On the Benefits of Multitask Learning: A Perspective Based on Task Diversity

by

Ziping Xu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2023

Doctoral Committee:

Professor Ambuj Tewari, Chair
Assistant Professor Nan Jiang
Professor Jian Kang
Assistant Professor Yixin Wang
Professor Ji Zhu

Ziping Xu
zipingxu@umich.edu
ORCID iD: 0000-0002-2591-0356

Dedicated to my family.

ACKNOWLEDGEMENTS

As I reach the end of my PhD journey, I am filled with gratitude for the incredible five years I have spent immersed in the world of Machine Learning research. None of this would have been possible without the support and collaboration of numerous people who have accompanied me throughout my graduate school years.

First and foremost, I extend my deepest appreciation to my advisor, Ambuj Tewari. Ambuj has not only been an exceptional advisor but also a profound inspiration on my academic path. His profound understanding of Statistics and Machine Learning, coupled with his great enthusiasm for research, has always inspired me. I still remember our initial meeting, where Ambuj energetically discussed the significant challenges in Reinforcement Learning and how our collaboration could contribute to solving these problems and making great impacts. In that moment, I aspired to become a researcher like Ambuj, someone who finds fulfillment in this lifelong pursuit. As I embark on my new role as an assistant professor, the transition from student to professor may be uneasy, but I find solace in having a perfect role model who exemplifies what a good advisor-student relationship should look like. Indeed, Ambuj will forever remain my role model in the realm of academia.

I would also like to express my gratitude to my fellow students and the esteemed professors in the department. Their unwavering care and support have made the Department of Statistics into a second home for me. Professor Ji Zhu has been both my faculty mentor and a valuable committee member throughout my PhD studies. He provided invaluable guidance and numerous suggestions whenever I felt uncertain about my research and personal life. I extend my heartfelt thanks to Rita Hu, my closest friend. As an international student experiencing life in the United States for the first time, Rita has selflessly assisted me in all aspects of my life. During the challenging times of job searching, Rita and her husband Yutong offered a lot of support by helping me practice my job talks and providing suggestions.

Above all, my family has been the most important pillar of strength, carrying me through the most difficult times. My brother, Zifan Xu, residing in Austin, has visited me many times during my PhD, providing invaluable emotional support and easing the burden of being away from my parents. Despite living on the other side of the world, my parents have

showered me with unconditional love and care. Their unwavering support for the crucial decisions in my life has been a source of reassurance. Knowing that I have their unwavering backing emboldens me to embark on any adventure life presents.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

Multitask learning (MTL) has achieved remarkable success in numerous domains, such as healthcare, computer vision, and natural language processing, by leveraging the relatedness across tasks. However, current theories of multitask learning fall short in explaining the success of some phenomena commonly observed in practice. For instance, many empirical studies have shown that having a diverse set of tasks improves both training and testing performance. This thesis aims at providing new theoretical insights into the significance of task diversity in two major learning settings: Supervised Learning and Reinforcement Learning. For supervised MTL, we focus on studying a popular learning paradigm known as multitask representation learning and provide a theoretical foundation that establishes diversity as a crucial condition for achieving good generalization performance. In the setting where tasks can be adaptively chosen, we propose an online learning algorithm that effectively achieves diversity with low regret. I then expand the discussion to Reinforcement Learning (RL), which involves making sequential decisions to optimize long-term rewards. Previous exploration designs in RL were either computationally intractable or lacked formal guarantees. We show that, in addition to the generalization benefits demonstrated in supervised learning, multitask reinforcement learning with a diverse set of tasks enables sample-efficient myopic exploration. This is surprising because myopic exploration is provably sample inefficient in the worst case even for a single task.

CHAPTER 1

Introduction

In the era of big data, there is an enormous demand for learning from multiple datasets/tasks with the hope to improve the overall performance by leveraging the similarities across different tasks. The widely used modern deep learning models are usually "data-hungry". Though the community has made great achievement in creating large-scale datasets for training large language models or computer vision models, datasets of that scale may not be available in other domains like healthcare and medical image, where collecting data is expensive. In these cases, Multitask Learning (MTL) can be an extremely useful tool as it exploits useful information from other related learning tasks to help alleviate this data sparsity problem. In this thesis, we review the commonly used algorithms for MTL and study the underlying theory that guides the practical algorithm designs. We study MTL on two major setups – **Supervised Learning** and **Reinforcement Learning**.

This thesis aims at providing theoretical understanding of the MTL. More specifically, we ask what property on the training task set guarantees a good generalization performance. For supervised learning, we bound the generalization error on the average of tasks or the downstream target tasks. We show that a diverse task set guarantees the worst-case generalization error and we provide hard instances on achieving diversity. For RL, we show that diversity again plays an important role in online exploration, a crucial part of RL literature. We provide a sample complexity bound for task set with diversity condition. Before we present our main results, we briefly review important concepts in MTL.

1.1 Practices and Theories for Supervised MTL

Chapter 2 and Chapter 3 in this thesis focuses on the Supervised Learning setting, where each task indexed by t is modeled as $\mathcal{D}_t$, a distribution over a shared input space $\mathcal{X}$ and output space $\mathcal{Y}$ (Zhang and Yang, 2017). Our goal is to achieve a low excess error on the average over all the tasks, or generalization error on a downstream target task. The understanding of the

benefits of MTL under this setup heavily relies on the crucial assumptions on task similarity. We review the commonly made assumptions in the supervised learning setting. We also note that many recent works on MTRL also inherit similar assumptions from supervised MTL.

1.1.1 Similarity Assumptions

Teshima et al. (2020) gives a brief summary of different similarity assumptions, upon which we compare more methods. The assumptions can be broadly categorized into five classes: 1) parametric assumptions; 2) invariant conditionals and marginals; 3) small discrepancy; 4) transferable parameters; and 5) restricted function class. In the following discussion, we denote the feature and response variables by X and Y.

(1) Parametric assumptions. This line of work directly assumes a parametric data generating distribution , e.g. Gaussian mixture model (Sugiyama and Storkey, 2007; Lawrence and Platt, 2004; Bonilla et al., 2008; Schwaighofer et al., 2005). Under the Gaussian mixture assumptions, the relatedness between tasks is characterized by a Gaussian Process prior over the mean function. Other parametric models assume a parametric distribution shift. Example includes location-scale transform (of $P_{X|Y}$) of class conditionals with a linear form (Zhang et al., 2013; Gong et al., 2016b), linearly dependent class conditionals (Zhang et al., 2019). The benefits of a parametric assumption is the clearness in the task relatedness despite of its worse generalization.

(2) Invariant conditionals and marginals. To ease the difficulties of transfer, some methods assume covariate shifts, where conditional distribution $P_{Y|X}$ remains the same (Sugiyama et al., 2007; Gretton et al., 2009; Kpotufe and Martinet, 2018). Other assumptions on marginals include target shifts with $P_{X|Y}$ staying the same (Zhang et al., 2019), conditional shifts with P_Y remaining the same, while $P_{X|Y}$ changing (Zhang et al., 2013; Nguyen et al., 2016). Furthermore, Pan et al. (2010) extends the covariate shift setting by allowing a transformation on the covariates, i.e. there exists a mapping $\mathcal{T}$ such that $P_{Y|\mathcal{T}(X)}$ matches.

(3) Small discrepancy. Another line of work relies on certain distributional. For example, Wang et al. (2019a); Shui et al. (2019); Kuroki et al. (2019); Cortes et al. (2019) assumes an upper bound on the discrepancy between some performance measures of different tasks using the same prediction; Courty et al. (2016); Redko et al. (2017) use an integral probability metric. Ben-David et al. (2010) directly measures the discrepancy for the data generating probabilities between tasks, also called $\mathcal{H}$-divergence. Small-discrepancy assumptions allow

great flexibility on the prediction function class, while can only handle tasks that are close under some metric.

(4) Transferable parameters. Some methods assume the parameters are transferable, i.e. the optimal parameters for different tasks are close (Kumagai, 2016; Xu et al., 2020). Fine-tuning is a widely applied domain adaptation method, which trains a model on source domain and fine-tune it on the target domain using small amount of data (Howard and Ruder, 2018; Hoaglin and Iglewicz, 1987). Xu et al. (2020) has shown that under some specific function class, e.g. linear model, transferable parameters and small discrepancy assumptions can be translated between each other.

(5) Joint function class. Joint function class assumption assumes a hypothesis class $\mathcal{F}$ over the joint function $\boldsymbol{f} = (f_1, \ldots, f_T)$ for a tasks set $[T]$. Note that some methods using small discrepancy also assume a hypothesis class. However, they use the same class to approximate all the tasks, i.e. their solution is $\boldsymbol{f} = (f, \ldots, f) \in \mathcal{F}^T$.

Different from previous methods, the similarities between tasks are characterized by the hypothesis class. For example, in the field of representation learning, Maurer et al. (2016); Du et al. (2020); Tripuraneni et al. (2020) use the class $\{(f_1 \circ h, \ldots, f_T \circ h) : h \in \mathcal{H}, f_t \in \mathcal{F}, \forall t \in [T]\}$. Tasks are similar under a common representation, while the task-specific function f_t can be irrelevant. The transferable parameters setting can be convert to the joint function class setting using the class $\{(\theta_1, \theta_2) : \|\theta_1 - \theta_2\| \leq d\}$, for two-task transfer learning setting.

Another commonly used transfer learning method is the off-set model, assuming that the optimal target function is a simple modification from the optimal function of source model (Torrey and Shavlik, 2010; Wang and Schneider, 2015). Using the restricted function class setting, the offset model can be written as $\{(f_1, f_2) : f_1 \in \mathcal{F}, f_2 \in \{f_1 + h : h \in \mathcal{H}\}\}$. Du et al. (2017) generalizes the setting by allowing a general transformation from f_1 to f_2.

1.1.2 MTL Algorithms

Different practices apply to MTL with different similarity assumptions. Zhang and Yang (2017) gives a brief summary of the popular MTL algorithms. We improve the review by corresponding them to different similarity assumptions.

Feature-based MTL. In this category, all MTL models assume that different tasks share a feature representation, which corresponds to shared representation assumption. More specifically, these methods can be further categorized into two approaches, including the

feature transformation approach, and the feature selection approach. Feature transformation approach learns a transformation of the original feature inputs. A popular choice is to learn a multi-layer neural network for each task and the parameters of the first K layers are shared by different tasks, resulting a shared feature transformation (Caruana, 1998; Argyriou et al., 2006, 2008). Another category aims to select a subset of original features as the shared feature representation for different tasks. Subset feature selection can be viewed as a shared transformation in a general sense. However, they can be significantly different in practice as the set of shared transformation is restricted to subset selection. Lozano and Swirszcz (2012) proposed a multi-level Lasso model whose sparsity is shared across different tasks.

Regularization-based approaches. If one believes that the true models of different tasks with a parametric assumption are closed, they may consider minimizing the average loss of different tasks by regularizing the distances between the parameters of different tasks Evgeniou and Pontil (2004). In Chapter 4 we also introduce a regularization-based approach to solve MTL with a special funnel structure in recommendation system (Xu et al., 2020).

Model fine-tuning. Another way of utilizing the closeness of the true model parameters is model Fine-tuning. Fine-tuning refers to the practice of fine-tuning a deep neural network with few steps of stochastic gradient descent method. A trendy area of meta-learning can be viewed as finding a global parameter, from which fine-tuning is effective for all the tasks (Vilalta and Drissi, 2002; Vanschoren, 2019). It implicitly assumes that the true parameters of different tasks are close.

1.1.3 Adaptive Task Scheduling

While previous literature often assumes a predetermined (and often equal) number of observations for all the tasks, in many applications, we are allowed to decide the *order* in which the tasks are presented and the *number of observations* from each task. Any strategy that tries to improve the performance with a adaptively chosen task scheduling is usually referred to **curriculum learning (CL)** (Bengio et al., 2009). The agent that schedules tasks at each step is often referred as the *task scheduler*.

Though curriculum learning has been extensively used in modern machine learning (Gong et al., 2016a; Sachan and Xing, 2016; Tang et al., 2018; Narvekar et al., 2020), there is very little theoretical understanding of the actual benefits of CL. We also do not know whether the heuristic methods used in many empirical studies can be theoretically justified. Chapter 3 summarizes my work on the theoretical justifications of curriculum learning.

1.2 Multitask Reinforcement Learning

Recent interests of machine learning community has been detoured to a bigger scope that learns how to make decisions to optimize a long-term goal. Inspired by behaviorist psychology, reinforcement learning studies how to take actions in an environment to maximize the cumulative reward and it shows good performance in many applications with AlphaGo, which beats humans in the Go game, as a representative application. When environments are similar, different reinforcement learning tasks can use similar policies to make decisions, which is a motivation of the proposal of multi-task reinforcement learning (Wilson et al., 2007; Li et al., 2009; Lazaric and Ghavamzadeh, 2010).

Many of these works (Wilson et al., 2007) solves different tasks through a hierarchical Bayesian infinite mixture model. Li et al. (2009) characterizes each task is characterized via a regionalized policy and a Dirichlet process is used to cluster tasks. Similarly to supervised multitask representation learning, the value functions in Calandriello et al. (2014) in different tasks are assumed to share sparse parameters and it applies the multi-task feature selection method.

One may understand the benefits of multitask Reinforcement Learning in the same way we understand the benefits for Supervised MTL. Many recent theoretical works have contributed to understanding the benefits of MTRL (Agarwal et al., 2022; Brunskill and Li, 2013; Calandriello et al., 2014; Cheng et al., 2022; Lu et al., 2021; Uehara et al., 2021; Yang et al., 2022; Zhang and Wang, 2021) by exploiting the shared structures across tasks. An earlier line of works Brunskill and Li (2013) assumes that tasks are clustered and the algorithm adaptively learns the identity of each task, which allows it to pool observations. For linear Markov Decision Process (MDP) settings (Jin et al., 2020b), Lu et al. (2021) shows a bound on the sub-optimality of the learned policy by assuming a full-rank least-square value iteration weight matrix from source tasks. Agarwal et al. (2022) makes a different assumption that the target transition probability is a linear combination of the source ones, and the feature extractor is shared by all the tasks. Our work differs from all these works as we focus on the reduced complexity of exploration design.

1.3 Diversity

A continuing topic throughout the thesis is the importance of having a diverse task set in achieving good generalization performance and better sample efficiency in RL. Indeed, this has been demonstrated by many empirical works. We give a brief review on how diversity is discussed in empirical studies and the previous theoretical guarantees through diversity.

Empirical evidence for supervised learning. In computer vision tasks, heterogeneous datasets are often available. For instance, tasks may target at goals ranging from image classification to harder ones like segmentation and object detection. Yu et al. (2020) construct a diverse driving dataset with various prediction structures and serve different aspects of a complete driving systems, which improves the performance by at least 20% on tasks with different goals. Similar evidence has been observed for biological and medical domains (Caruana et al., 1995; Mulyar et al., 2021; Aoki et al., 2022; Sun et al., 2022) and NLP (Subramanian et al., 2018; Radford et al., 2019). Some works even consider multitask learning across different domains. For instance, Nguyen and Okatani (2019) study multitask learning for a hierarchical vision-language representation. Recent interest of machine learning community on foundation model can also be understood a representation learning on an extremely diverse task set Yu et al. (2022).

Diversity for RL. There is also a large body of literature discussing the role of diversity for RL. For instance, in robotic learning, a common practice is to randomize training environment (Akkaya et al., 2019). Domain randomization is shown to significantly improve the generalization from policy trained on simulated environments to real-world environments in various different applications (Sadeghi and Levine, 2016; Tobin et al., 2017; Peng et al., 2018; Andrychowicz et al., 2020; Chen et al., 2021). Intuitively, if one thinks of training RL agent as a process of mastering skills, then it has to seen a diverse enough tasks to demonstrate different types of skills in order to achieve a good generalization performance.

Theories on diverse MTL. Despite the attention diversity received in empirical studies, we still lack a good theoretical understanding of diversity. Tripuraneni et al. (2020) proposed a first theoretical analysis on the generalization benefits of diversity in a multitask representation learning setting. We extend their results to a more general setting in Chapter 2 and 3. The role of diversity is even more under-explored for RL. We will make a first attempt to understand the "exploration" benefits of having a diverse task set.

1.4 Thesis Organization

This thesis is organized in the following manner. In Chapter 2, we study the benefits of multitask learning in a shared representation function setting. We first show that diverse task set provides a strong worst-case generalization error guarantee. We propose Eluder dimension as a measure for the number of tasks needed to achieve diversity. Chapter 3 closely follows Chapter 2 and consider the curriculum learning scenario, where tasks can be

adaptively chosen. We proposed an online learning algorithm that adaptively select task set that achieves diversity. Chapter 4 bridges supervised learning and reinforcement learning by considering a multitask learning on bandit setting. We show that under a special funnel structure, MTL can significantly benefit with a smaller regret bound guarantee. Chapter 5 studies MTRL setting. We show that diversity continue to play an important role in RL. Exploration is a core topic for online RL. We show that exploring on a diverse set of tasks allows sample-efficient myopic exploration, which has been shown sample-inefficient in the worst case. This has strong empirical implications as myopic exploration is easy to implement and generalizes well.

CHAPTER 2

Supervised Multitask Learning

Recent papers on the theory of representation learning has shown the importance of a quantity called diversity when generalizing from a set of source tasks to a target task. Most of these papers assume that the function mapping shared representations to predictions is linear, for both source and target tasks. In practice, researchers in deep learning use different numbers of extra layers following the pretrained model based on the difficulty of the new task. This motivates us to ask whether diversity can be achieved when source tasks and the target task use different prediction function spaces beyond linear functions. We show that diversity holds even if the target task uses a neural network with multiple layers, as long as source tasks use linear functions. If source tasks use nonlinear prediction functions, we provide a negative result by showing that depth-1 neural networks with ReLu activation function need exponentially many source tasks to achieve diversity. For a general function class, we find that eluder dimension gives a lower bound on the number of tasks required for diversity. Our theoretical results imply that simpler tasks generalize better. Though our theoretical results are shown for the global minimizer of empirical risks, their qualitative predictions still hold true for gradient-based optimization algorithms as verified by our simulations on deep neural networks. [1]

2.1 Introduction

It has become a common practice (Tan et al., 2018) to use a pre-trained network as the representation for a new task with *small* sample size in various areas including computer vision (Marmanis et al., 2015), speech recognition (Dahl et al., 2011; Jaitly et al., 2012; Howard and Ruder, 2018) and machine translation (Weng et al., 2020). Most representation learning is based on the assumption that the source tasks and the target task share the same low-dimensional representation.

[1] This chapter is based on the paper Xu and Tewari (2021) published at NeurIPS 2021 with Ambuj Tewari.

In this chapter, following the work of Tripuraneni et al. (2020), we assume that each task, indexed by t, generates observations noisily from the mean function $f_t^* \circ h^*$, where h^* is the true representation shared by all the tasks and f_t^* is called the prediction function. We assume $h^* \in \mathcal{H}$, the representation function space and $f_t \in \mathcal{F}_t$, the prediction function space. Tripuraneni et al. (2020) proposed a diversity condition which guarantees that the learned representation will generalize to target tasks with any prediction function. Assume we have T source tasks labeled by $1, \ldots, T$ and $\mathcal{F}_1 = \cdots = \mathcal{F}_T = \mathcal{F}_{so}$. We denote the target task by ta and let $\mathcal{E}_t(f, h) \in \mathbb{R}$ be the excess error of $f, h \in \mathcal{F}_t \times \mathcal{H}$ for task t. The diversity condition can be stated as follows: there exists some $\nu > 0$, such that for any $f_{ta}^* \in \mathcal{F}_{ta}$ and any $h \in \mathcal{H}$,

$$\underbrace{\inf_{f_{ta} \in \mathcal{F}_{ta}} \mathcal{E}_{ta}(f_{ta}, h)}_{\text{Excess error given } h \text{ for target task}} \leq \nu \underbrace{\inf_{f_t \in \mathcal{F}_{so}, t=1,\ldots,T} \frac{1}{T} \sum_{t=1}^{T} \mathcal{E}_t(f_t, h)}_{\text{Excess error given } h \text{ for source tasks}}.$$

The diversity condition relates the excess error for source tasks and the target task with respect to any fixed representation h. A smaller ν indicates a better transfer from source tasks to the target task. Generally, we say T source tasks achieve diversity over $\mathcal{F}_{ta}$ when ν is finite and relatively small.

The number of source tasks plays an important role here. To see this, assume $\mathcal{F}_{so} = \mathcal{F}_{ta} = \mathcal{F}$ is a discrete function space and each function can be arbitrarily different. In this case, we will need the target task to be the same as at least one source task, which in turn requires $T \geq |\mathcal{F}|$ and $\nu \geq |\mathcal{F}|$. So far, it has only been understood that when *both* source tasks and target task use *linear* prediction functions, it takes at least d source tasks to be diverse, where d is the number of dimension of the linear mappings. This chapter answers the following two open questions with a focus on the deep neural network (DNN) models:

How does representation learning work when $\mathcal{F}_{so} \neq \mathcal{F}_{ta}$?
How many source tasks do we need to achieve diversity with nonlinear prediction functions?

There are strong practical motivations to answer the above two questions. In practice, researchers use representation learning despite the difference in difficulty levels of source and target tasks. We use a more complex function class for substantially harder target task, which means $\mathcal{F}_{so} \neq \mathcal{F}_{ta}$. This can be reflected as extra layers when a deep neural network model is used as prediction functions. On the other hand, the source task and target task may have different objectives. Representation pretrained on a classification problem, say ImageNet, may be applied to object detection or instance segmentation problems. For instance, Oquab et al. (2014) trained a DNN on ImageNet and kept all the layers as the representation except for the last linear mapping, while two fully connected layers are used for the target task on

object detection.

Another motivation for our work is the mismatch between recently developed theories in representation learning and the common practice in empirical studies. Recent papers on the theory of representation learning all require multiple sources tasks to achieve diversity so as to generalize to any target task in $\mathcal{F}$ (Maurer et al., 2016; Du et al., 2020; Tripuraneni et al., 2020). However, most pretrained networks are only trained on a single task, for example, the ImageNet pretrained network. To this end, we will show that a single multi-class classification problem can be diverse.

Lastly, while it is common to simply use linear mapping as the source prediction function, there is no clear theoretical analysis showing whether or not diversity can be achieved with *nonlinear* prediction function spaces.

Main contributions. We summarize the main contributions made in this chapter.

1. We show that diversity over $\mathcal{F}_{so}$ implies diversity over $\mathcal{F}_{ta}$, when both $\mathcal{F}_{so}$ and $\mathcal{F}_{ta}$ are DNNs and $\mathcal{F}_{ta}$ has more layers. More generally, the same statement holds when $\mathcal{F}_{ta}$ is more complicated than $\mathcal{F}_{so}$, in the way that $\mathcal{F}_{ta} = \mathcal{F}'_{ta} \circ (\mathcal{F}_{so}^{\otimes m})$ for some positive integer m[2] and function class $\mathcal{F}'_{ta}$.

2. Turning our attention to the analysis of diversity for non-linear prediction function spaces, we show that for a depth-1 NN, it requires $\Omega(2^d)$ many source tasks to establish diversity with d being the representation dimension. For general $\mathcal{F}_{so}$, we provide a lower bound on the number of source tasks required to achieve diversity using the eluder dimension (Russo and Van Roy, 2013) and provide a upper bound using the generalized rank (Li et al., 2021).

3. We show that, from the perspective of achieving diversity, a single source task with multiple outputs can be equivalent to multiple source tasks. While our theories are built on empirical risk minimization, our simulations on DNNs for a multi-variate regression problem show that the qualitative predictions our theory makes still hold when stochastic gradient descent is used for optimization.

2.2 Preliminaries

We first introduce the mathematical setup of the problem studied In this chapter along with the two-phase learning method that we will focus on.

[2] $\mathcal{F}^{\otimes T}$ is the T times Cartesian product of $\mathcal{F}$.

Problem setup. Let $\mathcal{X}$ denote the input space. We assume the same input distribution P_X for all tasks, as covariate shift is not the focus of this work. In our representation learning setting, there exists a generic feature representation function $h^* \in \mathcal{H} : \mathcal{X} \mapsto \mathcal{Z}$ that is shared across different tasks, where $\mathcal{Z}$ is the feature space and $\mathcal{H}$ is the representation function space. Since we only consider the different prediction functions, each task, indexed by t, is defined by its prediction function $f_t^* \in \mathcal{F}_t : \mathcal{Z} \mapsto \mathcal{Y}_t \subset [0,1]$, where $\mathcal{F}_t$ is the prediction function space of task t and $\mathcal{Y}_t$ is the corresponding output space. The observations $Y_t = f_t^* \circ h^*(X) + \epsilon$ are generated noisily with mean function $f_t^* \circ h^*$, where $X \sim P_X$ and ϵ is zero-mean noise that is independent of X.

Our representation is learned on source tasks $\boldsymbol{f}_{so}^* \coloneqq (f_1^*, \ldots, f_T^*) \in \mathcal{F}_1 \times \cdots \times \mathcal{F}_T$ for some positive integer T. We assume that all the prediction function spaces $\mathcal{F}_t = \mathcal{F}_{so}$ are the same over the source tasks. We denote the target task by $f_{ta}^* \in \mathcal{F}_{ta} : \mathcal{Z} \mapsto \mathcal{Y}_t \subset [0,1]$, where $\mathcal{F}_{ta}$ is the target prediction function space. Unlike the previous papers (Du et al., 2020; Tripuraneni et al., 2020; Maurer et al., 2016), which all assume that the same prediction function space is used for all tasks, we generally allow for the possibility that $\mathcal{F}_{so} \neq \mathcal{F}_{ta}$.

Learning algorithm. We consider the same two-phase learning method as in Tripuraneni et al. (2020). In the first phase (the training phase), $\boldsymbol{n} = (n_1, \ldots, n_T)$ samples from each task are available to learn a good representation. In the second phase (the test phase), we are presented n_{ta} samples from the target task to learn its prediction function using the pretrained representation learned in the training phase.

We denote a dataset of size n from task f_t by $S_t^n = \{(x_{ti}, y_{ti})\}_{i=1}^n$. We use empirical risk minimization (ERM) for both phase. In the training phase, we minimize average risks over $\{S_t^{n_t}\}_{t=1}^T$:

$$\hat{R}(\boldsymbol{f}, h \mid \boldsymbol{f}_{so}^*) \coloneqq \frac{1}{\sum_t n_t} \sum_{t=1}^{T} \sum_{i=1}^{n_t} l_{so}(f_t \circ h(x_{ti}), y_{ti}),$$

where $l_{so} : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ is the loss function for the source tasks and $\boldsymbol{f} = (f_1, \ldots, f_T) \in \mathcal{F}_{so}^{\otimes T}$. The estimates are given by $(\hat{\boldsymbol{f}}_{so}, \hat{h}) \in \arg\min_{\boldsymbol{f},h} \hat{R}(\boldsymbol{f}, h \mid \boldsymbol{f}_{so}^*)$. In the second phase, we obtain the dataset $\{x_{ta,i}, y_{ta,i}\}_i^{n_{ta}}$ from the target task and our predictor $\hat{f}_{ta}$ is given by

$$\arg\min_{f \in \mathcal{F}_{ta}} \hat{R}(f, \hat{h} \mid f_{ta}^*) \coloneqq \frac{1}{n_{ta}} \sum_{i=1}^{n_{ta}} l_{ta}(f \circ \hat{h}(x_{ta,i}), y_{ta,i}),$$

for some loss function l_{ta} on the target task. We also use $R(\cdot, \cdot \mid \cdot)$ for the expectation of the above empirical risks. We denote the generalization error of certain estimates f, h by $\mathcal{E}(f, h \mid \cdot) \coloneqq R(f, h \mid \cdot) - \min_{f' \in \mathcal{F}, h' \in \mathcal{H}} R(f', h' \mid \cdot)$, where $\mathcal{F}$ can be either $\mathcal{F}_{so}^{\otimes T}$ or $\mathcal{F}_{ta}$

depending on the tasks. Our goal is to bound the generalization error of the target task.

For simplicity, our results are presented under square loss functions. However, we show that our results can generalize to different loss functions in Appendix 2.A. We also assume $n_1 = \cdots = n_T = n_{so}$ for some positive integer n_{so}.

2.2.1 Model complexity

As this chapter considers general function classes, our results will be presented in terms of the complexity measures of classes of functions. We follow the previous literature (Maurer et al., 2016; Tripuraneni et al., 2020), which uses Gaussian complexity. Note that we do not use the more common Rademacher complexity as the proofs require a decomposition theorem that only holds for Gaussian complexity.

For a generic vector-valued function class $\mathcal{Q}$ containing functions $q : \mathbb{R}^d \mapsto \mathbb{R}^r$ and N data points, $X_N = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^T$, the empirical Gaussian complexity is defined[3] as

$$\hat{\mathfrak{G}}_N(\mathcal{Q}) = \mathbb{E}_{\mathbf{g}}\left[\sup_{\mathbf{q}\in\mathcal{Q}} \frac{1}{\sqrt{N}} \sum_{i=1}^{N} g_i^T q(\boldsymbol{x}_i)\right], \quad g_i \sim \mathcal{N}(0, I_r) \quad i.i.d.$$

The corresponding population Gaussian complexity is defined as $\mathfrak{G}_N(\mathcal{Q}) = \mathbb{E}_{X_N}\left[\hat{\mathfrak{G}}_N(\mathcal{Q})\right]$, where the expectation is taken over the distribution of X_N.

2.2.2 Diversity

Source tasks have to be diverse enough, to guarantee that the representation learned from source tasks can be generalized to any target task in $\mathcal{F}_{ta}$. To measure when transfer can happen, we introduce the following two definitions, namely that of *transferability* and *diversity*.

Definition 2.1. *For some $\nu, \mu > 0$, we say the source tasks $f_1^*, \ldots, f_T^* \in \mathcal{F}_{so}$ are (ν, μ)-transferable to task f_{ta}^* if*

$$\sup_{h\in\mathcal{H}} \frac{\inf_{f\in\mathcal{F}_{ta}} \mathcal{E}(f, h \mid f_{ta}^*)}{\inf_{\boldsymbol{f}\in\mathcal{F}_{so}^{\otimes T}} \mathcal{E}(\boldsymbol{f}, h \mid \boldsymbol{f}_{so}^*) + \mu/\nu} \leq \nu.$$

Furthermore, we say they are (ν, μ)-diverse over $\mathcal{F}_{ta}$ if above ratio is bounded for any true

[3]Note that the standard definition has a $1/N$ factor instead of $1/\sqrt{N}$. We use the variant so that $\hat{\mathfrak{G}}_N$ does not scale with N in most of the cases we consider and only reflects the complexity of the class.

target prediction functions $f_{ta}^* \in \mathcal{F}_{ta}$*, i.e.*

$$\sup_{f_{ta}^* \in \mathcal{F}_{ta}} \sup_{h \in \mathcal{H}} \frac{\inf_{f \in \mathcal{F}_{ta}} \mathcal{E}(f, h \mid f_{ta}^*)}{\inf_{\boldsymbol{f} \in \mathcal{F}_{so}^{\otimes T}} \mathcal{E}(\boldsymbol{f}, h \mid \boldsymbol{f}_{so}^*) + \mu/\nu} \leq \nu.$$

When it is clear from the context, we denote $\mathcal{E}(f, h \mid f_{ta}^*)$ and $\mathcal{E}(\boldsymbol{f}, h \mid \boldsymbol{f}_{so}^*)$ by $\mathcal{E}_{ta}(f, h)$ and $\mathcal{E}_{so}(\boldsymbol{f}, h)$, respectively. We will call ν the transfer component and μ, the bias introduced by transfer. The definition of transferable links the generalization error between source tasks and the target task as shown in Theorem 2.1. The proof can be found in Appendix 2.A.

Theorem 2.1. *If source tasks* $f_1^*, \ldots, f_T^*$ *are* (ν, μ)*-diverse over* $\mathcal{F}_{ta}$*, then for any* $f_{ta}^* \in \mathcal{F}_{ta}$*, we have*

$$\mathcal{E}_{ta}(\hat{f}_{ta}, \hat{h}) \leq \nu \mathcal{E}_{so}(\hat{\boldsymbol{f}}_{so}, \hat{h}) + \mu + \frac{\sqrt{2\pi}\hat{\mathfrak{G}}_{n_{ta}}\left(\mathcal{F}_{ta} \circ \hat{h}\right)}{\sqrt{n_{ta}}} + \sqrt{\frac{9 \ln(2/\delta)}{2n_{ta}}}.$$

The first term in Theorem 2.1 can be upper bounded using the standard excess error bound of Gaussian complexity. The benefit of representation learning is due to the decrease in the third term from $\hat{\mathfrak{G}}_{n_{ta}}(\mathcal{F}_{ta} \circ \mathcal{H})/\sqrt{n_{ta}}$ without representation learning to $\hat{\mathfrak{G}}_{n_{ta}}(\mathcal{F}_{ta} \circ \hat{h})/\sqrt{n_{ta}}$ in our case. For the problem with complicated representations, the former term can be extremely larger than the later one.

In the rest of the paper, we discuss when we can bound (ν, μ), for nonlinear and nonidentical $\mathcal{F}_{so}$ and $\mathcal{F}_{ta}$.

2.3 Negative transfer when source tasks are more complex

Before introducing the cases that allow transfer, we first look at a case where transfer is impossible.

Let the source task use a linear mapping following the shared representation and the target task directly learns the representation. In other words, f_{ta}^* is identical mapping and known to the learner. We further consider $\mathcal{F}_{so} = \{z \mapsto w^T z : w \in \mathbb{R}^p\}$ and $\mathcal{H} = \{x \mapsto Hx : H \in \mathbb{R}^{p \times d}\}$. The interesting case is when $p \ll d$. Let the optimal representation be H^* and the true prediction function for each source task be $w_1^*, \ldots, w_T^*$. In the best scenario, we assume that there is no noise in the source tasks and each source task collects as many samples as possible, such that we will have an accurate estimation on each $w_t^{*T} H^* \in \mathbb{R}^{p \times d}$ which we denote by W_t^*.

However, the hypothesis class given the information from source tasks are

$$\{H \in \mathbb{R}^{p\times d} : \exists w \in \mathbb{R}^p, w^T H = W_t \text{ for all } t = 1, \ldots, T\}.$$

As H^* is in the above class, any QH^* for some rotation matrix $Q \in \mathbb{R}^{p\times p}$ is also in the class. In other words, a non-reducible error of learnt representation is $\max_Q \|H^* - QH^*\|_2^2$ in the worst case.

More generally, we give a condition for negative transfer. Consider a single source task f_{so}^*. Let $\mathcal{H}_{so}^* := \arg\min_{h\in\mathcal{H}} \min_{f\in\mathcal{F}_{so}} \mathcal{E}_{so}(f,h)$ and $\mathcal{H}_{ta}^* := \arg\min_{h\in\mathcal{H}} \min_{f\in\mathcal{F}_{ta}} \mathcal{E}_{ta}(f,h)$. In fact, if one hopes to get a representation learning benefit with zero bias ($\mu = 0$), we need all $h \in \mathcal{H}_{so}^*$ to satisfy $\inf_{f\in\mathcal{F}_{ta}} \mathcal{E}_{ta}(f,h) = 0$, i.e. any representation that is optimal in the source task is optimal in the target task as well. Equivalently, we will need $\mathcal{H}_{so}^* \subset \mathcal{H}_{ta}^*$. As shown in Proposition 2.1, the definition of *transferable* captures the case well.

Proposition 2.1. *If there exists a $h' \in \mathcal{H}_{so}^*$ such that $h' \notin \mathcal{H}_{ta}^*$, then $\min_{f_{ta}} \mathcal{E}_{ta}(f_{ta}, h') > 0$. Furthermore, there is no $\nu < \infty$, such that f_{so}^* is $(\nu, 0)$-transferable to f_{ta}^*.*

Proof. The first statement is by definition. Plugging g' into the Definition 2.1, we will have $\nu = \infty$ □

The negative transfer happens when the optimal representation for source tasks may not be optimal for the target task. As the case in our linear example, this is a result of more complex $\mathcal{F}_{so}$, which allows more flexibility to reduce errors. This inspires us to consider the opposite case where $\mathcal{F}_{ta}$ is more complex than $\mathcal{F}_{so}$.

2.4 Source task as a representation

Before discussing more general settings, we first consider a single source task, which we refer to as *so*. Assume that the source task itself is a representation of the target task. Equivalently, the source task has a known prediction function $f_{so}^*(x) = x$. This is a commonly-used framework when we decompose a complex task into several simple tasks and use the output of simple tasks as the input of a higher-level task.

A widely-used transfer learning method, called offset learning, which assumes $f_{ta}^*(x) = f_{so}^*(x) + w_{ta}^*(x)$ for some offset function w_{ta}^* falls within the framework considered here. The offset method enjoys its benefits when w_{ta} has low complexity and can be learnt with few samples. It is worth mentioning that our setting covers a more general setting in Du et al. (2017), which assumes $f_{ta}^*(x) = G(f_{so}^*(x), w_{ta}^*(x))$ for some known transformation function G and unknown w_{ta}^*.

We show that a simple Lipschitz condition on $\mathcal{F}_{ta}$ gives us a bounded transfer-component.

Assumption 2.1 (Lipschitz assumption)**.** *Any $f_{ta} \in \mathcal{F}_{ta}$ is L-Lipschitz with respect to L_2 distance.*

Theorem 2.2. *If Assumption 2.1 holds, task f^*_{so} is $(L, 0)$-transferable to task f^*_{ta} and we have with a high probability,*

$$\mathcal{E}_{ta}(\hat{f}_{ta}, \hat{h}) = \tilde{\mathcal{O}}\left(\frac{L\hat{\mathfrak{G}}_{n_{so}}(\mathcal{H})}{\sqrt{n_{so}}} + \frac{\hat{\mathfrak{G}}_{n_{ta}}(\mathcal{F}_{ta} \circ \hat{h})}{\sqrt{n_{ta}}}\right).$$

Theorem 2.2 bounds the generalization error of two terms. The first term that scales with $1/\sqrt{n_{so}}$ only depends on the complexity of $\mathcal{H}$. Though the second term scales with $1/\sqrt{n_{ta}}$, it is easy to see that $\hat{\mathfrak{G}}_{n_{ta}}(\mathcal{F}_{ta} \circ \hat{h}) = \hat{\mathfrak{G}}_{n_{ta}}(\mathcal{F}_{ta}) \ll \hat{\mathfrak{G}}_{n_{ta}}(\mathcal{F}_{ta} \circ \mathcal{H})$ with the dataset $\{\hat{h}(x_{ta,i})\}_{i=1}^{n_{ta}}$.

2.4.1 General case

Previously, we assume that a single source task is a representation of the target task. Now we consider a more general case: there exist functions in the source prediction space that can be used as representations of the target task. Formally, we consider $\mathcal{F}_{ta} = \mathcal{F}'_{ta} \circ (\mathcal{F}^{\otimes m}_{so})$ for some $m > 0$ and some target-specific function space $\mathcal{F}'_{ta} : \mathcal{Y}^{\otimes m}_{so} \mapsto \mathcal{Y}_{ta}$. Note that $\mathcal{F}_{ta}$ is strictly larger than $\mathcal{F}_{so}$ when $m = 1$ and the identical mapping $x \mapsto x \in \mathcal{F}'_{ta}$. In practice, ResNet (Tai et al., 2017) satisfies the above property.

Assumption 2.2. *Assume any $f'_{ta} \in \mathcal{F}'_{ta}$ is L'-Lipschitz with respect to L_2 distance.*

Theorem 2.3. *If Assumption 2.2 holds and the source tasks are (ν, μ)-diverse over its own space $\mathcal{F}_{so}$, then we have*

$$\mathcal{E}_{ta}\left(\hat{f}_{ta}, \hat{h}\right) = \tilde{\mathcal{O}}\left(L'm\left(\frac{\nu\hat{\mathfrak{G}}_{Tn_{so}}(\mathcal{F}^{\otimes T}_{so} \circ \mathcal{H})}{\sqrt{Tn_{so}}} + \mu\right) + \frac{\hat{\mathfrak{G}}_{n_{ta}}\left(\mathcal{F}_{ta} \circ \hat{h}\right)}{\sqrt{n_{ta}}}\right).$$

It is shown in Tripuraneni et al. (2020) that $\hat{\mathfrak{G}}_{\sum_s n_s}(\mathcal{F}^{\otimes S}_{so} \circ \mathcal{H})$ can be bounded by $\tilde{\mathcal{O}}(\hat{\mathfrak{G}}_{Tn_{so}}(\mathcal{H}) + \sqrt{T}\hat{\mathfrak{G}}_{n_{so}}(\mathcal{F}_{so}))$. Thus, the first term scales with $\hat{\mathfrak{G}}_{Tn_{so}}(\mathcal{H})/\sqrt{Tn_{so}} + \hat{\mathfrak{G}}_{n_{so}}(\mathcal{F}_{so})/\sqrt{n_{so}}$. Again the common part shared by all the tasks decreases with $\sqrt{Tn_{so}}$, while the task-specific part scales with $\sqrt{n_{so}}$ or $\sqrt{n_{ta}}$.

2.4.2 Applications to deep neural networks

Theorem 2.3 has a broad range of applications. In the rest of the section, we discuss its application to deep neural networks, where the source tasks are transferred to a target task with a deeper network.

We first introduce the setting for deep neural network prediction function. We consider the regression problem with $\mathcal{Y}_{so} = \mathcal{Y}_{ta} = \mathbb{R}$ and the representation space $\mathcal{Z} \subset \mathbb{R}^p$. A depth-$K$ vector-valued neural network is denoted by

$$f(\mathbf{x}) = \sigma\left(\mathbf{W}_K\left(\sigma\left(\ldots\sigma\left(\mathbf{W}_1\mathbf{x}\right)\right)\right)\right),$$

where each $\mathbf{W}_k$ is a parametric matrix of layer k and σ is the activation function. For simplicity, we let $\mathbf{W}_k \in \mathbb{R}^{p\times p}$ for $k = 1, \ldots K$. The class of all depth-K neural network is denoted by $\mathcal{M}_K$. We denote the linear class by $\mathcal{L} = \{x \mapsto \alpha^T x + \beta : \forall\alpha \in \mathbb{R}^p, \|\alpha\|_2 \leq M(\alpha)\}$ for some $M(\alpha) > 0$. We also assume

$$\max\{\|W_k\|_\infty, \|W_k\|_2\|\} \leq M(k),$$

where $\|\cdot\|_\infty$ and $\|\cdot\|_2$ are the infinity norm and spectral norm. We assume any $z \in \mathcal{Z}$, $\|z\|_\infty \leq D_Z$.

Deeper network for the target task. We now consider the source task with prediction function of a depth-K_{so} neural network followed by a linear mapping and target task with depth-K_{ta} neural network. We let $K_{ta} > K_{so}$. Then we have

$$\mathcal{F}_{ta} = \mathcal{L} \circ \mathcal{M}_{K_{ta}-K_{so}} \circ \mathcal{M}_{K_{so}} \text{ and } \mathcal{F}_{so} = \mathcal{L} \circ \mathcal{M}_{K_{so}}.$$

Using the fact that $\mathcal{M}_1 = \sigma(\mathcal{L}^{\otimes p})$, we can write $\mathcal{F}_{ta}$ as $\mathcal{L} \circ \mathcal{M}_{K_{ta}-K_{so}-1} \circ \sigma \circ (\mathcal{F}_{so}^{\otimes p})$. Thus, we can apply Theorem 2.3 and the standard Gaussian complexity bound for DNN models, which gives us Corollary 2.1.

Corollary 2.1. *Let $\mathcal{F}_{so}$ be depth-K_{so} neural network and $\mathcal{F}_{ta}$ be depth-K_{ta} neural network. If source tasks are $(\nu, 0)$-diverse over $\mathcal{F}_{so}$, we have*

$$\mathcal{E}_{ta}\left(\hat{f}_{ta}, \hat{h}\right) = \\ \tilde{\mathcal{O}}\left(p\nu M(\alpha)\Pi_{k=1}^{K_{ta}} M(k)\left(\frac{\hat{\mathfrak{G}}_{Tn_{so}}(\mathcal{H})}{\sqrt{Tn_{so}}} + \frac{D_Z\sqrt{K_{so}}}{\sqrt{n_{so}}}\right) + \frac{D_Z\sqrt{K_{ta}}\cdot M(\alpha)\Pi_{k=1}^{K_{ta}} M(k)}{\sqrt{n_{ta}}}\right). \quad (2.1)$$

Note that the terms that scales with $1/\sqrt{n_{so}}$ and $1/\sqrt{n_{ta}}$ have similar coefficients

$M(\alpha)\Pi_{k=1}^{K_{ta}} M(k)$, which do not depends on the complexity of $\mathcal{H}$. The term that depends on $\hat{\mathfrak{G}}_{Tn_{so}}(\mathcal{H})$ scales with $1/\sqrt{Tn_{so}}$ as we expected.

2.5 Diversity of non-linear function classes

While Corollary 2.1 considers the nonlinear DNN prediction function space under a diversity condition, it is not clearly understood how diversity can be achieved for nonlinear spaces. In this section, we first discuss a specific non-linear prediction function space and show a fundamental barrier to achieving diversity. Then we extend our result to general function classes by connecting eluder dimension and diversity. We end the section with positive results for achieving diversity under a generalized rank condition.

We consider a subset of depth-1 neural networks with ReLu activation function: $\mathcal{F} = \{x \mapsto [\langle x, w\rangle - (1-\epsilon/2)]_+ : \|x\|_2 \leq 1, \|w\| \leq 1\}$ for some $\epsilon > 0$. Our lower bound construction is inspired by the similar construction in Theorem 5.1 of Dong et al. (2021).

Theorem 2.4. *Let $\mathcal{T} = \{f_1, \ldots, f_T\}$ be any set of depth-1 neural networks with ReLu activation in $\mathcal{F}$. For any $\epsilon > 0$, if $T \leq 2^{d\log(1/\epsilon)-1}$, there exists some representation $h^*, h' \in \mathcal{H}$, some distribution P_X and a target function $f_{ta}^* \in \mathcal{F}$, such that*

$$\inf_{\boldsymbol{f}\in\mathcal{F}^{\otimes T}} \mathcal{E}_{so}(\boldsymbol{f}, h') = 0, \quad \textit{while } \inf_{f\in\mathcal{F}} \mathcal{E}_{ta}(f, h') = \epsilon^2/32.$$

Theorem 2.4 implies that we need at least $\Omega(2^d \log(1/\epsilon))$ source tasks to achieve diversity. Otherwise, we can always find a set of source tasks and a target task such that the generalization error in source tasks are minimized to 0 while that in target task is $\epsilon^2/32$. Though ReLu gives us a more intuitive result, we do show that similar lower bounds can be shown for other popular activation functions, for example, sigmoid function (see Appendix 2.A).

2.5.1 Lower bound using eluder dimension

We extend our result by considering a general function space $\mathcal{F}$ and build an interesting connection between diversity and eluder dimension (Russo and Van Roy, 2013). We believe we are the first to notice a connection between eluder dimension and transfer learning.

Eluder dimension has been used to measure the sample complexity in the Reinforcement Learning problem. It considers the minimum number of *inputs*, such that any two functions evaluated similarly at these inputs will also be similar at any other input. However, diversity considers the minimum number of functions such that any two representations with similar

outputs for these functions will also have similar output for any other functions. Thus, there is a kind of duality between eluder dimension and diversity.

We first formally define eluder dimension. Let $\mathcal{F}$ be a function space with support $\mathcal{X}$ and let $\epsilon > 0$.

