terms of regret. All variants of ProBALL-TS outperform TS, mmTS, and MixTS.

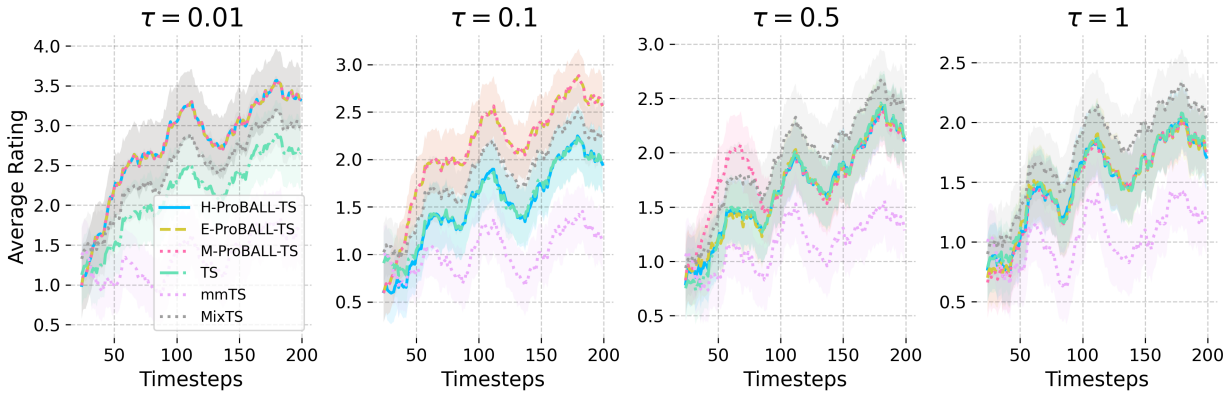

Figure D.9: Comparison of ProBALL-TS initialized with SOLD against TS, mmTS, and MixTS, for different choices of τ and confidence bound constructions, in terms of rolling average rating over 25 timesteps. All variants of ProBALL-TS outperform TS, mmTS, and MixTS. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ.

D.8.3.3 Comparison Against Low-Dimensional Ground Truth Subspaces

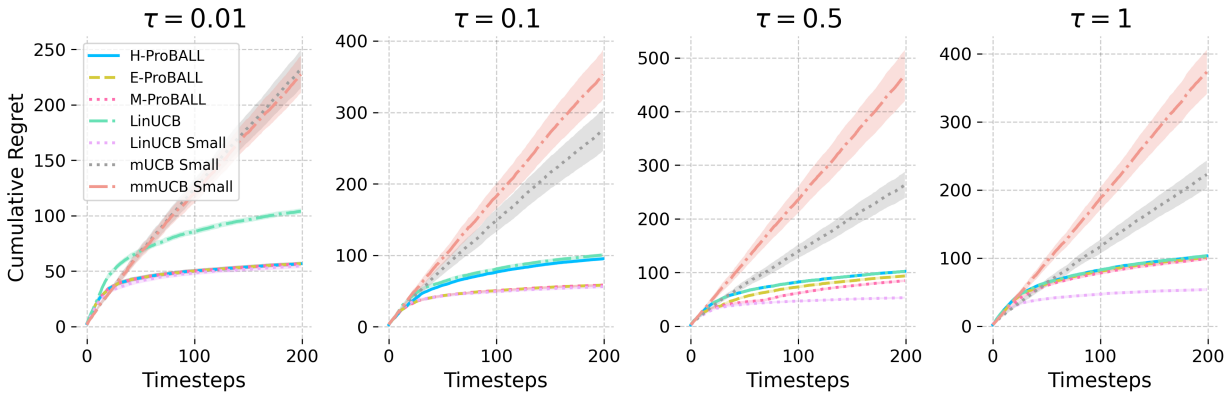

Figure D.10: Comparison of ProBALL-UCB initialized with SOLD against LinUCB, mUCB, and mmUCB on low-dimensional ground-truth features, for different choices of τ and confidence bound constructions. When τ is small enough, all variants of ProBALL-UCB perform no worse than low-dimensional LinUCB, and outperform mUCB and mmUCB, on ground truth features. This showcases the efficacy of SOLD, and demonstrates that we recover subspaces that are just as good as ground-truth. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ.