Definition 2.2 ($(\mathcal{F}, \epsilon)$-dependence and eluder dimension (Osband and Roy (2014)))**.** *We say that $x \in \mathcal{X}$ is $(\mathcal{F}, \epsilon)$-dependent on $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ iff*

$$\forall f, \tilde{f} \in \mathcal{F}, \quad \sum_{i=1}^{n} \left\| f(x_i) - \tilde{f}(x_i) \right\|_2^2 \leq \epsilon^2 \Longrightarrow \|f(x) - \tilde{f}(x)\|_2 \leq \epsilon.$$

We say $x \in \mathcal{X}$ is $(\mathcal{F}, \epsilon)$-independent of $\{x_1, \ldots, x_n\}$ iff it does not satisfy the definition of dependence.

The eluder dimension $\dim_E(\mathcal{F}, \epsilon)$ is the length of the longest possible sequence of elements in $\mathcal{X}$ such that for some $\epsilon' \geq \epsilon$ every element is $(\mathcal{F}, \epsilon')$-independent of its predecessors.

We slightly change the definition and let $\dim_E^s(\mathcal{F}, \epsilon)$ be the shortest sequence such that any $x \in \mathcal{X}$ is $(\mathcal{F}, \epsilon)$-dependent on the sequence.

Although $dim_E^s(\mathcal{F}, \epsilon) \leq dim_E(\mathcal{F}, \epsilon)$, it can be shown that in many regular cases, say linear case and generalized linear case, $dim_E(\mathcal{F}, \epsilon)$ is only larger then $dim_E^s(\mathcal{F}, \epsilon)$ up to a constant.

Definition 2.3 (Dual class)**.** *For any $\mathcal{F} : \mathcal{X} \mapsto \mathcal{Y}$, we call $\mathcal{F}^* : \mathcal{F} \mapsto \mathcal{Y}$ its dual class iff.* $\mathcal{F}^* = \{g_x : g_x(f) = f(x), \forall x \in \mathcal{X}\}$.

Theorem 2.5. *For any function class $\mathcal{F} : \mathcal{X} \mapsto \mathbb{R}$, and some $\epsilon > 0$, let $\mathcal{F}^*$ be the dual class of $\mathcal{F}$. Let $d_E = \dim_E^s(\mathcal{F}^*, \epsilon)$. Then for any sequence of tasks $f_1, \ldots, f_t$, $t \leq d_E - 1$, there exists a task $f_{t+1} \in \mathcal{F}$ such that for some data distribution P_X and two representations h, h^*,*

$$\frac{\inf_{f'_{t+1} \in \mathcal{F}} \mathbb{E}_X \|f'_{t+1}(h(X)) - f_{t+1}(h^*(X))\|_2^2}{\frac{1}{t} \inf_{f'_1, \ldots, f'_t} \sum_{i=1}^{t} \mathbb{E}_X \|f'_i(h(X)) - f_i(h^*(X))\|_2^2} \geq t/2.$$

Theorem 2.5 formally describes the connections between eluder dimension and diversity. To interpret the theorem, we first discuss what are good source tasks. Any T source tasks that are diverse could transfer well to a target task if the parameter ν could be bounded by some fixed value that is not increasing with T. For instance, in the linear case, $\nu = \mathcal{O}(d)$ no matter how large T is. While for a finite function class, in the worst case, ν will increase with T before T reaches $|\mathcal{F}|$. Theorem 2.5 states that if the eluder dimension of the dual space is at least d_E, then ν scales with T until T reaches d_E, as if the function space $\mathcal{F}$ is discrete with d_E elements.

Note that this result is consistent with what is shown in Theorem 2.9 as eluder dimension of the class discussed above is lower bounded by $\Omega(2^d)$ as well (Li et al., 2021).

2.5.2 Upper bound using approximate generalized rank

Though we showed that diversity is hard to achieve in some nonlinear function class, we point out that diversity can be easy to achieve if we restrict the target prediction function such that they can be realized by linear combinations of some known basis. Tripuraneni et al. (2020) has shown that any source tasks $f_1, \ldots, f_T \in \mathcal{F}$ are $(1/T, \mu)$-diverse over the space $\{f \in \mathcal{F} : \exists \tilde{f} \in \text{conv}(f_1, \ldots, f_T) \text{ such that } \sup_z \|f(z) - \tilde{f}(z)\| \leq \mu\}$, where $\text{conv}(f_1, \ldots, f_T)$ is the convex hull of $\{f_1, \ldots, f_T\}$. This can be characterized by a complexity measure, called generalized rank. Generalized rank is the smallest dimension required to embed the input space such that all hypotheses in a function class can be realizable as halfspaces. Generalized rank has close connections to eluder dimension. As shown in Li et al. (2021), eluder dimension can be upper bounded by generalized rank.

Definition 2.4 (Approximate generalized rank with identity activation)**.** *Let* $\mathcal{B}_d(R) \coloneqq \{x \in \mathbb{R}^d \mid \|x\|_2 \leq R\}$. *The* μ*-approximate* $\text{id-rk}(\mathcal{F}, R)$ *of a function class* $\mathcal{F} : \mathcal{X} \mapsto \mathbb{R}$ *at scale* R *is the smallest dimension* d *for which there exists mappings* $\phi : \mathcal{X} \mapsto \mathcal{B}_d(1)$ *and* $w : \mathcal{F} \mapsto \mathcal{B}_d(R)$ *such that*

$$\text{for all } (x, f) \in \mathcal{X} \times \mathcal{F} : |f(x) - \langle w(f), \phi(x) \rangle| \leq \mu.$$

Proposition 2.2. *For any* $\mathcal{F}$ *with* μ*-approximate* $\text{id-rk}(\mathcal{F}, R) \leq d_i$ *for some* $R > 0$*, there exists no more than* d_i *functions,* $f_1, \ldots, f_{d_i}$*, such that* $w(f_1), \ldots w(f_{d_i})$ *span* $\mathbb{R}^{d_{d_i}}$*. Then* $f_1, \ldots, f_{d_i}$ *are* (d_i, μ)*-diverse over* $\mathcal{F}$*.*

Proposition 2.2 is a direct application of Lemma 7 in Tripuraneni et al. (2020). Upper bounding id-rank is hard for general function class. However, this notation can be useful for those function spaces with a known set of basis functions. For example, any function space $\mathcal{F}$ that is square-integrable has a Fourier basis. Though the basis is an infinite set, we can choose a truncation level d such that the truncation errors of all functions are less than μ. Then the hypothesis space $\mathcal{F}$ has μ-approximate id-rank less than d.

2.6 Experiments

In this section, we use simulated environments to evaluate the actual performance of representation learning on DNNs trained with gradient-based optimization methods. We also test the impact of various hyperparameters. Our results indicate that even though our theory is shown to hold for ERM estimators, their qualitative theoretical predictions still hold when Adam is applied and the global minima might not be found. Before we introduce our experimental setups, we discuss a difference between our theories and experiments: our

experiments use a single regression task with multiple outputs as the source task, while our analyses are built on multiple source tasks.

2.6.1 Diversity of problems with multiple outputs

Though we have been discussing achieving diversity from multiple source tasks, we may only have access to a single source task in many real applications, for example, ImageNet, which is also the case in our simulations. In fact, we can show that diversity can be achieved by a single multiclass classification or multivariate regression source task. The diversity for the single multi-variate regression problem with L2 loss is trivial as the loss function is decomposable. For multi-class classification problems or regression problems with other loss functions, we will need some assumptions on boundedness and continuous. We refer the readers to Appendix 2.A for details.

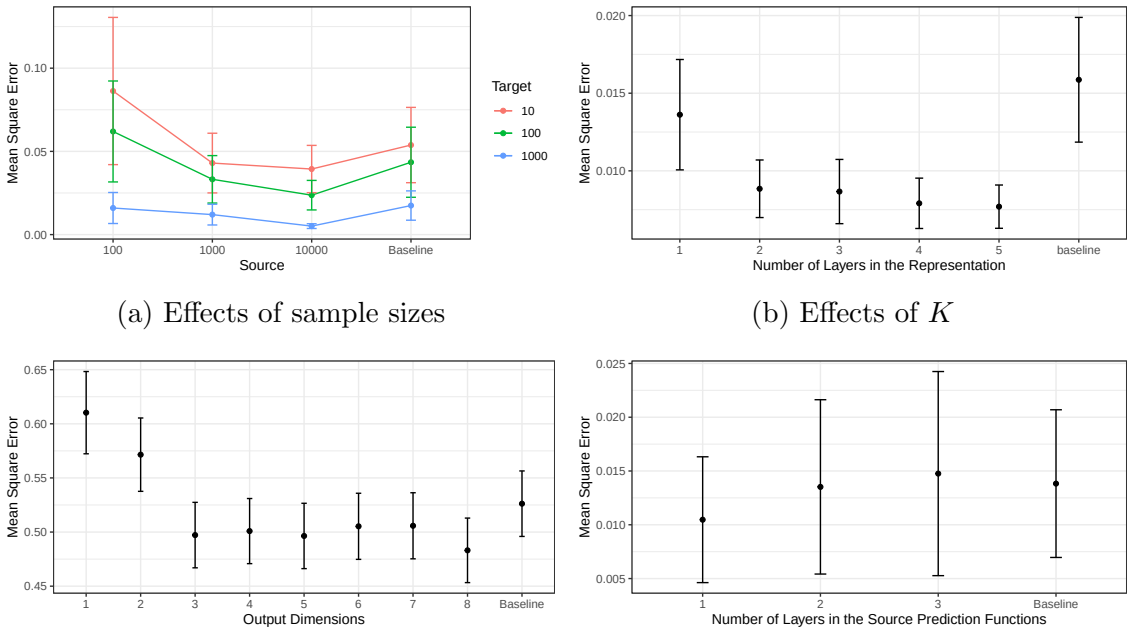
2.6.2 Experiments setup

Our four experiments are designed for the following goals. The first two experiments (Figure 2.1, a and b) target on the actual dependence on n_{so}, n_{ta}, K_{so} and K_{ta} in Equation (2.1) that upper bounds the errors of DNN prediction functions. Though the importance of diversity has been emphasized by various theories, no one has empirically shown its benefits over random tasks selection, which we explore in our third experiment (Figure 2.1, c). Our fourth experiment (Figure 2.1, d) verifies the theoretical negative results of nonlinearity of source prediction functions showed in Theorem 2.4 and 2.5.

Though the hyper-parameters vary in different experiments, our main setting can be summarized below. We consider DNN models of different layers for both source and target tasks. The first K layers are the shared representation. The source task is a multi-variate regression problem with output dimension p and K_{so} layers following the representation. The target task is a single-output regression problem with K_{ta} layers following the representation. We used the same number of units for all the layers, which we denote by n_u. A representation is first trained on the source task using n_{so} random samples and is fixed for the target task, trained on n_{ta} random samples. In contrast, the baseline method trains the target task directly on the same n_{ta} samples without the pretrained network. We use Adam with default parameters for all the training. We use MSE (Mean Square Error) to evaluate the performance under different settings.

2.6.3 Results

Figure 2.1 summaries our four experiments, with all the Y axes representing the average MSE's of 100 independent runs with an error bar showing the standard deviation. The X axis varies depending on the goal of each experiment. In subfigure (a), we test the effects of the numbers of observations for both source and target tasks, while setting other hyperparameters by default values. The X axis represents n_{so} and the colors represents n_{ta}. In subfigure (b), we test the effects of the number of shared representation layers K. To have comparable MSE's, we keep the sum $K + K_{ta} = 6$ and run $K = 1, \ldots, 5$ reflected in the X axis, while keeping $K_{so} = 1$. In subfigure (c), we test the effects of diversity. The larger p we have, the more diverse the source task is. We keep the actual number of observations $n_{so} \cdot p = 4000$ for a fair comparison. Lastly, in subfigure (d), we test whether diversity is hard to achieve when the source prediction function is nonlinear. The X axis is the number of layers in source prediction function K_{so}. The nonlinearity increases with K_{so}. We run $K_{so} = 1, 2, 3$ and add an activation function right before the output such that the function is nonlinear even if $K_{so} = 1$.



(a) Effects of sample sizes

(b) Effects of K

(c) Effects of diversity

(d) Effects of K_{so}

Figure 2.1: **(a)** Effects of the numbers of observations for both source (n_{so}) and target tasks (n_{ta}). **(b)** Effects of the number of shared representation layers K. **(c)** Effects of diversity determined by the output dimensions p. We keep the actual number of observations $n_{so} \cdot p = 4000$. **(d)** Effects of nonlinearity of the source prediction function. Higher K_{so} indicates higher nonlinearity.

In Figure 2.1 (a), MSE's decrease with larger numbers of observations in both source and target tasks, while there is no significant difference between $n_{so} = 1000$ and 10000. The baseline method without representation learning performs worst and it performs almost the same when n_{ta} reaches 1000. In (b), there are positive benefits for all different numbers of the shared layers and the MSE is the lowest at 5 shared layers. As shown in Figure 2.1 (c), the MSE's and their variances are decreasing when the numbers of outputs increase, i.e., higher diversity. Figure 2.1 (d) shows that there is no significant difference between baseline and $K_{so} = 1, 2, 3$. When $K_{so} = 2, 3$ there is a negative effects.

2.7 Discussion

In this chapter, we studied representation learning beyond linear prediction functions. We showed that the learned representation can generalize to tasks with multi-layer neural networks as prediction functions as long as the source tasks use linear prediction functions. We show the hardness of being diverse when the source tasks are using nonlinear prediction functions by giving a lower bound on the number of source tasks in terms of eluder dimension. We further give an upper bound depending on the generalized rank. This generalization works for source and target tasks with distinct label spaces. For example, regression task can generalize to classification tasks with a logistic regression model (see Lemma 2.4 for a change of loss function result).

Different prediction functions for source and target tasks as we studied in this chapter introduce asymmetry in terms of the model design. As we discussed, we may add more layers of neural network on top of the learned representation function. We argue that this asymmetry exists in many real-world applications, e.g. vision tasks, when we target at downstream task generalization. In practice, when we aim at improving a set of source tasks performance, where asymmetry can be an issue, we may choose the same source task prediction function class and make it large enough to include the functions of interest for all the tasks.

Future works. Focusing on future work, we need better tools to understand the recently proposed complexity measure generalized rank. Our analyses rely on the ERM, while in practice as well as in our simulations, gradient-based optimization algorithms are used. Further analyses on the benefits of representation learning that align with practice on the choice of optimization method should be studied. Our analyses assume arbitrary source tasks selection, while, in real applications, we may have limited data from groups that are under-represented, which may lead to potential unfairness. Algorithm that calibrates this

potential unfairness should be studied.

Another direction of future work is to propose reasonable assumptions for a guarantee of generalization even when the source prediction functions are nonlinear. Possibly the interesting functions are within a subset of the target prediction function class, which makes generalization easier.

In this work, we consider provided and fixed source tasks. In practice, many empirical methods (Graves et al., 2017) are proposed to adaptively select unknown source tasks. An interesting direction is to ask whether adaptively task selection could achieve the optimal level of diversity and to design algorithms for that purpose.

2.A Missing Proofs

Proof of Theorem 2.1

Proof. Note that the second phase is to find the best function within the class $\mathcal{F}_{ta} \circ \hat{h}$. We first apply the standard bounded difference inequality (Bartlett and Mendelson, 2002) as shown in Theorem 2.6.

Theorem 2.6 (Bartlett and Mendelson (2002))**.** *With a probability at least* $1-\delta$*,*

$$\sup_{f \in \mathcal{F}_{ta}} |R_{ta}(f \circ \hat{h}) - \hat{R}_{ta}(f \circ \hat{h})| \leq \frac{\sqrt{2\pi}\hat{\mathfrak{G}}_{n_{ta}}(\mathcal{F}_{ta} \circ \hat{h})}{\sqrt{n_{ta}}} + \sqrt{\frac{9\ln(2/\delta)}{2n_{ta}}} =: \epsilon(\hat{h}, n_{ta}, \delta),$$

furthermore, the total generalization error can be upper bounded by

$$\mathcal{E}_{ta}(\hat{f}_{ta} \circ \hat{h}) \leq \underbrace{\inf_{f \in \mathcal{F}_{ta}} \mathcal{E}_{ta}(f, \hat{h})}_{\text{approximation error}} + \underbrace{\epsilon(\hat{h}, n_{ta}, \delta)}_{\text{generalization error over } \mathcal{F}_{ta} \circ \hat{h}}. \tag{2.2}$$

Theorem 2.6 is stated in terms of Gaussian complexity. It is more common to use Radamecher complexity, which can be upper bounded by $\sqrt{2\pi}$ of the corresponding Gaussian complexity. For the generalization bound in terms of Rademacher complexity, Theorem 26.5 of Shalev-Shwartz and Ben-David (2014) has a full proof. Then recall that $\mathcal{Y}_t$ and $\mathcal{Y}_{ta} \subset [0, 1]$, we get rid of the loss function by the contraction lemma, which leads to Theorem 2.6. The result follows by the definition,

$$\inf_{f \in \mathcal{F}_{ta}} \mathcal{E}_{ta}(f, \hat{h}) \leq \frac{\inf_{f_{ta} \in \mathcal{F}_{ta}} \mathcal{E}_{ta}(f_{ta}, \hat{h})}{\inf_{\boldsymbol{f}_{so} \in \mathcal{F}_{so}^{\otimes S}} \mathcal{E}_{so}(\boldsymbol{f}_{so}, \hat{h}) + \mu/\nu} (\mathcal{E}_{so}(\hat{\boldsymbol{f}}_{so}, \hat{h}) + \mu/\nu) \leq \nu \mathcal{E}_{so}(\hat{\boldsymbol{f}}_{so}, \hat{h}) + \mu.$$

□

Proof of Theorem 2.2

Proof. To show f_{so}^* is $(L, 0)$-transferable to f_{ta}^*, we bound the approximation error of the target task given any fixed $h \in \mathcal{H}$.

$$\begin{aligned}
&\mathcal{E}_{ta}(f_{ta}^*, h) \\
&= \mathbb{E}_{X,Y}\left[l_{ta}(f_{ta}^* \circ h(X), Y) - l_{ta}(f_{ta}^* \circ h^*(X), Y)\right] \\
&= \mathbb{E}_X \| f_{ta}^* \circ h(X) - f_{ta}^* \circ h^*(X)\|_2^2 \\
&\leq L\mathbb{E}_X \|h(X) - h^*(X)\|_2^2. \qquad (2.3)
\end{aligned}$$

Now using Assumption 2.1, we have

$$\begin{aligned}
\mathcal{E}_{so}(h) &= \mathbb{E}_{X,Y}[l_{so}(h(X), Y) - l_{so}(h^*(X), Y)] \\
&= E_X \|h^*(X) - h(X)\|_2^2
\end{aligned}$$

Combined with Equation (2.3), we have $\sup_{h\in\mathcal{H}}[\mathcal{E}_{ta}(f_{ta}^*, h)/\mathcal{E}_{so}(h)] \leq L$.

Firstly, using Theorem 2.6 on source tasks solely, we have $\mathcal{E}_{so}(\hat{h}) = \tilde{\mathcal{O}}(\hat{\mathfrak{G}}_{n_{so}}(\mathcal{G})/\sqrt{n_{so}})$. Definition 2.1 gives us

$$\inf_{f\in\mathcal{F}_{\text{ta}}} \mathcal{E}_{\text{ta}}(f, \hat{h}) \leq L\mathcal{E}_{so}(\hat{h}) = \tilde{\mathcal{O}}(L\hat{\mathfrak{G}}_{n_{so}}(\mathcal{G})/\sqrt{n_{so}}).$$

Combined with (2.2), we have

$$\mathcal{E}_{ta}\left(\hat{f}_{ta}, \hat{h}\right) = \tilde{\mathcal{O}}\left(L\frac{\hat{\mathfrak{G}}_{n_{so}}(\mathcal{G})}{\sqrt{n_{so}}} + \frac{\hat{\mathfrak{G}}_{n_{\text{ta}}}\left(\mathcal{F}_{ta} \circ \hat{h}\right)}{\sqrt{n_{ta}}}\right).$$

□

Proof of Theorem 2.3

Proof. Let $f_{ta}^*(x) = f_{ta}'^*(f_1^* \circ h^*(x), \ldots f_m^* \circ h^*(x))$. Define new tasks $t_1, \ldots t_m$. Each t_i has the prediction function f_i^*.

By Definition 2.1, the source tasks are (ν, μ)-transferable to each t_i. By Theorem 2.1, we have

$$\inf_{f_i\in\mathcal{F}_{so}} \mathcal{E}_{t_i}(f_i, \hat{h}) \leq \nu\mathcal{E}_{so}(\hat{\boldsymbol{f}}_{so}, \hat{h}) + \mu.$$

Since we use L_2 loss,

$$\inf_{f_i \in \mathcal{F}_{so}} \mathbb{E}_X \|f_i \circ \hat{h}(X) - f_i^* \circ h^*(X)\|_2^2 = \inf_{f_i \in \mathcal{F}_{so}} \mathcal{E}_{t_i}(f_i, \hat{h}) \leq \nu \mathcal{E}_{so}(\hat{\boldsymbol{f}}_{so}, \hat{h}) + \mu.$$

As this holds for all $i \in [m]$, we have

$$\inf_{f_1,\ldots,f_m \in \mathcal{F}_{so}} \mathbb{E}_X \|(f_1 \circ \hat{h}(X), \ldots, f_m \circ \hat{h}(X)) - (f_1^* \circ h^*(X), \ldots, f_m^* \circ \hat{h}^*(X))\|_2^2 \leq m(\nu \mathcal{E}_{so}(\hat{\boldsymbol{f}}_{so}, \hat{h}) + \mu).$$

Using Assumption 2.2, we have

$$\inf_{f \in \mathcal{F}_{ta}} \mathcal{E}_{ta}(f, \hat{h}) \leq L' m(\nu \mathcal{E}_{so}(\hat{\boldsymbol{f}}_{so}, \hat{h}) + \mu).$$

Theorem 2.3 follows by plugging this into (2.2). □

Proof of Corollary 2.1 This section we prove Corollary 2.1 using Theorem 2.3, the standard bound for Gaussian complexity of DNN model and the Gaussian complexity decomposition from Tripuraneni et al. (2020).

The following theorem bounds the Rademacher complexity of a deep neural network model given an input dataset $\boldsymbol{X}_N = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^T \in \mathbb{R}^{N \times d}$.

Theorem 2.7 (Golowich et al. (2018))**.** *Let σ be a 1-Lipschitz activation function with $\sigma(0) = 0$. Recall that $\mathcal{M}_K$ is the depth K neural network with d-dimensional output with bounded input $\|x_{ji}\| \leq D_Z$ and $\|W_k\|_\infty \leq M(k)$ for all $k \in [K]$. Recall that $\mathcal{L} = \left\{x \mapsto \alpha^T x + \beta : \forall \alpha \in \mathbb{R}^p, \|\alpha\|_2 \leq M(\alpha)\right\}$ is the linear class following the depth-K neural network. Then,*

$$\mathfrak{R}_n(\mathcal{L} \circ M_K; \boldsymbol{X}_N) \leq \frac{2 D_Z \sqrt{K + 2 + \log d} \cdot M(\alpha) \Pi_{k=1}^K M(k)}{\sqrt{n}}.$$

Since for any function class $\mathcal{F}$, $\hat{\mathfrak{G}}_n(\mathcal{F}) \leq 2\sqrt{\log n} \cdot \hat{\mathfrak{R}}_n(\mathcal{F})$, we also have the bound for the Gaussian complexity under the same conditions.

Applying Theorem 2.7, we have an upper bound for the second term in Theorem 2.3:

$$\frac{\hat{\mathfrak{G}}_{n_{ta}}\left(\mathcal{F}_{ta} \circ \hat{h}\right)}{\sqrt{n_{ta}}} \leq \frac{2 D_Z \sqrt{\log(n_{ta})} \sqrt{K_{ta} + 2 + \log(d)} M_\alpha \Pi_{k=1}^{K_{ta}} M(k)}{\sqrt{n_{ta}}} = \tilde{\mathcal{O}}\left(\frac{D_Z \sqrt{K_{ta}} M(\alpha) \Pi_{k=1}^{K_{ta}} M(k)}{\sqrt{n_{ta}}}\right).$$

It only remains to bound $\hat{\mathfrak{G}}_{T n_{so}}\left(\mathcal{F}_{so}^{\otimes T} \circ \mathcal{H}\right) / \sqrt{T n_{so}}$ in Theorem 2.3. To proceed, we

introduce the decomposition theorem for Gaussian complexity (Tripuraneni et al., 2020).

Theorem 2.8 (Theorem 7 in Tripuraneni et al. (2020))**.** *Let the function class $\mathcal{F}$ consist of functions that are $L(\mathcal{F})$-Lipschitz and have boundedness parameter $D_X = \sup_{f,f',x,x'} \|f(x) - f'(x')\|_2$. Further, define $\mathcal{Q} = \{h(\bar{X}) : h \in \mathcal{H}, \bar{X} \in \cup_{j=1}^{T}\{X_j\}\}$. Then the Gaussian complexity of the function class $\mathcal{F}^{\otimes T}(\mathcal{H})$ satisfies,*

$$\hat{\mathfrak{G}}_{\mathbf{X}}\left(\mathcal{F}^{\otimes T}(\mathcal{H})\right) \leq \frac{4D_{\mathbf{X}}}{(nT)^{3/2}} + 128C\left(\mathcal{F}^{\otimes T}(\mathcal{H})\right) \cdot \log(nT),$$

where $C\left(\mathcal{F}^{\otimes t}(\mathcal{H})\right) = L(\mathcal{F})\hat{\mathfrak{G}}_{\mathbf{X}}(\mathcal{H}) + \max_{\mathbf{q}\in\mathcal{Q}} \hat{\mathfrak{G}}_{\mathbf{q}}(\mathcal{F})$.

With Theorem 2.8 applied, we have

$$\frac{\hat{\mathfrak{G}}_{Tn_{so}}\left(\mathcal{F}_{so}^{\otimes T} \circ \mathcal{H}\right)}{\sqrt{Tn_{so}}} \leq \frac{8D_X}{(Tn_{so})^2} + \frac{128\left(L(\mathcal{F}_{so})\hat{\mathfrak{G}}_{Tn_{so}}(\mathcal{H}) + \max_{\mathbf{q}\in\mathcal{Q}} \hat{\mathfrak{G}}_{\mathbf{q}}(\mathcal{F}_{so})\right) \cdot \log(Tn_{so})}{\sqrt{Tn_{so}}}. \quad (2.4)$$

The second term relies on the Lipschitz constant of DNN, which we bound with the following lemma. Similar results are given by Scaman and Virmaux (2018); Fazlyab et al. (2019).

Lemma 2.1. *If the activation function is 1-Lipschitz, any function in $\mathcal{L} \circ \mathcal{M}_K$ is $M(\alpha)\Pi_{k=1}^{K} M(k)$-Lipschitz with respect to L_2 distance.*

Proof. The linear mapping $x \mapsto W_k x$ is $\|W_k\|_2$-Lipschitz. Combined with the Lipschitz of activation function we have $\sigma(W_k x)$ is also $\|W_k\|_2$-Lipschitz. Then the composition of different layers has Lipschitz constant $\Pi_k M_k$. The Lemma follows by adding the Lipschitz of the last linear mapping. □

Thus, we have

$$L(\mathcal{F}_{so}) \leq M(\alpha)\Pi_{k=1}^{K_{so}} M(k).$$

By Theorem 2.7,

$$\max_{\mathbf{q}\in\mathcal{Q}} \hat{\mathfrak{G}}_{\mathbf{q}}(\mathcal{F}_{so}) = \tilde{\mathcal{O}}\left(\frac{D_Z\sqrt{K_{so}} M(\alpha)\Pi_{k=1}^{K_{so}} M(k)}{\sqrt{n_{so}}}\right).$$

Plug the above two equations into (2.4), we have

$$\frac{\hat{\mathfrak{G}}_{Tn_{so}}\left(\mathcal{F}_{so}^{\otimes T} \circ \mathcal{H}\right)}{\sqrt{Tn_{so}}} = \tilde{\mathcal{O}}\left(\frac{D_X}{(Tn_{so})^2} + M(\alpha)\Pi_{k=1}^{K_{so}} M(k)\left(\frac{\hat{\mathfrak{G}}_{Tn_{so}}(\mathcal{H})}{\sqrt{Tn_{so}}} + \frac{D_Z\sqrt{K_{so}}}{\sqrt{n_{so}}}\right)\right),$$

where $D_X = \sup_{(h,f,x),(h',f',x')\in\mathcal{H}\times\mathcal{F}_{so}\times\mathcal{X}} \|h \circ f(x) - h' \circ f'(x')\|$.

Lemma 2.2. *The boundedness parameter D_X satisfies $D_X \leq D_Z M(\alpha)\Pi_{k=1}^{K_{so}} M(k)$.*

Proof. The proof is given by induction. Let r_k denote the vector-valued output of the k-th layer of the prediction function. First note that

$$D_X \leq 2 \sup_{f\in\mathcal{F}_{so}, z\in\mathcal{Z}} \|f(z)\|^2 \leq 2M(\alpha)\|r_{K_{so}}\|^2.$$

For each output of the k-th layer, we have

$$\|r_k\|^2 = \|\sigma(W_k r_{k-1})\|^2 \leq \|W_k r_{k-1}\|_2^2 \leq \|W_k\|_2^2 \|r_{k-1}\|_2^2,$$

where the first inequality is by the 1-Lipschitz of the activation function. By induction, we have

$$D_X \leq 2D_Z M(\alpha)\Pi_{k=1}^{K_{so}} M(k).$$

□

Recall that $\mathcal{F}_{ta} = \mathcal{L} \circ \mathcal{M}_{K_{ta}-K_{so}-1} \circ (\mathcal{F}_{so}^{\otimes p})$ and the Lipschitz constant $L' \leq M(\alpha)\Pi_{K_{so}+2}^{K_{ta}} M(k)$. Using Theorem 2.3 and apply Lemma 2.2, we have

$$\mathcal{E}_{ta}\left(\hat{f}_{ta}, \hat{h}\right) =$$
$$\tilde{\mathcal{O}}\left(p\nu\Pi_{k=K_{so}+2}^{K_{ta}} M(k)\left(M(\alpha)\Pi_{k=1}^{K_{so}} M(k)\left(\frac{\hat{\mathfrak{G}}_{Tn_{so}}(\mathcal{H})}{\sqrt{Tn_{so}}} + \frac{D_Z\sqrt{K_{so}}}{\sqrt{n_{so}}}\right)\right) + \frac{D_Z\sqrt{K_{ta}}\cdot M(\alpha)\Pi_{k=1}^{K_{ta}} M(k)}{\sqrt{n_{ta}}}\right).$$

Lower bound results for the diversity of depth-1 NN We first give the proof using ReLu activation function (Theorem 2.4), as the result is more intuitive before we extend the similar results to other activation functions.

Proof. As we consider arbitrary representation function and covariate distribution, for simplicity we write $X' = h^*(X)$ and $Y' = h(X)$.

We consider a subset of depth-1 neural networks with ReLu activation function: $\mathcal{F} = \{x \mapsto [\langle x, w\rangle - (1-\epsilon/4)]_+ : \|x\|_2 \leq 1, \|w\| \leq 1\}$. Let f_w be the function with parameter w. Consider $U \subset \{x : \|x\|_2 = 1\}$ such that $\langle u, v\rangle \leq 1-\epsilon$ for all $u, v \in U, u \neq v$.

Lemma 2.3. *For any $\mathcal{T} \subset \mathcal{F}$, $|\mathcal{T}| \leq \lfloor |U|/2 \rfloor$, there exists a $V \subset U$, $|V| \geq \lfloor |U|/2 \rfloor$ such that any $f \in \mathcal{T}$, $f(v) = 0$ for all $v \in V$.*

Proof. For any set $\mathcal{T}$, let $U_{\mathcal{T}} = \{u : \exists t \in \mathcal{T}, u \in \arg\max_{u\in U}\langle u, f_t\rangle\}$ be a subset of U. Thus, $|U_{\mathcal{T}}| \leq T \leq \lfloor |U|/2 \rfloor$. Let $V = U \setminus U_{\mathcal{T}}$.

For any $f \in \mathcal{T}$, let u_f be its closed point in U. Let v_f be its closed point in V. Let θ_f be the angle between u_f and v_f. By the definition of U, we have $cos(\theta_f) = \langle u_f, v_f \rangle \leq 1 - \epsilon$. We will show that $\langle f, v_f \rangle \leq 1 - \epsilon/4$.

Note that since $\langle f, v_f \rangle \leq \langle f, u_f \rangle$, we have the angle between f and v is larger than $\theta_f/2$. By the simple fact that $cos(\theta_f/2) \leq 1 - (1 - cos(\theta_f))/4$, we have $\langle f, v_f \rangle \leq 1 - \epsilon/4$. Thus, $f(v_f) = 0$ and $f(v) = 0$ for all $v \in V$. □

For any set of prediction functions in source tasks, let V be the set defined in the above lemma. Consider any $u \in U \setminus V$ and let $V' = U \setminus (V \cup u)$. By this construction, we have $f_u(u) = \epsilon/4$, while all $f \in \mathcal{T}, f(u) = 0$. Note that

$$\inf_{f \in \mathcal{F}} \frac{1}{|V'|} \sum_{x \in V'} (f_u(u) - f(x))^2 \geq \frac{|V'| - 1}{16|V'|} \epsilon^2 \geq \frac{1}{32} \epsilon^2,$$

while

$$\sum_{f \in \mathcal{T}} \frac{1}{|V'|} \sum_{x \in V'} (f(u) - f(x))^2 = 0.$$

Thus, we let $X' = u$ almost surely and Y' follows a uniform distribution over V'. This is true when the covariate distribution is the same as Y' and $h = x \mapsto x$ and $h^* = x \mapsto u$. Recalling the definition of diversity, we have

$$\inf_{f \in \mathcal{F}} \mathbb{E}_{X',Y'} (f_u(X') - f(Y'))^2 = \epsilon^2/32 \text{ and } \frac{1}{T} \sum_{f_t \in \mathcal{T}} \inf_{f'_t \in \mathcal{F}} \mathbb{E}_{X',Y'} (f_s(X') - f'_s(Y'))^2 = 0.$$

Note that the same result holds when the bias $b \leq -(1 - \epsilon/4)$. For general bounded $\|b\|_2 \leq 1$, one can add an extra coordinate in x as an offset. □

In Theorem 2.4, we show that in depth-1 neural network with ReLu activation function, we will need exponentially many source tasks to achieve diversity. Similar results can be shown for other non-linear activation functions that satisfies the following condition:

Assumption 2.3. *Let $\sigma : \mathbb{R} \mapsto \mathbb{R}$ be an activation function. We assume there exists $x_1, x_2 \in \mathbb{R}$, $x_1 > x_2$, such that $|\sigma(x_1)| \geq \sup_{x \leq x_2} |\sigma(x)| M$ for some $M > 0$.*

ReLu satisfies the assumption with any $M > 0$ for any $x_1 > 0$ and $x_2 \leq 0$. Also note that any continuous activation function that is lower bounded and increasing satisfies this assumption.

Theorem 2.9. *Let σ satisfies the above assumption with M for some x_1 and x_2. Let $\mathcal{F} = \{x \mapsto \sigma(8(x_1 - x_2)\langle x, w \rangle - 7x_1 + 8x_2)) : \|x\|_2 \leq 1, \|w\|_2 \leq 1\}$. Let $\mathcal{T} = \{f_1, \ldots, f_T\}$ be*

any set of depth-1 neural networks with ReLu activation in $\mathcal{F}$. If $T \leq 2^{d\log(2)-1}$, there exists some representation $h^*, h' \in \mathcal{H}$, some distribution P_X and a target function $f^*_{ta} \in \mathcal{F}$, such that

$$\frac{\inf_{f\in\mathcal{F}} \mathcal{E}_{ta}(f, h')}{\inf_{\boldsymbol{f}\in\mathcal{F}} \mathcal{E}_{so}(\boldsymbol{f}, h')} \geq \frac{(M-1)^2}{8}.$$

Proof. We follow the construction in the proof of Theorem 2.9, fix an $\epsilon = 1/2$ and let $U \subset \{x : \|x\|_2 = 1\}$ such that $\langle u, v\rangle \leq 1 - \epsilon$ for all $u, v \in U, u \neq v$.

For any source tasks set $\mathcal{T}$, let $U_{\mathcal{T}} = \{u : \exists t \in \mathcal{T}, u \in \arg\max_{u\in U}\langle u, f_t\rangle\}$ be a subset of U. Thus, $|U_{\mathcal{T}}| \leq T \leq \lfloor |U|/2 \rfloor$. Let $V = U \setminus U_{\mathcal{T}}$. For any $f \in \mathcal{T}$, $v \in V$, similarly to the previous argument, we have $\langle f, v\rangle \leq 1 - \epsilon/4 = 1/8$. Therefore, $\langle f, v\rangle \leq 1 - \epsilon/4 = 1/8 \leq x_2$.

For any set of prediction functions in source tasks, let V be the set defined in the above lemma. Consider any $u \in U \setminus V$ and let $V' = U \setminus (V \cup u)$. By this construction, we have $f_u(u) = \sigma(x_1)$, while all $f \in \mathcal{T}, f(u) = \sigma(x_2)$. Note that

$$\inf_{f\in\mathcal{F}} \frac{1}{|V'|}\sum_{x\in V'}(f_u(u) - f(x))^2 \geq \frac{|V'| - 1}{|V'|} \geq \frac{1}{2}(\sigma(x_1) - \sigma(x_2))^2,$$

while

$$\sum_{f\in\mathcal{T}} \frac{1}{|V'|}\sum_{x\in V'}(f(u) - f(x))^2 \leq 4\sup_{x\leq x_2}\sigma(x_2)^2.$$

Thus

$$\frac{\inf_{f\in\mathcal{F}} \frac{1}{|V'|}\sum_{x\in V'}(f_u(u) - f(x))^2}{\sum_{f\in\mathcal{T}} \frac{1}{|V'|}\sum_{x\in V'}(f(u) - f(x))^2} \geq \frac{(M-1)^2}{8}$$

□

Proof of Theorem 2.5

Proof. Since $\dim^s(\mathcal{F}^*)$ is at least d_E, for any set $\{f_1, \ldots, f_t\}$, there exists a f_{t+1} that is $(\mathcal{F}^*, \epsilon)$-independent of $\{f_1, \ldots, f_t\}$. By definition, we have

$$\exists x_1, x_2 \in \mathcal{X}, \sum_{i=1}^{t} \|f_i(x_1) - f_i(x_2)\|_2^2 \leq \epsilon^2, \text{ while } \|f_{t+1}(x_1) - f_{t+1}(x_2)\|_2^2 \geq \epsilon^2.$$

We only need to construct appropriate data distribution P_X and representation g, g^* to finish the proof. As we do not make any assumption on g, g^* and P_X, it would be simple to let $X_1 = g(X)$ and $X_2 = g^*(X)$.

We let the distribution of X_1 be the point mass on x_1. Let X_2 be the uniform distribution over $\{x_1, x_2\}$.

For the excess error of source tasks, we have

$$\begin{aligned}
&\inf_{f'_1,\ldots,f'_t} \sum_{i=1}^{t} \mathbb{E}_{X_1,X_2} \|f'_i(X_1) - f_i(X_2)\|_2^2 \\
&\leq \sum_{i=1}^{t} \mathbb{E}_{X_1,X_2} \|f_i(X_1) - f_i(X_2)\|_2^2 \\
&= \sum_{i=1}^{t} \frac{1}{2} \|f_i(x_1) - f_i(x_2)\|_2^2 \leq \frac{\epsilon^2}{2}.
\end{aligned}$$

For the excess error of the target task f_{t+1}, we have

$$\begin{aligned}
&\inf_{f'_{t+1} \in \mathcal{F}} \mathbb{E}_X \|f'_{t+1}(X_1) - f_{t+1}(X_2)\|_2^2 \\
&= \inf_{f'_{t+1} \in \mathcal{F}} [\frac{1}{2} \|f'_{t+1}(x_1) - f_{t+1}(x_2)\|_2^2 + \frac{1}{2} \|f'_{t+1}(x_1) - f_{t+1}(x_1)\|_2^2] \\
&\geq \inf_{a \in \mathbb{R}} [\frac{1}{2} \|a - f_{t+1}(x_2)\|_2^2 + \frac{1}{2} \|a - f_{t+1}(x_1)\|_2^2] \\
&= \frac{1}{4} \|f_{t+1}(x_1) - f_{t+1}(x_2)\|_2^2 \geq \frac{\epsilon^2}{4}.
\end{aligned}$$

The statement follows. □

Extending to general loss functions In all the above analyses, we assume the square loss function for both source and target tasks. We first show that diversity under square loss implies diversity under any convex loss function. Let $\nabla l(x, y)$ be the gradient of function $\nabla l(\cdot, y)$ evaluated at x.

Lemma 2.4. *Any task set $\mathcal{F}$ that is (ν, μ)-diverse over any prediction space under square loss is also $(\nu/c_1, \mu/c_1)$-diverse over the same space under loss l, if l is c_1 strongly-convex and for all $x \in \mathcal{X}$*

$$\mathbb{E}[\nabla l(g^*(X), Y) \mid X = x] = 0 \tag{2.5}$$

Proof. Using the definition of the strongly convex and (2.5),

$$\begin{aligned}
&\mathbb{E}_{X,Y}[l(f_t \circ h(X), Y) - l(f_t^* \circ h^*(X), Y)] \\
&\geq \mathbb{E}_{X,Y}[\nabla l \left(f_t^* \circ h^*(X), Y\right)^T \left(f_t^* \circ h^*(X) - f_t \circ h(X)\right) + c_1 \left\|f_t^* \circ h^*(X) - f_t \circ h(X)\right\|_2^2] \\
&= c_1 \mathbb{E}_{X,Y}[\|f_t^* \circ h^*(X) - f_t \circ h(X)\|_2^2],
\end{aligned}$$

which is the generalization error under the square loss. □

Note that Equation (2.5), is a common assumption made in various analyses of stochastic gradient descent (Jin et al., 2021b).

On the other direction, we show that any established diversity over the target task with square loss also implies the diversity over the same target task with any loss l if $\nabla^2 l \succ c_2 I$ for some $c_2 > 0$.

Lemma 2.5. *Any task set $\mathcal{F}$ that is (ν, μ)-diverse over a target prediction space under square loss is also $(\nu c_2, \mu c_2)$-diverse over the same space under loss l, if $\nabla^2 l(\cdot, y) \succ c_2 I$ for all $y \in \mathcal{Y}_{ta}$ and for all $x \in \mathcal{X}$ we have $\mathbb{E}[\nabla l(g^*(X), Y) \mid X = x] = 0$.*

Proof. The proof is the same as the proof above except for changing the direction of inequality. Using the definition of the strongly convex and (2.5),

$$\begin{aligned}
&\mathbb{E}_{X,Y}[l(f_t \circ h(X), Y) - l(f_t^* \circ h^*(X), Y)] \\
&\leq \mathbb{E}_{X,Y}[\nabla l \left(f_t^* \circ h^*(X), Y\right)^T \left(f_t^* \circ h^*(X) - f_t \circ h(X)\right) + c_2 \left\| f_t^* \circ h^*(X) - f_t \circ h(X) \right\|_2^2] \\
&= c_2 \mathbb{E}_{X,Y}[\| f_t^* \circ h^*(X) - f_t \circ h(X) \|_2^2],
\end{aligned}$$

which is the generalization error under the square loss. □

Missing proofs in Section 6 Assume we have T tasks, which is (ν, μ)-diverse over $\mathcal{F}_{so}$, and $\mathcal{Y}_{so} \subset \mathbb{R}$. Then we can construct a new source task *so* with multivariate outputs, i.e. $\mathcal{Y}_{so} \subset \mathbb{R}^T$, such that $\mathcal{H}_{so} = \mathcal{F}_{so}^{\otimes T}$ and each dimension k on the output, given an input x, is generated by

$$Y_k(X) = f_k^* \circ h^*(X) + \epsilon.$$

Intuitively, this task is equivalent to T source tasks of a single output, which is formally described in the following Theorem.

Theorem 2.10. *Let so be a source task with $\mathcal{Y}_{so} \subset \mathcal{R}^K$ and $f_{so}^*(\cdot) = (m_1^*(\cdot), \ldots, m_K^*(\cdot))$ for some class $\mathcal{M} : \mathcal{Z} \mapsto \mathbb{R}$. Then if the task set $t_1, \ldots, t_K$ with prediction functions $m_1^*, \ldots, m_K^*$ from hypothesis class $\mathcal{M}$ is (ν, μ)-diverse over $\mathcal{M}$, then so is $(\frac{\nu}{K}, \frac{\mu}{K})$-diverse over the same class.*

Proof. This can be derived directly from the definition of diversity. We use t to denote the

new task. By definition,

$$\begin{aligned}\inf_{f_{so}\in\mathcal{F}_{so}} \mathcal{E}_{so}(f_{so},h) &= \inf_{\boldsymbol{h}_{so}} \mathbb{E}_X\|(m_1\circ(X),\ldots,m_K\circ h(X)) - (m_1^*\circ h^*(X),\ldots,m_K^*\circ h^*(X))\|_2^2 \\ &= \sum_{k=1}^K \inf_{m_k\in\mathcal{M}} \|m_k\circ h(X) - m_k^*\circ h^*(X)\|_2^2 \\ &= \sum_{k=1}^K \inf_{m_k\in\mathcal{M}} \mathcal{E}_{t_k}(f_{t_k},h)\end{aligned}$$

As $(t_1,\ldots,t_K)$ is (ν,μ)-diverse, we have

$$\frac{\sup_{m^*\in\mathcal{M}}\inf_{m\in\mathcal{M}}\mathcal{E}_{m^*}(m,h)}{\inf_{h_{so}\in\mathcal{F}_{so}}\mathcal{E}_{so}(f_{so},h)+\mu/\nu} = \frac{1}{K}\frac{\sup_{m^*\in\mathcal{M}}\inf_{m\in\mathcal{M}}\mathcal{E}_{m^*}(m,h)}{\frac{1}{K}\sum_{k=1}^K\inf_{m_k\in\mathcal{M}}\mathcal{E}_{t_k}(h_k,h)+\frac{\mu}{\nu K}} \le \frac{\nu}{K}.$$

□

For the multiclass classification problem, we try to explain the success of the pretrained model on ImageNet, a single multi-class classification task. For a classification problem with K-levels, a common way is to train a model that outputs a K-dimensional vector, upon which a Softmax function is applied to give the final classification result. A popular choice of the loss function is the cross-entropy loss.

Now we formally introduce our model. Let the Softmax function be $q:\mathbb{R}^K\mapsto[0,1]^K$. Assume our response variable $y\in\mathbb{R}^K$ is sampled from a multinomial distribution with mean function $q(f_{so}^*\circ h^*(x))\in[0,1]^K$, where $h^*\in\mathcal{H}:\mathcal{X}\mapsto\mathcal{Z}$ and $f_{so}^*\in\mathcal{F}_{so}:\mathcal{Z}\mapsto\mathbb{R}^K$. We use the cross-entropy loss $l:[0,1]^K\times[0,1]^K\mapsto\mathbb{R}$, $l(p,q)=-\sum_{k=1}^K p_k\log(q_k)$.

Assumption 2.4. *[Boundedness] We assume that any $f\circ h(x)\in\mathcal{F}_{so}\times\mathcal{H}$ is bounded in $[-\log(B),\log(B)]$ for some constant positive B. We also assume the true function $\min_k U(f^*\circ h^*(x))_k\ge 1/B_*$ for some $B_*>0$.*

Theorem 2.11. *Under Assumption 2.4, a K-class classification problem with $f_{so}^*(\cdot)=(m_1^*(\cdot),\ldots,m_K^*(\cdot))$ for some $m_1^*,\ldots,m_K^*\in\mathcal{M}$ and Softmax-cross-entropy loss function is $(2B_*^2B^4\nu,B_*^2\mu)$-diverse over any the function class $\mathcal{M}$ as long as f_{so}^* with L_2 loss is (ν,μ)-diverse over $\mathcal{M}$.*

Proof. We consider any target task with prediction function from $\mathcal{M}^{\otimes K'}$. Let $U:\mathbb{R}^{K'}\mapsto[0,1]^{K'}$ be the softmax function. We first try to remove the cross-entropy loss. By definition,

the generalization error of any $f \circ h \in \mathcal{M}^{\otimes K'} \times \mathcal{H}$ is

$$\begin{aligned}
&\mathcal{E}_{ta}(f \circ h) - \mathcal{E}_{ta}(h^* \circ h^*) \\
&= \mathbb{E}_{X,Y}[-\sum_{i=1}^{K'} \mathbb{1}(Y = i) \log(\frac{U(f \circ h)}{U(f^* \circ h^*)})] \\
&= \mathbb{E}_X[-\sum_{i=1}^{K'} U(f^* \circ h^*) \log(\frac{U(f \circ h)}{U(f^* \circ h^*)})], \qquad (2.6)
\end{aligned}$$

which gives us the KL-divergence between two distributions $U(f \circ h)$ and $U(f^* \circ h^*)$.

Lemma 2.6. *For any two discrete distributions $p, q \in [0,1]^K$, we have*

$$KL(p,q) \geq \frac{1}{2}(\sum_{i=1}^{K} |p - q|)^2 \geq \frac{1}{2}\sum_{i=1}^{K}(p_i - q_i)^2.$$

On the other hand, if $\min_i p_i \geq b$ for some positive b, then

$$KL(p,q) \leq \frac{1}{b^2}\sum_{i=1}^{K'}(p_i - q_i)^2.$$

Proof. The first inequality is from Theorem 2 in Dragomir and Gluscevic (2000). The second inequality is by simple calculus. □

By the assumption 2.4, we have that for any h, g, x,

$$U(f \circ h(x))_i \in [\frac{1}{KB^2}, \frac{1}{1 + (K-1)/B^2}].$$

We also have $\sum_{i=1}^{K} \exp(f \circ h(x)_i) \in [K/B, KB]$. *To proceed,*

$$
\begin{aligned}
&\frac{\sup_{f_{ta}^* \in \mathcal{M}^{\otimes K_{ta}}} \inf_{\hat{f}_{ta}} \mathcal{E}_{ta}(\hat{f}_{ta} \circ h) - \mathcal{E}_{ta}(f_{ta}^* \circ h^*)}{\inf_{\hat{f}_{so} \in \mathcal{M}^{\otimes K_{so}}} \mathcal{E}_{so}(\hat{f}_{so} \circ h) - \mathcal{E}_{so}(f_{so}^* \circ h^*) + \frac{B_*^2 \mu}{2B_*^2 B^4 \nu}} \\
&\quad \textit{(Applying (2.6) and Lemma 2.6)} \\
&\leq 2B_*^2 \frac{\sup_{f_{ta}^* \in \mathcal{M}^{\otimes K_{ta}}} \inf_{\hat{f}_{ta}} \mathbb{E}_X \|U(\hat{f}_{ta} \circ h(X)) - U(f_{ta}^* \circ h^*(X))\|_2^2}{\inf_{\hat{f}_{so} \in \mathcal{M}^{\otimes K_{so}}} \mathbb{E}_X \|U(\hat{f}_{so} \circ h(X)) - U(f_{so}^* \circ h^*(X))\|_2^2 + \frac{\mu}{B^4 \nu}} \\
&\quad \textit{(Using the boundedness of } \sum_{i=1}^{K} \exp\left(f \circ h(x)_i\right)\textit{)} \\
&\leq 2B_*^2 B^2 \frac{\sup_{f_{ta}^* \in \mathcal{M}^{\otimes K_{ta}}} \inf_{\hat{f}_{ta}} \mathbb{E}_X \|\exp(\hat{f}_{ta} \circ h(X)) - \exp(f_{ta}^* \circ h^*(X))\|_2^2}{\inf_{\hat{f}_{so} \in \mathcal{M}^{\otimes K_{so}}} \mathbb{E}_X \|\exp(\hat{f}_{so} \circ h(X)) - \exp(f_{so}^* \circ h^*(X))\|_2^2 + \frac{\mu}{B^2 \nu}} \\
&\quad \textit{(Using the Lipschitz and convexity of exp)} \\
&\leq 2B_*^2 B^4 \frac{\sup_{f_{ta}^* \in \mathcal{M}^{\otimes K_{ta}}} \inf_{\hat{f}_{ta}} \mathbb{E}_X \|\hat{f}_{ta} \circ h(X) - f_{ta}^* \circ h^*(X)\|_2^2}{\inf_{\hat{f}_{so} \in \mathcal{M}^{\otimes K_{so}}} \mathbb{E}_X \|\hat{f}_{so} \circ h(X) - f_{so}^* \circ h^*(X)\|_2^2 + \mu/\nu} \\
&\leq 2B_*^2 B^4 \nu.
\end{aligned}
$$

The diversity follows.

□

2.B Experimental details

Each dimension of inputs is generated from $\mathcal{N}(0, 1)$. We use Adam with default parameters for all the training with a learning rate 0.001. We choose ReLu as the activation function.

True parameters. The true parameters are initialized in the following way. All the biases are set by 0. The weights in the shared representation are sampled from $\mathcal{N}(0, 1/\sqrt{n_u})$. The weights in the prediction function for the source task are set to be orthonormal when $K_{so} = 1$ and $p \leq n_u$. For the target prediction function or source prediction function if $K_{so} > 1$, the weights are sampled from $\mathcal{N}(0, 1/\sqrt{n_u})$ as in the representation part.

Hyperparameters. Without further mentioning, we use the number of hidden units, $n_u = 4$, input dimension $p = 4$, $K = 5$, $K_{ta} = K_{so} = 1$, the number of observations $n_{so} = 1000$ and $n_{ta} = 100$ by default. Note that since p is set to be 4 by default, equivalently we will have $n_{so} \cdot p = 4000$ observations.

CHAPTER 3

Adaptive Task Scheduling

In the previous chapter, we consider the setting where the task set and the number of observations from each task are given to the algorithm. We show that if the task set is diverse, we can give strong generalization error guarantee. However, it might occurs in practice that one is given a sufficiently large task set and a simple random choice of tasks does not lead to diversity. For example, if one have T tasks with $T \gg d$ in a linear representation learning setting with linear prediction functions. Consider a hard case, where all $T - d$ tasks have the same coefficient for the prediction function, while the rest of the d tasks satisfy the diversity condition. If one have total number of observations N equally allocated for different tasks, we have a small ν, aka. bad diversity. However, there exists a potentially good allocation that achieves diversity. The question is whether we can adaptively choose tasks (when they are unknown) to achieve that optimal rate.

To this end, we study curriculum learning (CL), a commonly used machine learning training strategy that tries to improve the performance by adaptively choosing good tasks for learning. We study CL in the multitask linear regression problem under both structured and unstructured settings. For both settings, we derive the minimax rates for CL with the oracle that provides the optimal curriculum and without the oracle, where the agent has to adaptively learn a good curriculum. Our results reveal that adaptive learning can be fundamentally harder than the oracle learning in the unstructured setting, but it merely introduces a small extra term in the structured setting. To connect theory with practice, we provide justification for a popular empirical method that selects tasks with highest local prediction gain by comparing its guarantees with the minimax rates mentioned above. [1]

[1] This chapter is based on my paper (Xu and Tewari, 2022) with Ambuj Tewari published at ICML 2022.

3.1 Introduction

It has long been realized that we can design more efficient learning algorithms if we can make them learn on multiple tasks. Transfer learning, multitask learning and meta-learning are just few of the sub-areas of machine learning where this idea has been pursued vigorously. Often the goal is to minimize the weighted average losses over a set of tasks that are expected to be similar. While previous literature often assumes a predetermined (and often equal) number of observations for all the tasks, in many applications, we are allowed to decide the *order* in which the tasks are presented and the *number of observations* from each task. Any strategy that tries to improve the performance with a better task scheduling is usually referred to **curriculum learning (CL)** (Bengio et al., 2009). The agent that schedules tasks at each step is often referred as the *task scheduler*.

Though curriculum learning has been extensively used in modern machine learning (Gong et al., 2016a; Sachan and Xing, 2016; Tang et al., 2018; Narvekar et al., 2020), there is very little theoretical understanding of the actual benefits of CL. We also do not know whether the heuristic methods used in many empirical studies can be theoretically justified. Even the problem itself has not been rigorously formulated. To address these challenges, we first formulate the curriculum learning problem in the context of the linear regression problem. We analyze the minimax optimal rate of CL in two settings: an unstructured setting where parameters of different tasks are arbitrary and a structured setting where they have a low-rank structure. Finally we discuss the theoretical justification of a popular heuristic task scheduler that greedily selects tasks with highest local prediction gain.

Related literature. Despite the lack of the literature on curriculum learning theory, we found the following problems under a similar setting. Active learning (Settles, 2009, 2011), addresses the problem of actively selecting unlabeled data points to maximize model accuracy. Active learning can be treated as a curriculum learning when each unlabeled point is a task with point mass. Active learning has also been used for domain adaptation or transfer learning setting Persello and Bruzzone (2012). Multi-source domain adaptation Wang and Deng (2018); Sun et al. (2015) also considers multiple candidate source domains for a given target task. Bhatt et al. (2016) proposed an iterative adaptation methods to integrate the source data from each domain. However, such adaptive procedure has not been theoretically understood.

3.2 Background

We would like to point out previous work on three crucial aspects of CL: two types of benefits one may expect from CL, task similarities assumptions, and task scheduler used in empirical studies.

Two types of benefits. There are two distinct ways to understand the benefits of CL. From the perspective of optimization, some papers argue that the benefits of curriculum can be interpreted as learning from more convex and more smooth objective functions, which serves as a better initialization point for the non-convex target objective function (Bengio et al., 2009). The order of task scheduling is essential here. As an example, Figure 3.1 shows the objective functions of a problem with four source tasks and one target task with increasing difficulty (non-convexity). Directly minimizing the target task (marked in purple line) using gradient descent can be hard due to the non-convexity. However, the simple gradient descent algorithm can converge to the global optima of the current task if it starts from the global optima of the previous task. We refer to the benefit that involves a faster convergence in optimization as *optimization benefit*. Optimization benefits highly depend on the order of scheduling. Generally speaking, if one directly considers the empirical risk minimizer (ERM) which requires global minimization of empirical risk, there may not be any optimization benefit.

The second type of benefit concerns the benefit brought by carefully choosing the number of observations from each task while independent of the order, which we call *statistical benefit*. For example, we have two linear regression problems that are identical except for the standard deviation of the Gaussian noise on response variables. If we consider the OLS estimator on the joint dataset of the two tasks, there is a reduction in noise level when more samples are allocated to the task with a lower level, and the benefit is independent of the order by the nature of OLS. A *statistical benefit* can be seen as any benefit one can get except for the reduction in the difficulties of optimization. Weinshall and Amir (2020) focused on a special curriculum learning task where each sample is considered a task and they analyzed the error rate on the samples of different noise levels. They analyzed the benefits on the error rate of linear models, whose global minimizer can be easily found. Thus, it should also count as the *statistical benefits*.

In general, the two types of benefits can coexist. A good curriculum should account for both the non-convexity and the noise levels. However, due to the significantly different underlying mechanism in the two learning benefits. it is natural to study them separately. This chapter will focus on the analysis of the **statistical benefits**. Thus, we analyze

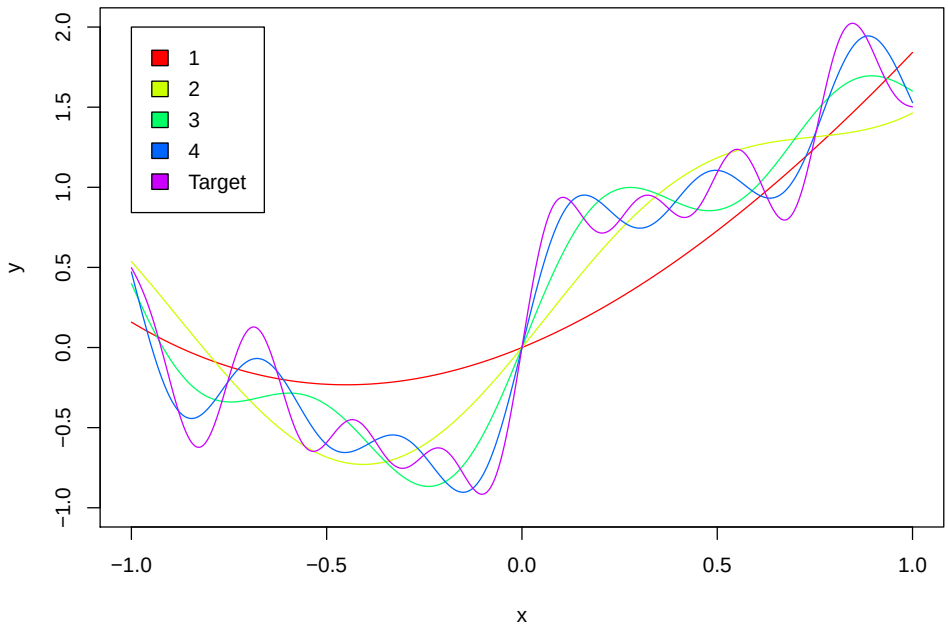


Figure 3.1: An example of tasks with increasing non-convexity. Solid lines of different colors represent the true object functions of different tasks.

algorithms that map datasets to an estimator for each task that may involve finding global minima of the empirical errors for non-convex functions.

Similarity assumptions. We discussed the problem with two almost identical tasks, where we can achieve perfect transfer and the trivial curriculum that allocates all the samples to the simpler task is optimal. However, tasks are generally not identical. Understanding how much benefits the target task can gain from learning source tasks has been a central problem in transfer learning and multitask learning literature. The key is to propose meaningful similarity assumptions.

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output space. Assume we have T tasks with data distributions $D_1 \ldots, D_T$ over $\mathcal{X} \times \mathcal{Y}$. Let (X_t, Y_t) be a sample from task t. Let $f_t = \mathbb{E}[Y_t \mid X_t]$, a mapping $\mathcal{X} \mapsto \mathcal{Y}$, be the mean function. In this chapter, we adopt the simple parametric model on the mean function with f_t represented by parameter $\theta_t^* \in \mathbb{R}^d$.

We consider two scenarios: structured and unstructured. In Section 3.3, we adopt simple linear regression models and do not assume any further internal structure on the true parameters. Two tasks are similar if $\|\theta_{t_1}^* - \theta_{t_2}^*\|_2$ is small. A learned parameter is directly transferred to the target task. This setting has been applied in many previous studies (Yao et al., 2018; Bengio et al., 2009; Xu et al., 2021). In Section 3.4, we study the multitask representation learning setting (Maurer et al., 2016; Tripuraneni et al., 2020; Xu and Tewari, 2021), where a stronger internal structure is assumed. To be specific, we write $f_t(x) = x^T B^* \beta_t^*$, where $x \in \mathbb{R}^d$, $B^* \in \mathbb{R}^{d \times k}$ is the linear representation mapping and $\beta_t^* \in \mathbb{R}^k$ is the task specific parameter. Generally, the input dimension is much larger than the representation dimension ($d \gg k$).

These two settings, while representative, do not exhaust all of the settings in the literature. We refer the reader to Teshima et al. (2020) for a brief summary of theoretical assumptions on the task similarity.

Task schedulers. Many empirical methods have been developed to automatically schedule tasks. Liu et al. (2020) designed various heuristic strategies for task selection for computer vision tasks. Cioba et al. (2021) discussed several meta-learning scenarios where the optimal data allocations are different, which interestingly aligns with our theoretical results. For a more general use, one major family of task scheduler is based on the intuition that the task scheduler should select the task that leads to the highest local gain on the target loss (Graves et al., 2017). Since the accurate prediction gain is not accessible, online decision-making algorithms (bandit and reinforcement learning) are frequently used to adaptively allocate samples (Narvekar et al., 2020). However, there is no theoretical guarantee that

such greedy algorithms can lead to the optimal curriculum.

Notation. For any positive integer n, we let $[n] = \{1, \ldots, n\}$. We use the standard $O(\cdot), \Omega(\cdot)$ and $\Theta(\cdot)$ notation to hide universal constant factors. We also use $a \lesssim b$ and $a \gtrsim b$ to indicate $a = O(b)$ and $b = \Omega(b)$.

3.3 Unstructured Linear Regression

In this section, we study the problem of learning from T tasks to generate an estimate for a single target task.

3.3.1 Formulations

We consider linear regression tasks. Let $1 \ldots, T$ denote T tasks. Let $\theta_t^* \in \mathbb{R}^d$ denote the true parameter of task t. The response Y_t of task t is generated in the following manner

$$Y_t = X_t^T \theta_t^* + \epsilon_t,$$

where ϵ_t is assumed to be the Gaussian noise with $\mathbb{E}[\epsilon_t^2] = \sigma_t^2$ and $X_t \sim \mathcal{N}(\mathbf{0}, \Sigma_t)$, where $\Sigma_t \in \mathbb{R}^{d \times d}$ is the covariance matrix that is positive definite. Any task, therefore, can be fully represented by a triple $(\theta_t^*, \sigma_t, \Sigma_t)$.

Throughout the chapter, we are more interested in the unknown parameters rather than the covariate distribution or the noise level. We simply denote $\boldsymbol{\theta}^* \in \mathbb{R}^{d \times T}$ the parameters of a *problem* (T tasks) and let $\boldsymbol{\theta}_t^*$ be the t-th column of the matrix.

We make a uniform assumption on the covariance matrix of input variables. The same assumption is also used by Du et al. (2020).

Assumption 3.1 (Coverage of covariate distribution)**.** *We assume that all* $C_0 I_d \succ \Sigma_t \succ C_1 I_d$ *for some constant* $C_0, C_1 > 0$ *and any* $t \in [T]$.

Goal. Let S_t^n be random n samples from task t. Let $l : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ be a loss function and $L_t(\theta) = \mathbb{E}[l(X^T\theta, Y)]$ be the expected loss of a given hypothesis θ evaluated on task t. Moreover, we denote the excess risk by

$$G_t(\theta) = L_t(\theta) - \inf_{\theta'} L_t(\theta').$$

Our goal in this section is to minimize the expected loss of the last task T, which we call the **target task**. Throughout the chapter, we use square loss function.

Transfer distance. Algorithms tend to perform better when the tasks are similar to each other, such that any observations collected from non-target task bear less transfer bias. We define *transfer distance* between tasks t_1, t_2 as $\Delta_{t_1,t_2} = \|\theta^*_{t_1} - \theta^*_{t_2}\|_2$.

It is not fair to compare the performances between problems with different transfer distances. To study a minimax rate, we are interested in the worst performance over a set of problems with similar transfer distance. Let $\boldsymbol{Q} \in \mathbb{R}^T$ be the distance vector encoding the upper bounds on the distance between the target task to any task. We define the hypothesis set with known transfer distance as $\Theta(\boldsymbol{Q}) = \{\boldsymbol{\theta}^* : \|\boldsymbol{\theta}^*_t - \boldsymbol{\theta}^*_T\|_2 \leq \boldsymbol{Q}_t\} \subset \mathbb{R}^{T\times d}$. The hypothesis set with unknown transfer distance can be defined as $\tilde{\Theta}(\boldsymbol{Q}) = \cup_p\{\boldsymbol{\theta}^* : \|\boldsymbol{\theta}^*_t - \boldsymbol{\theta}^*_T\|_2 \leq \boldsymbol{Q}_{p_t}\}$, where $p \in [T]^T$ is any permutation of $[T]$. We say this hypothesis set has unknown transfer distance because even if there exists some small $\boldsymbol{Q}_t$ such that the transfer distance is low, an agent does not know which task has the low transfer distance.

Curriculum learning and task scheduler. In this chapter, we concern only the statistical learning benefits. Since the order of selecting tasks does not affect the outcome of the algorithm, we denote a curriculum by $\boldsymbol{c} \in [N]^T$, where each $\boldsymbol{c}_t$ is the total number of observations from task t and $\sum_t \boldsymbol{c}_t = N$. Note that $\boldsymbol{c}$ can consist of random variables depending on the task scheduler. The set of all the curriculum with a total number of observations N is denoted by $\mathcal{C}_N = \{\boldsymbol{c} \in [N]^T : \sum_t \boldsymbol{c}_t = N\}$.

Any curriculum learning involves a multitask learning algorithm, which is defined as a mapping $\mathcal{A}$ from a set of datasets $(S_1^{n_1}, \ldots, S_T^{n_T})$ to a hypothesis θ for the target task.

A task scheduler runs the following procedure. At the start of the step $i \in [N]$, we have $n_{i,1}, \ldots, n_{i,T}$ observations from each task. The task scheduler $\mathcal{T}$ at step i is defined as a mapping from the past observations $(S_1^{n_{i,1}}, \ldots, S_T^{n_{i,T}})$ to a task index. Then a new observation from the selected task $\mathcal{T}(S_1^{n_{i,1}}, \ldots, S_T^{n_{i,T}}) \in [T]$ is sampled.

Minimax optimality and adaptivity. One of the goals of this chapter is to understand the minimax rate of the excess risk on the target task over all the possible combinations of multitask learning algorithms and task schedulers. We first attempt to understand a limit of that rate by considering an oracle scenario that provides the optimal curriculum for any problem.

Rigorously, we denote the loss of a fixed curriculum $\boldsymbol{c} \in \mathcal{C}_N$ with respect to a fixed algorithm $\mathcal{A}$ and problem $\boldsymbol{\theta}$ by

$$R_T^N(\boldsymbol{c}, \mathcal{A} \mid \boldsymbol{\theta}) = \mathbb{E}_{S_1^{\boldsymbol{c}_1}, \ldots, S_T^{\boldsymbol{c}_T}} G_T(\mathcal{A}(S_1^{\boldsymbol{c}_1}, \ldots, S_T^{\boldsymbol{c}_T})).$$

We define the following oracle rate, which takes infimum over all the possible fixed cur-

riculum designs given a fixed task set with different $\boldsymbol{\theta}$ in a hypothesis set Θ.

$$R_T^N(\Theta) \coloneqq \inf_{\mathcal{A}} \sup_{\boldsymbol{\theta}\in\Theta} \inf_{\boldsymbol{c}\in\mathcal{C}_N} R_T^N(c, \mathcal{A} \mid \boldsymbol{\theta}). \tag{3.1}$$

In general, the above oracle rate considers an ideal case, because the optimal curriculum depends on the unknown problem and any learning algorithm has to adaptively learn the problem to decide the optimal curriculum.

We ask the following question: can adaptively learned curriculum perform as well as the optimal one as in Equ. (3.1)? To answer the question, we define the minimax rate for adaptive learning:

$$\tilde{R}_T^N(\Theta) \coloneqq \inf_{\mathcal{A}} \inf_{\mathcal{T}} \sup_{\boldsymbol{\theta}\in\Theta} \mathbb{E} G_T(\mathcal{A}(S_1^{\boldsymbol{c}_\mathcal{T},1}, \ldots, S_T^{\boldsymbol{c}_\mathcal{T},T})), \tag{3.2}$$

where $\boldsymbol{c}_\mathcal{T} \in \mathcal{C}_N$ is the curriculum adaptively selected by the task scheduler $\mathcal{T}$ and the expectation is taken over both datasets $S_1, \ldots, S_T$ and $\boldsymbol{c}_\mathcal{T}$.

In this section, we are interested in the oracle rate in (3.1) compared to some naive strategy that allocates all the samples to one task. This answers how much benefits one can achieve compared to some naive learning schedules. We are also interested the gap between Equ. (3.1) and Equ. (3.2).

3.3.2 Oracle rate

In this section, we analyze the oracle rate defined in Equ. (3.1). We first give an overview of our results. For any problem instance, there exists a single task t such that the naive curriculum with $\boldsymbol{c}_t = N$ matches a lower bound for the oracle rate defined in Equ. (3.1).

For any task $t \in [T]$, its direct transfer performance of the OLS estimator on the target task T can be roughly bounded by $\Delta_{t,T}^2 + d\sigma_t^2/N$.

Thus, our result implies that essentially, the goal of curriculum learning is to identify the best task that balance the transfer distance and the noise level.

Theorem 3.1. *Let $\boldsymbol{Q}$ be a fixed distance vector defined above. The oracle rate within $\Theta(\boldsymbol{Q})$ in Equ. (3.1) can be lower bounded by*

$$R_T^N(\Theta(\boldsymbol{Q})) \gtrsim C_0 \min_t \{\boldsymbol{Q}_t^2 + \frac{d\sigma_t^2}{N}\}. \tag{3.3}$$

Proof highlights. Kalan et al. (2020) showed a minimax rate of the transfer learning problem with only one source task. They considered three scenarios, which can be uniformly

lower bounded by the right hand side of Equ. (3.3). Our analysis can be seen as an extension of their results to multiple source tasks. In general, let the rate in (3.3) be δ and $C > 0$ be a constant. Let t^* be the best task indicated by (3.3). Any task t with a large distance ($\boldsymbol{Q}_t > C\delta$) is not helpful to learn the target task. Thus, samples from these tasks can be discarded without reducing the performance. For any task t with $\boldsymbol{Q}_t \leq C\delta$, we will show that any sample from task t gives almost as much as information as the best task t^* gives. Thus, one can replace them with a random sample from the best task t^* without reducing the loss. Then the problem can be converted to a single source task problem, from where we follow the lower bound construction in Kalan et al. (2020).

3.3.3 Minimax rate for adaptive learning

The problem can be hard when the transfer distance is unknown. We introduce an intuitive example to help understand our theoretical result. Assume we have three tasks including one target task and two source tasks. One of the two source tasks is identical to the target task. We have n samples for both source tasks, while no observations from the target task. In this example, even if one of the source tasks is identical to the target task, no algorithm can decide which source task should be adopted, since we have no information from the target task. In other words, any algorithm can be as bad as the worst out of the two source tasks. This is not an issue when the transfer distance is known to the agent in the oracle scenario. This example implies that to adaptively gain information from source tasks, we will need sufficient information from the target task. Otherwise, there is risk of including information from tasks that contaminates the target task. Similarly, David et al. (2010) also showed that without any observations from the target task, domain adaptation is impossible.

More generally, even if we have some data from the target task, we will show that one is not able to avoid σ_T^2 term, the learning difficulty of the target task. Now we formally introduce our results.

Theorem 3.2. *Assume $T \geq 4$. Let $Q_{sub} > 0$. Let $\boldsymbol{Q}$ be a fixed distance vector that satisfies $\boldsymbol{Q}_1 = 0$ and $\boldsymbol{Q}_t = Q_{sub} > 0$ for all $t = 2, \ldots, T-1$. The minimax rate in Equ. (3.2) can be lower bounded by*

$$\tilde{R}_T^N(\tilde{\Theta}(\boldsymbol{Q})) \gtrsim \min\{\frac{\sigma_T^2 \log(T)}{N}, Q_{sub}^2\} + \min_t \frac{d\sigma_t^2}{N}. \tag{3.4}$$

Theorem 3.2 implies that without knowing the transfer distance, any adaptively learned curriculum of any multitask learning algorithm will suffer an unavoidable loss of $\sigma_T^2 \log(T)/N$, when Q_{sub} is large. Compared to the rate $\sigma_T^2 d/N$ without transfer learning, there is still a

potential improvement of a factor of $d/\log(T)$ when Q_{sub} and σ_t^2 are small.

Upper bound. As we showed above, there is a potential improvement of $d/\log(T)$. This is because given the prior information that one of the source tasks is identical to the target task, the problem reduces from estimating a d-dimensional vector to identifying the best task from a candidate set, whose complexity reduces to $\log(T)$.

In fact, a simple fixed curriculum could achieve the above minimax rate. Assume that any $\|\theta_t^*\|_2 \leq C_2$ for some constant $C_2 > 0$. Let $\boldsymbol{c}_T = N/2$ and for all the other tasks $\boldsymbol{c}_t = N/(2T-2)$. For each $t = 1, \dots T-1$, let $\tilde{\theta}_t$ be the OLS estimator using only its own samples. Let $\hat{\theta}_t$ be the projection of $\tilde{\theta}_t$ onto $\{\theta : \|\theta\|_2 \leq C_2\}$. Then we choose one estimator from $t = 1, \dots, T-1$, that minimizes the empirical loss for the target task:

$$t^* = \arg\min_{t \in [T-1]} \sum_{i=1}^{N/2} (Y_{T,i} - X_{T,i}^T \hat{\theta}_t)^2. \tag{3.5}$$

Theorem 3.3. *Assume there exists a task t such that $\Delta_{t,T} = 0$ and $\|\theta_t^*\|_2 \leq C_2$. With a probability at least $1-\delta$, $\hat{\theta}_{t^*}$ satisfies*

$$G_T(\hat{\theta}_{t^*}) \lesssim C_0 C_2^2 \log(Td/\delta) \left(\frac{C_2 \sigma_T^2}{N} + \frac{dT\sigma_{t^*}^2}{C_1 N} + \sqrt{\frac{d}{N}} \right). \tag{3.6}$$

Under a mild condition that $\sqrt{d}\sigma_T^2 \gg \sqrt{N}$, the first two terms dominate.

Note that t^* is a random value. However, when all t satisfy $dT\sigma_t^2 < \Delta_{max}\sigma_T^2 \log(T)$, the first term is the dominant term and our bound matches the lower bound in (3.4). This could happen when $\sigma_T^2 \gg \sigma_t^2$ for all $t = 1, \dots, T-1$.

General function class. As we mentioned before, though it is difficult to identify the good source tasks, the complexity of doing so is still lower than learning the parameters directly. We remark that this result can be generalized to any function class beyond linear functions. Keeping all the other setup unchanged, we assume that the mean function $f_t^* \in \mathcal{F}_t : \mathcal{X} \mapsto \mathcal{Y}$ for some input space $\mathcal{X}$ and output space $\mathcal{Y}$ shared by all the tasks. For convenience, we assume there is no covariate shift, i.e. the input distributions are the same. We give an analogy of Theorem 3.3.

Assumption 3.2 (Assumption B in Jin et al. (2021c))**.** *Assume $l(\cdot, y)$ is L_2-strongly convex and L_1-Lipschitz at any $y \in \mathcal{Y}$. Furthermore, for all $x \in \mathbb{R}^d$ and $t \in [T]$,*

$$\mathbb{E}[\nabla l(f_t^*(X), Y) \mid X = x] = 0.$$

Assume we have $N/(2T-2)$ observations for all tasks $t = 1, \ldots, T$ and $N/2$ observations for the target task. Let $\hat{f}_t$ be the empirical risk minimizer of the task t. Similarly to (3.5), let

$$t^* = \arg\min_{t \in [T-1]} l_T^N(\hat{f}_t),$$

where l_T^N is the empirical loss on task T. Let $L^* = \min_{t \in [T-1]} L_T(f_t^*)$ and $t' = \arg\min_{t \in [T-1]} L_T(f_t^*)$. We will use Rademacher complexity to measure the hardness of learning a function class. We refer readers to Bartlett and Mendelson (2002) for the detailed definition of Rademacher complexity.

Proposition 3.1. *Given the above setting and Assumption 3.2, we have with a probability at least* $1 - \delta$*,*

$$G_T(\hat{f}_{t^*}) \lesssim L^* + \frac{L_1}{L_2}\left(\mathcal{R}_{N/T}(\mathcal{F}_{t'}) + \sqrt{\frac{\log(1/\delta)}{N/T}}\right),$$

where $\mathcal{R}_N(\mathcal{F})$ *is the Rademacher complexity of function space* $\mathcal{F}$*.*

This bound improves the bound for single target task learning, which scales with $\mathcal{R}_N(\mathcal{F}_T)$, when $\mathcal{R}_N(\mathcal{F}_T) \gg \mathcal{R}_{N/T}(\mathcal{F}_{t^*})$. The underlying proof idea is still that identifying good tasks is easier than learning the model itself.

3.4 Structured Linear Regression

Now we consider a slightly different setting, where we want to learn a shared linear representation that generalizes to any target task within a set of interest.

A lot of recent papers have shown that to achieve a good generalization ability of the learned representation, the algorithm have to choose diverse source tasks (Tripuraneni et al., 2020; Du et al., 2020; Xu and Tewari, 2021). They all study the performance of a given choice of source tasks, while it has been unclear whether an algorithm can adaptively select diverse tasks.

3.4.1 Problem setup

We adopt the setup in Du et al. (2020). Let $d, k > 0$ be the dimension of input and representation, respectively ($k \ll d$). We also set $T \leq d$. Let $B^* \in \mathbb{R}^{d \times k}$ be the shared representation. Let $\beta_1^*, \ldots, \beta_T^* \in \mathbb{R}^k$ be the linear coefficients for prediction functions. The model setup is essentially the same as the setup in Section 3.3.1 except for the true parameters

being $B^*\beta_t$. We call this setting structured because if one stacks the true parameters as a matrix, the matrix has a low-rank structure. To be specific, the output of task t given by

$$Y_t = X_t^T B^* \beta_t^* + \epsilon_t.$$

We use the same setup for the covariate X_t as in Section 3.3 and we consider $\sigma_1^2 = \cdots = \sigma_T^2 = \sigma^2$ for some $\sigma^2 > 0$.

Diversity. Let t_i be the task selected by the scheduler at step i. It has been well understood that to learn a representation that could generalize to any target task t' with arbitrary $\beta_{t'}^*$, we will need a lower bound on the following term

$$\lambda_k \left(\sum_{i=1}^{N} \beta_{t_i}^* \beta_{t_i}^{*T} \right) =: \lambda_{N,k}, \tag{3.7}$$

where λ_k is the k-th largest eigenvalue of a matrix, i.e. the smallest eigenvalue. Basically, we hope the source tasks cover all the possible directions such that any new task could be similar to at least some of the source tasks. Equ. (3.7) serves as an assumption in Du et al. (2020). When the true β_t^* are known, we can simply diversely pick tasks. When the β_t^* are unknown, the trivial strategy that equally allocates samples will perform badly. For example, let $T \gg k$ and let all the $\beta_t, t = k+1, \ldots, T$ be identical. The trivial strategy will only cover one direction sufficiently, which ruins the generalization ability.

In this section, we will show that it is possible to adaptively schedule tasks to achieve the diversity even in the hard case discussed above.

3.4.2 Lower bounding diversity

In this section, we introduce an OFU (optimism in face of uncertainty) algorithm that adaptively selects diverse source tasks.

Two-phase estimator. We first introduce an estimator on the unknown parameters. Assume up to step i, we have dataset $S_1^{n_{i,1}}, \ldots, S_T^{n_{i,T}}$ for each task t. We evenly split each dataset $S_t^{n_{i,t}}$ to two datasets $S_{i,t}^{(1)}$ and $S_{i,t}^{(2)}$, both with a sample size of $\lfloor n_{i,t}/2 \rfloor$. We solve the optimization problem below:

$$\hat{B}_i = \operatorname*{arg\,min}_{B \in \mathbb{R}^{d \times k}} \min_{\beta_t \in \mathbb{R}^k, t \in [T]} \sum_{t \in [T]} \sum_{(x,y) \in S_{i,t}^{(1)}} \|y - x^T B \beta_t\|^2,$$

$$\text{and } \hat{\beta}_{i,t} = \arg\min_{\beta_t \in \mathbb{R}^k} \sum_{(x,y)\in S_{i,t}^{(2)}} \|y_j - x_j B \beta_{t_j}\|^2.$$

Note that we split the dataset such that $\hat{B}_i$ and $\hat{\beta}_{i,t}$ are independent.

Algorithm 1 CL by optimistic scheduling

1: **Input:** T tasks and total number of observations N and constant $\gamma > 0$.
2: Sample $\lceil \gamma(d + \log(N/\delta)) \rceil$ samples for each task.
3: **for** $i = T\lceil \gamma(d + \log(N/\delta)) \rceil + 1, \ldots, N$ **do**
4: Construct confidence set $\mathcal{B}_{i,t}$ for each $t \in [T]$ according to Equ. (3.8).
5: Select t_i according to Equ. (3.9).
6: **end for**
7: **return:** Curriculum $(t_1, \ldots, t_N)$.

Optimistic task scheduler. Our algorithm runs by keeping a confidence bound for $B^*\beta_t^*$ for each $t \in [T]$ and each step $i \in [N]$. Lemma 3.1 introduces a suitable upper bound construction. Lemma 3.1 holds under the following assumptions.

Lemma 3.1. *Let $\kappa = C_0/C_1$. Assume Assumption 3.1 hold. There exists universal constants $\gamma > 0, \alpha > 0$ such that, at all step $i > T\lceil \gamma(d + \log(N/\delta)) \rceil$, with a probability $1 - \delta$, we have for all $t \in [T]$,*

$$\|\hat{B}_i^T \hat{\beta}_{i,t} - B^* \beta_{i,t}^*\|_2^2 \lesssim \frac{\alpha C_5 \sigma^2 dk \log(\kappa N \delta / T)}{C_1^2 n_{i,t}},$$

where $n_{i,t}$ is the number of observations from task t up to step i.

Following the bound in Lemma 3.1, we construct the confidence set with width

$$\mathcal{W}_{i,t} := \frac{C_5 \sigma^2 dk \log(\kappa N \delta / T)}{C_1^2 n_{i,t}}.$$

At each step i for each task t, we construct a confidence set around $\hat{B}_i \hat{\beta}_{i,t}$,

$$\mathcal{B}_{i,t} = \{\theta \in \mathbb{R}^d : \|\hat{B}_i \hat{\beta}_{i,t} - \theta\|_2^2 \le \mathcal{W}_{i,t}\}. \tag{3.8}$$

Then following the principle of optimism in face of uncertainty, we select the task t_i such that

$$t_i \in \arg\max_{t\in[T]} \max_{\theta \in \mathcal{B}_{i,t}} \lambda_k\Big(\sum_{j=1}^{i-1} \tilde{\theta}_j \tilde{\theta}_j^T + \theta\theta^T\Big) \tag{3.9}$$

and $\tilde{\theta}_i = \arg\max_{\theta\in\mathcal{B}_{i,t}} \lambda_k(\sum_{j=1}^{i-1} \tilde{\theta}_j \tilde{\theta}_j^T + \theta\theta^T)$. Here $\tilde{\theta}_i$ is our belief for task t at the step i.

Now we are ready to present our lower bound results for diversity. Our results hold under two assumptions. The first assumption require the representation matrix B^* is not degenerated. We also assume boundedness on β_t^*'s.

Assumption 3.3. *Assume the largest singular value of B^* is smaller than C_4 for some $C_4 > 0$.*

Assumption 3.4 (Boundedness)**.** *We also assume that $\|\beta_t^*\|^2 \leq C_5$ for all $t \in [T]$.*

Theorem 3.4. *Suppose Assumption 3.3 and 3.4 hold. Assume for all $\nu \in \mathbb{R}^k, \|\nu\|_2 = 1$, there exists some task t such that $\nu^T B^* \beta_t^* \beta_t^{*T} B^{*T} \nu \geq \lambda$ for some $\lambda > 0$. Let $t_i, i = 1, \ldots, N$ be the tasks select by Algorithm 1 for some constant α. Then there exists some $\alpha > 0$, such that with a probability at least $1 - \delta$,*

$$\frac{\lambda_{N,k}}{N} \gtrsim \frac{\lambda}{C_4 k} - \sqrt{\frac{\sigma^2 C_5^2 dkT \log(\kappa N/(T\delta))}{C_1^2 C_4^2 \lambda N}}.$$

If we are provided with the oracle, we will only have the first term above. When N is sufficiently large, the second term in Theorem 3.4 is negligible and we will achieve diversity asymptotically as long as $dkT \ll N$. Our proof follows the standard framework for OFU algorithms. We first show the correctness of the confidence set implied by Lemma 3.1. Then the key steps are to show the optimism, i.e. $\lambda_k(\sum_{i=1}^N \tilde{\theta}_i \tilde{\theta}_i^T) = \Omega(\lambda/k)$ and to bound the difference term between the belief $\lambda_k(\sum_{i=1}^N \tilde{\theta}_i \tilde{\theta}_i^T)$ and the actual value $\lambda_{N,k}$. We provide the proof in Appendix 3.A.

3.4.3 Upper bound results

Though the lower bound in Theorem 3.4 is already satisfying, we still want to shed some light on whether the dependency on $\sqrt{1/N}$ is avoidable by showing an upper bound result in Theorem 3.5.

Theorem 3.5. *For any curriculum learning algorithm, there exists T tasks ($T > k$) such that for all $\nu \in \mathbb{R}^k, \|\nu\|_2 = 1$, there exists some β_t^*, $\|\beta_t^* \nu\| \geq 1$ and*

$$\mathbb{E}[\frac{\lambda_{N,k}}{N}] \lesssim \frac{\max_{t_1,\ldots,t_N \in [T]} \lambda_k(\sum_{i=1}^N \beta_{t_i}^* \beta_{t_i}^{*T})}{N} - \sqrt{\frac{\sigma^2 T}{Nk^3}}.$$

Theorem 3.5 states that the $\sqrt{1/N}$ dependency is unavoidable, while there is still a gap of dk^4 between the upper bound and the lower bound. Our hard case construction is inspired by the case where the naive strategy that allocates samples evenly. To be specific, we consider

T tasks such that k of them are diversely specified and all the other $T-k$ tasks are identical. Naive strategies will fail by having $\lambda_{N,k} \approx \frac{1}{kT}$. We divide T tasks into $\lfloor T/k \rfloor$ blocks. Then we construct similar problems. Different problems have the diverse tasks in different blocks. The difficulty of the problem becomes identifying the block with diverse tasks, which is analogous to the idea of bandit model in a general sense. From here, we follow a similar proof of stochastic bandits (Lattimore and Szepesvári, 2020). The full proofs can be found in Appendix 3.A.

3.5 Analysis of Prediction Gain

In this section, we give some theoretical guarantees on prediction-gain driven task scheduler under the unstructured setting discussed in Section 3.3. We do not consider the structured setting because it is not clear how to apply the prediction-gain driven method to multitask representation learning setting.

Prediction Gain and convergence rate. We define prediction gain in the following way. At the step i, a multitask learning algorithm $\mathcal{A}$ maps any trajectory $\mathcal{H}_i = \{x_{t_i,j}, y_{t_i,j}\}_{j=1}^{i}$ to a parameter $\theta \in \mathbb{R}^d$ for the target task. Let the estimate at step i be θ_i. The prediction gain is defined as

$$G(\mathcal{A}, \mathcal{H}_{i+1}) \coloneqq L_T(\theta_i) - L_T(\theta_{i+1}).$$

At the start of the round i, the prediction-gain based task scheduler selects $t_i \in [T]$ such that $G(\mathcal{A}, \mathcal{H}_i)$ is maximized.

Note that in general, prediction gain is not observable to the algorithm before $x_{t_i,i}$ and $y_{t_i,i}$ are actually sampled. There are simple ways to estimate prediction gain, for example, from several random samples from each task.

In a linear model, the prediction gain is equivalent to convergence rate.

$$L_T(\theta_i) - L_T(\theta_{i+1}) = \|\theta_i - \theta_T^*\|_{\Sigma_T}^2 - \|\theta_{i+1} - \theta_T^*\|_{\Sigma_T}^2.$$

Weinshall and Amir (2020) discussed various benefits of curriculum learning by show that their strategy gives higher local convergence rate. It is not clear from the context that the greedy strategy that selects the highest local prediction gain gives the best total prediction gain in long run.

Decomposing prediction gain. Considering a identical covariance matrix $\Sigma_t = I$, the loss over a given parameter θ can be written as $\|\theta - \theta_T^*\|_2^2 + \sigma_T^2$.

Assume the gradient is calculated from a sample from the task t. According to the update of SGD, at the step i, we have

$$\theta_{i+1} - \theta_T^* = (I - \eta_i x_i^{(t)} x_i^{(t)T})(\theta_i - \theta_T^*) + \eta_i x_i^{(t)}(\epsilon_i + x_i^T \theta_{t,T}^{\Delta}),$$

where $\theta_{t,T}^{\Delta} = \theta_t^* - \theta_T^*$.

The one-step prediction gain is

$$\begin{aligned}
&\|\theta_i - \theta_T^*\|^2 - \|\theta_{i+1} - \theta_T^*\|^2 \\
&= \eta_i \|\theta_i - \theta_T^*\|^2_{(2-\eta_i \|x_i^{(t)}\|_2^2) x_i x_i^T} - \eta_i^2 \|x_i^{(t)}(\epsilon_i + x_i^T \theta_{t,T}^{\Delta})\|_2^2 \\
&\quad - \eta_i (\theta_i - \theta_T^*)^T (I - \eta_i x_i^{(t)} x_i^T) x_i^{(t)} (\epsilon_i + x_i^T \theta_{t,T}^{\Delta}).
\end{aligned}$$

The first term on the R.H.S is the absolute gain shared by all the tasks. On expectation, the second term is

$$-\mathbb{E}\eta_i^2 \|x_i(\epsilon_i - x_i^T \theta_{t,T}^{\Delta})\|_2^2 = -\mathbb{E}\eta_i^2 \|x_i\|_2^2 (\sigma_t^2 + \|\theta_{t,T}^{\Delta}\|^2_{x_i x_i^T}). \tag{3.10}$$

In expectation, the third term is

$$-\mathbb{E}\eta_i (\theta_i - \theta_T^*)^T (I - \eta_i x_i x_i^T) x_i x_i^T \theta_{t,T}^{\Delta} = -\mathbb{E}(1 - \eta_i \|x_t\|_2^2) \eta_i (\theta_i - \theta_T^*)^T x_i x_i^T \theta_{t,T}^{\Delta}. \tag{3.11}$$

Now we discuss term (3.10) and (3.11), respectively. (3.11) is independent of σ_t^2 and it is a dynamic effects depending on the current estimate θ_i. That means (3.11) is independent of the task difficulty and its constantly changes. When $(\theta_i - \theta_T^*)^T x_t x_t^T (\theta_t^* - \theta_T^*)^T < 0$, the task t has a larger prediction gain. This is when the gradient descent direction is consistent in both target and the task t.

For term (3.10), we notice that task difficulty σ_t^2 and transfer distance $\Delta_{t,T}$ play equal importance in the prediction gain measure regardless of the number of observations.

Optimality of prediction gain. Let t^* be the optimal task defined by

$$t^* = \arg\min_t \Delta_{t,T}^2 + \frac{d\sigma_t^2}{N}.$$

We consider an averaging SGD algorithm with a step size $\eta_i = 1/i$. In general, let $\bar{\theta}_N = \sum_{i=1}^N \theta_i / N$. The following Theorem shows that the performance of the averaging SGD with an accurate prediction-gain based task scheduler matches the minimax lower bound in Theorem 3.1.

Theorem 3.6. *Assume $\Sigma_1 = \cdots = \Sigma_T = I$. Assume $\|\theta_t^*\|_2^2 \leq C_5$ for all t. Given T tasks with noise levels $\sigma_1^2, \ldots, \sigma_T^2$ and transfer distance $\Delta_{1,T}, \ldots, \Delta_{T,T}$, let $\bar{\theta}_N$ be the averaging SGD estimator with an accurate prediction-gain based task scheduler defined above. We have*

$$G_T(\bar{\theta}_N) \lesssim \Delta_{t^*,T}^2 + \frac{(d\sigma_{t^*}^2 + C_5)\log(N)}{N}. \tag{3.12}$$

Theorem 3.6 gives an upper bound on $G_T(\bar{\theta}_N)$ that matches the lower bound in Theorem 3.1.

3.5.1 Simulation Studies

To compliment the theoretical analyses, we conduct simulations studies by applying actual SGD with tasks chosen to maximize the local prediction gain. We consider two SGD scenarios: 1) assuming the algorithm has the accurate estimate on the prediction gain as in our analysis; 2) algorithms that have to estimate prediction gain.