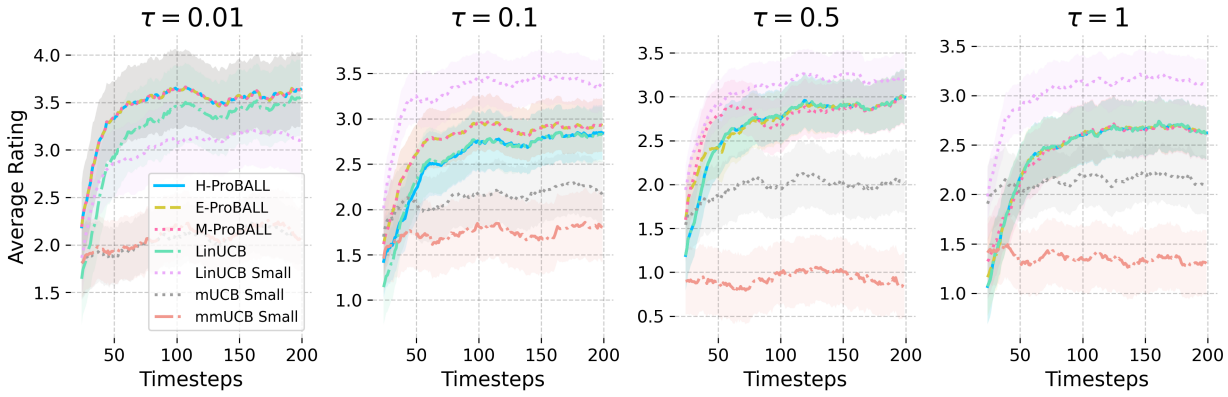

Figure D.11: Comparison of ProBALL-UCB initialized with SOLD against LinUCB, mUCB, and mmUCB on low-dimensional ground-truth features, for different choices of τ and confidence bound constructions. When τ is small enough, all variants of ProBALL-UCB perform no worse than low-dimensional LinUCB, and outperform mUCB and mmUCB, on ground truth features. This showcases the efficacy of SOLD, and demonstrates that we recover subspaces that are just as good as ground-truth. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ.

D.8.3.4 No Usage of SOLD

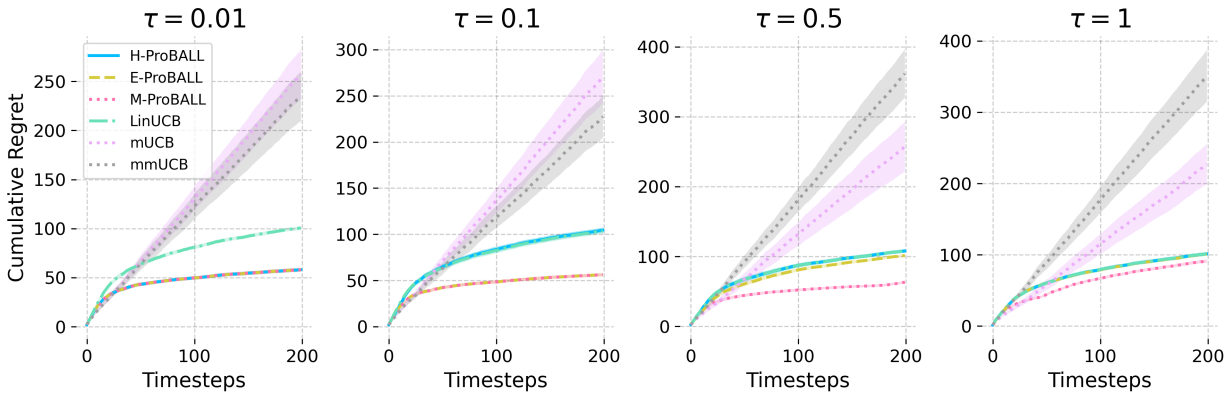

Figure D.12: Comparison of ProBALL-UCB initialized with ground truth subspaces against LinUCB, mUCB, and mmUCB, for different choices of τ and confidence bound constructions. All variants of ProBALL-UCB perform no worse than LinUCB, and outperform mUCB and mmUCB. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ.

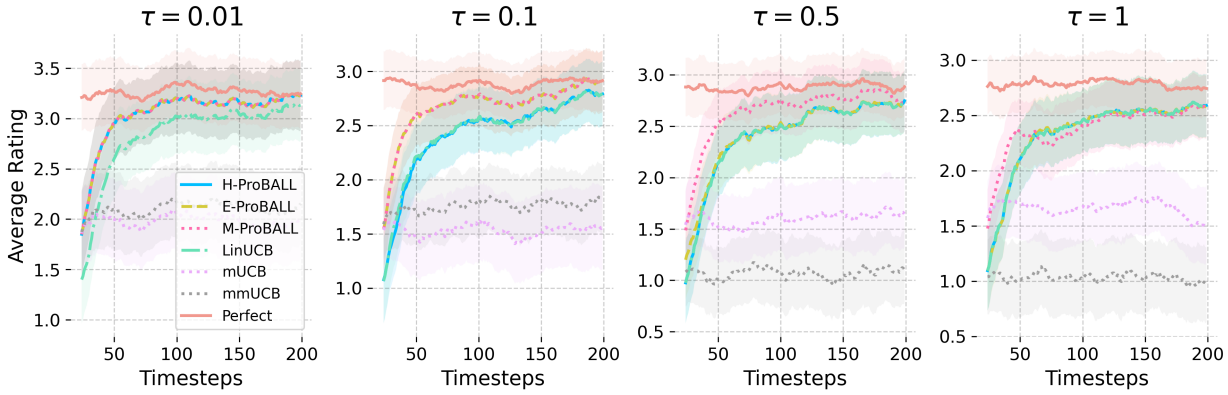

Figure D.13: Comparison of ProBALL-UCB initialized with ground truth subspaces against LinUCB, mUCB, and mmUCB, for different choices of τ and confidence bound constructions, in terms of rolling average rating over 25 timesteps. All variants of ProBALL-UCB perform no worse than LinUCB, and outperform mUCB and mmUCB. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ. The confidence intervals on regret thus account for the variation in frequentist regret for changing θ.