For the second scenario, we follow Graves et al. (2017), which regard the task scheduling as a sequential decision-making problem. A popular choice of agent is to use adversarial bandit model. To be specific, we use EXP3 algorithm. See Appendix 3.B for details of the algorithm. Bandit algorithm runs by maximizing rewards. In our experiments, let θ_i be the estimate at the step i. We sample one observation (x_i, y_i) from the target task after each gradient descent, and the reward r_i at the step i is given by

$$r_i = (y_i - \theta_{i-1}^T x_i)^2 - (y_i - \theta_i^T x_i)^2.$$

To evaluate the accurate prediction gain, we directly calculate the distance $\|\theta_i - \theta_t^*\|^2$.

Following the setup throughout the chapter, we consider a multitask linear regression problem. We set $T = 5$ and $\sigma_t^2 = 0.001, 0.01, 0.1, 1, 1$ for $t = 1, \ldots, 5$, respectively. Note that the 5-th task is the target task. We test the effects of total number of observations $n = 10, 50, 100, 500, 1000$ and the effects of dimension $d = 5, 10, 50, 100$. By default, we set $n = 1000$ and $d = 5$. The true parameters of all the tasks are sampled from $\mathcal{N}(0, 0.001 I_d)$. On expectation, the transfer distance $\Delta_{t,T}^2$ between task t and the target task is about $0.01d$. The input x's are sampled from the same distribution $\mathcal{N}(0, I_d)$ for all the tasks.

Figure 3.2 shows the L_2 distance of the final estimate and the true parameters of the target task. Our simulation results suggest that 1) prediction-gain based task scheduler can significantly improve the performance over the target task, when there exists some source task with low transfer distance and low noise; 2) there is still benefits when scheduler has to adaptive select tasks which coincidences our Theorem 3.3. The results are robust under

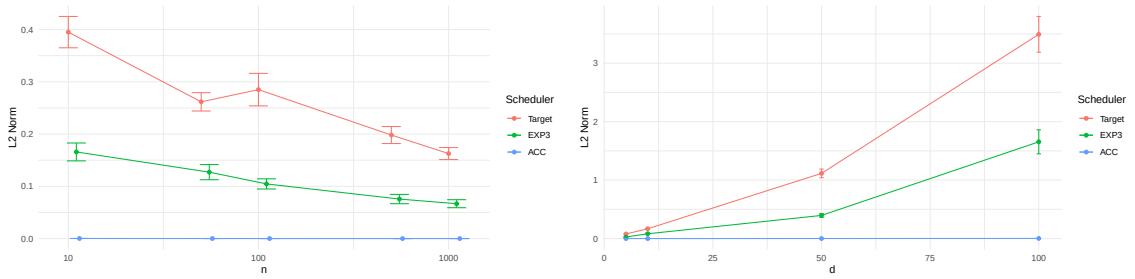


Figure 3.2: L_2 distance between the final estimate θ_i and θ_T^* under different total numbers of observations n and different numbers of dimensions d. The confidence intervals are the standard deviation of 1000 independent runs.

different choices of n and d.

3.6 Discussion

In this chapter, we discussed the benefits of Curriculum Learning under two special settings: multitask linear regression and multitask representation learning. In the multitask linear regression setting, it is fundamentally hard to adaptively identify the optimal source task to transfer. In the multitask representation learning setting, a good curriculum is the curriculum that diversifies the source tasks. We show that the extra error caused by the adaptive learning is small and it is possible to achieve a near-optimal curriculum. Then we provided theoretical justification for the popular prediction-gain driven task scheduler that has been used in the empirical work.

Our results suggest some natural directions for future work. We show a lower bound (Thm. 3.5) on the diversity in the multitask representation learning setting, while leaving a gap of d compared to our upper bound (Thm. 3.4). We believe this gap is because a loose construction of the hard cases that ignores the difficulty of learning the shared representation. Another direction is to show whether prediction-gain methods with no accurate gain estimation could still have performance close to lower bounds for the adaptive learning setting.

An important direction is to consider the how the order of presenting tasks affects the learning performance. Since the order of tasks are irrelevant for the analysis on empirical risk minimizer, one have to analyze the actual benefits in terms of optimization.

3.A Missing Proofs

Proof of Theorem 1

Proof. Our proof is inspired by the proof of Kalan et al. (2020), which gives a lower bound construction for the two-tasks transfer learning problem. Our results can be seen as an extension of their constructions to multiple-source tasks setting.

We define the optimal task

$$t^* = \arg\min_t \{\boldsymbol{Q}_t^2 + \frac{d\sigma_t^2}{N}\}.$$

Let $\delta^2 = (\boldsymbol{Q}_{t^*}^2 + \frac{d\sigma_{t^*}^2}{N})/64$. In general, we construct $T \times M$ parameters $\{\theta_{t,i}\}_{t\in[T],i\in[M]}$ with the t-th row corresponding to the hypothesis set of the t-th task.

We start by constructing the the hypothesis set of the target task and the task t^*. Let $\delta' = \boldsymbol{Q}_{t^*}/16 + \delta$. By definition, we have $\delta' \leq 1.5\delta$.

Consider the set $\Theta = \{\theta : \|\theta\|_2 \leq 2\delta'\}$. Let $\{\theta_{t^*,1}, \ldots, \theta_{t^*,M}\}$ be a δ'-packing of the set in the L_2-norm ($\|\theta_{t^*,i} - \theta_{t^*,j}\|_2 \geq \delta'$). We can find the packing with $\log(M) \leq d\log(2)$. Since $\theta_{t^*,i}, \theta_{t^*,j} \in \Theta$, we also have $\|\theta_{t^*,i} - \theta_{t^*,j}\|_2 \leq 4\delta'$ for any $i, j \in [M]$.

Now we construct hypothesis set for the target task. For all $i \in [M]$, we choose $\theta_{T,i}$ such that $\|\theta_{T,i} - \theta_{t^*,i}\|_2 = \boldsymbol{Q}_{t^*}/16$. So the construction for the target tasks satisfies

$$\|\theta_{T,i} - \theta_{T,j}\|_2 \geq \delta' - \boldsymbol{Q}_{t^*}/16 \geq \delta/2 \quad \text{and} \quad \|\theta_{T,i} - \theta_{T,j}\|_2 \leq 4\delta' + \boldsymbol{Q}_{t^*}/16 \leq 5\delta'.$$

Now we discuss two cases. For any task t with $\boldsymbol{Q}_t \geq 5\delta'$, we randomly pick a parameter in the hypothesis set of the target task which we denote by $\tilde{\theta}_t$ and we set all $\theta_{t,i} = \tilde{\theta}_t$ for all $i \in [M]$. This construction is valid since any $\|\theta_{t,i} - \theta_{T,i}\|_2 \leq 5\delta' \leq \boldsymbol{Q}_t$.

For any task t with $\boldsymbol{Q}_t \leq 5\delta'$, we will use the same construction as we use for t^*.

Let J be a random variable uniformly over $[M]$ representing the true hypothesis. The samples for each task t is i.i.d. generated from the linear model described in Section 3.3.1 with a parameter $\theta_{t,J}$. Our goal is to show that on expectation, any algorithm will perform badly as in Theorem 3.1.

Let E_t be a random sample from task t given the true parameter being $\theta_{t,J}$. Similarly to (5.2) in Kalan et al. (2020), using Fano's inequality, we can conclude that

$$R_T^N(\Theta(\boldsymbol{Q})) \geq \delta^2 \left(1 - \frac{\log(2) + \sum_{t=1}^T n_t I(J; E_t)}{\log(M)}\right). \tag{3.13}$$

We proceed by giving an uniform bound on the mutual information. We will need the following lemma to upper bound the mutual information term.

Lemma 3.2 (Lemma 1 in Kalan et al. (2020))**.** *The mutual information between J and any*

sample E_t can be upper bounded by $I(J; E_t) \leq \frac{1}{M^2} \sum_{i,j} D_{KL}\left(\mathbb{P}_{\theta_{t,i}} \| \mathbb{P}_{\theta_{t,j}}\right)$, where $\mathbb{P}_{\theta_{t,i}}$ is the induced distribution by the parameter $\theta_{t,i}$. Furthermore we have

$$D_{KL}\left(\mathbb{P}_{\theta_{t,i}} \| \mathbb{P}_{\theta_{t,j}}\right) = \|\Sigma_t^{1/2}(\theta_{t,i} - \theta_{t,j})\|_2^2/(2\sigma_t^2) \leq C_0 \|\theta_{t,i} - \theta_{t,j}\|_2^2/(2\sigma_t^2).$$

Using Lemma 3.2, we bound the mutual information of any task t.

Lemma 3.3. *Under the constructions introduced above, the mutual information*

$$I(J; E_t) \leq \frac{512 C_0}{7\sigma_{t^*}^2} \delta'^2 \text{ for all } t \in [T].$$

Proof. For any task in the first case discussed above ($Q_t \geq 5\delta'$), the mutual information $I(J; E_t)$ is 0. Thus the statement holds trivially.

Now we discuss the second case above. By definition, we have

$$\boldsymbol{Q}_t^2 + \frac{d\sigma_t^2}{N} \geq \boldsymbol{Q}_{t^*}^2 + \frac{d\sigma_{t^*}^2}{N} = 64\delta^2. \tag{3.14}$$

Note that

$$\boldsymbol{Q}_t \leq 5\delta' = 5(\boldsymbol{Q}_{t^*,T}/16 + \delta) \leq 7.5\delta.$$

Plugging back into (3.14), we have $d\sigma_t^2/N \geq 7\delta^2$, and by definition we have $d\sigma_{t^*}^2/N \leq 64\delta^2$. Therefore, we have $\sigma_t^2 \geq 7\sigma_{t^*}^2/(64)$.

Since the constructions are the same for the second case, the mutual information can be uniformly bounded by

$$I(J, E_t) \leq \frac{1}{M^2} \sum_{i,j} \frac{32 C_0}{7\sigma_{t^*}^2} \|\theta_{t^*,i} - \theta_{t^*,j}\|_2^2 \leq \frac{512 C_0}{7\sigma_{t^*}^2} \delta'^2.$$

□

Finally, we follow the analysis in Section 7.4 of Kalan et al. (2020). Using Lemma 3.A on Equation (3.13), we have

$$R_T^N(\Theta(\boldsymbol{Q})) \geq \delta^2 \left(1 - \frac{\log(2) + N\frac{512 C_0}{7\sigma_{t^*}^2}\delta'^2}{\log(M)}\right).$$

Plugging in $\delta' = Q_{t^*}/16 + \delta$, we can conclude

$$R_T^N(\Theta(\boldsymbol{Q})) \gtrsim \boldsymbol{Q}_{t^*}^2 + \frac{d\sigma_{t^*}^2}{N}.$$

□

Proof of Theorem 2

Proof. We first show the lower bound of the first term within the maximization. We construct the following problem: we have $T-1$ tuples of parameters $\{(\theta_{1,i},\ldots,\theta_{T,i}\}_{i=1}^{T-1}$, where $\theta_{t,i}$ corresponds to the parameters of the t-th task. Let $\{\tilde{\theta}_i\}_{i=1}^{T-1}$ be a set of parameters that are 2δ-separated for some $\delta > 0$. The parameters of our source and target tasks are chosen in the following manners:

1. $\theta_{t,i} = \tilde{\theta}_t$ for all $t \in [T-1]$.

2. $\theta_{T,i} = \tilde{\theta}_i$ for all $i \in [T-1]$.

Two important properties of this construction is that 1) there is always one source task that is identical to the target task; 2) the information from source tasks can not help learn the target task.

Let J follow the uniform distribution over $[T-1]$. Assume we have $n_1,\ldots,n_M$ and n_T be the number of observations for $T-1$ source tasks and target task from parameter $(\theta_{1,J},\ldots,\theta_{T-,J})$, respectively.

Proposition 3.2. *Since J is independent of $(\theta_{1,J},\ldots,\theta_{T-1,J})$, we have the mutual information $I(J;\theta_{1,J},\ldots,\theta_{T-1,J}) = 0$.*

Let ψ be any test statistics that maps our dataset to an index. For all ψ, by Fano's Lemma, we can conclude that

$$\begin{aligned}\tilde{R}_T^N(\tilde{\Theta}(\boldsymbol{Q})) &\geq \delta^2 \frac{1}{M}\sum_{i=1}^{M}\mathbb{P}\{\psi(S_1^{n_1},\ldots,S_M^{n_M},S_T^{n_T}) \neq j\} \\ &\geq \delta^2\left(1 - \frac{I(J;\psi(S_1^{n_1},\ldots,S_T^{n_T})) + \log(2))}{\log(T-1)}\right) \end{aligned} \tag{3.15}$$

To proceed, we analyze the mutual information

$$\begin{aligned} & I(J;\psi(S_1^{n_1},\ldots,S_T^{n_T})) \\ \leq\ & I(J;S_1^{n_1},\ldots,S_T^{n_T}) \\ & \text{(By the independence of } S_1^{n_1},\ldots,S_{T-1}^{n_{T-1}} \text{ and } S_T^{n_T}) \\ \leq\ & I(J;S_1^{n_1},\ldots,S_{T-1}^{n_{T-1}}) + I(J;S_T^{n_T}) \\ =\ & I(J;S_T^{n_T}). \end{aligned}$$

Let E be a random sample from the target task. We follow the analysis from Kalan et al. (2020), which construct $\tilde{\theta}$ by the 2δ-packing of the set

$$\{\theta : \theta \in \mathbb{R}^d, \|\theta\|_2 \leq 4\delta\}.$$

Then we can find such packing as long as $T - 1 \leq d \log(2)$. The mutual information by the above construction gives

$$I(J; S_T^{n_T}) \leq n_T I(J; E) \leq n_T \frac{32\delta^2}{\sigma_T^2}.$$

By choosing the optimal $\delta^* = \log((T-1)/2)\sigma_T^2/(64 n_T)$, we have for some $c > 0$,

$$\tilde{R}_T^N \gtrsim \frac{\sigma_T^2 \log(T-1)}{n_T}.$$

Note that the lower bound by Theorem 3.1 still applies here. In total, since $n_T \leq N$, we have

$$\tilde{R}_T^N \gtrsim \frac{\sigma_T^2 \log(T)}{N} + \min_t \frac{d\sigma_t^2}{N}.$$

The above analysis only works when $Q_{sub}^2 \geq \delta^* = \log((T-1)/2)\sigma_T^2/(64 n_T)$. Otherwise, one will at least suffer Q_{sub}^2 plus the learning difficulty term $\min_t \frac{d\sigma_t^2}{N}$. □

Proof of Theorem 3.3 Since the number of observations for each source task is $N/(2T-2)$, we notice that

$$L_t(\hat{\theta}_{t^*}) - L_t(\theta_{t^*}^*) \lesssim \frac{C_0 d T \sigma_{t^*}^2 \log(T/\delta)}{N}.$$

Using Assumption 3.1 and the definition of the loss function, we have

$$\|\hat{\theta}_{t^*} - \theta_{t^*}^*\|_2^2 \leq \frac{C_0 d T \sigma_{t^*}^2 \log(T/\delta)}{C_1 N}.$$

The proof of Theorem 3.3 is similar to many proofs of generalization bound. We let the empirical loss on the target task w.r.t θ be

$$\hat{L}_T(\theta) = \frac{2}{N} \sum_{i=1}^{N/2} \left(Y_{T,i} - X_{T,i}^T \theta\right)^2.$$

Write $Y_{T,i} = X_{T,i}^T \theta_T^* + \epsilon_{T,i}$. Let $\hat{\Sigma}_T = \frac{1}{n_T} \sum_{i=1}^{n_T} X_{T,i} X_{T,i}^T$ be the sample covariance matrix. We

start with

$$\begin{aligned}
&L_T(\hat{\theta}_{t^*}) - \min_{t\in[T-1]} L_T(\hat{\theta}_t) \\
&\leq \hat{L}_T(\hat{\theta}_{t^*}) - \min_{t\in[T-1]} \hat{L}_T(\hat{\theta}_t) + 2\max_{t\in[T-1]} |L_T(\hat{\theta}_t) - \hat{L}_T(\hat{\theta}_t)| \\
&= 2\max_{t\in[T-1]} |L_T(\hat{\theta}_t) - \hat{L}_T(\hat{\theta}_t)|,
\end{aligned}$$

where the last equality is based on the definition of t^*.

Now we bound the difference term. Let $n_T = N/2$.

Lemma 3.4. *With a probability at least $1-\delta$, we have*

$$|L_T(\hat{\theta}_t) - \hat{L}_T(\hat{\theta}_t)| \lesssim \frac{C_1 C_2^2 d\log(T/\delta)\sigma_T^2}{n_T} + \sqrt{\frac{d+\log(\delta)}{n_T}} \text{ for all } t \in [T-1].$$

Proof. We make the following decomposition.

$$\begin{aligned}
&|L_T(\hat{\theta}_t) - \hat{L}_T(\hat{\theta}_t)| \\
&= |\|\hat{\theta}_t - \theta_T^*\|_{\Sigma_T}^2 + \sigma_T^2 - \frac{1}{n_T}\sum_{i=1}^{n_T}[X_{T,i}^T(\hat{\theta}_t - \theta_T^*) - \epsilon_{T,i}]^2| \\
&\leq \|\hat{\theta}_t - \theta_T^*\|_{\Sigma_T - \hat{\Sigma}_T}^2 + |\sigma_T^2 - \frac{1}{n_T}\sum_{i=1}^{n_T}\epsilon_{T,i}^2| + \frac{1}{n_T}|\sum_{i=1}^{n_T} X_{T,i}^T(\hat{\theta}_t - \theta_T^*)\epsilon_{T,i}|.
\end{aligned}$$

Now we bound the three terms above separately. The second term is the concentration for $\chi^2(n_T)$ distribution. We have with a probability at least $1-\delta/(3T)$, $|\sigma_T^2 - \frac{1}{n_T}\sum_{i=1}^{n_T}\epsilon_{T,i}^2| \lesssim \sigma_T^2(\sqrt{\log(3T/\delta)/n_T} + \log(3T/\delta)/n_T)$.

To proceed, we consider the concentration of sample covariance matrix.

Lemma 3.5 (Matrix Hoeffding's inequality)**.** *Let $X_1, \ldots, X_n$ be centered, independent, symmetric, $d\times d$ random matrices that are sub-Gaussian with parameters $V_1, \ldots, V_n$. Then for all $\delta > 0$ with a probability $1-\delta$,*

$$\|\frac{1}{n}\sum_{i=1}^n X_i\|_{op} \leq \sqrt{\log(2d/\delta)\frac{2\sigma^2}{n}}, \text{ where } \sigma^2 = \|\frac{1}{n}\sum_{i=1}^n V_i\|.$$

Using the boundedness of both $\hat{\theta}_t$ and θ_T^*, $\|\hat{\theta}_t - \theta_T^*\|_2 \lesssim C_2$. Then applying Lemma 3.5, we have with a probability at least $1-\delta/(3T)$,

$$\|\hat{\theta}_t - \theta_T^*\|_{\Sigma_T - \hat{\Sigma}_T}^2 \lesssim C_2^2 C_0 \frac{\log(Td/\delta)}{n_T}.$$

For the third term, we apply the martingale concentration inequality on the sum $\sum_{i=1}^{n_T} X_{T,i}^T(\hat{\theta}_t - \theta_T^*)\epsilon_{T,i}$, we have with a probability at least $1-\delta/(3T)$, we have for all $t \in [T-1]$

$$\frac{1}{n_T}|\sum_{i=1}^{n_T} X_{T,i}^T(\hat{\theta}_t - \theta_T^*)\epsilon_{T,i}| \lesssim \frac{C_0 C_2 \sigma_T^2 \log(T/\delta)}{n_T}.$$

□

Proof of Proposition 3.1

Proof. The target task is basically minimizing the empirical loss over $T-1$ estimators. We first apply the standard generalization bound with Radermacher complexity

$$L_T(\hat{f}_{t^*}) \leq \min_{t\in[T-1]} L_T(\hat{f}_t) + \sqrt{\frac{2\log(T-1)}{N}} + c\sqrt{\frac{2\log(1/\delta)}{N}},$$

where c is a universal constant.

To proceed, we bound $\min_{t\in[T-1]} L_T(\hat{f}_t)$. By the assumption, we have $L^* = \min_t L_T(f_t^*)$. Let the task that realizes the minimization be t'. Using Assumption 3.2, we have

$$\min_{t\in[T-1]} L_T(\hat{f}_t) \leq L_T(\hat{f}_{t'}) = \mathbb{E}_{(X_T,Y_T)} l(\hat{f}_{t'}, Y_T) \leq L^* + L_1 \mathbb{E}_{X_T}\|\hat{f}_{t'} - f_{t'}^*\|^2 \leq L^* + \frac{L_1}{L_2}(L_{t'}(\hat{f}_{t'}) - L_{t'}^*).$$

We can apply the generalization bound on $L_{t'}(\hat{f}_{t'}) - L_{t'}^*$, which gives us the result.

□

Proof of Theorem 3.4

Proof of Lemma 3.1 We will borrow some techniques from Du et al. (2020) for the proof of Theorem 3.4. We start with the proof of Lemma 3.1, which provides a valid confidence set for the unknown parameters $B^*\beta_t^*$. First, we let $X_{i,t}^{(1)}$ be the covariance matrix of the first split $S_{i,t}^{(1)}$.

Claim 3.1 (Covariance concentration on the first split.)**.** *For $\delta \in (0,1)$, there exists a constant $\gamma_1 > 0$ such that with a probability at least $1-\delta/10$, we have*

$$0.9\Sigma_t \prec \frac{2}{n_{i,t}} X_{i,t}^{(1)T} X_{i,t}^{(1)} \prec 1.1\Sigma_t \quad \textit{for all } i \in \{i' \in [N] : n_{i',t} \geq \gamma_1(d + \log(N/\delta))\}.$$

Claim 3.2 (Covariance concentration on the second split.)**.** *For $\delta \in (0,1)$, there exists some $\gamma_2 > 0$ such that for any given $B \in \mathbb{R}^{d\times 2k}$ that is independent of $X_{i,t}^{(2)}$, with a probability at*

least $1-\delta/10$*, we have*

$$0.9B^T\Sigma_t B \prec \frac{2}{n_{i,t}} B^T X_{i,t}^{(2)T} X_{i,t}^{(2)} B \prec 1.1 B^T \Sigma_t B \quad \textit{for all } i \in \{i' \in [N] : n_{i',t} \geq \gamma_2(d+\log(N/\delta)).$$

Let $\gamma = \max\{\gamma_1, \gamma_2\}$. Recall that $\kappa = C_0/C_1$. Then the good events in Claim 3.1 and 3.2 hold for all $i : n_{i,*} \geq \lceil \gamma(d+\log(N/\delta))\rceil$. We first apply the Claim A.3 in Du et al. (2020), which guarantees the loss on the source training data. We rephrase it here as Lemma 3.6. Note that the only difference is that we require the good events hold for all $i : n_{i,t}$'s are sufficiently large.

Lemma 3.6 (Claim A3 in Du et al. (2020))**.** *With a probability at least* $1-\delta/5$*, we have*

$$\sum_{t=1}^{T} \|X_{i,t}^{(1)}(\hat{B}_i\hat{\beta}_{i,t} - B^*\beta_t^*)\|_2^2 \lesssim \sigma^2 \left(kT + dk\log(\kappa i/T) + \log(N/\delta)\right), \tag{3.16}$$

for all $i : n_{i,*} > \lceil \gamma(d+\log(N/\delta))\rceil$

Note that $X_{i,t}^{(1)}\hat{B}_i\hat{\beta}_{i,t} = P_{X_{i,t}^{(1)}\hat{B}_i} Y_t = P_{X_{i,t}^{(1)}\hat{B}_i}(X_{i,t}^{(1)}B^*\beta_t^* + z_t)$. To proceed, for any fixed $t' \in [T]$, we have

$$\begin{aligned}
&\sigma^2 \left(kT + dk\log(\kappa i/T) + \log(N/\delta)\right) \\
\gtrsim &\sum_{t=1}^{T} \|X_{i,t}^{(1)}(\hat{B}_i\hat{\beta}_{i,t} - B^*\beta_t^*)\|_2^2 \\
= &\sum_{t=1}^{T} \|P_{X_t\hat{B}_i}(X_{i,t}^{(1)}B^*\beta_t^* + z_t)\|_2^2 \\
\geq &\sum_{t=1}^{T} \|P_{X_t\hat{B}_i} X_{i,t}^{(1)}B^*\beta_t^*\|_2^2 \\
\geq &\, 0.9\sum_{t=1}^{T} \frac{n_{i,t}}{2} \|P_{\Sigma_t^{1/2}\hat{B}_i}\Sigma_t B^*\beta_t^*\|_2^2 \quad \text{(Using Claim 3.1)} \\
\geq &\, 0.45C_1 \sum_{t=1}^{T} n_{i,t} \|P_{\Sigma_{t'}^{1/2}\hat{B}_i}\Sigma_{t'} B^*\beta_t^*\|_2^2 \\
= &\, 0.45C_1 \sum_{j=1}^{i} \|P_{\Sigma_{t'}^{1/2}\hat{B}_i}\Sigma_{t'} B^*\beta_{t_j}^*\|_2^2
\end{aligned}$$

Then we have

$$\begin{aligned}
&\|\hat{B}_i\hat{\beta}_{t'} - B^*\beta^*_{t'}\|_2^2 \\
&\leq \frac{1}{C_1}\|\Sigma^2_{t'}(\hat{B}_i\hat{\beta}_{t'} - B^*\beta^*_{t'})\|_2^2 \\
&\leq \frac{1}{0.9C_1 n_{i,t'}}\|X^{(2)}_{i,t'}(\hat{B}_i\hat{\beta}_{t'} - B^*\beta^*_{t'})\|_2^2 \quad \text{(Using Claim 3.2)} \\
&= \frac{1}{0.9C_1 n_{i,t'}}\left(\|P_{X^{(2)}_{i,t'}\hat{B}_i}X^{(2)}_{i,t'}B^*\beta^*_{t'}\|_2^2 + \|P_{X^{(2)}_{i,t'}\hat{B}}z_{t'}\|_2^2\right)
\end{aligned}$$

For the second term above,

$$\frac{1}{\sigma^2}\|P_{X^{(2)}_{i,t'}\hat{B}}z_{t'}\|_2^2 \sim \chi^2(k).$$

Thus with a probability at least $1-\delta$, $\|P_{X^{(2)}_{i,t'}\hat{B}}z_{t'}\|_2^2 \lesssim k + \log(NT/\delta)$ for all $t' \in [T]$ and $i > T\lceil\gamma(d+\log(N/\delta))\rceil$. Therefore, we obtain the bound: for all $i > T\lceil\gamma(d+\log(N/\delta))\rceil$ and all $t' \in [T]$, it holds that

$$\begin{aligned}
\|\hat{B}_i\hat{\beta}_{t'} - B^*\beta^*_{t'}\|_2^2 &\lesssim \frac{\sigma^2}{C_1}\left(\frac{kT + dk\log(\kappa i/T) + \log(N/\delta)}{C_1 n_{i,t'}/C_5} + \frac{k+\log(NT/\delta)}{n_{i,t'}}\right) \\
&\lesssim \frac{C_5\sigma^2 dk\log(\kappa N\delta/T)}{C_1^2 n_{i,t'}}.
\end{aligned}$$

Proof of the full theorem Now we prove the full theorem. By Assumption 3.3, we convert our target $\lambda_d(\sum_{i=1}^N \beta^*_{t_i}\beta^{*T}_{t_i})$ to $\lambda_d(\sum_{i=1}^N B^*\beta^*_{t_i}\beta^{*T}_{t_i}B^{*T})$:

$$\frac{1}{N}\lambda_d(\sum_{i=1}^N \beta^*_{t_i}\beta^{*T}_{t_i}) \geq \frac{1}{NC_4}\lambda_d(\sum_{i=1}^N B^*\beta^*_{t_i}\beta^{*T}_{t_i}B^{*T}).$$

Then we follow the standard decomposition framework of UCB analysis:

$$\frac{1}{N}\lambda_d(\sum_{i=1}^N B^*\beta^*_{t_i}\beta^{*T}_{t_i}B^{*T}) = \frac{1}{N}\left(\lambda_k(\sum_{i=1}^N \tilde{\theta}_i\tilde{\theta}_i^T) + \lambda_k(\sum_{i=1}^N B^{*T}\beta^*_{t_i}\beta^{*T}_{t_i}B^*) - \lambda_k(\sum_{i=1}^N \tilde{\theta}_i\tilde{\theta}_i^T)\right). \tag{3.17}$$

Our proof proceeds by first showing

$$\frac{1}{N}\lambda_k(\sum_{i=1}^N \tilde{\theta}_i\tilde{\theta}_i^T) \geq \lambda/d,$$

which is usually interpreted as optimism. Then we bound the difference term

$$\frac{1}{N}\left(\lambda_k(\sum_{i=1}^{N} B^*\beta_{t_i}^*\beta_{t_i}^{*T}B^{*T}) - \lambda_k(\sum_{i=1}^{N}\tilde{\theta}_i\tilde{\theta}_i^T)\right),$$

which is expected to vanish when N becomes large.

Proof of optimism. We apply Lemma 3.1 and have $\tilde{\theta}_i \in \mathcal{B}_{i,t}^{\alpha}$. Since $B^*\beta_t^* \in \mathcal{B}_{i,t}^{\alpha}$ for all $t \in [T], i \in [N]$, it is easy to show that the greedy selection over $\mathcal{B}_{i,t}^{\alpha}$ will lead to

$$\lambda_k(\sum_{i=1}^{N}\tilde{\theta}_i\tilde{\theta}_i^T) \geq \lambda(\frac{N}{k} - 1).$$

We prove by induction. Assume at any step n, we have for all $\|\nu\|_2 = 1$,

$$\nu^T\sum_{i=1}^{n}\tilde{\theta}_i\tilde{\theta}_i^T\nu \geq \lambda(i/k - 1).$$

We will show that at the step $n + k$, we will at least have

$$\nu^T\sum_{i=1}^{n}\tilde{\theta}_i\tilde{\theta}_i^T\nu \geq \lambda(i/k).$$

The proof is simple, if there exists a ν such that the above inequality fails, we will select a task that brings it to $\lambda(i/k)$. This process can be done at most k times.

Upper bounding the differences. We first write the difference of eigenvalues in terms of the difference of the matrices. We will use a trick here.

$$
\begin{aligned}
&\lambda_k(\sum_{i=1}^{N} B^*\beta_{t_i}^*\beta_{t_i}^{*T}B^{*T}) - \lambda_k(\sum_{i=1}^{N}\tilde{\theta}_i\tilde{\theta}_i^T) \\
&= \min_{\|\nu\|_2=1} \nu^T \sum_{i=1}^{N} B^*\beta_{t_i}^*\beta_{t_i}^{*T}B^{*T}\nu - \min_{\|\nu\|_2=1} \nu^T \sum_{i=1}^{N}\tilde{\theta}_i\tilde{\theta}_i^T\nu \\
&\geq \min_{\|\nu\|_2=1} \nu^T \sum_{i=1}^{N} B^*\beta_{t_i}^*\beta_{t_i}^{*T}B^{*T}\nu - \min_{\|\nu\|_2=1} \nu^T \sum_{i=1}^{N}\tilde{\theta}_i\tilde{\theta}_i^T\nu \\
&\geq \min_{\|\nu\|_2=1} \left(\nu^T \sum_{i=1}^{N}(B^*\beta_{t_i}^*\beta_{t_i}^{*T}B^{*T} - \tilde{\theta}_i\tilde{\theta}_i^T)\nu\right) \\
&\geq \sum_{i=1}^{N} \min_{\|\nu\|_2=1} \nu^T(B^*\beta_{t_i}^*\beta_{t_i}^{*T}B^{*T} - \tilde{\theta}_i\tilde{\theta}_i^T)\nu \\
&\geq -\sum_{i=1}^{N} \|B^*\beta_{t_i}^* - \tilde{\theta}_i\|_2(\|B^*\beta_{t_i}^*\|_2 + \|\tilde{\theta}_i\|_2) \\
&\geq -2C_5\sum_{i=1}^{N} \|B^*\beta_{t_i}^* - \tilde{\theta}_i\|_2.
\end{aligned}
$$

Applying Lemma 3.1 and by the construction of the confidence set $\mathcal{B}_{i,t}^{\alpha}$, we have

$$
\|B^*\beta_{t_i}^* - \tilde{\theta}_i\|_2 \lesssim \sqrt{\frac{C_5\sigma^2 dk\log(\kappa N\delta/T)}{C_1^2 n_{i,t}}}.
$$

Thus,

$$
\begin{aligned}
&\lambda_k(\sum_{i=1}^{N} B^*\beta_{t_i}^*\beta_{t_i}^{*T}B^{*T}) - \lambda_k(\sum_{i=1}^{N}\tilde{\theta}_i\tilde{\theta}_i^T) \\
&\qquad\qquad \gtrsim -\sum_i \sqrt{\frac{C_5^2\sigma^2 dk\log(\kappa N\delta/T)}{C_1^2 n_{i,t}}} \gtrsim -\sum_i \sqrt{\frac{C_5^2\sigma^2 dkTN\log(\kappa N\delta/T)}{C_1^2}}.
\end{aligned}
$$

Plugging this back to the decomposition term (3.17) we arrive the final bound.

Proof of Theorem 3.5 Assume we have T tasks in total. We pick a set of orthogonal vectors $\{\beta_1, \ldots, \beta_k\} \in \mathbb{R}^k$ with $\|\beta_i\|_2^2 = \lambda$. We first construct a simple instance in the following way: the first k tasks are diverse such that $(\beta_i^*, \ldots, \beta_k^*) = (\beta_1, \ldots, \beta_k)$. Then all the other tasks share the same parameter β_1. We denote the instance by v. This construction is

hard for naive task scheduler that evenly allocates samples to all the tasks since the direction for β_1^* will be over-exploited.

We evenly divide T tasks into $M = \lfloor T/k \rfloor$ blocks. Let T_m be the total number of visits in the m-th block. For any task scheduler, there exists $m' \in [M]$ such that $\mathbb{E}[T_m] \leq \frac{N}{M}$ by pigeonhole theorem.

Then we construct another instance denoted by v' such that v is the same as v' except for that the m-th block has the parameters $(2\beta_1, \ldots, 2\beta_k)$ for the k tasks in the block.

Let P_v and $P_{v'}$ be the probability measure on the linear regression model with true parameter defined in Section 3.3.1 for v and v'.

Define $\Delta_{N,k}(\mathcal{T}, v)$ be the expected difference using task scheduler $\mathcal{T}$ on instance v, i.e.

$$\Delta_{N,k}(\mathcal{T}, v) \coloneqq \max_{t_1,\ldots,t_N} \lambda_k(\sum_{i=1}^{N} \beta_{t_i}^* \beta_{t_i}^{*T}) - \lambda_{N,k}.$$

Thus, applying Bretagnolle–Huber inequality (Theorem 14.2 (Lattimore and Szepesvári, 2020)) we have

$$\Delta_{N,k}(\mathcal{T}, v) + \Delta_{N,k}(\mathcal{T}, v') \geq \frac{N\lambda}{2k}(P_v(T_1 \leq N/M) + P_{v'}(T_1 > N/M)) \geq \frac{N\lambda}{2k} \exp(-D(P_v, P_{v'})).$$

where $D(P, Q)$ is the relative entropy between distributions P and Q.

Then we apply Lemma 15.1 (Lattimore and Szepesvári, 2020), which we rephrase here.

Lemma 3.7. *Let P_t and P_t' be the probability measure of the t-th task using true parameters from v and v', respectively. We also let $\bar{T}_t$ be the number of observations on the t-th task. Then we have*

$$D(P_v, P_{v'}) = \sum_{t=1}^{T} \mathbb{E}_v[\bar{T}_t] D(P_t, P_t') \leq \mathbb{E}_v[T_m] \max_{t=m(k-1)+1,\ldots,mk} D(P_t, P_t').$$

Now since $D(P_t, P_t') = \|\beta_t - \beta_t^*\|^2/(2\sigma^2) = \lambda^2/(2\sigma^2)$, we have

$$\Delta_{N,k}(\mathcal{T}, v) + \Delta_{N,k}(\mathcal{T}, v') \geq \frac{N\lambda}{2k} \exp(\frac{N\lambda^2}{2M\sigma^2}).$$

Choosing $\lambda = \frac{2M\sigma^2}{N}$, we have

$$\Delta_{N,k}(\mathcal{T}, v) + \Delta_{N,k}(\mathcal{T}, v') \geq \sigma\sqrt{NM}/k = \sigma\sqrt{NT/k^3}.$$

Proof of Theorem 3.6

Proof. We follow the standard procedure of the convergence analysis of SGD. Let t_i be the task that the task scheduler chooses at the step i. Let $v_i^{(t)}$ be the virtual gradient calculated at the step i if task t is scheduled, i.e.

$$v_i^{(t)} = x_i^{(t)T}(\theta_t^* - \theta_i)x_i^{(t)} + \epsilon_i^{(t)}x_i^{(t)},$$

where $\epsilon_i^{(t)}$ and $x_i^{(t)}$ is the random noise and the input sampled at the step i from task t. To start with, let $\theta_{i+1}^{(t)}$ be the virtual next step if task t is scheduled. We have

$$\theta_{i+1}^{(t)} - \theta_T^* = \theta_i - \theta_T^* - \eta_i v_i^{(t)}.$$

By algebra, we derive

$$\begin{aligned}
&\|\theta_{i+1}^{(t)} - \theta_T^*\|^2 - \|\theta_i - \theta_T^*\|^2 \\
&= -2\eta_i(\theta_i - \theta_T^*)^T x_i^{(t)} x_i^{(t)T}(\theta_i - \theta_T^*) + \\
&\quad 2\eta_i(\theta_i - \theta_T^*)^T x_i^{(t)} x_i^{(t)T}(\theta_t^* - \theta_T^*) - \\
&\quad 2\eta_i x_i^{(t)T}(\theta_i - \theta_T^*)\epsilon_i + \eta_i^2\|v_i^{(t)}\|^2.
\end{aligned}$$

Taking expectations over $x_i^{(t)}$ and $\epsilon_i^{(t)}$ and arrange the equation, we have

$$\|\theta_i - \theta_T^*\|^2 = \frac{\|\theta_i - \theta_T^*\|^2 - \mathbb{E}_{t,i}\|\theta_{i+1}^{(t)} - \theta_T^*\|^2}{2\eta_i} + (\theta_i - \theta_T^*)^T(\theta_t^* - \theta_T^*) + \frac{\eta_i}{2}\|v_i^{(t)}\|^2., \tag{3.18}$$

where $\mathbb{E}_{t,i}$ takes marginal expectation over the randomness of $x_i^{(t)}$ and $\epsilon_i^{(t)}$. Note that

$$L_T(\theta_i) - \sigma_T^2 = \|\theta_i - \theta_T^*\|^2.$$

Plugging this into Equ. (3.18), we have

$$L_T(\theta_i) - \sigma_T^2 = \frac{\|\theta_i - \theta_T^*\| - \mathbb{E}_{t,i}\|\theta_{i+1}^{(t)} - \theta_T^*\|^2}{2\eta_i} + (\theta_i - \theta_T^*)^T(\theta_t^* - \theta_T^*) + \frac{\eta_i}{2}\|v_i^{(t)}\|^2.$$

Since this holds for any t, we let $t = t^*$ and note that by the definition of the task scheduler with accurate prediction gain estimate,

$$\mathbb{E}\|\theta_i - \theta_T^*\|^2 \leq \mathbb{E}\|\theta_i^{(t)} - \theta_T^*\|^2.$$

We have

$$\begin{aligned}
&\mathbb{E}[L_T(\theta_i)] - \sigma_T^2 \\
&\leq \mathbb{E}\frac{\|\theta_i^{(t^*)} - \theta_T^*\|^2 - \|\theta_{i+1}^{(t^*)} - \theta_T^*\|^2}{2\eta_i} + \mathbb{E}(\theta_i - \theta_T^*)^T(\theta_{t^*}^* - \theta_T^*) + \frac{\eta_i}{2}\mathbb{E}\|v_i^{(t^*)}\|^2.
\end{aligned}$$

Summing over all $i = 1, \ldots, N$, we have

$$\begin{aligned}
&\sum_i \mathbb{E}[L_T(\theta_i)] - N\sigma_T^2 \\
&\leq \sum_{i=1}^N \mathbb{E}\frac{\|\theta_i^{(t^*)} - \theta_T^*\|^2 - \|\theta_{i+1}^{(t^*)} - \theta_T^*\|^2}{2\eta_i} + \sum_{i=1}^N \mathbb{E}(\theta_i - \theta_T^*)^T(\theta_{t^*}^* - \theta_T^*) + \sum_{i=1}^N \frac{\eta_i}{2}\mathbb{E}\|v_i^{(t^*)}\|^2.
\end{aligned}$$

Note that taking $\eta_i = 1/i$, the first term on the right hand side collapses to $-N\mathbb{E}\|\theta_{N+1}^{(t)} - \theta_T^*\|^2 \leq 0$.

To proceed, we have

$$\begin{aligned}
&\sum_{i=1}^N \mathbb{E}(\theta_i - \theta_T^*)^T(\theta_i - \theta_T^*) \\
&\leq \sum_{i=1}^N \mathbb{E}(\theta_i - \theta_T^*)^T(\theta_{t^*}^* - \theta_T^*) + \sum_i \frac{\eta_i}{2}\mathbb{E}\|v_i^{(t^*)}\|^2 \\
&\leq \sum_{i=1}^N \mathbb{E}(\theta_i - \theta_T^*)^T(\theta_{t^*}^* - \theta_T^*) + \log(N)(\sigma_{t^*}^2 d + C_5) \\
&\leq \sqrt{\sum_{i=1}^N \mathbb{E}\|\theta_i - \theta_T^*\|^2}\sqrt{\sum_{i=1}^N \|\theta_{t^*}^* - \theta_T^*\|^2} + \log(N)(\sigma_{t^*}^2 d + C_5).
\end{aligned}$$

By solving the inequality, we have

$$\begin{aligned}
\sum_{i=1}^N \mathbb{E}\|\theta_i - \theta_T^*)^T\|^2 &\leq \sum_{i=1}^N \mathbb{E}\|\theta_{t^*}^* - \theta_T^*\|^2 + \log(N)(\sigma_{t^*}^2 d + C_5) \\
&\leq \sum_{i=1}^N \Delta_{t^*,T}^2 + \log(N)(\sigma_{t^*}^2 d + C_5)
\end{aligned}$$

Divided by N on both side, we reach Theorem 3.6. □

3.B Additional details for simulations

We provide the details for EXP3 algorithm. The EXP3 task scheduler picks tasks from $\{1, \ldots, T\}$, which is the action set for the bandit problem. In our experiment, $\eta = 0.85$.

Algorithm 2 Exponential-weight Algorithm for Exploration and Exploitation (Exp3)

Input: T, K, η
Set $\hat{S}_{0,t} = 0$ for all $t \in [T]$.
for $i = 1, \ldots, n$ **do**
    Calculate $P_{it} \leftarrow \frac{\exp\left(\eta \hat{S}_{i-1,t}\right)}{\sum_{t' \in [T]} \exp\left(\eta \hat{S}_{i-1,t'}\right)}$ for all $t \in [T]$
    Sample t_i from P_i and receive reward r_i
    Calculate $\hat{S}_{i,t} \leftarrow \hat{S}_{i-1,t} + 1 - \frac{\mathbb{1}_{\{t_i = a\}}(1 - r_i)}{P_{it}}$
end for

CHAPTER 4

Multitask Contextual Bandits

In this chapter, we take one step beyond supervised learning and study contextual bandit problem, a sequential decision-making problem, which is a special case of reinforcement learning. We identify an interesting structural assumption called *Funnel Structure* that allows significant MTL improvement. Funnel structure, a well-known concept in the marketing field, occurs in those systems where the decision maker interacts with the environment in a layered manner receiving far fewer observations from deep layers than shallow ones. For example, in the email marketing campaign application, the layers correspond to Open, Click and Purchase events. Conversions from Click to Purchase happen very infrequently because a purchase cannot be made unless the link in an email is clicked on.

We formulate this challenging decision making problem as a contextual bandit with funnel structure and develop a multi-task learning algorithm that mitigates the lack of sufficient observations from deeper layers. We analyze both the prediction error and the regret of our algorithms. We verify our theory on prediction errors through a simple simulation. Experiments on both a simulated environment and an environment based on real-world data from a major email marketing company show that our algorithms offer significant improvement over previous methods. [1]

4.1 Introduction

We consider decision making problems arising in online recommendation systems or advertising systems (Pescher et al., 2014; Manikrao and Prabhakar, 2005). Traditional approaches to these problems only optimize a single reward signal (usually purchase or final conversion), whose positive rate can be extremely low in some real recommendation systems. This reward sparsity can lead to a slow learning speed and unstable models. Nevertheless, some non-sparse signals are usually available in these applications albeit in a layered manner. These

[1] This chapter is based on my paper published at AISTATS 2020 with Amirhossein Meisami and Ambuj Tewari (Xu et al., 2020)

signals can be utilized to boost the performance of the final sparse signal. As a special case, funnel structure generates a sequence of binary signals by layers and the observations are cumulative products of the sequence. An example of email conversion funnel is shown in Figure 4.1.

Funnel structure characterizes a wide range of problems in advertising systems. In the email campaign problem, the learning agent decides the time to send emails to maximize purchases. Apart from the final reward on purchase, we also observe the opening and clicking status of an email. There are also papers studying the participation funnel in MOOCs (Clow, 2013; Borrella et al., 2019). Students go through the layers of Awareness, Registration, Activity, Progress and Completion until they drop or complete the course. For both of the funnels, the drop-off fraction at each layer is large. For example, in email campaigns, the conversion rates are typically 10% for Open, 4% for Click and 2.5% for Purchase.

Funnel structure studies in the marketing field. Conversion funnel has been at the center of the marketing literature for several decades (Howard and Sheth, 1969; Barry, 1987; Mulpuru, 2011). This line of work focuses on the attribution of advertising effects and is more interested in analyzing buyers' behavior at each layer. Schwartz et al. (2017) learns contextual bandit with a Funnel Structure. However, their model directly learns on the final purchase signal and signals on other stages are only used for performance evaluation. Hence, it lacks a comprehensive method that exploits the structural information of a funnel.

Contextual bandits. The decision making problem is modelled as a contextual bandit problem (Li et al., 2010, 2011; Beygelzimer et al., 2011) in this chapter. Previous works on contextual bandits mainly focus on a single reward. Drugan and Nowe (2013); Turğay et al. (2018) study multi-objective bandits by considering a Pareto regret, which optimizes the vector of rewards for different objectives, while our work focuses on optimizing the final reward by exploiting the whole task set.

Related works This work is the first work that applies the multi-task learning to contextual bandits with funnel structure. There has been works considering contextual bandits with a sequential transfer (Lazaric et al., 2013; Soare et al., 2014), a multi-task learning approach (Deshmukh et al., 2017) and a multi-goal setting (Drugan and Nowe, 2013; Turğay et al., 2018). However, none of their algorithms adapts to the funnel structure due to the imbalance in the sample sizes of different layers. On the other side, Schwartz et al. (2017) focuses on funnel structure contextual bandits without multi-task learning. Some Reinforcement Learning algorithms with auxiliary rewards (Jaderberg et al., 2016; Lin et al., 2019) can

also be applied to this problem. This line of work does not use the full information as it either learns on the output of the last layer or learns on an weighted average of different layers. Our experiments also verify that using the full information could improve performances.

4.2 Formulation

In this section, we introduce the formulation for funnel structure and discuss how the formulation applies to our email campaign problem. We also introduce the generalized linear model and the assumptions for our theoretical analyses.

Funnel structure. A funnel, denoted by $F = \{J, \mathcal{X}, (Z_1, \ldots, Z_J)\}$, consists of the number of layers $J \in \mathbb{N}$, feature space $\mathcal{X} \in \mathbb{R}^d$ for some $d > 0$ and a sequence of J mappings $(Z_1, \ldots, Z_J)$. Each Z_j is a mapping from feature space to $[0, 1]$. On each interaction, a funnel takes an input feature $x \in \mathcal{X}$ and generates a sequence of binary variables $z_1, \ldots, z_J$ from Bernoulli distributions with parameters $Z_1(x), \ldots, Z_J(x)$, respectively. Then it returns $r_1, \ldots r_J$, for $r_j = \prod_{s=1}^{j} z_s$, to the learning agent.

Email conversion funnel. We illustrate how the formulation applies to the email conversion funnel. Our email conversion funnel, as shown in Figure 4.1, has 3 layers representing Open, Click and Purchase, respectively. Every email sent to a user randomly generates r_1, r_2, r_3 representing whether the email is actually opened, clicked or purchased using the mechanism described above, while z_1, z_2, z_3 are the indicators for the three events given previous events happened.

On the sparsity of the funnel, if an email is never opened, neither click or purchase could happen. Out of all the emails sent to users, 10% of them were opened, 0.4% were clicked, and 0.01% led to a purchase. More generally, when there exists a $r_j = 0$, all the successors r_i's, $i > j$, become 0, which leads to unobservable $z_{j+1}, \ldots, z_J$. On average, given a feature x, the probability of observing z_j is $P_{j-1}(x)$, which decreases **exponentially** as the layers go deeper.

Contextual bandit with funnel structure. Our contextual bandit with funnel structure is denoted by $M = \{\mathcal{A}, \mathcal{X}, P_x, \{F_a\}_{a \in \mathcal{A}}\}$, where $\mathcal{A}$ is the finite action space with $|\mathcal{A}| = A$, $\mathcal{X}$ is the common context space, P_x is the context distribution and each arm $a \in \mathcal{A}$ is assigned a funnel denoted by $F_a = \{J, \mathcal{X}, (Z_1^a, \ldots, Z_J^a)\}$. On each round of t-th interaction, the environment generates a context $x_t \sim P_x$, the agent takes an action a_t and the funnel

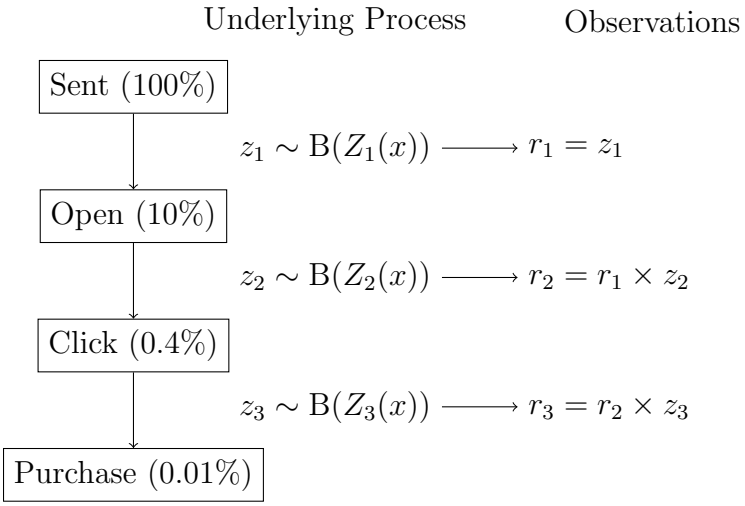


Figure 4.1: An illustration of the email conversion funnel. Given any input x, the profile information of the user, the funnel generates, $z_1, \dots z_3$, from Bernoulli distributions with parameter $Z_1(x), Z_2(x), Z_3(x)$, representing whether the email would be opened, clicked or purchased given the conversion of the previous layers happened. The observations $r_1, \dots, r_3$ represent whether the email is actually opened, clicked or purchased, respectively.

F_{a_t} returns the reward vector $(r_{1t}, \dots, r_{Jt})$ taken the input context x_t based on the process described above.

Note that our setting, when $J = 1$, differs from the contextual bandit setting in Chu et al. (2011); Filippi et al. (2010), where each arm has a unique context but the same mapping from context to reward function. Filippi et al. (2010) assumes a canonical exponential family density function. We consider a binomial signal, whose density might not be in canonical exponential family.

Assumptions. Most analyses on multi-task learning assume some similarities among tasks set to allow knowledge transfer. Here we assume a generalized linear model (GLM) and a prior-known hypothesis class over the unknown parameters for all the layers. The hypothesis class characterizes the relatedness across layers.

Assumption 4.1 (Generalized linear model)**.** *Assume all $Z_j = \mu(x^T \theta_j^*)$ for some mean function $\mu : \mathbb{R} \mapsto [0, 1]$ and θ_j^* is the true parameter of layer j. A throughout example of this chapter is the model for logistic regression, where $\mu(y) = 1/(1 + \exp(-y))$.*

We also assume that the mean function μ is Lipschitz continuous and convex.

Assumption 4.2. *We assume that μ is monotonically increasing and $\mu'(x) \geq c_\mu$ for all $x \in \mathcal{X}$. We also require that function μ satisfies $|\mu'(x)| \leq \kappa$.*

Assumption 4.3. *Assume* $\mathcal{X} \subset \{x \in \mathbb{R} : \|x\| \leq d_x\}$.

Generally, we assume that the joint parameter is from a hypothesis class. Two special cases of interest are introduced, upon which we design our practical algorithms.

Assumption 4.4 (Similarity assumption)**.** *Let* $\boldsymbol{\theta} = (\theta_1^T, \theta_2^T, \ldots, \theta_j^T)^T \in \mathbb{R}^{dJ}$ *and* $\boldsymbol{\theta}^*$ *is the joint vector for the true parameters. We assume* $\boldsymbol{\theta}^* \in \Theta_0 \subset \mathbb{R}^{dJ}$*. Throughout the chapter, we discuss two special cases:*

1. ***Sequential dependency:*** $\Theta_0 \coloneqq \{\boldsymbol{\theta} \in \mathbb{R}^{dJ} : \|\theta_j - \theta_{j-1}\|_2 \leq q_j$, *for* $j > 1$ *and* $\|\theta_1\| \leq q_1$ *for some* $q_1, \ldots q_J \in \mathbb{R}^+$.

2. ***Clustered dependency:*** $\Theta_0 \coloneqq \{\boldsymbol{\theta} \in \mathbb{R}^{dJ} : \exists \theta_0 \in \mathbb{R}^d, \|\theta_j - \theta_0\|_2 \leq q_j, \forall j \in [J]\}$ *for some* $q_1, \ldots q_J \in \mathbb{R}^+$.

For any set $\Theta \subset \mathbb{R}^{dJ}$, we denote the marginal set of task j by $\Theta[j]$ i.e., $\Theta[j] = \{\theta \in \mathbb{R}^d : \exists \boldsymbol{\theta} \in \Theta, \boldsymbol{\theta}_j = \theta\}$.

We first note that a hypothesis class over the joint parameters is a common assumption in multi-task learning literature (Maurer et al., 2016; Zhang and Yang, 2017; Pentina et al., 2015). Also, in another line of work focusing on transfer learning, the theoretical analyses often assume a discrepancy between tasks (Wang et al., 2019a). We argue in Appendix 4.A that under our GLM assumptions, the discrepancy assumption is almost the same as ours.

The role of diversity. Note that diversity continues to play an important role in the proposed dependencies. If the environment only has two tasks, then there will be a large distance between parameters $\|\theta_j - \theta_{j-1}\|$ or $\|\theta_j - \theta_0\|$. By adding more tasks to fill in the gap between original task set, we improve the diversity whiling having a smaller set of discrepancy q's.

4.3 Supervised learning

Before discussing the contextual bandits with funnel structure, we first consider the supervised learning scenario for a single funnel and seek a bound for the prediction error of each layer:

$$\mathrm{PE}_j = |\mu(x^T\theta_j^*) - \mu(x^T\hat{\theta}_j)|,$$

for some estimates $\hat{\theta}_1, \ldots, \hat{\theta}_J \in \mathbb{R}^d$.

For a single funnel, our algorithm learns on the dataset $\{x_i, r_{1i}, \ldots, r_{Ji}\}_{i=1}^n$ of size n. Let $n_j = \sum_{i=1}^n \mathbf{1}(r_{j-1,i} = 1)$ be the number available observations for layer j and $j_1, \ldots j_{n_j}$ be

the indices of these n_j samples, i.e. $r_{j-1,j_i} > 0$ for all $i \in [n_j]$. Let $z_{j,j_i} = r_{j,j_i}/r_{j-1,j_i}$. We denote the square loss function of layers j by

$$l_j(\theta) \coloneqq \sum_{i=1}^{n_j} (z_{j,j_i} - \mu(x_{j_i}^T \theta))^2. \tag{4.1}$$

4.3.1 Implications from a single-layered case

We first investigate a single-layered case and see how prior knowledge helps improve the upper bound on prediction error. Lemma 4.1 bounds the prediction error using either prior knowledge or collected samples. The proof of Lemma 4.1 is provided in Appendix 4.A.

Lemma 4.1. *Using the model defined above, assume $J = 1$ and the true parameter $\theta^* \in \Theta_0$. Let the dataset be $\{x_i, z_i\}_{i=1}^n$ and let $\tilde{\theta}$ be the solution that maximizes the L_2 function in (4.1) and $\hat{\theta}$ be its projection onto Θ_0. Let $q \coloneqq \sup_{\theta_1,\theta_2 \in \Theta_0} \|\theta_1 - \theta_2\|_2$. Then with a probability at least $1 - \delta$, we have*

$$\begin{aligned} |\mu(x^T\hat{\theta}) - \mu(x^T\theta^*)| &\le \kappa \min\{ \sup_{\theta_1,\theta_2 \in \Theta_0} |x^T(\theta_1 - \theta_2)|, \|x\|_{M_n^{-1}} \frac{4c_\delta}{c_\mu} \sqrt{\frac{d}{n \vee 1}} \} \\ &\le \kappa \min\{ d_x q, \|x\|_{M_n^{-1}} \frac{c_\delta}{c_\mu} \sqrt{\frac{d}{n}} \}, \end{aligned} \tag{4.2}$$

furthermore, we have a confidence set on θ^*,

$$\theta^* \in \{\theta : \|\hat{\theta} - \theta\|_{M_n} \le \frac{c_\delta}{c_\mu} \sqrt{\frac{d}{n}} \}, \tag{4.3}$$

where $M_n = \frac{1}{n} \sum_{i=1}^n x_i x_i^T$, $c_\delta = 80 d_x \sqrt{2 \ln(8/\delta)}$.

Equation (4.2) bounds the prediction error with a minimum of two terms. The first term in (4.2) is directly derived from prior knowledge. The second term is a parametric bound without any regularization (Srebro et al., 2010). In fact, (4.2) is a tight upper bound on prediction error as shown in Appendix 4.A.

Lemma 4.1 implies a **"transfer or learn"** scenario: when the sample size for the new task is not large enough, i.e. $n = o(1/q^2)$, it is more beneficial to directly apply the prior knowledge. Otherwise, one can drop the prior knowledge and use parametric bound.

4.3.2 Multi-task learning algorithm

Our algorithm, inspired by the "transfer or learn" idea, consists of two steps: 1) optimize the loss function for each layer within its marginal set and calculate the confidence set defined in

Algorithm 3 Regularized MTL for funnel structure

Input: number of layers J, hypothesis set Θ_0, dataset $\{x_i, r_{1i}, \ldots, r_{Ji}\}_{i=1}^n$ generated from the funnel, accuracy $\delta > 0$.
 # Calculate confidence set.
for $j = 1$ to J **do**
Solve $\bar{\theta}_j = \text{Proj}_{\Theta_0[j]}(\arg\min l_j(\theta))$.
Calculate $\hat{\Theta}_j$ defined in Equation (4.3) by

$$\hat{\Theta}_j = \{\theta : \|\theta - \bar{\theta}_j\|_{M_{j,n_j}} \leq \frac{c_\delta}{c_\mu}\sqrt{\frac{d}{n_j}}\},$$

for $M_{j,n_j} = \sum_{i=1}^{n_j} x_{j,j_i} x_{j,j_i}^T$.
end for
 # Re-estimate parameters.
Calculate joint set $\hat{\Theta} = \{\boldsymbol{\theta} : \boldsymbol{\theta}_j \in \hat{\Theta}_j \text{ for all } j \in [J]\}$.
Set $\Theta_1 \leftarrow \Theta_0 \cap \hat{\Theta}$.
for $j = 1$ to J **do**
Solve $\hat{\theta}_j = \text{Proj}_{\Theta_1[j]}(\arg\min l_j(\theta))$.
end for
Return $\hat{\theta}_1, \ldots, \hat{\theta}_J$.

(4.3); 2) take the intersection between the confidence set and Θ_0, which generates Θ_1. Then project the unconstrained solution onto the new set Θ_1. The details are shown in Algorithm 3, where $\text{Proj}_\Theta(\theta)$ denotes the projection of θ onto the set Θ.

4.3.3 Upper bound on prediction error

Directly applying Lemma 4.1 within the set Θ_1 gives us a bound on prediction error depending on the marginal set $\Theta_1[j]$ and n_j as shown in Corollary 4.1.

Corollary 4.1. *Let $\hat{\theta}_1, \ldots, \hat{\theta}_J$ be the estimates from Algorithm 3 and Θ_1 be the set defined in Algorithm 3. With a probability at least $1 - \delta$, for all $j \in [J]$, we have*

$$PE_j \leq \kappa \min\{\sup_{\theta_1, \theta_2 \in \Theta_1[j]} |x^T(\theta_1 - \theta_2)|, \|x\|_{M_n^{-1}} \frac{c_\delta}{c_\mu}\sqrt{\frac{d}{n_j}}\}. \tag{4.4}$$

However, it is more interesting to discuss the actual form of $\Theta_1 = \Theta_0 \cap \hat{\Theta}$ and its interactions with n_j under some special assumptions on Θ_0. As mentioned in Assumption 4.4, we consider two cases: sequential dependency and clustered dependency.

Recall that for the sequential dependency, we assume $\boldsymbol{\theta}^* \in \Theta_0 := \{\boldsymbol{\theta} \in \mathbb{R}^{dJ} : \|\theta_j - \theta_{j-1}\|_2 \leq$

q_j, for $j > 1$ and $\|\theta_1\|_2 \leq q_1\}$ for some $q_1, \dots q_J > 0$. Before presenting our results, we need an extra assumption on the distribution of covariates.

Assumption 4.5. *Assume the minimum eigenvalue of M_{j,n_j} is lower bounded by a constant $\lambda > 0$ for all $j \in [J]$.*

Assumption 4.5 guarantees that the distribution of the covariate covers all dimensions.

Theorem 4.1 (Prediction error under sequential dependency)**.** *For any funnel with a sequential dependency of parameters $q_1, \dots, q_J$, let $\hat{\theta}_1, \dots, \hat{\theta}_J$ be the estimates from Algorithm 3. If $n_{j+1} \leq n_j/4$, $q_1 \geq, \dots, \geq q_J$ and Assumption 5 is satisfied, then with a probability at least $1 - \delta$, for any $j_0 \in [J]$, we have*

$$PE_j \leq \begin{cases} \kappa\|x\|_2 \frac{c_{\delta/J}}{c_\mu \lambda} \sqrt{\frac{d}{n_j}}, & \text{if } j < j_0, \\ \kappa\|x\|_2 (\frac{c_{\delta/J}}{c_\mu \lambda} \sqrt{\frac{d}{n_{j_0}}} + \sum_{i=j_0+1}^{j} q_i), & \text{if } j \geq j_0, \end{cases}$$

where we let $n_0 = \infty$. The smallest bound of all choices of j_0 is achieved when j_0 is the smallest $j \in [J]$, such that

$$\frac{c_\delta \sqrt{d}}{c_\mu \lambda} (\frac{1}{\sqrt{n_j}} - \frac{1}{\sqrt{n_{j-1}}}) \geq q_j, \tag{4.5}$$

if none of j's in $[J]$ satisfies (4.5), $j_0 = J + 1$.

Theorem 4.1 shows that for some funnel under sequential dependency assumption, there exists a threshold layer j_0, before which the bounds without multi-task learning are tighter. After j_0, we use the bounds depending on prior knowledge. For small n_j's, $j_0 = 1$ and for sufficient large n_j's, $j_0 = J+1$. Figure 4.2 shows an example of how the threshold j_0 changes when total number n increases in a 5-layered funnel.

For the clustered dependency, the prediction error bound can be characterized by Theorem 4.2.

Theorem 4.2 (Prediction error under clustered dependency)**.** *For any funnel with a clustered dependency of parameters $q_1, \dots, q_J$, let $\hat{\theta}_1, \dots, \hat{\theta}_J$ be the estimates from Algorithm 3. With a probability at least $1 - \delta$,*

$$PE_j \leq \kappa\|x\|_2 \min\{\frac{c_{\delta/J}}{c_\mu \lambda} \sqrt{\frac{d}{n_{j_0}}} + q_j, \frac{c_{\delta/J}}{c_\mu \lambda} \sqrt{\frac{d}{n_j}}\},$$

where $j_0 = \arg\min_{j \in [J]} \frac{c_\delta}{c_\mu \lambda} \sqrt{\frac{d}{n_j}} + q_j$.

Under the clustered dependency, there is a single layer that gives the tightest confidence set on the unknown center, which is used by all the other layers.

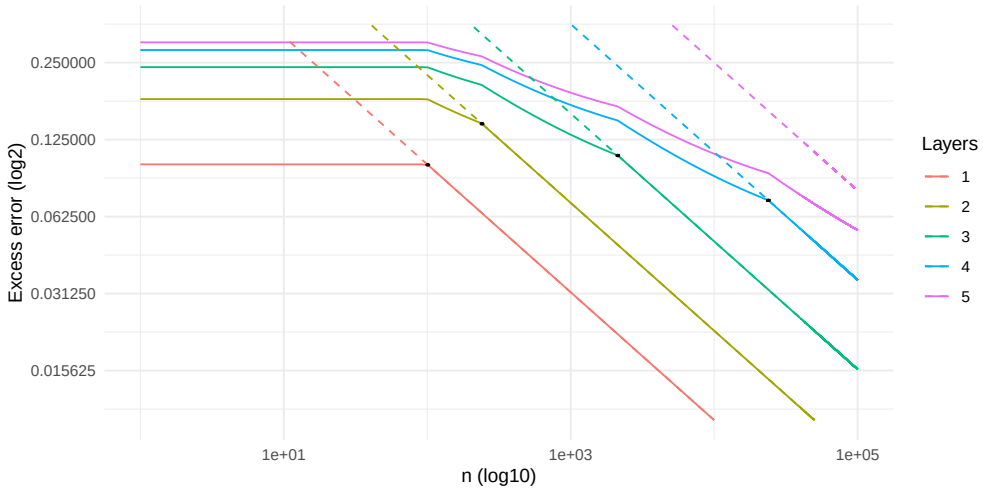


Figure 4.2: An example of how the prediction error bound in Theorem 4.1 changes when the number of observations increases in a 5-layered funnel. We set $\|x\|_2 c_\delta \sqrt{d}/(c_\mu \lambda) = 1$, $q_j = (12-2j)/100$ and $n_j = 0.2^{j-1}n$. Solid lines mark the prediction error bound defined in Theorem 4.1 and dashed lines mark the prediction error without multi-task learning (second term in (4.3)). Black points marked the change of j_0.

4.4 Regret Analysis for Contextual Bandit

In this section, we bound regrets for contextual bandits with funnel structure and discuss the benefits of multi-task learning.

Extra notations. For simpler demonstration, we define $P_j : \mathcal{X} \mapsto [0,1]$, such that $P_j(x) = \prod_{i=1}^{j} Z_i(x)$. For any $\boldsymbol{\theta} = (\theta_1^T, \theta_2^T, \ldots, \theta_j^T)^T \in \mathbb{R}^{dJ}$, let $P_j(x, \boldsymbol{\theta}) = \prod_{i=1}^{j} \mu(x^T \theta_j)$. To account for multiple funnels, we let $n_{a,j}^t$ be the number of observations for the j-th layer of funnel F_a up to step t . Let $\theta_{a,j}^*$ be the true parameters and $\boldsymbol{\theta}_a^*$ be the joint vector. Further we let $\lambda_{a,j}^t$ be the sample minimum eigenvalue covariance matrix of layer j of funnel F_a, $\lambda_{a,j}$ be the minimum eigenvalue of its expectation and $\bar{\lambda}$ be a lower bound over all a and j.

Regret of contextual bandits with funnel structure is defined as

$$\sum_{t=1}^{T} \left[P_J(x_t, \boldsymbol{\theta}_{a_t^*}^*) - P_J(x_t, \boldsymbol{\theta}_{a_t}^*) \right],$$

where a_t^* is the optimal action for input x_t, a_t is the the action chosen by the agent at step t and T is the total steps.

Algorithm 4 Contextual Bandit with a Funnel Structure

$t \to 1$, total number of steps T, memory $\mathcal{H}_a = \{\}$ for all $a \in [A]$. Initialize $\theta_{a,\star}$ with zero vectors.
for $t = 1$ to T **do**
  Receive context x_t.
  Compute $P_J^+(x_t, \hat{\boldsymbol{\theta}}_a)$ based on (4.6) for all $a \in [A]$.
  Choose $a_t = \arg\max_{a \in \mathcal{A}} \hat{P}_J^+(x_t, \hat{\theta}_{a,j})$.
  Receive $r_{t,1}, \ldots, r_{t,J}$ from funnel F_{a_t}.
  Set $\mathcal{H}_{a_t} \to \mathcal{H}_{a_t} \cup \{(x_t, (r_{t,1}, \ldots, r_{t,J}))\}$.
  Update $\hat{\theta}_{a_t,\star}$ by algorithm 3 with dataset $\mathcal{H}_{a_t}$.
end for

4.4.1 Optimistic algorithm

We propose a variation of the famous UCB (upper confidence bound) algorithm that adds bonuses based on the uncertainty of the whole funnel. The prediction error of funnel F_a of a given input x is $|P_J(x, \hat{\boldsymbol{\theta}}_a^t) - P_J(x, \boldsymbol{\theta}_a^*)|$ for $\hat{\boldsymbol{\theta}}_a^t$ which is the joint vector of estimates $\hat{\theta}_{a,j}^t$ at the step t.

From Lemma 4.1, we define $\Delta\mu_{a,j}^t$ by

$$\kappa \|x_t\|_2 \min\{\sup_{\theta_1, \theta_2 \in \Theta_{a,1}^t} \|\theta_1 - \theta_2\|_2, \frac{c_{\delta/3AJT}}{c_\mu \lambda_{a,j}^t} \sqrt{\frac{d}{n_{a,j}^t \vee 1}}\},$$

where $\Theta_{a,1}^t$ is the intersection set from Algorithm 3 for funnel a at step t. Using simple Taylor's expansion, we have Lemma 4.2.

Lemma 4.2. *Using the estimates from Algorithm 3, we have, with the same probability in (4.4),*

$$\begin{aligned} &|P_J(x, \hat{\boldsymbol{\theta}}_a^t) - P_J(x, \boldsymbol{\theta}_a^*)| \leq \\ &\sum_j \frac{P_J(x, \hat{\boldsymbol{\theta}}_a^t)}{\mu(x^T \hat{\theta}_{a,j}^t)} \Delta\mu_{a,j}^t + \sum_{i \neq j} \Delta\mu_{a,j}^t \Delta\mu_{a,i}^t =: \Delta\mu_a^t. \end{aligned} \tag{4.6}$$

Define $P_J^+(x, \hat{\boldsymbol{\theta}}_a^t) = P_J(x, \hat{\boldsymbol{\theta}}_a^t) + \Delta\mu_a^t$. We are able to derive an optimistic algorithm shown in Algorithm 4. Now we use the prediction error on the whole funnel to analyze the regret.

4.4.2 Regret analysis

Theorem 4.3 bounds regrets for Algorithm 4. The regret in Theorem 4.3 can be bounded by three terms in (4.7). The first term in (4.7) represents the normal regret without any

multi-task learning with an order $\mathcal{O}(\sum_{a,j}\sqrt{n_{a,j}^T})$, which reduces to the standard $\sqrt{AT}$ when $J=1$. Note that the impact of one layer on regret is bounded with the square root of its number of observations. The second term is a constant term that does not depend on T. The third term represents the benefits of multi-task learning. Full version of Theorem 4.3 is given and proved in Appendix 4.A.

Theorem 4.3. *Using Algorithm 4, under the Assumptions 1-4, with high probability, the total regret*

$$\sum_{t=1}^{T}\left[P_J(x_t,\boldsymbol{\theta}_{a_t^*}^*)-P_J(x_t,\boldsymbol{\theta}_{a_t}^*)\right]=\mathcal{O}(c_0\sum_{a,j}\sqrt{n_{a,j}^T}+\sum_{a,j}\frac{c_0^2Jd_x^4}{\bar{p}_{a,j}^2})-\sum_{a,j}\Delta_{a,j} \tag{4.7}$$

where $\mathcal{O}$ ignores all the constant terms and logarithmic terms for better demonstrations, $c_0=(\kappa d_x c_{\delta/3AJT}\sqrt{d})/(c_\mu\bar{\lambda})$, $\bar{p}_{a,j}\coloneqq\mathbb{E}_x P_{j-1}(x,\boldsymbol{\theta}_a^*)$ and

$$\Delta_{a,j}=\sum_{t=1;a_t=a}^{T}P_j(x_t^T\hat{\theta}_{a_t}^t)\left[c_0\frac{1}{\sqrt{n_{a,j}^t\vee 1}}-\Delta\mu_{a,j}^t\right].$$

represents the benefits of transfer learning.

We discuss the actual form of benefits under sequential dependency. If the hypothesis class is truly sequential dependency with parameters $q_{a,1},\ldots,q_{a,J}$ for each funnel F_a, we have

$$\Delta_{a,j}=\mathcal{O}(\sum_{j'\le j}1/q_{a,j'}),$$

for sufficient large T. Generally, for sufficient large total steps, the benefits scale with $\sum_{a,j}(J-j+1)/q_{a,j}$.

4.5 Experiments

In this section, we first present a simulation on the power of multi-task learning using Algorithm 3. Then we propose a more practical contextual bandit algorithm, which is tested on a simulated environment and our real-data email campaign environment.

4.5.1 Simulation on supervised learning

We use a simulation to verify the bound in Theorem 4.1 and the curves in Figure 4.2. We consider a 5-layered funnel with sequential dependency. The link function is $\mu(x)=$

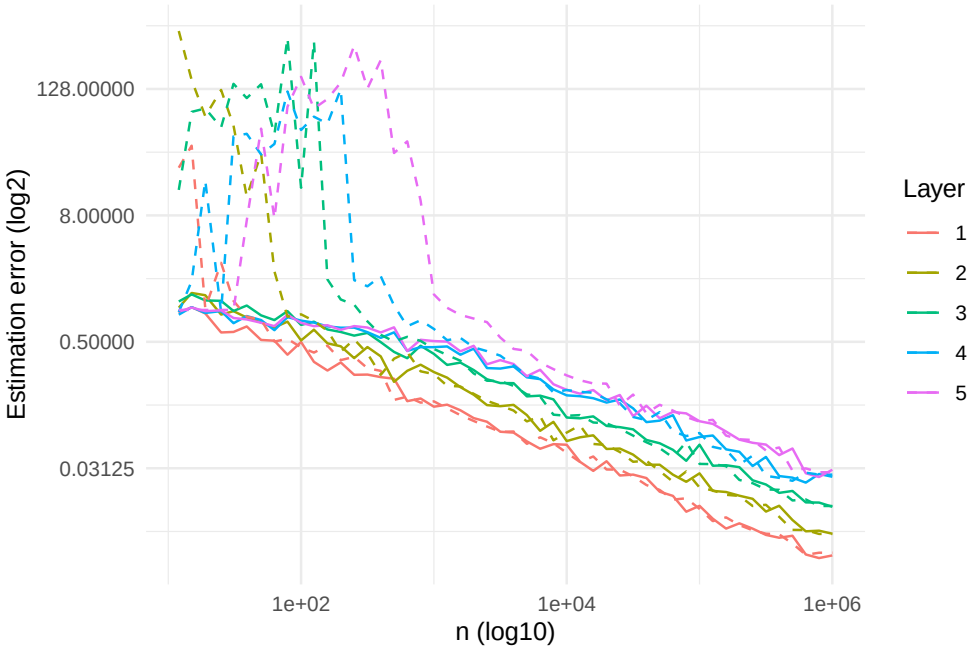


Figure 4.3: Estimation errors (L_2 distance to θ_j^*) of $\bar{\theta}_j$ and $\hat{\theta}_j$ under different number of interactions with the funnel. Colors represents the layers. Solid (Dashed) lines represents the estimation errors of $\hat{\theta}_j$ ($\bar{\theta}_j$). Each point in the plot is an average over 10 independent runs.

$1/(1+\exp(-x))$. We set $d = 5$ and $\theta_{j+1} = \theta_j + u_j q_j$, where $u_j \in \mathbb{R}^d$ is a unit vector with a random direction and $q_j = 1.2 - 0.2j$. In the simulation, we apply Algorithm 3 under the sequential dependency and calculate the estimation error of the estimates without parameter transfer ($\bar{\theta}_j$) and the final estimates $\hat{\theta}_j$. The results are shown in Figure 4.3. We observe a similar pattern as in Figure 4.2 and the errors of deeper layers are controlled well despite of their small sample sizes.

4.5.2 Simulations on contextual bandits

Practical algorithm. Calculating the intersection between two sets is not easy when Θ_0 has a complex form. Also, we may not have access to Θ_0 in real data analysis. We, therefore, develop practical algorithms especially for the sequential dependency and clustered dependency. Both of the algorithms reduced to optimizing parameters under a L_2 regularization. An equivalent form is to optimize under L_2 penalty. Our practical contextual bandit algorithm optimize loss function using an L_2 penalty controlled by tuned hyper-parameters (Algorithm 5 in Appendix 4.B). Since exploration is not the primary interest of this work, we also adopt a ϵ-greedy exploration for the simplicity of implementation.

Compared algorithms. Apart from the naive algorithm that directly learns on the signals r_J, some methods learn on the averaged rewards across the funnel. This approach is commonly used in Reinforcement Learning with auxiliary rewards (Jaderberg et al., 2016; Lin et al., 2019), where the true reward is sparse and there are some non-sparse auxiliary rewards that can accelerate learning. We call this method *Mix* in the following experiments.

Inspired by the idea of *Mix* and that of curriculum learning Bengio et al. (2009), we also test the method that learns on the signals for each layer sequentially.

To sum up, we compare the following five strategies:

1. *Target*: we train a single model that only predicts the reward from the last stage, i.e. r_J.

2. *Mix*: we train a single model that only predicts the average rewards from all the stages, i.e. $\frac{1}{J}\sum_{j=1}^{J} r_j$)

3. *Sequential*: for a total steps T, we train a single model on rewards $r_1, \ldots, r_{J-1}$ sequentially for equal number of steps $\alpha T/(J-1)$ for some constant $\alpha \in [0,1]$ and train the model on the final reward r_J for the rest of steps.

4. *Multi-layer clustered*: Algorithm 5 under clustered dependency.

5. *Multi-layer sequential*: Algorithm 5 under sequential dependency.

Simulated Environments. We first tested the performance of multi-task learning algorithms on the contextual bandit setting. In our contextual bandit setting, number of action is set to be $A = 50$, for each action, we independently generate a funnel with $J = 8$ stages. The link function is from the logistic regression, where we sample the unknown parameter $\theta_{a,j}$ sequentially. In this case, $\theta_{a,1}$ is sampled from $N(0, \sigma^2)$ and $\theta_{a,j}$ is sampled from $N(\theta_{a,j-1}, \sigma^2/j)$ for for $j > 1$. This gives us a funnel with decreasing uncertainty. Context x is sampled from a Gaussian distribution $N(0, \sigma_x^2)$.

We set the parameters for the environment to be $A = 50; J = 8; \sigma = 1; \sigma_x = 0.08; d = 45; T = 3000$.

Model setup. For *Target*, *Mix* and *Sequential*, we use a Neural Network model with one hidden layer, d-dimensional input and A-dimensional output. Each dimension on the output vector represents the predicted conversion probability for an action. The number of units for the hidden layer is searched in $\{8, 16, 32, 64\}$. *Sequential* has the hyper-parameter a, which is searched in $\{0.1, 0.2, 0.4, 0.6\}$.

For *Multi-layer clustered* and *Multi-layer sequential*, each stage is modeled with the same one-layer Neural Networks defined above. The number of units for the hidden layer is searched in $\{1, 4, 8, 16\}$. The penalty parameter λ is searched in $\{0.001, 0.005, 0.01, 0.05\}$. An ϵ-greedy exploration is applied.

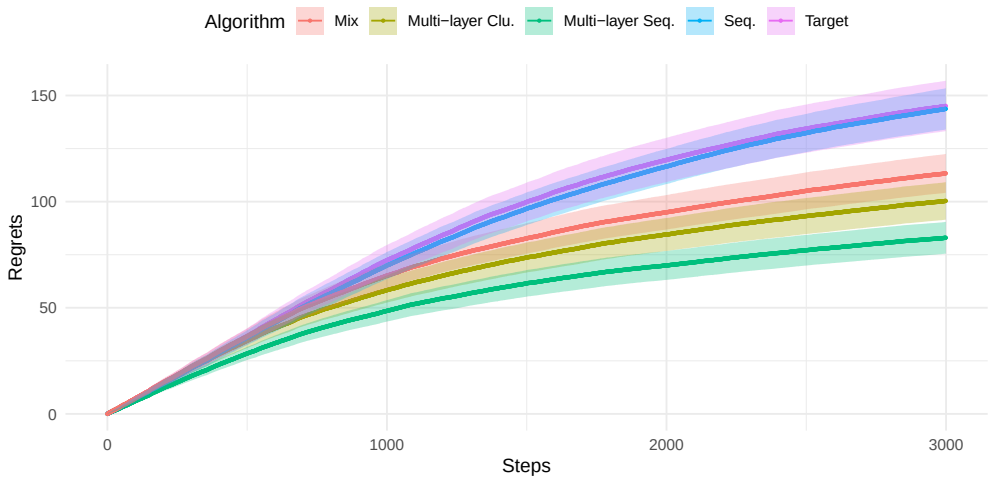


Figure 4.4: Cumulative regrets over 3000 steps using the best hyper-parameters for each of the five algorithms. The confidence interval is calculated from independent runs.

As shown in Figure 4.4, our practical algorithms beat all the other algorithms in terms of cumulative regrets. Algorithm 5 under sequential has lower regrets compared to that under clustered dependency.

4.5.3 Email campaign environment

We further test our algorithms on the Email Campaign problem, which aims at the act of sending a commercial message, typically to a group of people, using email.

Dataset. We randomly selected 5609706 users, who were active up to the data collection date and then tracked all the interactions of those users in the following 51 days, which adds up to 39488647 emails. Each email has a five-dimensional context, consisting of: *NumSent*, the number of emails sent to the user since 2019-12-01; *NumOpen*, the number of emails opened by the user since 2019-12-01; *NumClick*, the number of emails whose links were clicked by the user since 2019-12-01; *BussinessGroup*, categorical variable indicating the business group of the user; and *Recency*, number of hours since last email was sent to the user, which is categorized into '0-12', '13-24', '25-36', '37-48', '49+'.

The agent takes actions of the time to send an email. The action space is divided into six blocks: 00:00-04:00, 04:00-08:00, 08:00-12:00, 12:00-16:00, 16:00-20:00, 20:00-24:00.

Three rewards are available for each email, indicating whether the email is opened, whether the email is clicked and whether the email leads to a purchase respectively. Note that we say an email leads a purchase if the user purchases in the following 30 days.

The email campaign problem defines a funnel with three layers. On average, the rates for opening, clicking and purchasing are about 10%, 0.4% and 0.01%, respectively in the dataset. As we can see, the signals for learning purchasing behaviour are sparse. We will show in our experiments that how multi-task learning can improve the sample efficiency.

Data-based environment. We first build a data-based contextual bandit environment using population distribution. At each step, the environment randomly samples a context from the dataset and the agent takes an action from the six blocks. The environment then samples a reward vector from the set of rewards with the same action and context.

	Purchase (10^{-2})	Click	Open
Target	4.09	3.88	24.6
Mix	4.23	6.63	47.0
Sequential	2.1	0.981	0.495
Multi-layer clu.	6.34	0.753	-18.1
Multi-layer seq.	1.06	2.04	3.03

Table 4.1: Average increases in the number of Purchase, Click or Open over 10000 steps compared to the Random policy using 20 independent runs. The standard deviations are all less than 10^{-3} for Purchase and 10^{-1} for Click and Open.

Results. We searched the same set of hyper-parameters as used in the previous experiments except for the number of hidden units. The hidden units for the two multi-layer algorithms are searched in $\{8, 16, 32, 64\}$ instead.

We first tested the decrease in prediction error for the five algorithms with actions randomly selected. Hyper-parameters with the lowest cumulative square prediction errors were selected. As shown in Figure 4.5, our multi-task learning algorithms have much lower cumulative prediction errors. The total prediction errors converge after 2500 steps.

We further applied our practical algorithm in Algorithm 5 and selected the hyper-parameters with the lowest regret. Table 1 showed the average increased number of Purchase, Click and Open within 10000 steps with respect to a purely random policy over 10 independent runs. The results are shown in Table 4.1. As one can see in the table, Multi-layer

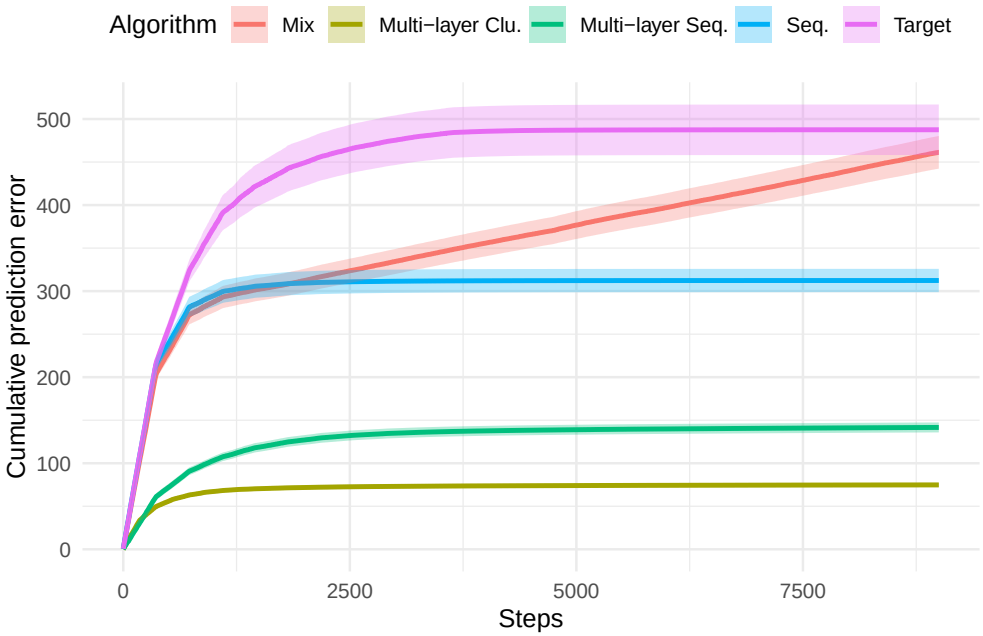


Figure 4.5: The cumulative square errors for five algorithms. The solid lines are averaged over 10 independent runs and regions mark the 1 standard deviation over the 10 runs.

clustered improved Purchase rate the most by 0.0634 over 10000 steps. However, Mix improved the average number of Click or Open the most. This indicates that the three tasks are not exactly the same.

The advantages of our algorithms over the compared three algorithms are the use of full information and the appropriate regularization, which allows knowledge transfer between layers. *Target*, *Seq.* and *Mix* all use a single model for the whole funnel, where the similarities between layers are not clear and they all lose some information while processing signals. *Target* and *Seq.* learn only on a single signal from one layer on each step. *Mix* adopts a weighted average.

4.6 Discussion

In this chapter, we formulated an important problem, funnel structure, from the marketing field. We used a multi-task learning algorithm to solve the contextual bandit problem with a funnel structure and offered its regret analysis. We verified our theorem using a simple simulation environment and tested the performances of our algorithm on both simulation and real-data environment.

Note that our bounds on prediction error in Theorem 4.1 and Theorem 4.2 do not scale

with $1/\sqrt{\sum_j n_j}$ under the special case when $\theta_1 = \cdots = \theta_J$. However, the sparsity of the funnel implies that the optimal rate is only smaller than our bound by a constant factor that does not depend on J. To see this, assuming that $n_{j+1}/n_j = q$, we have $\sum_j n_j \to n/(1-q)$, which is a constant value.

In terms of the real-data environment, we adopted the population model that may lead to high variance in context and action pairs that do not have sufficient observations and our environment may not fully reflect the true environment. A better data-based environment may be proposed using rare event simulation Rubino and Tuffin (2009).

4.A Missing Proofs

Connection to discrepancy measure In this section, we discuss how our assumption relates to discrepancy assumptions. Consider $\mathcal{Y}$-discrepancy that measures the maximum absolute distance between the loss function: $\text{dist}\left(\mathcal{D}_1, \mathcal{D}_2\right) \coloneqq \sup_{h\in\mathcal{H}} \left|\mathcal{L}_{\mathcal{D}_1}(h) - \mathcal{L}_{\mathcal{D}_2}(h)\right|$, where $\mathcal{D}_1$ and $\mathcal{D}_1$ represents the source domain and target domain and $\mathcal{L}_{\mathcal{D}_1}$ and $\mathcal{L}_{\mathcal{D}_2}$ are expected loss for two domains.

Note that under the GLM assumption, the L_2 distance in unknown parameters resembles the discrepancy using square loss. Consider a funnel with two layers and $\|\theta_1 - \theta_2\|_2 = q$. Lemma 4.3 indicates that $q \approx \text{dist}(\mathcal{D}_1, \mathcal{D}_2)$.

Lemma 4.3. *We have under square loss function,* $\text{dist}(\mathcal{D}_1, \mathcal{D}_2) \leq 4\kappa d_x q$.

Proof.

We first show the second inequality.

$$
\begin{aligned}
&\text{dist}\left(\mathcal{D}_1, \mathcal{D}_2\right) \\
&= \sup_\theta \left|\mathbb{E}_x(\mu(x^T\theta)) - \mu(x^T\theta_1^*))^2 - \mathbb{E}_x(\mu(x^T\theta) - \mu(x^T\theta_2^*))^2\right| \\
&\leq \sup_\theta \left|\mathbb{E}_x \mu(x^T\theta)(\mu(x^T\theta_1^*) - \mu(x^T\theta_2^*))\right| + \left|\mathbb{E}_x(\mu^2(x^T\theta_1^*) - \mu^2(x^T\theta_2^*))\right| \\
&\leq 4\left|\mathbb{E}_x(\mu(x^T\theta_1^*) - \mu(x^T\theta_2^*))\right| \\
&\leq 4\mathbb{E}_x\left|\mu(x^T\theta_1^*) - \mu(x^T\theta_2^*)\right| \\
&\leq 4\kappa\mathbb{E}_x\left|x^T(\theta_1^* - \theta_2^*)\right| \\
&\leq 4\kappa d_x q
\end{aligned}
$$

On the other hand, an lower bound of dist $(\mathcal{D}_1, \mathcal{D}_2)$ is also closely related to q.

$$
\begin{aligned}
&\text{dist}\,(\mathcal{D}_1, \mathcal{D}_2) \\
&= \sup_\theta |\mathbb{E}_x(\mu(x^T\theta)) - \mu(x^T\theta_1^*))^2 - \mathbb{E}_x(\mu(x^T\theta) - \mu(x^T\theta_2^*))^2| \\
&= \sup_\theta |\mathbb{E}_x(\mu(x^T\theta_1^*) - \mu(x^T\theta_2^*))(\mu(x^T\theta_1^*) + \mu(x^T\theta_2^*) + \mu(x^T\theta))| \\
&= \sup_\theta |\mathbb{E}_x \int_t \mu'\left(tx^T\theta_1^* + (1-t)x^T\theta_2^*\right) dt (x^T(\theta_1^* - \theta_2^*))(\mu(x^T\theta_1^*) + \mu(x^T\theta_2^*) + \mu(x^T\theta))| \\
&= \sup_\theta |(\theta_1^* - \theta_2^*)^T[\mathbb{E}_x x \int_t \mu'\left(tx^T\theta_1^* + (1-t)x^T\theta_2^*\right) dt (\mu(x^T\theta_1^*) + \mu(x^T\theta_2^*) + \mu(x^T\theta))]| \\
&\quad (\text{Let } \theta \to -\infty) \\
&\geq |(\theta_1^* - \theta_2^*)^T \nu_{\theta_1^*, \theta_2^*}| \text{ (letting } \nu_{\theta_1^*, \theta_2^*} = [\mathbb{E}_x x \int_t \mu'\left(tx^T\theta_1^* + (1-t)x^T\theta_2^*\right) dt (\mu(x^T\theta_1^*) + \mu(x^T\theta_2^*))]).
\end{aligned}
$$

Let $\theta_2^* = \theta_1^* + \|\theta_1^* - \theta_2^*\|_2 \mu$, where μ is a unit vector. For sufficient small $\|\theta_1^* - \theta_2^*\|_2$, $\nu_{\theta_1^*, \theta_2^*} \to 2\mathbb{E}_x[x\mu'(x^T\theta_1^*)\mu(x^T\theta_1^*)] =: \nu_{\theta_1^*}$, which is a constant vector. Thus

$$
\lim_{\|\theta_1^* - \theta_2^*\|_2 \to 0} \frac{\text{dist}(\mathcal{D}_1, \mathcal{D}_2)}{\|\theta_1^* - \theta_2^*\|_2} = |\mu^T \nu_{\theta_1^*}|.
$$

For sufficient small $\|\theta_1^* - \theta_2^*\|$, discrepancy scales with $\|\theta_1^* - \theta_2^*\|$.

Proof of Lemma 4.1 In this subsection, we introduce the proof of Lemma 4.1. Many proofs could achieve a very similar bound. Here we use the idea of local Rademacher complexity.

Proof.

We discuss two cases: 1) $\hat{\theta} \in \text{int}(\Theta_0)$. 2) $\hat{\theta} \notin \text{int}(\Theta_0)$.

In both cases, one simply has

$$
|\mu(x^T\hat{\theta}) - \mu(x^T\theta^*)| \leq \kappa |x^T(\hat{\theta} - \theta^*)| \leq \kappa \sup_{\theta_1, \theta_2 \in \Theta_0} |x^T(\theta_1 - \theta_2)|,
$$

which completes the first term in the minimum.

Now we prove the parametric bound. We first assume that case 1 holds. In this case, the constraint does not come into effects and $\hat{\theta}$ is the global minimal. By Theorem 26.5 in Shalev-Shwartz and Ben-David (2014), we have under an event, whose probability is at least $1 - \delta$,

$$
L(\hat{\theta}) - L(\theta^*) \leq 2R_n(\boldsymbol{z}) + 5\sqrt{\frac{2\ln(8/\delta)}{n}}, \tag{4.8}
$$

where $R(\boldsymbol{z})$ is the Rademacher complexity defined by

$$R_n(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{\sigma}} \frac{1}{n} \sup_{\theta \in \Theta} \sum_{i=1}^{n} \|z_i - \mu(x_i^T \theta)\|_{M_n}^2 \sigma_i,$$

and the variables in $\boldsymbol{\sigma}$ are distributed i.i.d. from Rademacher distribution. Let us call the event E_A.