D.8.4 Sample Complexity of SOLD

We perform an empirical study of the sample complexity of SOLD on the MovieLens dataset. To do so, we compare the end-to-end regret at $T = 200$ timesteps of both ProBALL-UCB and ProBALL-TS, against LinUCB and Linear Thompson sampling using ground-truth low-dimensional features. When τ is small enough, we see that the end-to-end regret of both ProBALL-UCB and ProBALL-TS converges to that of LinUCB and Linear Thompson sampling using ground-truth low-dimensional features. This shows that we lose little from needing to estimate the subspace with SOLD when enough offline samples are present.

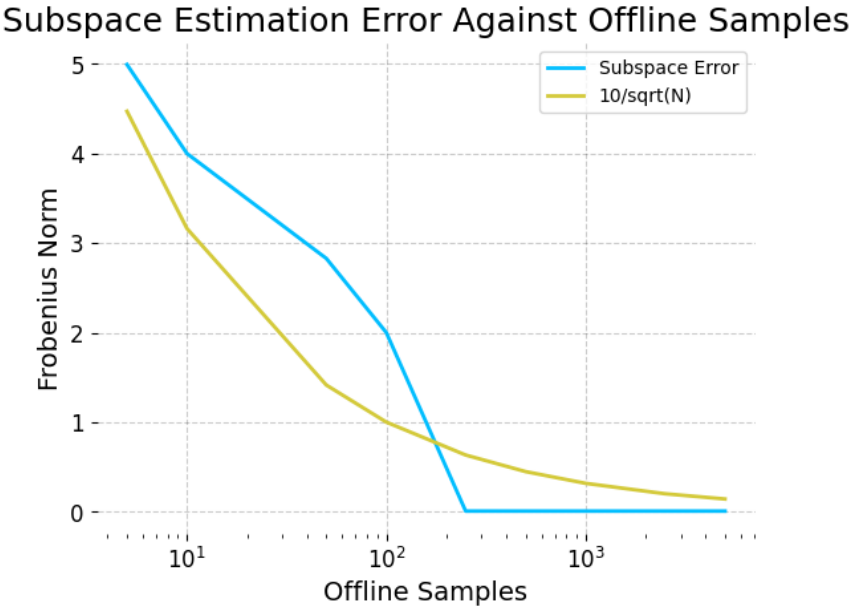

Figure D.14: Subspace estimation error of SOLD against the number of offline samples, in the Frobenius norm. This was performed on the MovieLens dataset. We compare the error of SOLD against the parametric rate of $1/\sqrt{N}$. This shows that the error of SOLD indeed decreases very quickly in practice.

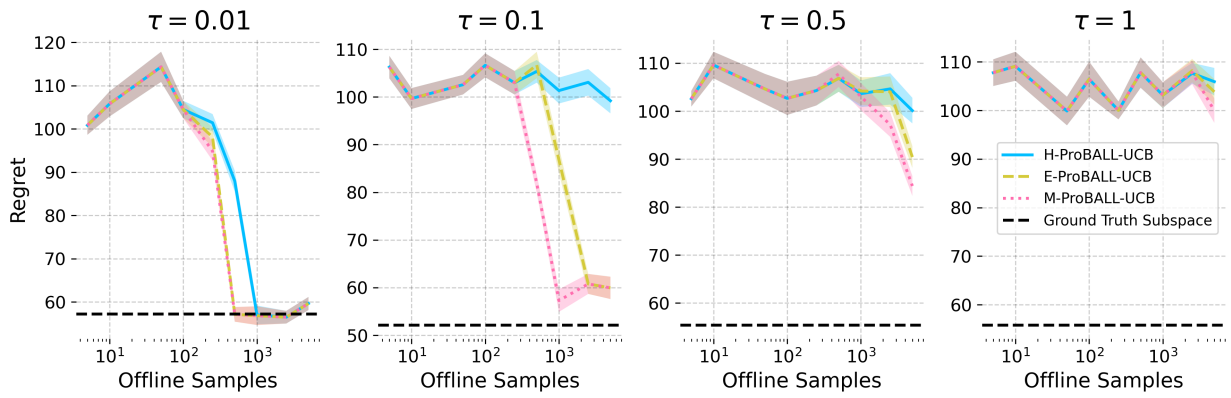

Figure D.15: End-to-end regret at $T = 200$ timesteps of ProBALL-UCB initialized with SOLD, against the number of offline samples used in fitting SOLD. With a low enough τ, the regret of ProBALL-UCB approaches the regret of LinUCB on ground-truth low-dimensional features, showing that we lose next to nothing from needing to estimate the subspace with SOLD. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ.

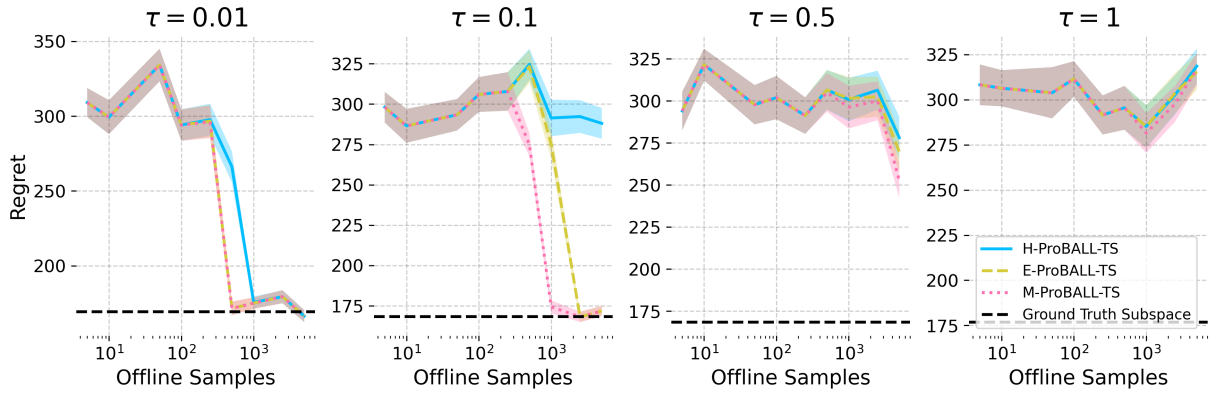

Figure D.16: End-to-end regret at $T = 200$ timesteps of ProBALL-TS initialized with SOLD, against the number of offline samples used in fitting SOLD. With a low enough τ, the regret of ProBALL-TS approaches the regret of TS on ground-truth low-dimensional features, showing that we lose next to nothing from needing to estimate the subspace with SOLD. Shaded area depicts 1-standard error confidence intervals over 30 trials with fresh θ.

BIBLIOGRAPHY

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Riad Akrour, Marc Schoenauer, and Michèle Sebag. April: Active preference learning-based reinforcement learning. In *European Conerence on Machine Learning and Knowledge Discovery in Databases*, 2012.

Paul S. Albert. A two-state markov mixture model for a time series of epileptic seizure counts. *Biometrics*, 47(4):1371–1381, 1991. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2532392.

Anima Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(80):2773–2832, 2014. URL https://jmlr.org/papers/v15/anandkumar14b.html.

Karl Johan Åström. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, 2017.

Andrew Bennett, Nathan Kallus, Lihong Li, and Ali Mousavi. Off-policy evaluation in infinite-horizon reinforcement learning with latent confounders. *CoRR*, abs/2007.13893, 2020. URL https://arxiv.org/abs/2007.13893.

Eric Brochu, Matthew W. Hoffman, and Nando de Freitas. Portfolio allocation for bayesian optimization, 2011.

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, 2019.

David Bruns-Smith and Angela Zhou. Robust fitted-q-evaluation and iteration under sequentially exogenous unobserved confounders, 2023.

David A Bruns-Smith. Model-free and model-based policy evaluation when causality is uncertain. In *International Conference on Machine Learning*, pages 1116–1126. PMLR, 2021.

Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning. *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013*, 09 2013.

Kirsten Bulteel, Francis Tuerlinckx, Annette Brose, and Eva Ceulemans. Clustering vector autoregressive models: Capturing qualitative differences in within-person dynamics. *Frontiers in Psychology*, 7, 2016. ISSN 1664-1078. doi: 10.3389/fpsyg.2016.01540. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2016.01540.

Róbert Busa-Fekete and Eyke Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *Algorithmic Learning Theory*, 2014.

Changxiao Cai, T. Tony Cai, and Hongzhe Li. Transfer learning for contextual multi-armed bandits, 2024. URL https://arxiv.org/abs/2211.12612.

Qi Cai, Zhuoran Yang, and Zhaoran Wang. Reinforcement learning from partial observation: Linear function approximation with provable sample efficiency. In *International Conference on Machine Learning*, 2022.

Leonardo Cella, Karim Lounici, and Massimiliano Pontil. Meta representation learning with contextual linear bandits, 2022. URL https://arxiv.org/abs/2205.15100.

Iadine Chades, Josie Carwardine, Tara Martin, Samuel Nicol, Regis Sabbadin, and Olivier Buffet. Momdps: A solution for modelling adaptive management problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1):267–273, Sep. 2021. doi: 10.1609/aaai.v26i1.8171. URL https://ojs.aaai.org/index.php/AAAI/article/view/8171.

Jeffrey Chan, Aldo Pacchiano, Nilesh Tripuraneni, Yun S Song, Peter Bartlett, and Michael I Jordan. Parallelizing contextual linear bandits. *arXiv preprint arXiv:2105.10590*, 2021.