As for any $i \in [n]$, let $\phi_i(t) \coloneqq (z_i - \mu(t))^2$, which satisfies $|\phi_i'(t)| = |2(z_i - \mu(t))\mu'(t)| \leq \kappa$, using Contraction lemma (Shalev-Shwartz and Ben-David, 2014), we have

$$\begin{aligned} R_n(\boldsymbol{z}) &\leq \mathbb{E}_{\boldsymbol{\sigma}} \frac{1}{n} \sup_{\theta \in \Theta} \sum_{i} \kappa(x_i^T \theta) \sigma_i \\ &= \kappa \mathbb{E}_{\boldsymbol{\sigma}} \frac{1}{n} \sup_{\theta \in \Theta} \sum_{i=1}^{n} x_i^T (\theta - \theta^*) \sigma_i. \\ &\leq \kappa \mathbb{E}_{\boldsymbol{\sigma}} \frac{1}{n} \sup_{\theta \in \Theta} \| \sum_{i} x_i \sigma_i \|_{M_n^{-1}} \|\theta - \theta^*\|_{M_n} \\ &\leq \kappa \mathbb{E}_{\boldsymbol{\sigma}} \frac{1}{n} \| \sum_{i} x_i \sigma_i \|_{M_n^{-1}} \sup_{\theta \in \Theta_0} \|\theta - \theta^*\|_{M_n}. \end{aligned} \tag{4.9}$$

Next, using Jensen's inequality we have that

$$\begin{aligned} & \mathbb{E}_{\boldsymbol{\sigma}} \frac{1}{n} \| \sum_{i} x_i \sigma_i \|_{M_n^{-1}} \\ &\leq \frac{1}{n} \left(\mathbb{E}_{\boldsymbol{\sigma}} \| \sum_{i} x_i \sigma_i \|_{M_n^{-1}}^2 \right)^{1/2} \\ &= \frac{1}{n} \left(\mathbb{E}_{\boldsymbol{\sigma}} tr[M_n^{-1} (\sum_{i} x_i \sigma_i)(\sum_{i} x_i \sigma_i)^T] \right)^{1/2} \\ &= \frac{1}{n} \left(tr[M_n^{-1} \mathbb{E}_{\boldsymbol{\sigma}} (\sum_{i} x_i \sigma_i)(\sum_{i} x_i \sigma_i)^T] \right)^{1/2} \end{aligned} \tag{4.10}$$

Finally, since the variables $\sigma_1, \ldots, \sigma_m$ are independent we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\sigma}}(\sum_i x_i\sigma_i)(\sum_i x_i\sigma_i)^T \\
&= \mathbb{E}_{\boldsymbol{\sigma}} \sum_{k,l\in[n]} \sigma_k\sigma_l x_k x_l^T \\
&= \mathbb{E}_{\boldsymbol{\sigma}} \sum_{i\in[n]} \sigma_i^2 x_i x_i^T \\
&= \sum_{i\in[n]} x_i x_i^T = nM_n.
\end{aligned}$$

Plugging this into (4.10), assuming M_n is full rank, we have

$$(4.9) \le \sqrt{d/n} \sup_{\theta\in\Theta_0} \|\theta - \theta^*\|_{M_n}. \tag{4.11}$$

Lemma 4.4. *Under the notation in Lemma 4.1 and Assumption 4.2, if an estimate $\hat{\theta}$ satisfies $L(\hat{\theta}) \le L(\theta^*) + b_n$, then*

$$\|\hat{\theta} - \theta^*\|_{M_n}^2 \le \frac{d_x b_n}{c_\mu}.$$

Proof. Let $g_n(\theta) = \sum_i x_i(\mu(x_i^T\theta) - \mu(x_i^T\theta^*))$. For any θ, $\nabla g_n(\theta) = \sum_i x_i x_i^T \mu'(x_i^T\theta)$. By simple calculus,

$$g_n(\theta^*) - g_n(\hat{\theta}) = \int_0^1 \nabla g_n\left(s\theta^* + (1-s)\hat{\theta}\right) ds(\theta^* - \hat{\theta}).$$

As $\mu(t) \ge c_\mu$, we have $\int_0^1 \nabla g_n\left(s\theta^* + (1-s)\hat{\theta}\right) ds \succ c_\mu M_n$. Plugging this into the inequality above we have

$$\|\theta^* - \hat{\theta}\|_{M_n}^2 \le \frac{1}{c_\mu}(\sum_i x_i(\mu(x_i^T\theta) - \mu(x_i^T\theta^*)))^2 = \frac{1}{c_\mu}\epsilon^T M_n \epsilon \le \frac{d_x}{c_\mu}\epsilon^T\epsilon = \frac{d_x}{c_\mu}(L(\hat{\theta}) - L(\theta^*)),$$

where $\epsilon \coloneqq (\mu(x_i^T\hat{\theta}) - \mu(x_i^T\theta^*))_{i=1}^n$.

Applying (4.8) and Lemma 4.4, we complete the proof by

$$\begin{aligned}
\|\hat{\theta} - \theta^*\|_{M_n} &\le \sqrt{\frac{2d_x\sqrt{d}}{c_\mu\sqrt{n}} \sup_{\theta\in\Theta} \|\theta - \theta^*\|_{M_n} + 5\sqrt{\frac{2\ln(8/\delta)}{n}}} \\
&\le \sqrt{\frac{20\sqrt{2\ln(8/\delta)}d_x\sqrt{d}\sup_{\theta\in\Theta_0}\|\theta - \theta^*\|_{M_n}}{c_\mu\sqrt{n}}}.
\end{aligned} \tag{4.12}$$

□

We apply (4.12) iteratively [2]. Let $\Theta_{(1)} \coloneqq \Theta_0$. For any $t > 1$, let $\Theta_{(t)} = \{\theta \in \mathbb{R}^d : \|\theta - \hat{\theta}\|_{M_n} \leq \sqrt{\frac{20\sqrt{2d\ln(8/\delta)}}{c_\mu\sqrt{n}} \sup_{\theta\in\Theta_{(t-1)}} \|\theta - \theta^*\|_{M_n}}\}$. When $t \to \infty$, we have

$$\Theta_{(\infty)} = \frac{20 d_x \sqrt{2d\ln(8/\delta)}}{c_\mu\sqrt{n}}.$$

By (4.12), we have $\theta^* \in \cap_{t\geq 1}\Theta_{(\infty)}$ and $\|\hat{\theta} - \theta^*\|_{M_n} \leq \frac{40 d_x\sqrt{2d\ln(8/\delta)}}{c_\mu\sqrt{n}}$, which completes the second part of Lemma 4.1.

For any $x \in \mathcal{X}$, we have

$$|\mu(x^T\hat{\theta}) - \mu(x^T\theta^*)| \leq \kappa\|x\|_{M_n^{-1}} \frac{40 d_x\sqrt{2d\ln(8/\delta)}}{c_\mu\sqrt{n}}. \tag{4.13}$$

When case 2 holds, let $\hat{\theta}'$ be the global minimizer. Using the analysis above, we have

$$\|\hat{\theta}' - \theta^*\|_{M_n} \leq \frac{40 d_x\sqrt{2d\ln(8/\delta)}}{c_\mu\sqrt{n}}.$$

Then by triangle inequality

$$\|\hat{\theta} - \theta^*\|_{M_n} \leq \|\hat{\theta} - \hat{\theta}'\|_{M_n} + \|\hat{\theta}' - \theta^*\|_{M_n} \leq \frac{80 d_x\sqrt{2d\ln(8/\delta)}}{c_\mu\sqrt{n}}.$$

Tightness of Lemma 4.1 We use an example to show the tightness of Lemma 4.1. Assume a linear predictor, i.e. $\mu(t) = t$. Consider the following distribution, let X be uniform over the d-standard basis vector e_m, for $m = 1, \ldots, d$. Let $Z \mid (X = e_i) \sim Bern(r_i)$, where $r_i \in [0, 1]$ is pre-determined and unknown. The optimal parameter $\theta^* = (r_1, \ldots, r_d)^T$. Let n_m be the number of samples collected for dimension m. Let $\Theta_0 \coloneqq \{\theta : \|\theta\|_2 \leq q\}$.

When n is sufficiently large $n > 1/q^2$, $\hat{\theta}$ is the regularized minimizer. It can be shown that for any $\hat{\theta}$, there exists θ^* such that $\mathbb{E}[\hat{\theta}_i - \theta_i^*]^2 \geq (r_m(1 - r_m))/n_m$. Then $\mathbb{E}\|\hat{\theta} - \theta^*\|_2^2 \geq \sum_{m=1}^d \frac{r_i(1-r_i)}{n_m} \geq \frac{d^2(r_i(1-r_i))}{n} = \Omega(\frac{d^2}{n})$.

Then we also see that when n is small ($\leq \frac{1}{q^2}$), the estimation error is $\Omega(q)$. We use the same example as above. This time, we assume $\|\theta^*\| \leq \frac{q}{2}$. If we have a $\|\hat{\theta}\| = q$, then $\|\theta^* - \hat{\theta}\| \geq q/2 = \Omega(q)$. Otherwise, we use the lower bound above: $\|\theta^* - \hat{\theta}\| \geq \Omega(\frac{d}{\sqrt{n}}) = \Omega(dq)$.

[2]Note that (4.12) holds under the same event E_A as the estimates $\hat{\theta}$ keeps the same each round as it is the global minimizer.

The above argument corresponds to the upper bound in Lemma 4.1, where we use prior knowledge when n is small and use the parametric bound when n is large.

Proof of Theorem 4.1 In this subsection, we show the missing proof for Theorem 4.1.

Theorem 4.4 (Prediction error under sequential dependency)**.** *For any funnel with a sequential dependency of parameters $q_1, \ldots, q_J$, let $\hat{\theta}_1, \ldots, \hat{\theta}_J$ be the estimates from Algorithm 3. If $n_{j+1} \leq n_j/4$, $q_1 \geq, \ldots, \geq q_J$ and Assumption 5 is satisfied, then with a probability at least $1-\delta$, for any $j_0 \in [J]$, we have*

$$PE_j \leq \begin{cases} \kappa\|x\|_2 \frac{c_\delta}{c_\mu \lambda}\sqrt{\frac{d}{n_j}}, \text{ if } j < j_0, \\ \kappa\|x\|_2 (\frac{c_\delta}{c_\mu\lambda}\sqrt{\frac{d}{n_{j_0}}} + \sum_{i=j_0+1}^{j} q_j), \text{ if } j \geq j_0, \end{cases} \tag{4.14}$$

where we let $n_0 = \infty$. The bound is smallest when j_0 is the smallest $j \in [J]$, such that

$$\frac{4c_\delta\sqrt{d}}{c_\mu\lambda}(\frac{1}{\sqrt{n_j}} - \frac{1}{\sqrt{n_{j-1}}}) \geq q_j, \tag{4.15}$$

if none of j's in $[J]$ satisfies (4.15), $j_0 = J+1$.

Proof. First we reshape the ellipsoid in (4.3) to a ball.

Lemma 4.5 (Reshape)**.** *For any vector $x \in \mathbb{R}^d$ and any matrix $M \succ 0 \in \mathbb{R}^{d\times d}$, $\|x\|_2 \leq \frac{1}{\lambda}\|x\|_M$, where λ is the minimum eigenvalue of M.*

Proof. We directly use the definition of positive definite matrix: $\lambda^2\|x\|_2^2 - \|x\|_M^2 = x^T(\lambda^2 I - M)x \leq 0$. Thus, $\|x\|_2 \leq \frac{1}{\lambda^2}\|x\|_M$. #

Using Lemma 4.5 and Assumption 4.5, we have $\|\bar{\theta}_j - \theta_j^*\|_2 \leq \frac{1}{\lambda}\|\bar{\theta}_j - \theta_j^*\|_{M_n} \leq \frac{4c_\delta}{c_\mu\lambda}\sqrt{\frac{d}{n}}$. Thus the set $\hat{\Theta}_j \subset \{\theta : \|\theta - \bar{\theta}_j\|_2 \leq \frac{4c_\delta}{c_\mu\lambda}\sqrt{\frac{d}{n}}\} =: \hat{\Theta}_j^{ball}$.

For every j, one can derive two bounds. First we can directly apply Corollary 4.1 and get $PE_j \leq \kappa\|x\|_2\frac{4c_\delta}{c_\mu\lambda}\sqrt{\frac{d}{n_j}}$. Second, for any j_0, we have $\theta_j^* \in \Theta_1[j] \subset \{\theta : \|\bar{\theta}_{j_0} - \theta\|_2 \leq \frac{4c_s}{c_\mu\lambda}\sqrt{\frac{d}{n_{j_0}}} + \sum_{j_0+1\leq i\leq j} q_i\}$ and get $PE_j \leq \kappa\|x\|_2(\frac{c_\delta}{c_\mu\lambda}\sqrt{\frac{d}{n_{j_0}}} + \sum_{i=j_0+1}^{j} q_j)$.

Now we show the second argument: of all those bounds the one defined in (4.14) with j_0 defined in (4.15) is the smallest. For any $j \leq j_0$ and $j_1 \leq j$, we have

$$\frac{4c_\delta}{c_\mu\lambda}\sqrt{\frac{d}{n_j}} = \frac{4c_\delta\sqrt{d}}{c_\mu\lambda}\left(\sum_{i=j_1+1}^{j}(\frac{1}{\sqrt{n_i}} - \frac{1}{\sqrt{n_{i-1}}}) + \frac{1}{\sqrt{n_{j_1}}}\right) \leq \frac{4c_\delta}{c_\mu\lambda}\sqrt{\frac{d}{n_{j_1}}} + \sum_{i=j_1+1}^{j} q_i. \tag{4.16}$$

The second inequality is given by $(\frac{1}{\sqrt{n_i}} - \frac{1}{\sqrt{n_{i-1}}}) \leq q_i$ for all $i < j_0$. For any $j \geq j_0$ and $j_1 \leq j_0$, by (4.16), we have

$$\frac{4c_\delta}{c_\mu \lambda}\sqrt{\frac{d}{n_{j_0}}} + \sum_{i=j_0+1}^{j} q_i \leq \frac{4c_\delta}{c_\mu \lambda}\sqrt{\frac{d}{n_{j_1}}} + \sum_{i=j_1+1}^{j} q_i.$$

Now we prove that for all $i \geq j_0$,

$$\frac{4c_\delta\sqrt{d}}{c_\mu \lambda}(\frac{1}{\sqrt{n_i}} - \frac{1}{\sqrt{n_{i-1}}}) \geq q_i. \tag{4.17}$$

We use induction. Assume for some i_1, (4.17) is satisfied. Under the assumption that $n_{i_1-1} \leq n_{i_1}/4$ and $q_{i_1} \geq q_{i_1+1}$, we have

$$\begin{aligned}
\frac{4c_\delta\sqrt{d}}{c_\mu \lambda}(\frac{1}{\sqrt{n_{i_1+1}}} - \frac{1}{\sqrt{n_{i_1}}}) &= \frac{4c_\delta\sqrt{d}}{c_\mu \lambda}(\frac{1}{\sqrt{n_{i_1+1}}} + \frac{1}{\sqrt{n_{i_1-1}}} - \frac{2}{\sqrt{n_{i_1}}} + \frac{1}{\sqrt{n_{i_1}}} - \frac{1}{\sqrt{n_{i_1-1}}}) \\
&\geq \frac{4c_\delta\sqrt{d}}{c_\mu \lambda}(\frac{1}{\sqrt{n_{i_1+1}}} + \frac{2}{\sqrt{n_{i_1}}} - \frac{2}{\sqrt{n_{i_1}}} + \frac{1}{\sqrt{n_{i_1}}} - \frac{1}{\sqrt{n_{i_1-1}}}) \\
&\geq \frac{4c_\delta\sqrt{d}}{c_\mu \lambda}(+\frac{1}{\sqrt{n_{i_1}}} - \frac{1}{\sqrt{n_{i_1-1}}}) \\
&\geq q_{i_1} \geq q_{i_1+1}.
\end{aligned}$$

Using 4.17, for any $j \geq j_1 > j_0$,

$$\frac{4c_\delta}{c_\mu \lambda}\sqrt{\frac{d}{n_{j_0}}} + \sum_{i=j_0+1}^{j} q_i = \frac{4c_\delta\sqrt{d}}{c_\mu \lambda}\left(\sum_{i=j_0+1}^{j_1}(\frac{1}{\sqrt{n_{i-1}}} - \frac{1}{\sqrt{n_i}}) + \frac{1}{\sqrt{n_{j_1}}}\right) + \sum_{i=j_0+1}^{j} q_i \leq \frac{4c_\delta}{c_\mu \lambda}\sqrt{\frac{d}{n_{j_1}}} + \sum_{i=j_1+1}^{j} q_i.$$

Finally, we conclude that j_0 gives the smallest bound. # □

Similar argument can be used to show Theorem 4.2. For any $j_0 \in [J]$, we have

$$PE_j \leq \kappa \|x\|_2 \min\left\{\frac{c_{\delta/J}}{c_\mu \lambda}\sqrt{\frac{d}{n_{j0}}} + q_j, \frac{c_{\delta/J}}{c_\mu \lambda}\sqrt{\frac{d}{n_j}}\right\}.$$

Out of all the choices of j_0, the best one is achieved by $j_0 = \arg\min_{j\in[J]} \frac{c_\delta}{c_\mu \lambda}\sqrt{\frac{d}{n_j}} + q_j$.

Proof of Theorem 4.3

Theorem 4.5. *Using Algorithm 4, under the Assumptions 1-4, with a probability at least*

$1-\delta$, *the total regret*

$$\begin{aligned}&\sum_{t=1}^{T}\left[P(x_t,\boldsymbol{\theta}^*_{a_t^*})-P(x_t,\boldsymbol{\theta}^*_{a_t})\right]\\&\leq 2\sqrt{2}c_0\sum_{a,j}\sqrt{n^T_{a,j}}+\sum_{a,j}\frac{8c_0^2Jd_x^4\log(6AJT/\delta)}{\bar{p}^2_{a,j}}-\sum_{a,j}\Delta_{a,j}.\end{aligned}\tag{4.18}$$

where $\mathcal{O}$ ignores all the constant terms and logarithmic terms for better demonstrations, $c_0=(\kappa d_x c_{\delta/AJT}\sqrt{d})/(c_\mu\bar{\lambda})$, $\bar{p}_a\coloneqq\mathbb{E}_x P_{J-1}(x^T\boldsymbol{\theta}^*_a)$ *and*

$$\Delta_{a,j}=\sum_{t=1;a_t=a}^{T}P_j(x_t^T\hat{\theta}^t_{a_t})\left[c_0\frac{1}{\sqrt{n^t_{a,j}\vee 1}}-\Delta\mu^t_{a,j}\right].$$

represents the benefits of transfer learning.

Let $\bar{p}_{a,j}\coloneqq\mathbb{E}_x P_{j-1}\left(x^T\theta^*_a\right)$. We first show that upper bound the number of steps t with $\lambda^t_{a_t,j}\leq\bar{\lambda}/2$ or $n^t_{a,j}\leq\frac{1}{2}n^t_{a,1}\bar{p}_{a,j}$. These steps are considered bad events.

Lemma 4.6 shows that with high probability, the number of observations for each layer is close to its expectation.

Lemma 4.6. *With a probability at least $1-\delta$, we have $n^t_{a,j}\geq n^t_{a,1}\bar{p}_{a,j}-\sqrt{2n^t_{a,1}\log(1/\delta)}$. Especially, when $n^t_{a,1}>8\log(1/\delta)/\bar{p}^2_{a,j}=:c_{n,a}$, we have $n^t_{a,j}\geq\frac{1}{2}n^t_{a,1}\bar{p}_{a,j}$.*

Proof. This is a direct application of Hoeffding inequality. □

Lemma 4.7. *For any $x_1,\ldots,x_n$ i.i.d, $\|x_i\|\leq d_x$, let λ_n be the minimum eigenvalue of $\sum_i x_i x_i^T/n$ and $\bar{\lambda}$ be the minimum eigenvalue of its expectation. We have $\lambda_n\geq\bar{\lambda}/2$, when $n>d_x^4\log(1/\delta)/\bar{\lambda}^2$.*

Proof. For all $x_1,\ldots,x_n$, write $x_i=\sum_{s=1}^d\nu_{s,i}\tilde{x}_s$, where $\tilde{x}_1,\ldots,\tilde{x}_d$ are any basis of $\mathbb{R}^d$. We have $\mathbb{E}\nu^2_{s,i}\geq\bar{\lambda}$. For Hoeffding's inequality, since $\nu_{s,i}\leq d_x$, with a probability $1-\delta$, we have

$$\frac{1}{n}\sum_i\nu^2_{s,i}\geq\mathbb{E}\nu^2_{s,1}-d_x^2\sqrt{\frac{\log(1/\delta)}{n}}\geq\bar{\lambda}-d_x^2\sqrt{\frac{\log(1/\delta)}{n}}.$$

For $n>d_x^4\log(1/\delta)/\bar{\lambda}^2$, we have $\frac{1}{n}\sum_i\nu^2_{s,i}\geq\bar{\lambda}/2$. There exists a choice of $\tilde{x}_1,\ldots,\tilde{x}_d$ such that $\lambda_n=\frac{1}{n}\sum_i\nu^2_{s,i}$. □

Combining Lemma 4.6 and Lemma 4.7, we have with a probability at least $1-\delta/3$,

$\#\{t : \exists j, \lambda_{a_t,j}^t \leq \bar{\lambda}/2 \text{ or } n_{a,j}^t \leq \frac{1}{2} n_{a,1}^t \bar{p}_{a,j}\}$ can be upper bounded by

$$\sum_{a,j} \max\left\{8\log(6AJT/\delta)/\bar{p}_{a,j}^2, 2d_x^4 \log(6AJT/\delta)/(\bar{\lambda}^2 \bar{p}_{a,j})\right\}. \tag{4.19}$$

In the following proof, we assume for all t, $\lambda_{a,j}^t \geq \bar{\lambda}/2$ and $n_{a,j}^t \geq \frac{1}{2} n_{a,1}^t \bar{p}_{a,j}$. We also assume the event in Lemma 4.1 happens for all $a \in [A], j \in [J]$ and $t < T$. The probability is at least $1 - \delta/3$ as each probability is at least $1 - \delta/(3AJT)$.

The total regret is

$$\begin{aligned}
&\sum_{t=1}^{T} \left[P(x_t, \boldsymbol{\theta}_{a_t^*}^*) - P(x_t, \boldsymbol{\theta}_{a_t}^*)\right] \\
&\leq \sum_{t=1}^{T} \left[P(x_t, \boldsymbol{\theta}_{a_t^*}^*) - P^+(x_t, \hat{\boldsymbol{\theta}}_{a_t}) + P^+(x_t, \hat{\boldsymbol{\theta}}_{a_t}) - P(x_t, \boldsymbol{\theta}_{a_t}^*)\right] \\
&\text{(Using } P(x_t, \boldsymbol{\theta}_{a_t^*}^*) - P^+(x_t, \hat{\boldsymbol{\theta}}_{a_t}) \leq 0) \\
&\leq \sum_{t=1}^{T} \left[P^+(x_t, \hat{\boldsymbol{\theta}}_{a_t}^t) - P(x_t, \boldsymbol{\theta}_{a_t}^*)\right] \\
&\text{(Using Lemma 4.2)} \\
&\leq \sum_{t=1}^{T} \left[\sum_j \frac{P_J\left(x, \hat{\theta}_{a_t}^t\right)}{\mu\left(x^T \hat{\theta}_{a_t,j}^t\right)} \Delta\mu_{a_t,j}^t + \sum_{i \neq j} \Delta\mu_{a_t,j}^t \Delta\mu_{a_t,i}^t\right] \\
&\leq \sum_{t=1}^{T} \left[\sum_j P_j(x_t, \hat{\theta}_{a_t}^t) \Delta\mu_{a_t,j}^t + \sum_{i \neq j} \Delta\mu_{a_t,j}^t \Delta\mu_{a_t,i}^t\right] \\
&= \sum_{t=1}^{T} \left[\sum_j (P_j(x_t, \theta_{a_t}^*) + P_j(x_t, \hat{\theta}_{a_t}^t) - P_j(x_t, \theta_{a_t}^*)) \Delta\mu_{a_t,j}^t + \sum_{i \neq j} \Delta\mu_{a_t,j}^t \Delta\mu_{a_t,i}^t\right] \\
&\leq \underbrace{\sum_{t=1}^{T} \sum_j P_j\left(x_t, \theta_{a_t}^*\right) \frac{c_0}{\sqrt{n_{a_t,j}^t}}}_{①} - \underbrace{\sum_{t=1}^{T} \sum_j P_j\left(x_t, \theta_{a_t}^*\right) \left(\frac{c_0}{\sqrt{n_{a_t,j}^t}} - \Delta\mu_{a_t,i}^t\right)}_{②} + \underbrace{\sum_{t=1}^{T} \sum_{i \neq j} \Delta\mu_{a_t,j}^t \Delta\mu_{a_t,i}^t}_{③} \\
&\quad + \underbrace{\sum_{t=1}^{T} \left[\sum_j (P_j(x_t, \hat{\theta}_{a_t}^t) - P_j(x_t, \theta_{a_t}^*)) \Delta\mu_{a_t,j}^t\right]}_{④}.
\end{aligned}$$

We further bound the terms separately. The first term ① represents the bound one could have without multi-task learning.

$$\sum_{t=1}^{T}\sum_{j} P_j(x_t,\theta^*_{a_t})\frac{c_0}{\sqrt{n^t_{a_t,j}}}$$

$$\leq \sum_{t=1}^{T}\sum_{j} \mathbf{1}(r_{t,j-1}=1)\frac{c_0}{\sqrt{n^t_{a_t,j}}} + \sum_{t=1}^{T}\sum_{j}(P_j(x_t,\theta^*_{a_t}) - \mathbf{1}(r_{t,j-1}=1))\frac{c_0}{\sqrt{n^t_{a_t,j}}}$$

(Using Lemma 19 in Jaksch et al. (2010))

$$\leq c_0 2\sqrt{2}\sum_{a,j}\sqrt{n^T_{a,j}} + \sum_{t=1}^{T}\sum_{j}(P_j(x_t,\theta^*_{a_t}) - \mathbf{1}(r_{t,j-1}=1))\frac{c_0}{\sqrt{n^t_{a_t,j}}} \tag{4.20}$$

As $\mathbb{E}[P_j(x_t,\theta^*_{a_t}) - \mathbf{1}(r_{t,j-1}=1)] = 0$, the second term in (4.20) is a martingale. Using Azuma-Hoeffding inequality, with a probability at least $1-\delta/3$, for all T,

$$\sum_{t=1}^{T}\sum_{j}(P_j(x_t,\theta^*_{a_t}) - \mathbf{1}(r_{t,j-1}=1))\frac{c_0}{\sqrt{n^t_{a_t,j}}} \leq c_0\sqrt{2\log(3TJ/\delta)}. \tag{4.21}$$

Combined with (4.20),

$$① \leq 2\sqrt{2}c_0\sum_{a,j}\sqrt{n^T_{a,j}} + c_0\sqrt{2\log(3TJ/\delta)}. \tag{4.22}$$

Next we bound ③. We notice that this is a quadratic term. We first show Lemma 4.6 that lower bounds the number of observations for each layer. Lemma 4.6 is a direct application of Hoeffding's inequality.

For any pair i, j, we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \Delta\mu^t_{a_t,j}\Delta\mu^t_{a_t,i} \\
&\leq c_0^2 \sum_{t=1}^{T} \frac{1}{\sqrt{n^t_{a_t,i}}}\frac{1}{\sqrt{n^t_{a_t,j}}} \\
&\leq c_0^2 \sum_{t=1}^{T} \left[\mathbf{1}(n^t_{a_t,1} \leq c_{n,a_t})\frac{1}{\sqrt{n^t_{a_t,i}}}\frac{1}{\sqrt{n^t_{a_t,j}}} + \mathbf{1}(n^t_{a_t,1} > c_{n,a_t})\frac{1}{\sqrt{n^t_{a_t,i}}}\frac{1}{\sqrt{n^t_{a_t,j}}}\right] \\
&\leq c_0^2 \sum_{a} c_{n,a} + c_0^2 \sum_{t} \frac{4}{\bar{p}_a^2 n^t_{a_t,1}} \\
&\leq c_0^2 \sum_{a} c_{n,a} + c_0^2 \sum_{a} \frac{4\log(n^T_{a,1})}{\bar{p}_a^2} \\
&\leq 4c_0^2 \sum_{a} \frac{\log(n^T_{a,1}A/\delta)}{\bar{p}_a^2}. \qquad (4.23)
\end{aligned}
$$

where we let $\bar{p}_a \coloneqq \mathbb{E}_x P_J\left(x^T\theta^*_a\right)$.

Thus, ③ is upper bounded by $4c_0^2J^2\sum_a \frac{\log(n^T_{a,1}A/(3\delta))}{\bar{p}_a^2}$.

Finally we bound term ④. Using Lemma 4.2 on only first j layers, we have

$$
④ \leq \sum_t \sum_j [\sum_i \Delta\mu^t_{a_t,i} + \sum_{i,k} \Delta\mu^t_{a_t,k}\Delta\mu^t_{a_t,i}]\Delta\mu^t_{a_t,j} \leq (J+1) \times ③. \qquad (4.24)
$$

The proof is completed by combining Equations (4.19), (4.21), (4.22), (4.23) and (4.24).

4.B Experiments Details

Algorithm 5 Practical Algorithm for Contextual Bandit with a Funnel Structure

$t \to 1$, total number of steps T, memory $\mathcal{H}_a = \{\}$ for all $a \in [A]$. Initialize $\hat{\theta}_{a,\star}$ with zero vectors.

$\hat{\theta}_{a,0} \to 0$.

for $t = 1$ to T **do**

Receive context x_t.

Choose $a_t = \arg\max_{a \in \mathcal{A}} \hat{P}_J(x_t, \hat{\theta}_{a,j})$.

Set $a_t = \text{Unif}([A])$ with probability ϵ.

Receive $r_{t,1}, \ldots, r_{t,J}$ from funnel F_{a_t}.

Set $\mathcal{H}_{a_t} \to \mathcal{H}_{a_t} \cup \{(x_t, (r_{t,1}, \ldots, r_{t,J}))\}$.

for $j = 1, \ldots, J$ **do**

 # For sequential dependency

$$\hat{\theta}_{a_t,j} \to \arg\min_{\theta} l(\theta, \mathcal{H}_{a_t}) + \lambda_j \|\theta - \hat{\theta}_{a_t,j-1}\|_2$$

 # For clustered dependency

$$\hat{\theta}_{a_t,j} \to \arg\min_{\theta} l(\theta, \mathcal{H}_{a_t}) + \lambda_j \|\theta - \frac{1}{J}\sum_i \hat{\theta}_{a_t,i}\|_2$$

end for

end for

Practical algorithm

Simulated environment.

1. Target: units 16
2. Mix: units 32
3. Sequential: units 32
4. Multi-layer Clustered: units 4; λ 0.001
5. Multi-layer Sequential: units 8; λ 0.001

Data-based environment.

1. Target: units 64
2. Mix: units 64
3. Sequential: units 64
4. Multi-layer Clustered: units 64; λ 0.005
5. Multi-layer Sequential: units 16; λ 0.001

CHAPTER 5

Multitask Reinforcement Learning

Multitask Reinforcement Learning (MTRL) approaches have gained increasing attention for its wide applications in many important Reinforcement Learning (RL) tasks. However, while recent advancements in MTRL theory have focused on the improved statistical efficiency by assuming a shared structure across tasks, exploration–a crucial aspect of RL–has been largely overlooked. In this chapter, we address this gap by showing that when an agent is trained on a sufficiently *diverse* set of tasks, algorithms with myopic exploration design like ϵ-greedy that are inefficient in general can be sample-efficient for MTRL. To the best of our knowledge, this is the first theoretical demonstration of the "exploration benefits" of MTRL. It may also shed light on the enigmatic success of the wide applications of myopic exploration in practice. To validate the role of diversity, we conduct experiments on synthetic robotic control environments, where a more diverse training task set leads to improved performance.

5.1 Introduction

Reinforcement Learning often involves solving multitask problems. For instance, robotic control agents are trained to simultaneously solve multiple goals in multi-goal environments (Andreas et al., 2017; Andrychowicz et al., 2017). In mobile health applications, RL is employed to personalize sequences of treatments, treating each patient as a distinct task (Yom-Tov et al., 2017; Forman et al., 2019; Ghosh et al., 2023; Liao et al., 2020). Many algorithms (Andreas et al., 2017; Andrychowicz et al., 2017; Hessel et al., 2019; Yang et al., 2020) have been designed to jointly learn from multiple tasks, which shows significant improvement over these that learn each task individually. Towards the potential explanations of such improvement, recent advancements in Multitask Reinforcement Learning (MTRL) theory study the improved statistical efficiency in estimating unknown parameters by assuming a shared structure across tasks (Agarwal et al., 2022; Brunskill and Li, 2013; Calandriello et al., 2014; Cheng et al., 2022; Lu et al., 2021; Uehara et al., 2021; Xu et al., 2021; Yang

et al., 2022; Zhang and Wang, 2021). Similar setups originate from Multitask Supervised Learning, where it has been shown that learning from multiple tasks reduces the generalization error by a factor of $1/\sqrt{N}$ compared to single-task learning with N being the total number of tasks (Maurer et al., 2016; Du et al., 2020). Nevertheless, these studies overlook an essential aspect of RL–exploration.

Exploration design plays an important role in achieving sample-efficient learning. To understand how learning from multiple tasks, as opposed to single-task learning, could potentially benefit exploration design, we consider a generic MTRL scenario, where an algorithm interacts with a task set $\mathcal{M}$ in rounds T. *In each round, the algorithm chooses an exploratory policy π that is used to collect one episode of its own choice of $M \in \mathcal{M}$. A sample-efficient algorithm should output a near-optimal policy for each task in polynomial number of rounds.*

Previous sample-efficient algorithms for single-task learning ($|\mathcal{M}| = 1$) heavily rely on strategic design on the exploratory policies, such as Optimism in Face of Uncertainty (Auer et al., 2008; Bartlett and Tewari, 2009; Dann et al., 2017) and Posterior Sampling (Russo and Van Roy, 2014; Osband and Van Roy, 2017). Strategic design is criticized for either being restricted to environments with strong structural assumptions, or involving intractable computation oracle, such as non-convex optimization (Jiang, 2018; Jin et al., 2021a). In contrast, myopic exploration design like ϵ-greedy that injects random noise to a current greedy policy is easy to implement and performs well in a wide range of applications (Mnih et al., 2015; Kalashnikov et al., 2018), while it is shown to have exponential sample complexity in the worst case for single-task learning (Osband et al., 2019). Throughout the paper, we ask the main question:

Can algorithms with myopic exploration design be sample-efficient for MTRL?

In this chapter, we address the question by showing that *a simple algorithm that explores one task with ϵ-greedy policies from other tasks can be sample-efficient if the task set $\mathcal{M}$ is adequately diverse.* Our results may shed some light on the longstanding mystery that ϵ-greedy is successful in practice, while being shown sample-inefficient in theory. We argue that in a MTRL setting, ϵ-greedy policies may no longer behave myopically, as they explore myopically around the optimal policies from other tasks. When the task set is adequately diverse, this exploration may provide sufficient coverage.

To summarize our contributions, we discuss a sufficient diversity condition under a general value function approximation setting (Dann et al., 2022; Jin et al., 2021a). We show that the condition guarantees polynomial sample-complexity bound by running the aforementioned algorithm with myopic exploration design. We further discuss how to satisfy the diversity condition in different cases studies, including tabular cases, linear cases and linear quadratic

regulator cases. In the end, we validate our theory with experiments on synthetic robotic control environments, where we see that a diverse task set leads to a better policy learning and an improved generalization performance compared to that of non-diverse tasks.

5.2 Problem Setup

The following are notations that will be used throughout the paper.

Notation. For a positive integer H, we denote $[H] \coloneqq \{1, \dots, H\}$. For a discrete set $\mathcal{A}$, we denote $\Delta_{\mathcal{A}}$ by the set of distributions over $\mathcal{A}$. We use $\mathcal{O}$ and Ω to denote the asymptotic upper and lower bound notations and use $\tilde{\mathcal{O}}$ and $\tilde{\Omega}$ to hide the logarithmic dependence. Let $\{e_i\}_{i\in[d]}$ be the standard basis that spans $\mathbb{R}^d$. We let $N_{\mathcal{F}}(\rho)$ denote the ℓ_∞ covering number of a function class $\mathcal{F}$ at scale ρ. For a class $\mathcal{F}$, we denote the N-times Cartesian product of $\mathcal{F}$ by $(\mathcal{F})^{\otimes N}$.

5.2.1 Proposed Multitask Learning Scenario

Throughout the paper, we consider each task as an episodic MDP denoted by $M = (\mathcal{S}, \mathcal{A}, H, P_M, R_M)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $H \in \mathbb{N}$ is the horizon length in each episode, $P_M = (P_{h,M})_{h\in[H]}$ is the collection of transition kernels, and $R_M = (R_{h,M})_{h\in[H]}$ is the collection of immediate reward distributions. Note that we consider the setting, where all the tasks share the same state space, action space, and horizon length.

An agent interacts with an MDP M in the following way: starting with a fixed initial state s_1, at each step $h \in [H]$, the agent decides an action a_h and the environment samples the next state $s_{h+1} \sim P_{h,M}(\cdot \mid s_h, a_h)$ and next reward $r_h \sim R_{h,M}(s_h, a_h)$. An episode is a sequence of states, actions, and rewards $(s_1, a_1, r_1, \dots, s_H, a_h, r_H, s_{H+1})$. In general, we assume that the sum of r_h is upper bounded by 1 for any action sequence almost surely. The goal of an agent is to maximize the cumulative reward $\sum_{h=1}^{H} r_h$ by optimizing their actions.

The agent chooses actions based on *Markovian policies* denoted by $\pi = (\pi_h)_{h\in[H]}$ and each π_h is a mapping $\mathcal{S} \mapsto \Delta_{\mathcal{A}}$, where $\Delta_{\mathcal{A}}$ is the set of all distributions over $\mathcal{A}$. Let Π denote the space of all such policies. For a finite action space, we let $\pi_h(a \mid s)$ denote the probability of selecting action a given state s at the step h. In case of the infinite action space, we slightly abuse the notation by letting $\pi_h(\cdot \mid s)$ denote the density function.

Proposed learning scenario and objective. We consider the following multitask RL learning scenario. An algorithm interacts with a set of tasks $\mathcal{M}$ sequentially for T rounds.

At the each round t, the algorithm chooses an exploratory policy, which is used to collect one episode of its own choice of $M \in \mathcal{M}$. At the end of T rounds, the algorithm outputs a set of policies $\{\pi_M\}_{M\in\mathcal{M}}$. The goal of an algorithm is to learn a near-optimal policy π_M for each task $M \in \mathcal{M}$. The sample complexity of an algorithm is defined as follows.

Definition 5.1 (MTRL Sample Complexity)**.** *An algorithm is said to have sample-complexity of $\mathcal{C} : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{N}$ for a task set $\mathcal{M}$ if for any $\beta > 0, \delta \in (0, 1)$, it outputs a β-optimal policy π_M for each MDP $M \in \mathcal{M}$ with probability at least $1 - \delta$, by interacting with the task set for $\mathcal{C}(\beta, \delta)$ rounds.*

A sample-efficient algorithm should have a sample-complexity polynomial in the parameters of interests. For the tabular case, where the state space and action space are finite, $\mathcal{C}(\beta, \delta)$ should be polynomial in $|\mathcal{S}|$, $|\mathcal{A}|$, $|\mathcal{M}|$, H, and $1/\beta$ for a sample-efficient learning. Current state-of-the-art algorithm (Zhang et al., 2021) on a single-task tabular MDP achieves sample-complexity of $\tilde{\mathcal{O}}(|\mathcal{S}||\mathcal{A}|/\beta^2)$[1]. This bound translates to a MTRL sample-complexity bound of $\tilde{\mathcal{O}}(|\mathcal{M}||\mathcal{S}||\mathcal{A}|/\beta^2)$ by running their algorithm individually for each $M \in \mathcal{M}$. However, their exploration design closely follows the principle of Optimism in Face of Uncertainty, which is normally criticized for over-exploring.

5.2.2 Value Function Approximation

We consider the setting where value functions are approximated by general function classes. Denote the value function of an MDP M with respect to a policy π by

$$Q^{\pi}_{h,M}(s, a) = \mathbb{E}^{M}_{\pi}\left[r_h + V^{\pi}_{h+1,M}\left(s_{h+1}\right) \mid s_h = s, a_h = a\right]$$
$$V^{\pi}_{h,M}(s) = \mathbb{E}^{M}_{\pi}\left[Q^{\pi}_{h,M}\left(s_h, a_h\right) \mid s_h = s\right],$$

where by $\mathbb{E}^{M}_{\pi}$, we take expectation over the randomness of trajectories sampled by policy π on MDP M and we let $V^{\pi}_{H+1,M}(s) \equiv 0$ for all $s \in \mathcal{S}$ and $\pi \in \Pi$. We denote the optimal policy for MDP M by π^*_M. The corresponding value functions are denoted by $V^*_{h,M}$ and $Q^*_{h,M}$, which is shown to satisfy Bellman Equation $\mathcal{T}^M_h Q^*_{h+1,M} = Q^*_{h,M}$, where for any $g : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$, $\left(\mathcal{T}^M_h g\right)(s, a) = \mathbb{E}[r_h + \max_{a'\in\mathcal{A}} g\left(s_{h+1}, a'\right) \mid s_h = s, a_h = a]$.

The agent has access to a collection of function classes $\mathcal{F} = (\mathcal{F}_h : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R})_{h\in[H+1]}$. We assume that different tasks share the same set of function class. For each $f \in \mathcal{F}$, we denote $f_h \in \mathcal{F}_h$ the h-th component of the function f. We let $\pi^f = \{\pi^f_h\}_{h\in[H]}$ be the greedy policy with $\pi^f_h(s) = \arg\max_{a\in\mathcal{A}} f_h(s, a)$. When it is clear from the context, we slightly abuse the

[1]This bound is under the regime with $1/\beta \gg |\mathcal{S}|$

notation and let $f \in (\mathcal{F})^{\otimes|\mathcal{M}|}$ be a joint function for all the tasks. We further let f_M denote the function for the task M and let $f_{h,M}$ be its h-th component.

Define Bellman error operator $\mathcal{E}_h^M$ such that $\mathcal{E}_h^M f = f_h - \mathcal{T}_h^M f_{h+1}$ for any $f \in \mathcal{F}$. The goal of the learning algorithm is to approximate $Q_{h,M}^*$ through the function class $\mathcal{F}_h$ by minimizing the empirical Bellman error for each step h and task M.

To provide theoretical guarantee on this practice, we make the following realizability and completeness assumptions. The two assumptions and their variants are commonly used in the literature (Dann et al., 2017; Jin et al., 2021a).

Assumption 5.1 (Realizability and Completeness)**.** *For any MDP M considered in this chapter, we assume $\mathcal{F}$ is realizable and complete under the Bellman operator such that $Q_{h,M}^* \in \mathcal{F}_h$ for all $h \in [H]$ and for every $h \in [H]$, $f_{h+1} \in \mathcal{F}_{h+1}$ there is a $f_h \in \mathcal{F}_h$ such that $f_h = \mathcal{T}_h^M f_{h+1}$.*

5.2.3 Myopic Exploration Design

As opposed to carefully designed exploration, myopic exploration injects random noise to the current greedy policy. For a given greedy policy π, we use $\mathrm{expl}(\pi)$ to denote the myopic exploration policy based on π. Depending on the action space, the function expl can take different forms. The most common choice for finite action spaces is ϵ-greedy, which mixes the greedy policy with a random action: $\mathrm{expl}(\pi_h)(a \mid s) = (1-\epsilon_h)\pi_h(a \mid s) + \epsilon_h/A$.[2] As it is is our main study of exploration strategies, we let expl be ϵ-greedy function if not explicitly specified. For a continuous action space, we consider exploration with Gaussian noise: $\mathrm{expl}(\pi_h)(a \mid s) = (1-\epsilon_h)\pi_h(a \mid s) + \epsilon_h \exp(-a^2/2\sigma_h^2)/\sqrt{2\pi\sigma_h^2}$. Gaussian noise is useful for Linear Quadratic Regulator (LQR) setting, which will be discussed in Appendix 5.C.

5.3 Generic Multitask RL Algorithm

In this section, we introduce a generic algorithm (Algorithm 6) for the proposed multitask RL scenario without any strategic exploration, whose theoretical properties will be studied throughout the paper. In a typical single-task learning, a myopically exploring agent samples trajectories by running its current greedy policy estimated from the historical data equipped with naive explorations like ϵ-greedy.

In light of the exploration benefits of MTRL, we study Algorithm 6 as a counterpart of the single-task learning scenario in the MTRL setting. Algorithm 6 maintains a dataset for each MDP separately and different tasks interact in the following way: in each round

[2]Note that we consider a more general setup, where the exploration probability ϵ can depend on h.

Algorithm 6 explores every MDP with an exploratory policy that is the mixture (defined in Definition 5.2) of greedy policies of all the MDPs in the task set (Line 8). One way to interpret Algorithm 6 is that we share knowledge across tasks by policy sharing instead of parameter sharing or feature extractor sharing in the previous literature.

Definition 5.2 (Mixture Policy)**.** *For a set of policies* $\{\pi_i\}_{i=1}^N$*, we denote* $\text{Mixture}(\{\pi_i\}_{i=1}^N)$ *by the mixture of* N *policies, such that before the start of an episode, it samples* $I \sim \text{Unif}([N])$*, then runs policy* π_I *for the rest of the episode.*

We provide some justifications for the choice of the mixture policy. In the multi-goal RL setting (Andrychowicz et al., 2017; Chane-Sane et al., 2021; Liu et al., 2022), where the reward distribution is a function of goal-parameters, it is a common practice (Andrychowicz et al., 2017) to relabel the rewards in trajectories in the experience buffer such that they were as if sampled for a different task. On expectation, this is equivalent to exploring one task with the mixture policy. More generally, many MTRL algorithms train one agent for all the tasks, where tasks may be implicitly distinguished by the state space or be indexed by a goal parameter (Portelas et al., 2020). In this scenario, the roll-outs of different tasks are often stored in the same replay buffer and the learning of one task uses the exploratory trajectories of all the other tasks.

The greedy policy is obtained from an offline learning oracle $\mathcal{Q}$ (Line 4) that maps a dataset $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$ to a function $f \in \mathcal{F}$, such that $\mathcal{Q}(\mathcal{D})$ is an approximate solution to the following minimization problem

$$\arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \left(f_{h_i}(s_i, a_i) - r_i - \max_{a' \in \mathcal{A}} f_{h_i+1}(s_i', a') \right)^2 .$$

In practice, one can run fitted Q-iteration for an approximate solution.

5.4 Generic Sample Complexity Guarantee

In this section, we rigorously define the diversity condition and provide a sample-complexity bound for Algorithm 6. We start with introducing an intuitive example on how diversity encourages exploration in a multitask setting.

Motivating example. Figure 5.1 introduces a motivating example of grid-world environment on a long hallway with $N+1$ states. Since this is a deterministic tabular environment, whenever a task collects an episode that visits its goal state, running an offline policy optimization algorithm with pessimism will output its optimal policy.