Niladri Chatterji, Aldo Pacchiano, Peter Bartlett, and Michael Jordan. On the theory of reinforcement learning with once-per-episode feedback. In *Advances in Neural Information Processing Systems*, 2021.

Fan Chen, Yu Bai, and Song Mei. Partially observable rl with b-stability: Unified structural condition and sharp sample-efficient algorithms. In *International Conference on Learning Representations*, 2022a.

Fan Chen, Huan Wang, Caiming Xiong, Song Mei, and Yu Bai. Lower bounds for learning in revealing pomdps. In *International Conference on Machine Learning*, 2023.

Xiaoyu Chen, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *International Conference on Machine Learning*, 2022b.

Yanxi Chen and H. Vincent Poor. Learning mixtures of linear dynamical systems. *CoRR*, abs/2201.11211, 2022. URL https://arxiv.org/abs/2201.11211.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

Philip J Clare, Timothy A Dobbins, and Richard P Mattick. Causal models adjusting for time-varying confounding—a systematic review of the literature. *International Journal of Epidemiology*, 48 (1):254–265, 10 2018. ISSN 0300-5771. doi: 10.1093/ije/dyy218. URL https://doi.org/10.1093/ije/dyy218.

R M Daniel, S N Cousens, B L De Stavola, M G Kenward, and J A C Sterne. Methods for dealing with time-dependent confounding. *Stat. Med.*, 32(9):1584–1618, April 2013.

Constantinos Daskalakis and Ioannis Panageas. The limit points of (optimistic) gradient descent in min-max optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/139c3c1b7ca46a9d4fd6d163d98af635-Paper.pdf.

Thomas Degris, Martha White, and Richard S. Sutton. Off-policy actor-critic, 2013.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, 2019.

Yaqi Duan and Mengdi Wang. Minimax-optimal off-policy evaluation with linear function approximation. *CoRR*, abs/2002.09516, 2020a. URL https://arxiv.org/abs/2002.09516.

Yaqi Duan and Mengdi Wang. Minimax-optimal off-policy evaluation with linear function approximation, 2020b.

Yonathan Efroni, Nadav Merlis, and Shie Mannor. Reinforcement learning with trajectory feedback. In *AAAI conference on artificial intelligence*, 2021.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Matthew Fitzpatrick and Michael Stewart. Asymptotics for markov chain mixture detection. *Econometrics and Statistics*, 22:56–66, 2022. ISSN 2452-3062. doi: https://doi.org/10.1016/j.ecosta.2021.11.004. URL https://www.sciencedirect.com/science/article/pii/S2452306221001337. The 2nd Special issue on Mixture Models.

Steven W Flavell, Nadine Gogolla, Matthew Lovett-Barron, and Moriel Zelikowsky. The emergence and influence of internal states. *Neuron*, 2022.

Zuyue Fu, Zhengling Qi, Zhaoran Wang, Zhuoran Yang, Yanxun Xu, and Michael R. Kosorok. Offline reinforcement learning with instrumental variables in confounded markov decision processes, 2022.

Noah Golowich, Ankur Moitra, and Dhruv Rohatgi. Learning in observable pomdps, without computationally intractable oracles. In *Advances in Neural Information Processing Systems*, 2022.

Rishi Gupta, Ravi Kumar, and Sergei Vassilvitskii. On mixtures of markov chains. In *NIPS*, pages 3441–3449, 2016. URL http://papers.nips.cc/paper/6078-on-mixtures-of-markov-chains.

David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 215–223, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098060. URL https://doi.org/10.1145/3097983.3098060.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), dec 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, and Craig Boutilier. Latent bandits revisited. In *Advances in Neural Information Processing Systems*, volume 33, pages 13423–13433, 2020.

Joey Hong, Branislav Kveton, Manzil Zaheer, Mohammad Ghavamzadeh, and Craig Boutilier. Thompson sampling with a mixture prior. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 7565–7586. PMLR, 2022.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

Lingxiao Huang, K Sudhir, and Nisheeth Vishnoi. Coresets for time series clustering. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22849–22862. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/c115ba9e04ab27fbbb664f932112246d-Paper.pdf.

Rodrigo Andrés Toro Icarte. *Reward Machines*. PhD thesis, University of Toronto, Canada, 2022.

Rodrigo Toro Icarte, Toryn Klassen, Richard Valenzano, and Sheila McIlraith. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *International Conference on Machine Learning*, pages 2107–2116. PMLR, 2018.

Rodrigo Toro Icarte, Ethan Waldie, Toryn Klassen, Rick Valenzano, Margarita Castro, and Sheila McIlraith. Learning reward machines for partially observable reinforcement learning. In *Advances in Neural Information Processing Systems*, 2019.

Rodrigo Toro Icarte, Toryn Q Klassen, Richard Valenzano, and Sheila A McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1704–1713. PMLR, 2017.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, 2018.

Chi Jin, Sham Kakade, Akshay Krishnamurthy, and Qinghua Liu. Sample-efficient reinforcement learning of undercomplete pomdps. In *Advances in Neural Information Processing Systems*, 2020.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. In *Advances in Neural Information Processing Systems*, 2021a.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021b.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998a.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998b. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(98)00023-X. URL https://www.sciencedirect.com/science/article/pii/S000437029800023X.

Nathan Kallus and Masatoshi Uehara. Statistically efficient off-policy policy gradients, 2020.

Nathan Kallus and Angela Zhou. Confounding-robust policy evaluation in infinite-horizon reinforcement learning. *arXiv preprint arXiv:2002.04518*, 2020.

Soummya Kar, H. Vincent Poor, and Shuguang Cui. Bandit problems in networks: Asymptotically efficient distributed allocation rules. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 1771–1778, 2011. doi: 10.1109/CDC.2011.6160719.

Chinmaya Kausik, Kevin Tan, and Ambuj Tewari. Learning mixtures of markov chains and mdps, 2022. URL https://arxiv.org/abs/2211.09403.

Chinmaya Kausik, Kevin Tan, and Ambuj Tewari. Learning mixtures of Markov chains and MDPs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15970–16017. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kausik23a.html.

Chinmaya Kausik, Yangyi Lu, Kevin Tan, Maggie Makar, Yixin Wang, and Ambuj Tewari. Offline policy evaluation and optimization under confounding. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 1459–1467. PMLR, 02–04 May 2024a. URL https://proceedings.mlr.press/v238/kausik24a.html.

Chinmaya Kausik, Mirco Mutti, Aldo Pacchiano, and Ambuj Tewari. A theoretical framework for partially observed reward-states in RLHF. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2024b. URL https://arxiv.org/abs/2402.03282.

Chinmaya Kausik, Kevin Tan, and Ambuj Tewari. Leveraging offline data in linear latent contextual bandits. In *Proceedings of the 42nd International Conference on Machine Learning*. PMLR, 2025. URL https://arxiv.org/abs/2405.17324.

Achim Klenke. Probability theory: A comprehensive course. In *Universitext, Springer Cham.*, 2008. URL https://api.semanticscholar.org/CorpusID:117791811.

Weihao Kong, Raghav Somani, Zhao Song, Sham M. Kakade, and Sewoong Oh. Meta-learning for mixed linear regression. *CoRR*, abs/2002.08936, 2020. URL https://arxiv.org/abs/2002.08936.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, 2016.

Jeongyeol Kwon, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. RL for latent mdps: Regret guarantees and a lower bound. *CoRR*, abs/2102.04939, 2021. URL https://arxiv.org/abs/2102.04939.

Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 633–638, 2017. doi: 10.1109/FOCS.2017.64.

Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *preprint*, 2018.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent converges to minimizers. 2016. doi: 10.48550/ARXIV.1602.04915. URL https://arxiv.org/abs/1602.04915.

Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*, 2021.

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, WWW '10. ACM, April 2010. doi: 10.1145/1772690.1772758. URL http://dx.doi.org/10.1145/1772690.1772758.

Luofeng Liao, Zuyue Fu, Zhuoran Yang, Yixin Wang, Mladen Kolar, and Zhaoran Wang. Instrumental variable value iteration for causal offline reinforcement learning, 2021.

T. Warren Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874, nov 2005. doi: 10.1016/j.patcog.2005.01.025. URL https://doi.org/10.1016%2Fj.patcog.2005.01.025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Qinghua Liu, Alan Chung, Csaba Szepesvári, and Chi Jin. When is partially observable reinforcement learning not scary? In *Conference on Learning Theory*, 2022a.

Qinghua Liu, Praneeth Netrapalli, Csaba Szepesvari, and Chi Jin. Optimistic mle–a generic model-based algorithm for partially observable sequential decision making. *arXiv preprint arXiv:2209.14997*, 2022b.

Yueyang Liu, Xu Kuang, and Benjamin Van Roy. A definition of non-stationary bandits, 2023.

Rui Lu, Gao Huang, and Simon S. Du. On the power of multitask representation learning in linear mdp, 2021a. URL https://arxiv.org/abs/2106.08053.

Yangyi Lu, Ziping Xu, and Ambuj Tewari. Bandit algorithms for precision medicine, 2021b.

E. A. Maharaj. Cluster of time series. *Journal of Classification*, 17(2):297–314, Jul 2000. ISSN 1432-1343. doi: 10.1007/s003570000023. URL https://doi.org/10.1007/s003570000023.

Mohammad Ali Mansournia, Mahyar Etminan, Goodarz Danaei, Jay S Kaufman, and Gary Collins. Handling time varying confounding in observational research. *BMJ: British Medical Journal*, 359, 2017. ISSN 09598138, 17561833. URL https://www.jstor.org/stable/26951608.