Algorithm 6 Generic Algorithm for Multitask Reinforcement Learning

Input: function class $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_{H+1}$, task set $\mathcal{M}$, exploration function expl
Initialize $\mathcal{D}_{0,M} \leftarrow \emptyset$ for all $M \in \mathcal{M}$
for round $t = 1, 2, \ldots, \lfloor T/|\mathcal{M}| \rfloor$ **do**
    Offline learning oracle outputs $\hat{f}_{t,M} = \mathcal{Q}(\mathcal{D}_{t-1,M})$ for each M ▷ Offline learning
    Set myopic exploration policy $\hat{\pi}_{t,M} \leftarrow \text{expl}(\pi^{\hat{f}_{t,M}})$ for each M
    Set $\hat{\pi}_t \leftarrow \text{Mixture}(\{\hat{\pi}_{t,M}\}_{M \in \mathcal{M}})$ ▷ Exploratory policy
    for $M \in \mathcal{M}$ **do**
        Sample one episode $\tau_{t,M}$ on MDP M with policy $\hat{\pi}_t$ ▷ Collect new trajectory
        Add $\tau_{t,M}$ to the dataset: $\mathcal{D}_{t,M} \leftarrow \mathcal{D}_{t-1,M} \cup \{\tau_{t,M}\}$
    end for
end for
Return $\{\hat{\pi}_{t,M}\}_{M \in \mathcal{M}, t \in \lfloor T/|\mathcal{M}| \rfloor}$

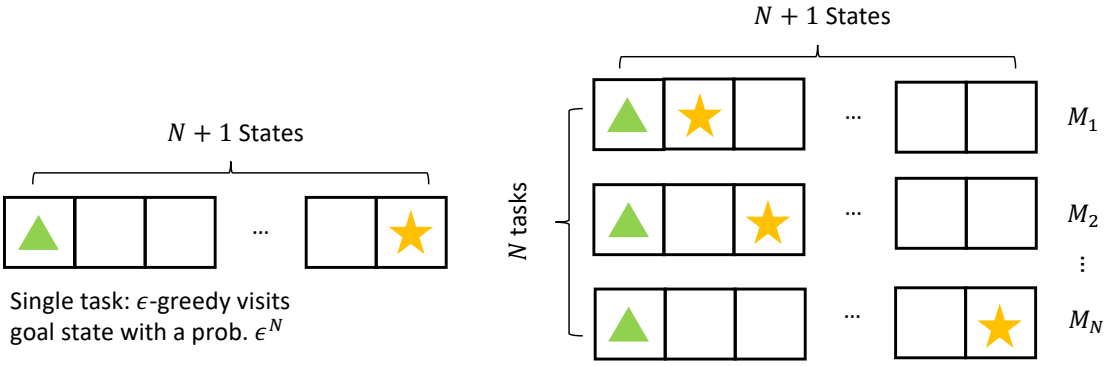


Figure 5.1: A diverse grid-world task set on a long hallway with $N+1$ states. From the left to the right, it represents a single-task and a multitask learning scenario, respectively. The triangles represent the starting state and the stars represent the goal states, where an agent receives a positive reward. The agent can choose to move forward or backward.

Left penal of Figure 5.1 is a single-task learning, where the goal state is N steps away from the initial state, making it exponentially hard to visit the goal state with ϵ-greedy exploration. Figure 5.1 on the right demonstrates a multitask learning scenario with N tasks, whose goal states "diversely" distribute along the hallway. A main message of this chapter is the advantage of exploring one task by running the ϵ-greedy policies of other tasks. To see this, consider any current greedy policies $(\pi_1, \pi_2, \ldots, \pi_N)$. Let i be the first non-optimal policy, i.e. all $j < i$, π_j is optimal for M_j. Since π_{i-1} is optimal, by running an ϵ-greedy of π_{i-1} on MDP M_i, we have a probability of $\prod_{h=1}^{i-1}(1-\epsilon_h)\epsilon_i$ to visit the goal state of M_i, allowing it to improve its policy to optimal in the next round. Such improvement can only happen for N times and all policies will be optimal within polynomial (in N) number of rounds if we choose $\epsilon_h = 1/(h+1)$. Hence, myopic exploration with a diverse task set

leads to sample-efficient learning. The rest of the section can be seen as generalizing this idea to function approximation.

5.4.1 Multitask Myopic Exploration Gap

Dann et al. (2022) proposed an assumption named Myopic Exploration Gap (MEG) that allows efficient myopic exploration for a single MDP under strong assumptions on the reward function, or on the mixing time. We extend this definition to the multitask learning setting. For the conciseness of the notation, we let $\text{expl}(f)$ denote the following mixture policy $\text{Mixture}(\{\text{expl}(\pi^{f_M})\}_{M\in\mathcal{M}})$ for a joint function $f \in (\mathcal{F})^{\otimes|\mathcal{M}|}$. Intuitively, a large myopic exploration gap implies that within all the policies that can be learned by the current exploratory policy, there exists one that can make significant improvement on the current greedy policy.

Definition 5.3 (Multitask Myopic Exploration Gap (Multitask MEG))**.** *For any $\mathcal{M}$, a function class $\mathcal{F}$, a joint function $f \in (\mathcal{F})^{\otimes|\mathcal{M}|}$ we say that f has $\alpha(f,\mathcal{M},\mathcal{F})$-myopic exploration gap, where $\alpha(f,\mathcal{M},\mathcal{F})$ is the value to the following maximization problem:*

$$\max_{M\in\mathcal{M}} \sup_{\tilde{f}\in\mathcal{F},c\geq 1} \frac{1}{\sqrt{c}}(V_{1,M}^{\tilde{f}} - V_{1,M}^{f_M}),\ \textit{s.t. for all } f^{'} \in \mathcal{F} \textit{ and } h \in [H],$$

$$\mathbb{E}_{\pi^{\tilde{f}}}^{M}[(\mathcal{E}_h^M f')(s_h,a_h)]^2 \leq c\mathbb{E}_{\text{expl}(f)}^{M}[(\mathcal{E}_h^M f')(s_h,a_h)]^2$$
$$\mathbb{E}_{\pi^{f_M}}^{M}[(\mathcal{E}_h^M f')(s_h,a_h)]^2 \leq c\mathbb{E}_{\text{expl}(f)}^{M}[(\mathcal{E}_h^M f')(s_h,a_h)]^2.$$

Let $M(f,\mathcal{M},\mathcal{F})$, $c(f,\mathcal{M},\mathcal{F})$ be the corresponding $M \in \mathcal{M}$ and c that attains the maximization.

Design of myopic exploration gap. To illustrate the spirit of this definition, we specialize to the tabular case, where conditions in Definition 5.3 can be replaced by a concentrability condition: for all $s,a,h \in \mathcal{S}\times\mathcal{A}\times[H]$, we require

$$\mu_{h,M}^{\pi^{\tilde{f}}}(s,a) \leq c\mu_{h,M}^{\text{expl}(f)}(s,a) \text{ and } \mu_{h,M}^{\pi^{f_M}}(s,a) \leq c\mu_{h,M}^{\text{expl}(f)}(s,a), \tag{5.1}$$

where $\mu_{h,M}^{\pi}(s,a)$ is the occupancy measure, i.e. the probability of visiting (s,a) at the step h by running policy π on MDP M.

The design of myopic exploration gap connects deeply to the theory of offline Reinforcement Learning. For a specific MDP M, Equation (5.1) defines a set of policies with concentrability assumption (Xie et al., 2021) that are the policies that can be accurately evaluated through the offline data collected by the current behavior policy. As an extension to the

definition in Dann et al. (2022), Multitask MEG considers the maximum myopic exploration gap over a set of MDPs and the behavior policy is a mixture of all the greedy policies.

5.4.2 Sample Complexity Guarantee

We propose Diversity in Definition 5.5, which relies on having a lower bounded Multitask MEG for any suboptimal policy. We then present Theorem 5.1 that provide an upper bound for sample complexity of Algorithm 6 by assuming diversity.

Definition 5.4 (Multitask Suboptimality)**.** *For a multitask RL problem with MDP set $\mathcal{M}$ and value function class $\mathcal{F}$. Let $\mathcal{F}_\beta \subset (\mathcal{F})^{\otimes|\mathcal{M}|}$ be the β-suboptimal class, such that for any $f \in \mathcal{F}_\beta$, there exists f_M and π^{f_M} is β-suboptimal for MDP M, i.e. $V_{1,M}^{\pi^{f_M}} \leq \max_{\pi\in\Pi} V_{1,M}^{\pi} - \beta$.*

Definition 5.5 (Diverse Tasks)**.** *For some function $\tilde{\alpha} : [0,1] \mapsto \mathbb{R}$, and $\tilde{c} : [0,1] \mapsto \mathbb{R}$, we say that a tasks set is $(\tilde{\alpha}, \tilde{c})$-diverse if any $f \in \mathcal{F}_\beta$ has multitask myopic exploration gap $\alpha(f, \mathcal{M}, \mathcal{F}) \geq \tilde{\alpha}(\beta)$ and $c(f, \mathcal{M}, \mathcal{F}) \leq \tilde{c}(\beta)$ for any constant $\beta > 0$.*

Theorem 5.1 (Upper Bound for Sample Complexity)**.** *Consider a multitask RL problem with MDP set $\mathcal{M}$ and value function class $\mathcal{F}$ such that $\mathcal{M}$ is $(\tilde{\alpha}, \tilde{c})$-diverse. Then Algorithm 6 with ϵ-greedy exploration function has a sample-complexity*

$$\mathcal{C}(\beta, \delta) = \mathcal{O}\left(|\mathcal{M}|^2 H^2 d_{\mathrm{BE}} \frac{\ln \tilde{c}(\beta)}{\tilde{\alpha}^2(\beta)\beta} \ln\left(\frac{N'_{\mathcal{F}}\left(T^{-1}\right) \ln T}{\delta}\right)\right),$$

where $d_{\mathrm{BE}} = \dim_{\mathrm{BE}}(\mathcal{F}, 1/\sqrt{T})$ is the Bellman-Eluder dimension of class $\mathcal{F}$ and $N'_{\mathcal{F}}(\rho) = \sum_{h=1}^{H-1} N_{\mathcal{F}_h}(\rho) N_{\mathcal{F}_{h+1}}(\rho)$. A definition of Bellman-Eluder dimension is deferred to Appendix 5.E.

Remark 5.1. *As a direct extension from Dann et al. (2022), we remark how their results translate to our setup. They provide a single-task learning sample complexity bound that depends on single-task MEG, which translates to a bound of $\tilde{\mathcal{O}}(|\mathcal{M}| H^2 d_{\mathrm{BE}}/(\alpha^2(\beta)\beta))$ for the multitask learning scenario. While there is an extra factor $|\mathcal{M}|$ in Theorem 5.1, Multitask MEG can be arbitrarily larger than single-task MEG, leading to potential exponential improvement on sample complexity.*

5.5 Lower Bounding Myopic Exploration Gap

Following the generic result in Theorem 5.1, the key to the problem is to lower bound myopic exploration gap $\tilde{\alpha}(\beta)$ and $c(\beta)$. Since $\tilde{c}(\beta) \leq 1/\tilde{\alpha}^2(\beta)$ and the upper bound in Theorem 5.1

depends logarithmically in $\tilde{c}(\beta)$, we only need to lower bound $\tilde{\alpha}(\beta)$. In this section, we lower bound $\tilde{\alpha}(\beta)$ for the linear MDP case. We defer an improved analysis for the tabular case and an analysis for the Linear Quadratic Regulator cases to Appendix 5.C.

Linear MDPs have been an important case study for the theory of RL (Chen et al., 2022b; Jin et al., 2020b; Wang et al., 2019b). It is a more general case than tabular MDP and has strong implication for Deep RL. In order to employ ϵ-greedy, we consider finite action space, while the state space can be infinite.

Definition 5.6 (Linear MDP (Jin et al., 2020b)). *An MDP is called linear MDP if its transition probability and reward function admit the following form.* $P_h(s' \mid s, a) = \langle \phi_h(s,a), \mu_h(s') \rangle$ *for some known function* $\phi_h : \mathcal{S} \times \mathcal{A} \mapsto (\mathbb{R}^+)^d$ *and unknown function* $\mu_h : \mathcal{S} \mapsto (\mathbb{R}^+)^d$. $R_h(s,a) = \langle \phi_h(s,a), \theta_h \rangle$ *for unknown parameters* θ_h [3]. *Without loss of generality, we assume* $\|\phi_h(s,a)\| \leq 1$ *for all* $s, a, h \in \mathcal{S} \times \mathcal{A} \times \mathcal{H}$ *and* $\max\{\|\mu_h(s)\|, \|\theta_h\|\} \leq \sqrt{d}$ *for all* $s, h \in \mathcal{S} \times [H]$.

An important property of Linear MDPs is that the value function also takes the linear form and the linear function class defined below satisfies Assumption 5.1.

Proposition 5.1 (Proposition 2.3 (Jin et al., 2020b)). *For linear MDPs, we have for any policy* π, $Q^\pi_{h,M}(s,a) = \langle \phi_h(s,a), w^\pi_{h,M} \rangle$, *where* $w^\pi_{h,M} = \theta_{h,M} + \int_{\mathcal{S}} V^\pi_{h+1,M}(s') \mu_h(s') ds' \in \mathbb{R}^d$. *Therefore, we only need to consider* $\mathcal{F}_h = \{(s,a) \mapsto \langle \phi_h(s,a), w \rangle : w \in \mathbb{R}^d, \|w\|_2 \leq 2\sqrt{d}\}$.

Now we are ready to define a diverse set of MDPs for the linear MDP case.

Definition 5.7 (Diverse MDPs for linear MDP case). *We say* $\mathcal{M}$ *is a diverse set of MDPs for the linear MDP case, if they share the same feature extractor* ϕ_h *and the same measure* μ_h *(leading to the same transition probabilities) and for any* $h \in [H]$, *there exists a subset* $\{M_{i,h}\}_{i \in [d]} \subset \mathcal{M}$, *such that the reward parameter* $\theta_{h, M_{i,h}} = e_i$ *and all the other* $\theta_{h', M_{i,h}} = 0$ *with* $h' \neq h$. [4]

We need the assumption that the minimum eigenvalue of the covariance matrix is strictly lower bounded away from 0. The feature coverage assumption is commonly use in the literature that studies Linear MDPs (Agarwal et al., 2022). Suppose Assumption 5.2 hold, we have Theorem 5.2, which lower bounds the multitask myopic exploration gap. Combined with Theorem 5.1, we have a sample-complexity bound of $\tilde{\mathcal{O}}(|\mathcal{M}|^3 H^3 d^2 |\mathcal{A}| / (\beta^3 b_1^2))$ with $|\mathcal{M}| \geq d$.

[3] Note that we consider non-negative functions only as μ_h is interpreted as a measure, which is normally non-negative.

[4] Note that this diversity definition is quite restricted even for the linear MDPs. We discuss a potential attempt to extend this definition and its technical challenge in Appendix 5.E.

Assumption 5.2 (Feature coverage)**.** *For any $\nu \in \mathbb{S}^{d-1}$ and $[\nu]_i > 0$ for all $i \in [d]$, there exists a policy π such that $\mathbb{E}_\pi[\nu^\intercal \phi_h(s_h, a_h)] \geq b_1$, for some constant $b_1 > 0$.*

Theorem 5.2. *Consider $\mathcal{M}$ to be a diverse set as in Definition 5.7. Suppose Assumption 5.2 holds and $\beta \leq b_1/2$, then we have for any $f \in \mathcal{F}_\beta$, $\alpha(f, \mathcal{F}, \mathcal{M}) = \Omega(\sqrt{\beta^2 b_1^2/(|\mathcal{A}||\mathcal{M}|H)}$ by setting $\epsilon_h = 1/(h+1)$.*

Proof highlight. We highlight the critical steps in the proof. The key is to show that near-optimal policies lead to a full-rank feature covariance matrix at each step. Fix a $f \in \mathcal{F}_\beta$ and let $\pi = \text{expl}(\{\pi^{f_M}\}_{M \in \mathcal{M}})$. Let h' be the step such that $\{M_{i,h'}\}_{i \in [d]}$ are all β-optimal, while some MDP in $\{M_{i,h'+1}\}_{i \in [d]}$ is not.

Let $\Phi_h^\pi = \mathbb{E}_\pi \phi_h(s_h, a_h)\phi_h(s_h, a_h)^\intercal$ be the covariance matrix of the embeddings at the step h by executing policy π. We first observe that if Φ_h^π at the step h' is full rank, the concentrability condition in Definition 5.3 is satisfied for any policy for MDPs with positive rewards at the step $h'+1$ (Lemma 5.1).

Lemma 5.1. *Let $\mathcal{F}$ be the function class in Proposition 5.1. For any policy π such that $\lambda_{min}(\Phi_h^\pi) \geq \underline{\lambda}$, then for any policy π' and $f' \in \mathcal{F}$, $\mathbb{E}_{\pi'}^M \left[\left(\mathcal{E}_h^M f' \right)^2 (s_h, a_h) \right] \leq \mathbb{E}_\pi^M \left[\left(\mathcal{E}_h^M f' \right)^2 (s_h, a_h) \right] / \underline{\lambda}$.*

The proof is complete by further showing that near-optimal policies at the step h' leads a full-rank $\Phi_{h'+1}^\pi$ (Lemma 5.2). The following lemma connects the coverage between two successive steps. We show that the feature covariance matrix at the step $h'+1$ is full-rank as long as the MDPs $\{M_{i,h}\}_{i \in [d]}$ are $b_1/2$-optimal.

Lemma 5.2. *Fix a step h and fix a $\beta < b_1/2$. Let $\{\pi_i\}_{i=1}^d$ be d policies such that π_i is a β-optimal policy for $M_{i,h}$ as in Definition 5.7. Let $\tilde{\pi} = \text{Mixture}(\{\text{expl}(\pi_i)\}_{i=1}^d)$. Then for any $\nu \in \mathbb{S}^{d-1}$, we have $\lambda_{\min}(\Phi_{h+1}^{\tilde{\pi}}) \geq \epsilon_h \prod_{h'=1}^{h-1}(1 - \epsilon_{h'}) b_1^2/(2dA)$.*

Connections to curriculum learning. Note that the proof reflects an interesting connection to curriculum learning, as we show that the near-optimal policies at the step h lead to a good coverage at the step $h+1$, which allows the algorithm to learn the optimal policies at the step $h+1$. The proof is proceeded as if it follows a curriculum from smaller steps to larger steps.

5.5.1 Discussions on the Tabular Case

Diverse tasks in Definition 5.7, when specialized to the tabular case, corresponds to $S \times H$ sparse-reward MDPs. Interestingly, similar constructions are used in reward-free exploration

(Jin et al., 2020a), which shows that by calling an online-learning oracle individually for all the sparse reward MDPs, one can generate a dataset that outputs a near-optimal policy for any given reward function. We want to point out the intrinsic connection between the two settings: our algorithm, instead of generating an offline dataset all at once, generates an offline dataset at each step h that is sufficient to learn a near-optimal policy for MDPs that corresponds to the step $h+1$.

Relaxing coverage assumption. Though feature coverage Assumption (Assumption 5.2) is handy for our proof as it guarantees that any β-optimal policy (with $\beta < b_1/2$) has a probability at least $b_1/2$ to visit their goal state, this assumption may not be reasonable for the tabular MDP case. Intuitively, without this assumption, a β-optimal policy can be an arbitrary policy and we can have at most S such policies in total leading to a cumulative error of $S\beta$. A naive solution is to request a $S^{-H}\beta$ accuracy at the first step, which leads to exponential sample-complexity. In Appendix 5.D, we show that an improved analysis can be done for the tabular MDP without having to make the coverage assumption. However, an extra price of SH has to be paid.

5.6 Implications of Diversity on Robotic Control Environments

In this section, we conduct simulation studies on robotic control environments with practical interests. Since myopic exploration has been shown empirically efficient in many problems of interest, we focus on the other main topic–diversity. We investigate how our theory guides a diverse task set selection. More specifically, our prior analysis on Linear MDPs suggests that a diverse task set should prioritize tasks with full-rank feature covariance matrices. We ask whether tasks with a more spread spectrum of the feature covariance matrix lead to a better training task set. *Note that the goal of this experiment is not to show the practical interests of Algorithm 6. Instead, we are revealing interesting implications of the highly conceptual definition of diversity in problems with practical interests.*

Environment and training setup. We adopt the BipedalWalker environment from Portelas et al. (2020). The learning agent is embodied into a bipedal walker whose motors are controllable with torque (i.e. continuous action space). The observation space consists of laser scan, head position, and joint positions. The objective of the agent is to move forward as far as possible, while crossing stumps with varying heights at regular intervals (see Figure 5.2 (a)). The agent receives positive rewards for moving forward and negative rewards for

torque usage. An environment or task, denoted as $M_{p,q}$, is controlled by a parameter vector (p, q), where p and q denote the heights of the stumps and the spacings between the stumps, respectively. Intuitively, an environment with higher and denser stumps is more challenging to solve. We set the parameter ranges for p and q as $p \in [0, 3]$ and $q \in [0, 6]$ in this study.

The agent is trained by Proximal Policy Optimization (PPO) (Schulman et al., 2017) with a standard actor-critic framework (Konda and Tsitsiklis, 1999) and with ϵ-greedy exploration. The architecture for the actor and critic feature extractors consists of two layers with 400 and 300 neurons, respectively, and Tanh (Rumelhart et al., 1986) as the activation function. Fully-connected layers are then used to compute the action and value. We keep the hyper-parameters for training the agent the same as Romac et al. (2021).

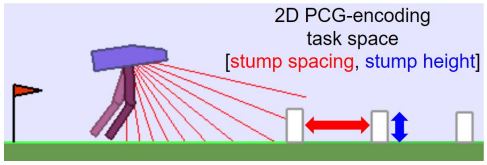
5.6.1 Investigating Feature Covariance Matrix

We denote by $\phi(s, a)$ the output of the feature extractor. We evaluate the extracted feature at the end of the training generated by near-optimal policies, denoted as π, on 100 tasks with different parameter vectors (p, q). We then compute the covariance matrix of the features for each task, denoted as $V_{p,q} = \mathbb{E}_{\pi}^{M_{p,q}} \sum_{h=1}^{H} \phi(s_h, a_h)\phi(s_h, a_h)^T$, whose spectrums are shown in Figure 5.2 (b) and (c).

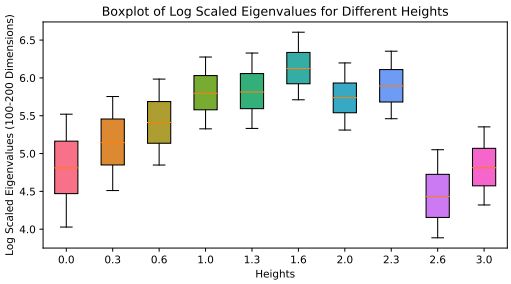
By ignoring the extremely large and small eigenvalues on two ends, we focus on the largest 100-200 dimension, where we observe that the height of the stumps p has a larger impact on the distribution of eigenvalues. In Figure 5.2 (b), we show the boxplot of the log-scaled eigenvalues of 100-200 dimensions for environments with different heights, where we average spacings. We observe that the eigenvalues can be 10 times higher for environments with an appropriate height (1.0-2.3), compared to extremely high and low heights. However, the scales of eigenvalues are roughly the same if we control the spacings and take average over different heights as shown in Figure 5.2 (c). This indicates that choosing an appropriate height is the key to properly scheduling tasks.

In fact, the task selection coincidences with the tasks selected by the state-of-the-art Automatic Curriculum Learning (ACL). We investigate the curricula generated by ALP-GMM (Portelas et al., 2020), a well-established curriculum learning algorithm, for training an agent in the BipedalWalker environment for 20 million timesteps. Figure 5.2 (d) gives the density plots of the ACL task sampler during the training process, which shows a significant preference over heights in the middle range, with little preference over spacing.

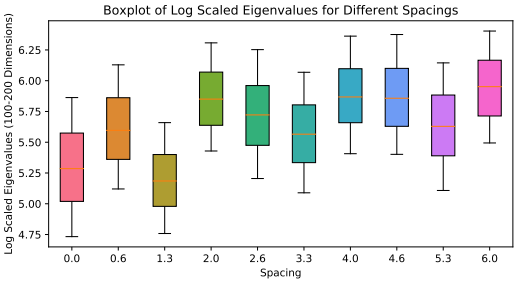
Training on different parameters. To further validate our finding, we train the same PPO agent with different means of the stump heights and see that how many tasks does the current agent master. As we argued in the theory, a diverse set of tasks provides good



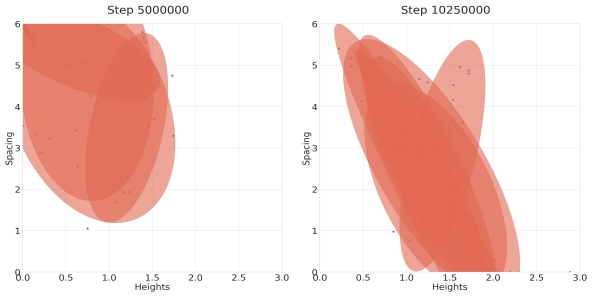
(a) BipedalWalker environment



(b) Controlling heights



(c) Controlling spacings



(d) Preference of automatic CL

Figure 5.2: **(a)** BipedalWalker Environment with different stump spacing and heights. **(b-c)** Boxplots of the log-scaled eigenvalues of sample covariance matrices of the trained embeddings generated by the near optimal policies for different environments. In (b), we take average over environments with the same height while in (c), over the same spacing. **(d)** Task preference of automatically generated curriculum at 5M and 10M training steps respectively. The red regions are the regions where a task has a higher probability to be sampled.

behavior policies for other tasks of interest. Therefore, we also test how many tasks it could further master if one use the current policy as behavior policy for fine-tuning on all tasks. The number of tasks the agent can master by learning on environments with heights ranging in [0.0, 0.3], [1.3, 1.6], [2.6, 3.0] are 28.1, 41.6, 11.5, respectively leading to a significant outperforming for diverse tasks ranging in [1.3, 1.6]. Table 5.1 in Appendix 5.F gives a complete summary of the results.

5.7 Discussions

In this chapter, we propose a new perspective of understanding the sample efficiency of myopic exploration design through diverse multitask learning. We show that by learning a diverse set of tasks, multitask RL algorithm with myopic exploration design can be sample-efficient. This chapter is a promising first step towards understanding the exploration benefits

of MTRL and it shed lights on the longstanding mystery of the empirical success of myopic exploration, which also leaves interesting future directions. A straightforward extension is to consider the setting where both transition functions and reward functions can differ.

Towards diversity for general function classes. Though Theorem 5.1 is presented for general value function approximators, we only studied the explicit form of diversity for Tabular MDPs, Linear MDPs and Linear Quadratic Regulator cases. How to achieve diversity for any general function class is an open problem. Recalling our proof for the Linear MDP case, a sufficient condition is to include a set of MDPs for each step h, such that the state distribution generated by their optimal policies satisfy the concentrability assumptions. In other words, any MDP with positive reward only at the step $h+1$ can be offline-learned through the dataset collected by these optimal policies. The diversity for general function classes poses the question on the number of tasks it takes to have sufficient coverage at the each step. We give a more detailed discussion of this topic in Appendix 5.E.

Improving sample-complexity bound. Our sample complexity bound can be sub-optimal. For instance, Theorem 5.1 specialized to the tabular case has an upper sample complexity bound of $|\mathcal{M}|^2 S^3 H^5 A^2/\beta^3$, which leaves a large gap between the current optimal bound of $|\mathcal{M}|SA/\beta^2$ if tasks are learned independently. We conjecture that this gap may originate from two factors. First, the nature of the myopic exploration makes it less efficient because the exploration are conducted in a layered manner. This might be complimented by a lower bound for myopic exploration. Second, our algorithm collects trajectories for every MDP with the mixture of all the policy in each round, which may be improved if a curriculum is prior known.

5.A Related Works

Multitask RL. Many recent theoretical works have contributed to understanding the benefits of MTRL (Agarwal et al., 2022; Brunskill and Li, 2013; Calandriello et al., 2014; Cheng et al., 2022; Lu et al., 2021; Uehara et al., 2021; Yang et al., 2022; Zhang and Wang, 2021) by exploiting the shared structures across tasks. An earlier line of works (Brunskill and Li, 2013) assumes that tasks are clustered and the algorithm adaptively learns the identity of each task, which allows it to pool observations. For linear Markov Decision Process (MDP) settings (Jin et al., 2020b), Lu et al. (2021) shows a bound on the sub-optimality of the learned policy by assuming a full-rank least-square value iteration weight matrix from source tasks. Agarwal et al. (2022) makes a different assumption that the target transition probability is a linear combination of the source ones, and the feature extractor is shared by all the tasks. Our work differs from all these works as we focus on the reduced complexity

of exploration design.

Curriculum learning. Curriculum learning refers to adaptively selecting tasks in a specific order to improve the learning performance (Bengio et al., 2009) under a multitask learning setting. Numerous studies have demonstrated improved performance in different applications (Jiang et al., 2015; Pentina et al., 2015; Graves et al., 2017; Wang et al., 2021). However, theoretical understanding of curriculum learning remains limited. Xu and Tewari (2022) study the statistical benefits of curriculum learning under Supervised Learning setting.

Myopic exploration. Myopic exploration, characterized by its ease of implementation and effectiveness in many problems (Kalashnikov et al., 2018; Mnih et al., 2015), is the most commonly used exploration strategy. Many theory works (Dabney et al., 2020; Dann et al., 2022; Liu and Brunskill, 2018; Simchowitz and Foster, 2020) have discussed the conditions, under which myopic exploration is efficient. However, all these studies consider a single MDP and require strong conditions on the underlying environment. Our paper closely follows Dann et al. (2022) where they define Myopic Exploration Gap. An MDP with low Myopic Exploration Gap can be efficiently learned by exploration exploration.

5.B Efficient Myopic Exploration for Deterministic MDP with known Curriculum

In light of the intrinsic connection between Algorithm 6 and curriculum learning. We present an interesting results for curriculum learning showing that any deterministic MDP can be efficiently learned through myopic exploration when a proper curriculum is given.

Proposition 5.2. *For any deterministic MDP M, with sparse reward, there exists a sequence of deterministic MDPs $M_1, M_2, \ldots, M_H$, such that the following learning process returns a optimal policy for M:*

1. *Initialize π_0 by a random policy.*
2. *For $t = 1, \ldots, n$, follow π_{t-1} with an ϵ-greedy exploration to collect $4At\log(H/\delta)$ trajectories denoted by $\mathcal{D}_t$. Compute the optimal policy π_t from the model learned by $\mathcal{D}_t$.*
3. *Output π_H.*

The above procedure will end in $O(AH^2 \log(H/\delta))$ episodes and with a probability at least $1-\delta$, π_H is the optimal policy for M.

Proof. We construct the sequence in the following manner. Let the optimal policy for an MDP M be π^*_M. Let the trajectory induced by π^*_M be $\{s^*_0, a^*_0, \ldots, s^*_H, a^*_H\}$. The MDP M receives a positive reward only when it reaches s^*_H. Without loss of generality, we assume that M is initialized at a fixed state s_0. We choose M_n such that

$$R_{M_i}(s,a) = \mathbb{1}(P_{M_n}(s,a) = s^*_i).$$

Furthermore, we set

$$P_{M_i}(s^*_i | s^*_i, a) = 1 \; \forall a \in \mathcal{A}$$

and

$$P_{M_i} = P_M$$

otherwise.

This ensures that any policy that reaches s^*_i on the i'th step is an optimal policy.

We first provide an upper bound on the expected number of episodes for finding an optimal policy using the above algorithm for M_i.

Fix $2 \leq i \leq H$. Let $\epsilon = \frac{1}{i}$. Define $k = |A|$. Then

Then the probability for reaching optimal reward for M_i is less than or equal to

$$(1 - \frac{1}{i})^{i-1}(\frac{1}{ki})$$

So the expected number of episodes to reach this optimal reward (and thus find an optimal policy) is

$$\frac{1}{(1-\frac{1}{i})^{i-1}(\frac{1}{ki})} = (i-1)k(\frac{i}{i-1})^i \leq 4k(i-1)$$

since $i \geq 2$ and $(\frac{i}{i-1})^i$ is decreasing. By Chebyshev's inequality, a successful visit can be found in $4k(i-1)\log(H/\delta)$ with a probability at least $1 - \delta/H$.

The expected total number of episodes for the all the MDP's is therefore

$$\sum_{j=2}^{H} 4k(j-1)\log(H/\delta) \leq \frac{H}{2}(4kH)\log(H/\delta)$$

which is $O(kH^2 \log(H/\delta))$. □

5.C Missing Proofs

Generic Upper Bound for Sample Complexity In this section, we prove the generic upper bound on sample complexity in Theorem 5.1. We first prove Lemma 5.3, which holds under the same condition of Theorem 5.1.

Lemma 5.3. *Consider a multitask RL problem with MDP set $\mathcal{M}$ and value function class $\mathcal{F}$ such that $\mathcal{M}$ is $(\tilde{\alpha}, \tilde{c})$-diverse. Then Algorithm 6 with exploration function* expl *satisfies that the total number of rounds, where there exists an MDP M, such that $\pi^{\hat{f}_{t,M}}$ is β-suboptimal for M, can be upper bounded by*

$$\mathcal{O}\left(|\mathcal{M}|^2 H^2 d \frac{\ln \tilde{c}(\beta)}{\tilde{\alpha}(\beta)^2} \ln\left(\frac{N'_{\mathcal{F}}\left(T^{-1}\right) \ln T}{\delta}\right)\right),$$

where $d = \dim_{BE}(\mathcal{F}, 1/\sqrt{T})$ is the Bellman-Eluder dimension of class $\mathcal{F}$ and $N'_{\mathcal{F}}(\rho) = \sum_{h=1}^{H-1} N_{\mathcal{F}_h}(\rho) N_{\mathcal{F}_{h+1}}(\rho)$.

Proof. Let us partition $\mathcal{F}_\beta$ into $\mathcal{F}_\beta = \{\mathcal{F}_{M,i}\}_{M\in\mathcal{M}, i\in[i_{\max}]}$ such that

$$\mathcal{F}_{M,i} \coloneqq \{f \in \mathcal{F}_\beta : c(f, \mathcal{M}, \mathcal{F}) \in [e^{i-1}, e^i] \text{ and } M(f, \mathcal{M}, \mathcal{F}) = M\}.$$

Furthermore, denote $(\hat{f}_{t,M})_{M\in\mathcal{M}}$ by $\hat{f}_t$. We define $\mathcal{K}_{M,i,t} = \{\tau \in [t], \hat{f}_\tau \in \mathcal{F}_{M,i}\}$. The proof in Dann et al. (2022) can be seen as bounding the sum of $\mathcal{K}_{M,i,t}$ for a specific M, while apply the same bound for each M, which leads to an extra $|\mathcal{M}|$ factor.

Lemma 5.4. *Under the same condition in Theorem 5.1 and the above definition, we have*

$$|\mathcal{K}_{M,i,T}| \le \mathcal{O}\left(\frac{H^2 d\left(\mathcal{F}'_i\right)}{\alpha_\beta^2} \ln \frac{N'_{\mathcal{F}}(1/T) \ln(T)}{\delta}\right).$$

Proof. In the following proof, we fix an MDP M and without further specification, the policies or rewards are with respect to the specific M. We study all the steps $t \in \mathcal{K}_{M,i,T}$.

For each $t \in \mathcal{K}_{M,i,T}$,

1. Recall that $\hat{\pi}_t$ is the mixture of exploration policy for all the MDPs: $\text{Mixture}(\{\text{expl}(\hat{\pi}_{t,M'})\}_{M'\in\mathcal{M}})$;
2. Define π'_t as the improved policy that attains the maximum in the multitask myopic exploration gap for $\hat{f}_t$ in Definition 5.3.

Note that π'_t is a policy for M since $t \in \mathcal{K}_{M,i,t}$. A key step in our proof is to upper bound the difference between the value of the current policy and the value of π'_t. By Lemma 5.5,

The total difference in return between the greedy policies and the improved policies can be bounded by

$$\sum_{t\in\mathcal{K}_{M,i,T}}(V_{1,M}^{\pi'_t}(s_1)-V_{1,M}^{\hat{\pi}_{t,M}}(s_1))\leq \tag{5.2}$$

$$\sum_{t\in\mathcal{K}_{M,i,T}}\sum_{h=1}^{H}\mathbb{E}_{\hat{\pi}_{t,M}}^{M}[(\mathcal{E}_h^M\hat{f}_{t,M})(s_h,a_h)]-\sum_{t\in\mathcal{K}_{M,i,T}}\sum_{h=1}^{H}\mathbb{E}_{\pi'_t}^{M}[(\mathcal{E}_h^M\hat{f}_{t,M})(s_h,a_h)], \tag{5.3}$$

where the exportation is taken over the randomness of the trajectory sampled for MDP M.

Under the completeness assumption in Assumption 5.1, by Lemma 5.6 we show that with a probability $1-\delta$ for all $(h,t)\in[H]\times[T]$,

$$\sum_{\tau=1}^{t-1}\mathbb{E}_{\hat{\pi}_\tau}^{M}\left[\left(\mathcal{E}_h f_{t,M}\right)(s_h,a_h)\right]^2\leq 3\frac{t-1}{T}+176\ln\frac{6N'_{\mathcal{F}}(1/T)\ln(2t)}{\delta}.$$

We consider only the event where this condition holds. Since $c(\hat{f}_t,\mathcal{M},\mathcal{F})\leq e^i$ for all $t\in\mathcal{K}_{M,i,T}$, by Definition 5.3 we bound

$$\begin{aligned}
&\sum_{\tau\in\mathcal{K}_{M,i,t-1}}\mathbb{E}_{\pi'_\tau}^{M}\left[\left(\mathcal{E}_h^M\hat{f}_{t,M}\right)(s_h,a_h)\right]^2\\
&\leq\sum_{\tau\in[t-1]}\mathbb{E}_{\pi'_\tau}^{M}\left[\left(\mathcal{E}_h^M\hat{f}_{t,M}\right)(s_h,a_h)\right]^2\\
&\leq e^i\sum_{\tau\in[t-1]}\mathbb{E}_{\hat{\pi}_\tau}^{M}\left[\left(\mathcal{E}_h^M\hat{f}_{t,M}\right)(s_h,a_h)\right]^2\\
&\leq 179e^i\ln\frac{6N'_{\mathcal{F}}(1/T)\ln(2t)}{\delta}.
\end{aligned}$$

Combined with the distributional Eluder dimension machinery in Lemma 5.8, this implies that

$$\sum_{t\in\mathcal{K}_{M,i,T}}\left|\mathbb{E}_{\pi'_t}^{M}\left[\left(\mathcal{E}_h^M\hat{f}_{t,M}\right)(s_h,a_h)\right]\right|\leq$$

$$\mathcal{O}\left(\sqrt{e^i d\left(\mathcal{F}'_i\right)\ln\frac{N'_{\mathcal{F}}(1/T)\ln(T)}{\delta}\left|\mathcal{K}_{M,i,T}\right|}+\min\left\{\left|\mathcal{K}_{M,i,T}\right|,d\left(\mathcal{F}'_i\right)\right\}\right),$$

Note that we can derive the same upper-bound for $\sum_{t\in\mathcal{K}_{M,i,T}}\left|\mathbb{E}_{\pi_t}^{M}\left[\left(\mathcal{E}_h^M\hat{f}_{t,M}\right)(s_h,a_h)\right]\right|$. Then

plugging the above two bounds into Equation (5.2), we obtain

$$\sum_{t\in\mathcal{K}_{M,i,T}}\left(V_{1,M}^{\pi_t'}(s_1)-V_{1,M}^{\hat{\pi}_{t,M}}(s_1)\right)\leq\mathcal{O}\left(\sqrt{e^i H^2 d\left(\mathcal{F}_i'\right)\ln\frac{N_{\mathcal{F}}'(1/T)\ln(T)}{\delta}\left|\mathcal{K}_{M,i,T}\right|}+Hd\left(\mathcal{F}_i'\right)\right)$$

By the definition of myopic exploration gap, we lower bound the LHS by

$$\sum_{t\in\mathcal{K}_{M,i,T}}\left(V_{1,M}^{\pi_t'}(s_1)-V_{1,M}^{\hat{\pi}_{t,M}}(s_1)\right)\geq\left|\mathcal{K}_{M,i,T}\right|\sqrt{e^{i-1}}\alpha_\beta.$$

Combining both bounds and rearranging yields

$$\left|\mathcal{K}_{M,i,T}\right|\leq\mathcal{O}\left(\frac{H^2 d\left(\mathcal{F}_i'\right)}{\alpha_\beta^2}\ln\frac{N_{\mathcal{F}}'(1/T)\ln(T)}{\delta}\right)$$

□

Summing over $M\in\mathcal{M}$ and $i\in[i_{max}]$, we conclude Theorem 5.1. □

Lemma 5.5 (**Lemma 3** Dann et al. (2022))**.** *For any MDP M, let $f=\{f_h\}_{h\in[H]}$ with $f_h:\mathcal{S}\times\mathcal{A}\mapsto\mathbb{R}$ and π^f is the greedy policy of f. Then for any policy π',*

$$V_1^{\pi'}(s_1)-V_1^{\pi^f}(s_1)\leq\sum_{h=1}^{H}\mathbb{E}_{\pi^f}^{M}\left[\left(\mathcal{E}_h f\right)(s_h,a_h)\right]-\sum_{h=1}^{H}\mathbb{E}_{\pi'}^{M}\left[\left(\mathcal{E}_h f\right)(s_h,a_h)\right].$$

Lemma 5.6 (Modified from **Lemma 4** Dann et al. (2022))**.** *Consider a sequence of policies $(\pi_t)_{t\in\mathbb{N}}$. At step τ, we collect one episode using $\hat{\pi}_\tau$ and define $\hat{f}_\tau$ as the fitted Q-learning estimator up to step t over the function class $\mathcal{F}=\{\mathcal{F}\}_{h\in[H]}$. Let $\rho\in\mathbb{R}^+$ and $\delta\in(0,1)$. If $\mathcal{F}$ satisfies Assumption 5.1, then with a probability at least $1-\delta$, for all $h\in[H]$ and $t\in\mathbb{N}$,*

$$\sum_{\tau=1}^{t-1}\mathbb{E}_{\hat{\pi}_\tau}^{M}[(\mathcal{E}_h\hat{f}_t)(s_h,a_h)]^2\leq 3\rho t+176\ln\frac{6N_{\mathcal{F}}'(\rho)\ln(2t)}{\delta},$$

where $N_{\mathcal{F}}'(\rho)=\sum_{h=1}^{H}N_{\mathcal{F}_h}(\rho)N_{\mathcal{F}_{h+1}}(\rho)$ is the sum of ℓ_∞ covering number of $\mathcal{F}_h\times\mathcal{F}_{h+1}$ w.r.t. radius $\rho>0$.

Proof. The only difference between our statement and the statement in Dann et al. (2022) is that they consider $\hat{\pi}_\tau=\mathrm{expl}(\hat{f}_\tau)$, while this statement holds for any data-collecting policy $\hat{\pi}_\tau$. To show this, we go through the complete proof here.

Consider a fixed $t\in\mathbb{N}$, $h\in[H]$ and $f=\{f_h,f_{h+1}\}$ with $f_h\in\mathcal{F}_h$, $f_{h+1}\in\mathcal{F}_{h+1}$. Let

$(x_{t,h}, a_{t,h}, r_{t,h})_{t\in\mathbb{N},h\in[H]}$ be the collected trajectory in $[t]$. Then

$$\begin{aligned}
Y_{t,h}(f) &= \left(f_h\left(x_{t,h}, a_{t,h}\right) - r_{t,h} - \max_{a'} f_{h+1}\left(x_{t,h+1}, a'\right)\right)^2 - \\
&\qquad \left(\left(\mathcal{T}_h f_{h+1}\right)\left(x_{t,h}, a_{t,h}\right) - r_{t,h} - \max_{a'} f_{h+1}\left(x_{t,h+1}, a'\right)\right)^2 \\
&= \left(f_h\left(x_{t,h}, a_{t,h}\right) - \left(\mathcal{T}_h f_{h+1}\right)\left(x_{t,h}, a_{t,h}\right)\right) \\
&\quad \times \left(f_h\left(x_{t,h}, a_{t,h}\right) + \left(\mathcal{T}_h f_{h+1}\right)\left(x_{t,h}, a_{t,h}\right) - 2r_{t,h} - 2\max_{a'} f_{h+1}\left(x_{t,h+1}, a'\right)\right).
\end{aligned}$$

Let $\mathfrak{F}_t$ be the σ-algebra under which all the random variables in the first $t-1$ episodes are measurable. Note that $|Y_{t,h}(f)| \le 4$ almost surely and the conditional expectation of $Y_{y,h}(f)$ can be written as

$$\mathbb{E}\left[Y_{t,h}(f) \mid \mathfrak{F}_t\right] = \mathbb{E}\left[\mathbb{E}\left[Y_{t,h}(f) \mid \mathfrak{F}_t, x_{t,h}, a_{t,h}\right] \mid \mathfrak{F}_t\right] = \mathbb{E}_{\pi_t}[\left(f_h - \mathcal{T}_h f_{h+1}\right)\left(x_h, a_h\right)^2].$$

The variance can be bounded by

$$\mathrm{Var}\left[Y_{t,h}(f) \mid \mathfrak{F}_t\right] \le \mathbb{E}\left[Y_{t,h}(f)^2 \mid \mathfrak{F}_t\right] \le 16\mathbb{E}\left[\left(f_h - \mathcal{T}_h f_{h+1}\right)\left(x_{t,h}, a_{t,h}\right)^2 \mid \mathfrak{F}_t\right] = 16\mathbb{E}\left[Y_{t,h}(f) \mid \mathfrak{F}_t\right],$$

where we used the fact that $|f_h(x_{t,h}, a_{t,h}) + (\mathcal{T}_h f_{h+1})(x_{t,h}, a_{t,h}) - 2r_{t,h} - 2\max_{a'} f_{h+1}(x_{h+1}, a')| \le 4$ almost surely. Applying Lemma 5.7 to the random variable $Y_{t,h}(f)$, we have that with probability at least $1-\delta$, for all $t \in \mathbb{N}$,

$$\begin{aligned}
\sum_{i=1}^{t} \mathbb{E}\left[Y_{i,h}(f) \mid \mathfrak{F}_i\right] &\le 2A_t \sqrt{\sum_{i=1}^{t} \mathrm{Var}\left[Y_{i,h}(f) \mid \mathfrak{F}_i\right]} + 12A_t^2 + \sum_{i=1}^{t} Y_{i,h}(f) \\
&\le 8A_t \sqrt{\sum_{i=1}^{t} \mathbb{E}\left[Y_{i,h}(f) \mid \mathfrak{F}_i\right]} + 12A_t^2 + \sum_{i=1}^{t} Y_{i,h}(f),
\end{aligned}$$

where $A_t = \sqrt{2\ln\ln(2t) + \ln(6/\delta)}$. Using AM-GM inequality and rearranging terms in the above we have

$$\sum_{i=1}^{t} \mathbb{E}\left[Y_{i,h}(f) \mid \mathfrak{F}_i\right] \le 2\sum_{i=1}^{t} Y_{i,h}(f) + 88A_t^2 \le 2\sum_{i=1}^{t} Y_{i,h}(f) + 176\ln\frac{6\ln(2t)}{\delta}.$$

Let $\mathcal{Z}_{\rho,h}$ be a ρ-cover of $\mathcal{F}_h \times \mathcal{F}_{h+1}$. Now taking a union bound over all $\phi_h \in \mathcal{Z}_{\rho,h}$ and

$h \in [H]$, we obtain that with probability at least $1-\delta$ for all ϕ_h and $h \in [H]$

$$\sum_{i=1}^{t} \mathbb{E}\left[Y_{i,h}\left(\phi_h\right) \mid \mathfrak{F}_i\right] \leq 2\sum_{i=1}^{t} Y_{i,h}\left(\phi_h\right) + 176 \ln \frac{6N'_{\mathcal{F}}(\rho)\ln(2t)}{\delta}.$$

This implies that with probability at least $1-\delta$ for all $f = \{f_h, f_{h+1}\} \in \mathcal{F}_h \times \mathcal{F}_{h+1}$ and $h \in [H]$,

$$\sum_{i=1}^{t} \mathbb{E}\left[Y_{i,h}(f) \mid \mathfrak{F}_i\right] \leq 2\sum_{i=1}^{t} Y_{i,h}(f) + 3\rho(t-1) + 176 \ln \frac{6N'_{\mathcal{F}}(\rho)\ln(2t)}{\delta}.$$

Let $\hat{f}_{t,h}$ be the h-th component of the function $\hat{f}_t$. The above inequality holds in particular for $f = \{\hat{f}_{t,h}, \hat{f}_{t,h+1}\}$ for all $t \in \mathbb{N}$. Finally, we have

$$\begin{aligned}
\sum_{i=1}^{t-1} Y_{i,h}\left(\widehat{f}_t\right) &= \sum_{i=1}^{t-1} \left(\widehat{f}_{t,h}\left(s_{i,h}, a_{i,h}\right) - r_{i,h} - \max_{a'} \widehat{f}_{t,h+1}\left(s_{i,h+1}, a'\right)\right)^2 \\
&\quad - \sum_{i=1}^{t-1} \left(\left(\mathcal{T}_h \widehat{f}_{t,h+1}\right)\left(s_{i,h}, a_{i,h}\right) - r_{i,h} - \max_{a'} \widehat{f}_{t,h+1}\left(s_{i,h+1}, a'\right)\right)^2 \\
&= \inf_{f' \in \mathcal{F}_h} \sum_{i=1}^{t-1} \left(f'\left(s_{i,h}, a_{i,h}\right) - r_{i,h} - \max_{a'} \widehat{f}_{t,h+1}\left(s_{i,h+1}, a'\right)\right)^2 \\
&\quad - \sum_{i=1}^{t-1} \left(\left(\mathcal{T}_h \widehat{f}_{t,h+1}\right)\left(s_{i,h}, a_{i,h}\right) - r_{i,h} - \max_{a'} \widehat{f}_{t,h+1}\left(s_{i,h+1}, a'\right)\right)^2 \\
&\leq 0,
\end{aligned}$$

where the last inequality follows from the completeness in Assumption 5.1. □

Lemma 5.7 (Time-Uniform Freedman Inequality)**.** *Suppose $\{X_t\}_{t=1}^{\infty}$ is a martingale difference sequence with $|X_t| \leq b$. Let*

$$\mathrm{Var}_\ell\left(X_\ell\right) = \mathrm{Var}\left(X_\ell \mid X_1, \cdots, X_{\ell-1}\right).$$

Let $V_t = \sum_{\ell=1}^{t} \mathrm{Var}_\ell(X_\ell)$ be the sum of conditional variances of X_t. Then we have that for any $\delta' \in (0,1)$ and $t \in \mathbb{N}$

$$\mathbb{P}\left(\sum_{\ell=1}^{t} X_\ell > 2\sqrt{V_t}A_t + 3bA_t^2\right) \leq \delta'$$

where $A_t = \sqrt{2\ln\ln(2(\max(V_t/b^2, 1))) + \ln(6/\delta')}$.