Robert McCulloch and Ruey S. Tsay. Statistical analysis of economic time series via markov switching models. *Journal of Time Series Analysis*, 15(5):523–539, 1994. ISSN 0143-9782. doi: 10.1111/j.1467-9892.1994.tb00208.x.

Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*, 2021.

Rui Miao, Zhengling Qi, and Xiaoke Zhang. Off-policy evaluation for episodic partially observable markov decision processes under non-parametric models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 593–606. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/03dfa2a7755635f756b160e9f4c6b789-Paper-Conference.pdf.

Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007a. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf.

Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007b. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(27):815–857, 2008. URL http://jmlr.org/papers/v9/munos08a.html.

Cameron Musco and Christopher Musco. Recursive sampling for the nyström method. 2016. doi: 10.48550/ARXIV.1605.07583. URL https://arxiv.org/abs/1605.07583.

Yash Nair and Nan Jiang. A spectral approach to off-policy evaluation for pomdps. *ArXiv*, abs/2109.10502, 2021.

Hongseok Namkoong, Ramtin Keramati, Steve Yadlowsky, and Emma Brunskill. Off-policy policy evaluation for sequential decisions under unobserved confounding. *arXiv preprint arXiv:2003.05623*, 2020.

Ellen Novoseller, Yibing Wei, Yanan Sui, Yisong Yue, and Joel Burdick. Dueling posterior sampling for preference-based reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, 2020.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models, 2019.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Art B. Owen. *Monte Carlo Theory, Methods and Examples*. 2013. Available at https://artowen.su.domains/mc/.

Aldo Pacchiano, Aadirupa Saha, and Jonathan Lee. Dueling rl: reinforcement learning with trajectory preferences. *arXiv preprint arXiv:2111.04850*, 2021.

Soumyabrata Pal, Arun Sai Suggala, Karthikeyan Shanmugam, and Prateek Jain. Optimal algorithms for latent bandits with cluster structure. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 7540–7577. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/pal23a.html.

Robert W. Platt, Enrique F. Schisterman, and Stephen R. Cole. Time-modified Confounding. *American Journal of Epidemiology*, 170(6):687–694, 08 2009. ISSN 0002-9262. doi: 10.1093/aje/kwp175. URL https://doi.org/10.1093/aje/kwp175.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Doina Precup, Richard S Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. In *International Conference on Machine Learning*, 2000.

William H. Press. Bandit solutions provide unified ethical models for randomized clinical trials and comparative effectiveness research. *Proceedings of the National Academy of Sciences*, 106 (52):22387–22392, 2009. doi: 10.1073/pnas.0912378106. URL https://www.pnas.org/doi/abs/10.1073/pnas.0912378106.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, 2013.

Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. Technical report, EECS Berkeley, 2017.

Aadirupa Saha, Aldo Pacchiano, and Jonathan Lee. Dueling rl: Reinforcement learning with trajectory preferences. In *International Conference on Artificial Intelligence and Statistics*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Chengchun Shi, Masatoshi Uehara, Jiawei Huang, and Nan Jiang. A minimax learning approach to off-policy evaluation in confounded partially observable markov decision processes. In *International Conference on Machine Learning*, 2021.

Daniel Shin, Anca D Dragan, and Daniel S Brown. Benchmarks and algorithms for offline preference-based reward learning. *arXiv preprint arXiv:2301.01392*, 2023.

Marta Soare, Ouais Alsharif, Alessandro Lazaric, and Joelle Pineau. Multi-task linear bandits. In *NIPS 2014 Workshop on Transfer and Multi-Task Learning*, 2014.

Yuda Song, Yifei Zhou, Ayush Sekhari, J. Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid RL: Using both offline and online data can make RL efficient. In *The Eleventh International Conference on Learning Representations*, 2023.

Lauren Steimle, David Kaufman, and Brian Denton. Multi-model markov decision processes. *IISE Transactions*, 53:1–39, 03 2021. doi: 10.1080/24725854.2021.1895454.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. *arXiv preprint arXiv:2401.04056*, 2024.

Csaba Szepesvári. Lecture notes in theoretical foundations of reinforcement learning: Lecture 23, tabular mdps. https://rltheory.github.io/lecture-notes/online-rl/lec23/, 2023.

Kevin Tan, Wei Fan, and Yuting Wei. Hybrid reinforcement learning breaks sample size barriers in linear mdps. In *Advances in Neural Information Processing Systems*, 2024.

Guy Tennenholtz, Shie Mannor, and Uri Shalit. Off-policy evaluation in partially observable environments. *CoRR*, abs/1909.03739, 2019. URL http://arxiv.org/abs/1909.03739.

Andrea Tirinzoni, Aymen Al-Marjani, and Emilie Kaufmann. Optimistic pac reinforcement learning: the instance-dependent view. In *International Conference on Algorithmic Learning Theory*, 2023.

Nilesh Tripuraneni, Chi Jin, and Michael I. Jordan. Provable meta-learning of linear representations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10434–10443. PMLR, 2021.

Masatoshi Uehara, Haruka Kiyohara, Andrew Bennett, Victor Chernozhukov, Nan Jiang, Nathan Kallus, Chengchun Shi, and Wen Sun. Future-dependent value-based off-policy evaluation in pomdps, 2022.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *J. Comput. Syst. Sci*, 68:2004, 2004.

Mathukumalli Vidyasagar. *Learning and Generalization: With Applications to Neural Networks*. Springer Publishing Company, Incorporated, 2nd edition, 2010. ISBN 1849968675.

Andrew Wagenmaker and Aldo Pacchiano. Leveraging offline data in online reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35300–35338. PMLR, 2023.

Andrew J Wagenmaker, Max Simchowitz, and Kevin Jamieson. Beyond no regret: Instance-dependent pac reinforcement learning. In *Conference on Learning Theory*, 2022.

Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Provably efficient causal reinforcement learning with confounded observational data. *CoRR*, abs/2006.12311, 2020. URL https://arxiv.org/abs/2006.12311.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Represent to control partially observed systems: Representation learning with provable sample efficiency. In *International Conference on Learning Representations*, 2023a.

Yuanhao Wang, Qinghua Liu, and Chi Jin. Is rlhf more difficult than standard rl? *arXiv preprint arXiv:2306.14111*, 2023b.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(1):1–27, 02 2023. ISSN 1369-7412. doi: 10.1093/jrsssb/qkad009. URL https://doi.org/10.1093/jrsssb/qkad009.

Yimin Wei, Yanhua Cao, and Hua Xiang. A note on the componentwise perturbation bounds of matrix inverse and linear systems. *Applied Mathematics and Computation*, 169(2):1221–1236, 2005. ISSN 0096-3003. doi: https://doi.org/10.1016/j.amc.2004.10.065. URL https://www.sciencedirect.com/science/article/pii/S0096300304008471.

Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Chun Shan Wong and Wai Keung Li. On a mixture autoregressive model. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62(1):95–115, 2000. ISSN 13697412, 14679868. URL http://www.jstor.org/stable/2680680.

Wing Hung Wong and Xiaotong Shen. Probability Inequalities for Likelihood Ratios and Convergence Rates of Sieve MLES. *The Annals of Statistics*, 23(2):339 – 362, 1995. doi: 10.1214/aos/1176324524. URL https://doi.org/10.1214/aos/1176324524.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*, 2023.

Tengyang Xie, John Langford, Paul Mineiro, and Ida Momennejad. Interaction-grounded learning. In *International Conference on Machine Learning (ICML)*, 2021.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint, 2024. URL https://arxiv.org/abs/2312.11456.

Wanqi Xue, Qingpeng Cai, Zhenghai Xue, Shuo Sun, Shuchang Liu, Dong Zheng, Peng Jiang, Kun Gai, and Bo An. Prefrec: Recommender systems with human preferences for reinforcing long-term user engagement. *arXiv preprint arXiv:2212.02779*, 2022.

Jiaqi Yang, Qi Lei, Jason D. Lee, and Simon S. Du. Nearly minimax algorithms for linear bandits with shared representation, 2022. URL https://arxiv.org/abs/2203.15664.

Ming Yin and Yu-Xiang Wang. Asymptotically efficient off-policy evaluation for tabular reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3948–3958. PMLR, 2020.

Bin Yu. Rates of Convergence for Empirical Processes of Stationary Mixing Sequences. *The Annals of Probability*, 22(1):94 – 116, 1994. doi: 10.1214/aop/1176988849. URL https://doi.org/10.1214/aop/1176988849.

Yi Yu, Tengyao Wang, and Richard J. Samworth. A useful variant of the davis–kahan theorem for statisticians, 2014.

Wenhao Zhan, Masatoshi Uehara, Wen Sun, and Jason D Lee. Pac reinforcement learning for predictive state representations. In *International Conference on Learning Representations*, 2022.

Wenhao Zhan, Masatoshi Uehara, Wen Sun, and Jason D Lee. How to query human feedback efficiently in rl? *arXiv preprint arXiv:2305.18505*, 2023.

Junzhe Zhang and Elias Bareinboim. Markov decision processes with unobserved confounders : A causal approach. 2016.

Ruiqi Zhang, Xuezhou Zhang, Chengzhuo Ni, and Mengdi Wang. Off-policy fitted q-evaluation with differentiable function approximators: Z-estimation and inference theory. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26713–26749. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zhang22al.html.

Han Zhong, Wei Xiong, Sirui Zheng, Liwei Wang, Zhaoran Wang, Zhuoran Yang, and Tong Zhang. Gec: A unified framework for interactive decision making in mdp, pomdp, and beyond. *arXiv preprint arXiv:2211.01962*, 2023.

Doudou Zhou, Yufeng Zhang, Aaron Sonabend-W, Zhaoran Wang, Junwei Lu, and Tianxi Cai. Federated offline reinforcement learning, 2024. URL https://arxiv.org/abs/2206.05581.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: Variational bayes-adaptive deep rl via meta-learning. *Journal of Machine Learning Research*, 22(289):1–39, 2021.