Lemma 5.8 (Lemma 41 Jin et al. (2021a)). *Given a function class Φ defined on $\mathcal{X}$ with $|\phi(x)| \le C$ for all $(\phi, x) \in \Phi \times \mathcal{X}$ and a family of probability measures Π over $\mathcal{X}$. Suppose sequences $\{\phi_i\}_{i\in[K]} \subset \Phi$ and $\{\mu_i\}_{i\in[K]} \subset \Pi$ satisfy for all $k \in [K]$ that $\sum_{i=1}^{k-1}(\mathbb{E}_{\mu_i}[\phi_k])^2 \le \beta$. Then for all $k \in [K]$ and $w > 0$,*

$$\sum_{t=1}^{k} |\mathbb{E}_{\mu_t}[\phi_t]| \le O\left(\sqrt{\dim_{DE}(\Phi, \Pi, \omega)\beta k} + \min\{k, \dim_{DE}(\Phi, \Pi, \omega)\} C + k\omega\right).$$

Proof of Theorem 5.1. Denote $B = \Theta\left(|\mathcal{M}|^2 H^2 d_{\mathrm{BE}} \frac{\ln \tilde{c}(\beta)}{\tilde{\alpha}(\beta)^2 \beta} \ln\left(\frac{\bar{N}_{\mathcal{F}}(T^{-1}) \ln T}{\delta}\right)\right)$. The following Corollary transform Lemma 5.3 to Theorem 5.1, whose proof directly follows by taking $T = B/\beta$. Since at most B rounds are suboptimal according to Lemma 5.3, the mixing of all T policies are β-optimal. This leads to a sample complexity

$$\mathcal{C}(\tilde{\alpha}, \tilde{c}) = \Theta\left(|\mathcal{M}|^2 H^2 d_{\mathrm{BE}} \frac{\ln \tilde{c}(\beta)}{\tilde{\alpha}(\beta)^2 \beta} \ln\left(\frac{\bar{N}_{\mathcal{F}}(T^{-1}) \ln T}{\delta}\right)\right)$$

Linear MDP case Note that in this section, we use $\mathbb{E}_\pi$ for the expectation over transition w.r.t a policy π.

Lemma 5.1. *Let $\mathcal{F}$ be the function class in Proposition 5.1. For any policy π such that $\lambda_{min}(\Phi_h^\pi) \ge \underline{\lambda}$, then for any policy π' and $f' \in \mathcal{F}$, $\mathbb{E}_{\pi'}[(\mathcal{E}_h^2 f')(s_h, a_h)] \le \mathbb{E}_\pi[(\mathcal{E}_h^2 f')(s_h, a_h)]/\underline{\lambda}$.*

Proof. Recall that $\Phi_h^\pi \coloneqq \mathbb{E}_\pi \phi_h(s_h, a_h)\phi_h(s_h, a_h)^\intercal$.

We derive the Bellman error term using the fact that f' is a linear function and the transitions admit the linear function as well. For any policy π, we have

$$\begin{aligned}
&\mathbb{E}_\pi[(\mathcal{E}_h^2 f')(s_h, a_h)] \\
&= \mathbb{E}_\pi\left[\left(f'_h(s_h, a_h) - \phi_h(s_h, a_h)^\intercal \theta_h - \max_{a'} \mathbb{E}_{s_{h+1}}[f'_{h+1}(s_{h+1}, a') \mid s_h, a_h]\right)^2\right] \\
&= \mathbb{E}_\pi\left[\left(\phi_h(s_h, a_h)^\intercal w_h - \phi_h(s_h, a_h)^\intercal \theta_h - \max_{a'} \mathbb{E}_{s_{h+1}}[\phi_{h+1}(s_{h+1}, a')^\intercal w_{h+1} \mid s_h, a_h]\right)^2\right] \\
&= \mathbb{E}_\pi\left[\left(\phi_h(s_h, a_h)^\intercal w_h - \phi_h(s_h, a_h)^\intercal \theta_h - \phi_h(s_h, a_h)^\intercal \int_{s'} \phi_{h+1}(s', \pi_{h+1}^{f'}(s'))^\intercal w_{h+1} \mu_h(s') ds'\right)^2\right] \\
&= \mathbb{E}_\pi\left[\left(\phi_h(s_h, a_h)^\intercal (w_h - \theta_h - w'_{h+1})\right)^2\right] \\
&= (w_h - \theta_h - w'_{h+1})^\intercal \mathbb{E}_\pi\left[\phi_h(s_h, a_h)\phi_h(s_h, a_h)^\intercal\right](w_h - \theta_h - w'_{h+1})
\end{aligned}$$

where $w'_{h+1} = \int_{s'} \phi_{h+1}(s', \pi_{h+1}^{f'}(s'))^\intercal w_{h+1} \mu_h(s') ds'$. Since by the assumption in Definition 5.6 that $\|\phi_h(s, a)\| \le 1$ for any s, a, we have $\Phi_h^{\pi'} \prec I$. The result follow by the condition that

$\lambda_{\min}(\Phi_h^\pi) \geq \underline{\lambda}$. $\square$

Lemma 5.2. *Fix a step h. Let $\{M_{i,h}\}_{i\in[d]}$ be the d MDPs such that $\theta_{h,M_{i,h}} = e_i$ as in Definition 5.6. Let $\{\pi_i\}_{i=1}^d$ be d policies such that π_i is a β-optimal policy for $M_{i,h}$ with $\beta < b_1/2$. Let $\tilde{\pi} = \text{Mixture}(\{\text{expl}(\pi_i)\}_{i=1}^d)$. Then for any $\nu \in \mathbb{S}^{d-1}$, we have $\lambda_{\min}(\Phi_{h+1}^{\tilde{\pi}}) \geq \epsilon_h \prod_{h'=1}^{h-1}(1-\epsilon_{h'})b_1^2/(2dA)$.*

Proof. Let π be any stationary policy and recall that Π is the set of all the stationary policies. We denote $A_h^\pi(s') \sim \pi_h(s')$ by the random variable for the action sampled at the step h using policy π given the state is s'. Let $\phi_h^\pi \coloneqq \mathbb{E}_\pi \phi_h(s_h, a_h)$.

We further define

$$a_{h+1}^\nu(s) \coloneqq \arg\max_{a\in\mathcal{A}} [\nu^\intercal \phi_{h+1}(s,a)\phi_{h+1}(s,a)^\intercal \nu].$$

Lower bound the following quadratic term for any unit vector $\nu \in \mathbb{R}^d$,

$$\begin{aligned}
&\max_{\pi\in\Pi} \nu^\intercal \Phi_{h+1}^\pi \nu \\
&= \max_{\pi\in\Pi} \mathbb{E}_\pi \left[\int_{s'} \nu^\intercal \phi_{h+1}(s', A_{h+1}^\pi(s'))\phi_{h+1}(s', A_{h+1}^\pi(s'))^\intercal \nu \mu_h(s')^\intercal \phi_h(s_h,a_h) ds' \right] \\
&= \max_{\pi} \mathbb{E}_\pi[\phi_h(s_h,a_h)^\intercal] \left(\int_{s'} \nu^\intercal \phi_{h+1}(s', a_{h+1}^\nu(s'))\phi_{h+1}(s', a_{h+1}^\nu(s'))^\intercal \nu \mu_h(s') ds' \right) \\
&= \max_{\pi\in\Pi} (\phi_h^\pi)^\intercal w_{h+1}^\nu.
\end{aligned}$$

where we let $w_{h+1}^\nu \coloneqq \int_{s'} \nu^\intercal \phi_{h+1}(s', a_{h+1}^\nu(s'))\phi_{h+1}(s', a_{h+1}^\nu(s'))^\intercal \nu \mu_h(s')^\intercal ds'$.

By Assumption 5.2, we have $\max_{\pi\in\Pi} \mathbb{E}_\pi[\phi_h(s_h,a_h)^\intercal] w_{h+1}^\nu \geq b_1^2$.

For the mixture policy $\tilde{\pi}$ defined in our lemma,

$$\begin{aligned}
\nu^\intercal \Phi_{h+1}^{\tilde{\pi}} \nu &= \frac{1}{d} \sum_{i=1}^d \mathbb{E}_{\text{expl}(\pi_i)}[\nu^\intercal \phi_{h+1}(s_{h+1}, a_{h+1})\phi_{h+1}(s_{h+1}, a_{h+1})^\intercal \nu] \\
&\geq \frac{\epsilon_h \prod_{h'=1}^{h-1}(1-\epsilon_{h'})}{Ad} \sum_{i=1}^d (\phi_h^{\pi_i})^\intercal w_{h+1}^\nu. \qquad (5.4)
\end{aligned}$$

Since π_i is a $b_1/2$-optimal policy for MDP $M_{i,h}$ and again by Assumption 5.2, we have

$$\theta_{h,M_{i,h}}^\intercal \phi_h^{\pi_i} \geq \frac{1}{2} \max_{\pi\in\Pi} \theta_{h,M_{i,h}}^\intercal \phi_h^\pi. \qquad (5.5)$$

For any vector $\nu \in \mathbb{R}^d$, let $[\nu]_i$ be the i-th dimension of the vector. Note that $\theta_{h,M_{i,h}} = e_i$, (5.5) indicates $[\phi_h^{\pi_i}]_i \geq \frac{1}{2}\max_\pi [\phi_h^\pi]_i$.

Combining the inequality (5.5) with (5.4), we have

$$\begin{aligned}
\nu^\intercal \Phi_{h+1}^{\tilde{\pi}} \nu &= \frac{\epsilon_h \prod_{h'=1}^{h-1}(1-\epsilon_{h'})}{dA} \sum_{i=1}^{d}\sum_{j=1}^{d} [\phi_h^{\pi_i}]_j [w_{h+1}^{\nu}]_j \\
&\geq \frac{\epsilon_h \prod_{h'=1}^{h-1}(1-\epsilon_{h'})}{dA} \sum_{i=1}^{d} [\phi_h^{\pi_i}]_i [w_{h+1}^{\nu}]_i \\
&\geq \frac{\epsilon_h \prod_{h'=1}^{h-1}(1-\epsilon_{h'})}{dA} \sum_{i=1}^{d} \max_{\pi} [\phi_h^{\pi}]_i [w_{h+1}^{\nu}]_i \\
&\geq \frac{\epsilon_h \prod_{h'=1}^{h-1}(1-\epsilon_{h'})}{2dA} \max_{\pi} (\phi_h^{\pi})^\intercal w_{h+1}^{\nu} \\
&\geq \frac{\epsilon_h \prod_{h'=1}^{h-1}(1-\epsilon_{h'}) b_1^2}{2dA}
\end{aligned}$$

□

Proof of Theorem 5.2 **Theorem 5.2.** *Consider $\mathcal{M}$ defined in Definition 5.7. With Assumption 5.2 holding and $\beta \leq b_1/2$, for any $f \in \mathcal{F}_\beta$, we have lower bound $\alpha(f, \mathcal{F}, \mathcal{M}) \geq \sqrt{e\beta^2 b_1^2/(2A|\mathcal{M}|H)}$ by setting $\epsilon_h = 1/h$.*

Proof. Let h' be the smallest h, such that there exists $M_{i,h}$, $\pi^{f_{M_{i,h}}}$ is β-suboptimal. Let (i', h') be the index of the MDP that has the suboptimal policy. We show that $M_{i',h'}$ has lower bounded myopic exploration gap.

By definition, f is β-optimal for any MDP $M_{i,h'-1}$. By Lemma 5.2, letting $\tilde{\pi} = \mathrm{expl}(f, \epsilon_{h'})$, we have

$$\nu^\intercal \Phi_{h'+1}^{\tilde{\pi}} \nu \geq \frac{\epsilon_{h'} \prod_{h''=1}^{h'-1}(1-\epsilon_{h''}) b_1^2}{2A|\mathcal{M}|}.$$

By Lemma 5.1, we have that the optimal value function f^* for MDP $M_{i',h'}$ satisfies that for any f'

$$\mathbb{E}_{\pi^{f^*}}^{M} \left[\left(\mathcal{E}_h^2 f' \right)(s_h, a_h) \right] \leq \frac{2A|\mathcal{M}|}{\epsilon_{h'} \prod_{h''=1}^{h'-1}(1-\epsilon_{h''}) b_1^2} \mathbb{E}_{\pi}^{M} \left[\left(\mathcal{E}_h^2 f' \right)(s_h, a_h) \right].$$

Thus, by Definition 5.3, the myopic exploration gap for f is lower bounded by

$$\beta \frac{1}{\sqrt{c}} = \beta \sqrt{\frac{\epsilon_{h'} \prod_{h''=1}^{h'-1}(1-\epsilon_{h''}) b_1^2}{2A|\mathcal{M}|}} \geq \sqrt{\frac{\beta^2 b_1^2}{2A|\mathcal{M}|eH}},$$

if we choose $\epsilon_h = 1/(h+1)$. □

Linear Quadratic Regulator To demonstrate the generalizability of the proposed framework, we introduce another interesting setting called Linear Quadratic Regulator (LQR). LQR takes continuous state space $\mathbb{R}^{d_s}$ and action space $\mathbb{R}^{d_a}$. In the LQR system, the state $s_h \in \mathbb{R}^{d_s}$ evolves according to the following transition: $s_{h+1} = A_h s_h + B_h a_h$, where $A_h \in \mathbb{R}^{d_s \times d_s}$, $B_h \in \mathbb{R}^{d_s \times d_a}$ are unknown system matrices that are shared by all the MDPs. We denote $s_h = (s_h, a_h)$ as the state-action vector. The reward function for an MDP M takes a known quadratic form $r_{h,M}(s,a) = s^\intercal R^s_{h,M} s + a^\intercal R^a_{h,M} a$, where $R^s_{h,M} \in \mathbb{R}^{d_s \times d_s}$ and $R^a_{h,M} \in \mathbb{R}^{d_a \times d_a}$ [5].

Note that LQR is more commonly studied for the infinite-horizon setting, where stabilizing the system is a primary concern of the problem. We consider the finite-horizon setting, which alleviates the difficulties on stabilization so that we can focus our discussion on exploration. Finite-horizon LQR also allows us to remain consistent notations with the rest of the paper. A related work (Simchowitz and Foster, 2020) states that naive exploration is optimal for online LQR with a condition that the system injects a random noise onto the state observation with a full rank covariance matrix $\Sigma \succ 0$. Though this is a common assumption in LQR literature, one may notice that the analog of this assumption in the tabular MDP is that any state and action pair has a non-zero probability of visiting any other state, which makes naive exploration sample-efficient trivially. In this section, we consider a deterministic system, where naive exploration does not perform well in general.

Properties of LQR. It can be shown that the optimal actions are linear transformations of the current state (Farjadnasab and Babazadeh, 2022; Li et al., 2022).

The optimal linear response is characterized by the discrete-time Riccati equation given by

$$P_{h,M} = A_h^\intercal (P_{h+1,M} - P_{h+1,M} \bar{R}^{-1}_{h+1,M} B_h^\intercal P_{h+1,M}) A_h + R^s_{h,M},$$

where $\bar{R}_{h+1,M} = R^a_h + B_h^\intercal P_{h+1,M} A_h$ and $P_{H+1} = \mathbf{0}$. Assume that the solution for the above equation is $\{P^*_{h,M}\}_{h \in [H+1]}$, then the optimal control actions takes the form

$$a_h = F^*_{h,M} s_h, \text{ where } F^*_{h,M} = -(R^s_{h,M} + B_h^\intercal P^*_{h,M} B_h)^{-1} B^\intercal P^*_{h,M} A_h.$$

and optimal value function takes the quadratic form: $V^*_{h,M}(s) = s^\intercal P^*_{h,M} s$ and

$$Q^*_{h,M}(x) = x^\intercal \begin{bmatrix} R^s_{h,M} + A_h^\intercal P^*_{h+1,M} A_h & A_h^\intercal P^*_{h+1,M} B_h \\ B_h^\intercal P^*_{h+1,M} A_h & R^a_{h,M} + B_h^\intercal P^*_{h+1,M} B_h \end{bmatrix} x.$$

[5] Note that LQR system often consider a cost function and the goal of the agent is to minimize the cumulative cost with $R^s_{h,M}$ being semi-positive definite. We formulation this as a reward maximization problem for consistency. Thus, we consider $R^s_{h,M} \prec \mathbf{0}$

This observation allows us to consider the following function approximation

$$\mathcal{F} = (\mathcal{F}_h)_{h\in[H+1]}, \text{ where each } \mathcal{F}_h = \{x \mapsto x^\intercal G_h x : G_h \in \mathbb{R}^{(d_s+d_a)\times(d_s+d_a)}\}.$$

The quadratic function class satisfies Bellman realiazability and completeness assumptions.

Definition 5.8 (Diverse LQR Task Set)**.** *Inspired by the task construction in linear MDP case, we construct the diverse LQR set by $\mathcal{M} = \{M_{i,h}\}_{i\in[d_s],h\in[H]}$ such that these MDPs all share the same transition matrices A_h and B_h and each $M_{i,h}$ has $R^s_{h',M_{i,h}} = \mathbb{1}[h' = h]e_i e_i^\intercal$ and $R^a_{h',M_{i,h}} = -I$.*

Assumption 5.3 (Regularity parameters)**.** *Given the task set in Definition 5.8, we define some constants that appears on our bound. Let $\pi^*_{i,h}$ be the optimal policy for $M_{i,h}$. Let*

$$b_4 = \max_{i,h} \mathbb{E}_{\pi^*_{i,h}} \max_{h'} \|s_{h'}\|_2, \text{ and } b_5 = \max_{i,h} \mathbb{E}_{\pi^*_{i,h}} \max_{h'} \|a_{h'}\|_2.$$

These regularity assumption is reasonable because the optimal actions are linear transformations of states and we consider a finite-horizon MDP, with F^*_h having upper bounded eigenvalues.

Similarly to the linear MDP case, we assume that the system satisfies some visibility assumption.

Assumption 5.4 (Coverage Assumption)**.** *For any $\nu \in \mathbb{R}^{d_s-1}$, there exists a policy π with $\|a_h\|_2 \le 1$ such that*

$$\max_\pi \mathbb{E}_\pi[s_h^\intercal \nu] \ge b_3, \text{ for } b_3 > 1.$$

Theorem 5.3. *Given Assumption 5.3, 5.4 and the diverse LQR task set in Definition 5.8, we have that for any $f \in \mathcal{F}_\beta$ with $\beta \le (b_3^2-1)b_5^2/2$,*

$$\alpha(f, \mathcal{F}, \mathcal{M}) = \Omega\left(\frac{\max\{b_4^2, b_5^2\}b_4^2}{d_s H \max\{(b_3^2-1)b_5^2, d_s\sigma^2\}(b_3^2-1)b_5^2}\right).$$

Proof of Theorem 5.3

Lemma 5.9. *Assume that we have a set of policies $\{\pi_i\}_{i\in[d]}$ such that the i-th policy is a $(b_3^2-1)b_5^2/2$-optimal policy for LQR with $R^s_{h,i} = e_i e_i^\intercal$ and $R^a_{h,i} = -I$. Let $\tilde{\pi} = Mixture(\text{expl}\{\pi_i\})$. Then we have*

$$\lambda_{min}(\mathbb{E}_{\tilde{\pi}} s_{h+1} s_{h+1}^\intercal) \ge \frac{d_s \max\{\underline{\lambda}, d\sigma^2\}}{2\max\{b_4^2, b_5^2\}\prod_{h'=1}^{h-1}(1-\epsilon_{h'})\epsilon_h}\underline{\lambda},$$

with $\underline{\lambda} = (b_3^2-1)b_5^2$.

Proof. We directly analyze the state covariance matrix at the step $h+1$. Let $\eta_h \sim \mathcal{N}(0, \sigma^2)$

$$\begin{aligned}
\mathbb{E}_{\tilde{\pi}} s_{h+1} s_{h+1}^\intercal &= \mathbb{E}_{\tilde{\pi}} (A_h s_h + B_h a_h)(A_h s_h + B_h a_h)^\intercal \\
&\succeq \frac{\prod_{h'=1}^{h-1}(1-\epsilon_{h'})\epsilon_h}{d_s} \sum_{i=1}^{d_s} (\mathbb{E}_{\pi_i}(A_h s_h + B_h \eta_h)(A_h s_h + B_h \eta_h)^\intercal) \\
&= \frac{\prod_{h'=1}^{h-1}(1-\epsilon_{h'})\epsilon_h}{d_s} \sum_{i=1}^{d_s} (A_h \mathbb{E}_{\pi_i} s_h s_h^\intercal A_h^\intercal + B_h \mathbb{E} \eta_h \eta_h^\intercal B_h^\intercal) \qquad (5.6)
\end{aligned}$$

To proceed, we show that $\sum_{i=1}^{d_s} \mathbb{E}_{\pi_i} s_h s_h^\intercal \succeq \underline{\lambda} I$.

From Assumption 5.4, we have $\mathbb{E}_{\pi_i^*}[s_h^\intercal e_i e_i^\intercal s_h - a_h a_h^\intercal] \succeq b_3^2 b_5^2 - b_5^2$, and by the fact that π_i is a $(b_3^2-1)b_5^2/2$-optimal policy, we have

$$\mathbb{E}_{\pi_i^*}[s_h^\intercal e_i e_i^\intercal s_h - a_h a_h^\intercal] \succeq (b_3^2 b_5^2 - b_5^2)/2.$$

Since $\mathbb{E}_{\pi_i} a_h a_h^\intercal \succeq 0$, we have $\mathbb{E}_{\pi_i}[s_h^\intercal e_i e_i^\intercal s_h] \succeq (b_3^2-1)b_5^2/2$. Therefore, $\sum_{i=1}^{d_s} \mathbb{E}_{\pi_i} s_h s_h^\intercal \succeq \underline{\lambda} I$ with $\underline{\lambda} = (b_3^2-1)b_5^2/2$.

Combined with (5.6), we have

$$\mathbb{E}_{\tilde{\pi}} s_{h+1} s_{h+1}^\intercal \succeq \frac{\prod_{h'=1}^{h-1}(1-\epsilon_{h'})\epsilon_h}{d_s} \left(\underline{\lambda} A_h A_h^\intercal + d_s \sigma^2 B_h B_h^\intercal\right).$$

Apply Assumption 5.4 again, for each $\nu_i = e_i, i = 1, \ldots, d_s$, there exists some policy π_i' with $\|a_h\|_2 \le b_5$, such that $\nu_i^\intercal \mathbb{E}_{\pi_i'} s_{h+1} s_{h+1}^\intercal \nu_i \ge b_3^2 b_5^2 - b_5^2$. Therefore, we have that $\sum_{i=1}^{d_s} \mathbb{E}_{\pi_i'} s_{h+1} s_{h+1}^\intercal \succeq (b_3^2-1)b_5^2 I$

The proof is completed by

$$\begin{aligned}
\sum_{i=1}^{d_s} \mathbb{E}_{\pi_i'} s_{h+1} s_{h+1}^\intercal &\preceq 2 \sum_{i=1}^{d_s} \left(A_h \mathbb{E}_{\pi_i'} s_h s_h^\intercal A_h^\intercal + B_h \mathbb{E}_{\pi_i'} a_h a_h^\intercal B_h^\intercal\right) \\
&\preceq 2 \sum_{i=1}^{d_s} \left(b_4^2 A_h A_h^\intercal + b_5^2 B_h \mathbb{E}_{\pi_i'} B_h^\intercal\right) \\
&\preceq \frac{2\max\{b_4^2, b_5^2\}}{\max\{\underline{\lambda}, d\sigma^2\}} \frac{\prod_{h'=1}^{h-1}(1-\epsilon_{h'})\epsilon_h}{d_s} \mathbb{E}_{\tilde{\pi}} s_{h+1} s_{h+1}^\intercal.
\end{aligned}$$

To complete the proof of Theorem 5.3, we combine Lemma 5.10 and Lemma 5.9.

□

Supporting lemmas Lemma 5.10 shows that having a full rank covariance matrix for the state s_h is a sufficient condition for bounded occupancy measure.

Lemma 5.10. *Let $\mathcal{F}$ be the function class described above. For any policy π and h such that*

$$\lambda_{min}(\mathbb{E}_\pi[s_h s_h^\intercal]) \geq \underline{\lambda},$$

we have for any π' such that $\max_h \|s_h\|_2 \leq b_4$, and for any $f' \in \mathcal{F}$,

$$\mathbb{E}_{\pi'}^M \left[\left(\mathcal{E}_h^2 f' \right)(s_h, a_h) \right] \leq \frac{b_4^2}{\underline{\lambda}^2} \mathbb{E}_\pi^M \left[\left(\mathcal{E}_h^2 f' \right)(s_h, a_h) \right].$$

Proof. Lemma 5.11 shows that the Bellman error also takes a quadratic form of s_h.

Lemma 5.11. *For any $f \in \mathcal{F}$, there exists some matrix $\tilde{G}_h$ such that $(\mathcal{E}_h f)(x) = x^\intercal \tilde{G}_h x$.*

To complete the proof of Lemma 5.10, let $w_h = s_h \otimes s_h$ be the Kronecker product between s_h and itself. By Lemma 5.11, we can write $(\mathcal{E}_h f)(s_h) = \mathrm{Vec}(\tilde{G}_h)^\intercal w_h$. Again, this is an analog of the linear form we had for thee linear MDP case. Thus, we can write $(\mathcal{E}_h^2 f)(s_h) = \mathrm{Vec}(\tilde{G}_h)^\intercal w_h w_h^\intercal \mathrm{Vec}(\tilde{G}_h)$.

By Lemma 5.12 and the fact that $\mathbb{E}_\pi(w_h w_h^\intercal) = \mathbb{E}_\pi(s_h s_h^\intercal) \otimes \mathbb{E}_\pi(s_h s_h^\intercal)$, we have $\lambda_{\min}(\mathbb{E}_\pi w_h w_h^\intercal) \geq \underline{\lambda}^2$.

For any other policy π', and using the fact that $\|w_h\| \leq b_4^2$, we have

$$\mathbb{E}_{\pi'}(\mathcal{E}_h^2 f)(s_h) = \mathbb{E}_{\pi'}[\mathrm{Vec}(\tilde{G}_h)^\intercal w_h w_h^\intercal \mathrm{Vec}(\tilde{G}_h)] \leq \frac{b_4^2}{\underline{\lambda}^2} \mathbb{E}_\pi[\mathrm{Vec}(\tilde{G}_h)^\intercal w_h w_h^\intercal \mathrm{Vec}(\tilde{G}_h)] \leq \frac{b_4^2}{\underline{\lambda}^2} \mathbb{E}_\pi(\mathcal{E}_h^2 f)(s_h).$$

□

Lemma 5.11. *For any $f \in \mathcal{F}$, there exists some matrix $\tilde{G}_h$ such that $(\mathcal{E}_h f)(x) = x^\intercal \tilde{G}_h x$.*

Proof. The Bellman error of the LQR can be written as

$$(\mathcal{E}_h f)(x) = \left(x^\intercal G_h x - s^\intercal R_h^s s - a^\intercal R_h^a a - \max_{a' \in \mathbb{R}^{d_a}} [(A_h s + B_h a)^\intercal, a'^\intercal] G_{h+1} \begin{bmatrix} A_h s + B_h a \\ a' \end{bmatrix} \right)$$

Note that the optimal a' can be written as some linear transformation of x. Thus we can write

$$\max_{a' \in \mathbb{R}^{d_a}} [(A_h s + B_h a)^\intercal, a'^\intercal] G_{h+1} \begin{bmatrix} A_h s + B_h a \\ a' \end{bmatrix} = x^\intercal G' x.$$

The whole equation can be written as a quadratic form as well. □

Lemma 5.12. *Let $A \in \mathbb{R}^{d_1 \times d_1}$ have eigenvalues $\{\lambda_i\}_{i \in [d]}$ and $B \in \mathbb{R}^{d_2 \times d_2}$ have eigenvalues $\{\mu_i\}_{i \in [d]}$. The eigenvalues of $A \otimes B$ are $\{\lambda_i \mu_j\}_{i \in [d_1], j \in d_2}$.*

5.D Relaxing Visibility Assumption

Tabular Case A simple but interesting case to study is the tabular case, where the value function class is the class of any bounded functions, i.e. $\mathcal{F}_h = \{f : \mathcal{S} \times \mathcal{A} \mapsto [0,1]\}$. A commonly studied family of multitask RL is the MDPs that share the same transition probability, while they have different reward functions, this problem is studied in a related literature called reward-free exploration (Jin et al., 2020a; Wang et al., 2020; Chen et al., 2022a). Specifically, Jin et al. (2020a) propose to learn $S \times H$ sparse reward MDPs separately and generates an offline dataset, with which one can learn a near-optimal policy for any potential reward function. With a similar flavor, we show that any superset of the $S \times H$ sparse reward MDPs has low myopic exploration gap. Though the tabular case is a special case of the linear MDP case, the lower bound we derive for the tabular case is slightly different, which we show in the following section.

We first give a formal definition on the sparse reward MDP.

Definition 5.9 (Sparse Reward MDPs)**.** *Let $\mathcal{M}$ be a set of MDPs sharing the same transition probabilities. We say $\mathcal{M}$ contains all the sparse reward MDPs if for each $s, h \in \mathcal{S} \times [H]$, there exists some MDP $M_{s,h} \in \mathcal{M}$, such that $R_{h',M_{s,h}}(s',a') = \mathbb{1}(s = s', h = h')$ for all s', a', h'.*

To show a lower bound on the myopic exploration gap, we make a further assumption on the occupancy measure $\mu_h^\pi(s,a) \coloneqq \mathbb{Pr}_\pi(s_h = s, a_h = a)$, the probability of visiting s, a at the step h by running policy π.

Assumption 5.5 (Lower bound on the largest achievable occupancy measure)**.** *For all $s, h \in \mathcal{S} \times [H]$, we assume that $\max_\pi \mu_h^\pi(s) \geq b_1$ for some constant b or $\max_\pi \mu_h^\pi(s) = 0$.*

Assumption 5.5 guarantees that any β-optimal policy (with $\beta < b_1$) is not a vacuous policy and it provides a lower bound on the corresponding occupancy measure. We will discuss later in Appendix 5.D on how to remove this assumption with an extra $S \times H$ factor on the sample complexity bound.

Proposition 5.3. *Consider a set of sparse reward MDP as in Definition 5.9. Assume Assumption 5.5 is true. For any $\beta \leq b_1/2$ and $f \in \mathcal{F}_\beta$, we have $\alpha(f, \mathcal{F}, \mathcal{M}) \geq \bar{\alpha}$ for some constant $\bar{\alpha} = \sqrt{\beta^2/(2e|\mathcal{M}|AH)}$ by choosing $\epsilon_h = 1/h$.*

Proof. We prove this lemma in a layered manner. Let h' be the minimum step such that there exists some $M_{s,h'}$ is β-suboptimal. By definition, in the layer $h'-1$, all the MDPs are β-suboptimal, in which case $\boldsymbol{\pi}_{M_{s,h'-1}}$ visits $(s, h'-1)$ with a probability at least $b/2$. Now we show that the optimal policy $\pi^*_{M_{s,h'}}$ of a suboptimal MDP $M_{s,h'}$ has lower bounded occupancy ratio.

For a more concise notation, we let $M' = M_{s,h'}$. Note that

$$\begin{aligned}
\mu_{h'}^{\pi^*_{M'}}(s) &= \sum_{s' \in \mathcal{S}} \mu_{h'-1}^{\pi^*_{M'}}(s') P_{h'-1}(s \mid s', \pi^*_{M'}(s')) \\
&\leq \sum_{s' \in \mathcal{S}} \max_{\pi \in \Pi} \mu_{h'-1}^{\pi}(s') P_{h'-1}(s \mid s', \pi^*_{M'}(s')) \\
&\quad \text{(By the fact that } \mu_{h'-1}^{\boldsymbol{\pi}_{M_{s',h'-1}}}(s') \text{ is } \beta\text{-optimal policy of } M_{s',h'-1}) \\
&\leq \sum_{s' \in \mathcal{S}} \frac{b_1}{b_1 - \beta} \mu_{h'-1}^{\boldsymbol{\pi}_{M_{s',h'-1}}}(s') P_{h'-1}(s \mid s', \pi^*_{M'}(s')) \\
&\leq \sum_{s' \in \mathcal{S}} \frac{b_1 |\mathcal{M}| A}{(b_1 - \beta)(1-\epsilon)^{h'-1}\epsilon} \mu_{h'-1}^{\mathrm{expl}(\boldsymbol{\pi})}(s') P_{h'-1}(s \mid s', \mathrm{expl}(\boldsymbol{\pi})(s')) \\
&= \frac{b_1 |\mathcal{M}| A}{(b_1 - \beta)(1-\epsilon)^{h'-1}\epsilon} \mu_{h'}^{\mathrm{expl}(\boldsymbol{\pi})}(s)
\end{aligned}$$

The occupancy measure ratio can be upper bounded by $c = \frac{b_1 |\mathcal{M}| A}{(b_1 - \beta)(1-\epsilon)^{h'-1}\epsilon}$. Then the myopic exploration gap can be lower bounded by

$$\frac{\beta}{\sqrt{c}} = \sqrt{\frac{(b_1 - \beta)\beta^2 (1-\epsilon)^{h'-1}\epsilon}{b_1 |\mathcal{M}| A}} \geq \sqrt{\frac{\beta^2 (1-\epsilon)^{h'-1}\epsilon}{2|\mathcal{M}| A}}.$$

To proceed, we choose $\epsilon_h = 1/h$, which leads to $(1-\epsilon_h)^{h-1}\epsilon \geq 1/(eH)$. □

Plugging this into Theorem 5.1, we achieve a sample complexity bound of $\mathcal{O}(S^2 A H^5 / \beta^2)$, with $|\mathcal{M}| = SH$. This is not a near-optimal bound for reward-free exploration (a fair comparison in our setup). This is because the sample complexity bound in Theorem 5.1 is not tailored for tabular case.

Removing coverage assumption Though Assumption 5.2 and Assumption 5.4 are relatively common in the literature, we have not seen an any like Assumption 5.5. In fact, Assumption 5.5 is not a necessary condition for sample-efficient myopic exploration as we will discuss in this section. The main technical invention is to construct a mirror transition probability that satisfies the conditions in Assumption 5.5. However, we will see that a inevitable price of an extra SH factor has to be paid.

To illustrate the obstacle of removing Assumption 5.5, recall that the proof of Proposition 5.3 relies on the fact that all β-optimal policies guarantee a non-zero probability of visiting the state corresponding to their sparse reward with $\beta < b_1/2$. Without Assumption 5.5, a β-optimal policy can be an arbitrary policy. At the step h, we have at most S such MDPs, which may accumulate an irreducible error of $S\beta$, which means that at the round $h+1$,

we can only guarantee $S\beta$-optimal policies. An naive adaptation will require us to set the accuracy $\beta' = \beta/S^H$ in order to guarantee a β error in the last step. The following discussion reveals that the error does not accumulate in a multiplicative way.

Mirror MDP construction. It is helpful to consider a mirror transition probability modified from our original transition probability. We denote the original transition probability by $P = \{P_h\}_{h\in[H]}$. Consider a new MDP with transition $P' = \{P'_h\}_{h\in[H]}$ and state space $\mathcal{S}' = \mathcal{S} \cup \{s_0\}$, where s_0 is a dummy state. We initialize P' such that

$$P'_h(s' \mid s, a) = P_h(s' \mid s, a) \text{ for all } s', s, a, h, \text{ where } s', s \neq s_0, \text{ and } P'_h(s_0 \mid s_0, \cdot) = 1 \tag{5.7}$$

Starting from $h = 1$, we update P'_h by a forward induction according to Algorithm 7. The design principle is to direct the probability mass of visiting $(s, h+1)$ to $(s_0, h+1)$, whenever the maximal probability of visiting $(s, h+1)$ is less than β.

Algorithm 7 Creating Mirror Transitions

Input: Original Transition P, threshold $\beta > 0$.
Initialize P' according to (5.7)
for $h = 1, 2, \ldots, H-1$ **do**
  for each $s \in \mathcal{S}$ such that $\max_\pi \mu'^{\pi}_{h+1}(s) \leq \beta$ **do**
    $P'_h(s_0 \mid \tilde{s}, \tilde{a}) \leftarrow P'_h(s_0 \mid \tilde{s}, \tilde{a}) + P'_h(s \mid \tilde{s}, \tilde{a})$ for each $\tilde{s}$, $\tilde{a}$.
    $P'_h(s \mid \tilde{s}, \tilde{a}) \leftarrow 0$ for each $\tilde{s}$, $\tilde{a}$.
  end for
end for
Return P'

By definition of P', we have two nice properties.

Proposition 5.4. *For any $h \in [H]$, $s \in \mathcal{S}$, we have $\max_\pi \mu'^{\pi}_h(s) = 0$ or $\max_\pi \mu'^{\pi}_h(s) > \beta$.*

Thus, P' nicely satisfies our Assumption 5.5. We also have that P' is not significantly different from P.

Proposition 5.5. *For any policy π, $\mu'^{\pi}_h(s) \geq \mu^{\pi}_h(s) - HS\beta$. Further more, any $(SH+1)\beta$-suboptimal policy for P is at least β-suboptimal for P' with respect to the same reward.*

Proof. We simply observe that $\max_\pi \mu'^{\pi}_h(s_0) \leq (h-1)S\beta$. This is true since at any round, we have at most S states with $\max_\pi \mu'^{\pi}(s) \leq \beta$, all the probability that goes to s will be deviated to s_0. Therefore, for any π

$$\mu'^{\pi}_{h+1}(s_0) \leq \mu'^{\pi}_h(s_0) + S\beta.$$

□

Therefore, any $(SH+1)\beta$-suboptimal policy for P has the myopic exploration gap of β-suboptimal policy for P'.

Theorem 5.4. *Consider a set of sparse reward MDP as in Definition 5.9. For any $\beta \in (0,1)$ and $f \in \mathcal{F}_\beta$, we have $\alpha(f, \mathcal{F}, \mathcal{M}) \geq \bar{\alpha}$ for some constant $\bar{\alpha} = \Omega(\sqrt{\beta^2/(|\mathcal{M}|AS^2H^3)})$ by choosing $\epsilon_h = 1/(h+1)$.*

5.E Connections to Diversity

Diversity has been an important consideration for the generalization performance of multitask learning. How to construct a diverse set, with which we can learn a model that generalizes to unseen task is studied in the literature of multitask supervised learning.

Tripuraneni et al. (2020); Xu and Tewari (2021) studied the importance of diversity in multitask representation learning. They assume that the response variable is generated through mean function $f_t \circ h$, where h is the representation function shared by different tasks and f_t is the task-specific prediction function of a task indexed by t. They showed that diverse tasks can learn the shared representation that generalizes to unseen downstream tasks. More specifically, if $f_t \in \mathcal{F}$ is a discrete set, a diverse set needs to include all possible elements in $\mathcal{F}$. If $\mathcal{F}$ is the set of all bounded linear functions, we need d tasks to achieve a diverse set. Note that these results align with the results presented in the previous section. *Could there be any connection between the diversity in multitask representation learning and the efficient myopic exploration?*

Xu and Tewari (2021) showed that Eluder dimension is a measure for the hardness of being diverse. Here we introduce a generalized version called distributional Eluder dimension (Jin et al., 2021a).

Definition 5.10 (ε-independence between distributions)**.** *Let $\mathcal{G}$ be a class of functions defined on a space $\mathcal{X}$, and $\nu, \mu_1, \ldots, \mu_n$ be probability measures over $\mathcal{X}$. We say ν is ε-independent of $\{\mu_1, \mu_2, \ldots, \mu_n\}$ with respect to $\mathcal{G}$ if there exists $g \in \mathcal{G}$ such that $\sqrt{\sum_{i=1}^n (\mathbb{E}_{\mu_i}[g])^2} \leq \varepsilon$, but $|\mathbb{E}_\nu[g]| > \varepsilon$*

Definition 5.11 (Distributional Eluder (DE) dimension)**.** *. Let $\mathcal{G}$ be a function class defined on $\mathcal{X}$, and Π be a family of probability measures over $\mathcal{X}$. The distributional Eluder dimension $\dim_{\mathrm{DE}}(\mathcal{G}, \Pi, \varepsilon)$ is the length of the longest sequence $\{\rho_1, \ldots, \rho_n\} \subset \Pi$ such that there exists $\varepsilon' \geq \varepsilon$ where ρ_i is ε'-independent of $\{\rho_1, \ldots, \rho_{i-1}\}$ for all $i \in [n]$.*

Definition 5.12 (Bellman Eluder (BE) dimension (Jin et al., 2021)). *Let $\mathcal{E}_h\mathcal{F}$ be the set of Bellman residuals induced by $\mathcal{F}$ at step h, and $\Pi = \{\Pi_h\}_{h=1}^H$ be a collection of H probability measure families over $\mathcal{X} \times \mathcal{A}$. The ε-Bellman Eluder dimension of $\mathcal{F}$ with respect to Π is defined as*

$$\dim_{\mathrm{BE}}(\mathcal{F}, \Pi, \varepsilon) := \max_{h\in[H]} \dim_{\mathrm{DE}}\left(\mathcal{E}_h\mathcal{F}, \Pi, \varepsilon\right)$$

Constructing a diverse set. For each $h \in [H]$, consider a sequence of functions $f_1, \ldots, f_d \in \mathcal{F}$, such that the induced policy $(\pi^{f_i})_{i\in[d]}$ generates probability measures $(\mu_{h+1}^{f_i})_{i\in[d]}$ at the step $h+1$. Let $(\mu_{h+1}^{f_i})_{i\in[d]}$ be ϵ-independence w.r.t the function class $\mathcal{E}_h\mathcal{F}$ between their predecessors. By the definition of BE dimension, we can only find at most $\dim_{\mathrm{DE}}\left(\mathcal{E}_h\mathcal{F}, \Pi, \varepsilon\right)$ of these functions. Then conditions in Definition 5.3 is satisfied with $c = 1/(dH)$.

Revisiting linear MDPs. The task set construction in 5.7 seems to be quite restricted as we require a set of standard basis. One might conjecture that a task set $M_{i,h}$ with full rank $[\theta_{1,h}, \ldots, \theta_{d,h}]$ will suffice. From what we discussed in the general case, we will need the occupancy measure generated by the optimal policies for these MDPs to be ϵ-independent and any other distribution is ϵ-dependent. This is generally not true even if the reward parameters are full rank. To see this, let us consider a tabular MDP case with two states $\{1, 2\}$, where at the step h, we have two tasks M_1, M_2, with $R_{h,M_1}(s, a) = \mathbb{1}[s = 1]$ and $R_{h,M_2}(s, a) = 0.51\mathbb{1}[s = 1] + 0.49\mathbb{1}[s = 2]$. This gives $\theta_{h,M_1} = [1, 0]$ and $\theta_{h,M_2} = [0.49, 0.51]$ as shown in Figure 5.3.

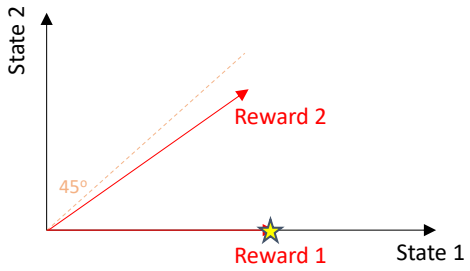


Figure 5.3: An illustration of why a full-rank set of reward parameters does not achieve diversity. The red arrows are two reward parameters and the star marks the generated state distributions of the optimal policies corresponding to the two rewards at the step h. Since both optimal policies only visit state 1, they may not provide a sufficient exploration for the next time step $h + 1$.

Construct the MDP such that the transition probability and action space any policy either visit state 1 or state 2 at the step h. Then the optimal policies for both tasks are the same

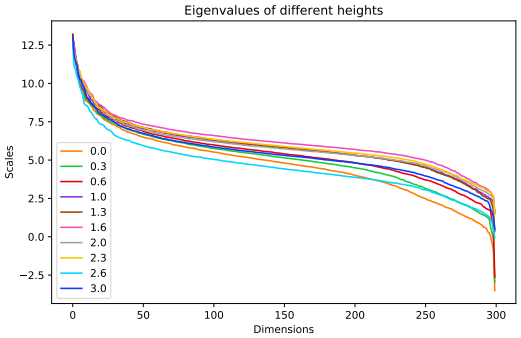
policy which visits state 1 with probability one, even if the reward parameters $[\theta_{h,M_1}, \theta_{h,M_2}]$ are full-rank.

5.F Experiment Details

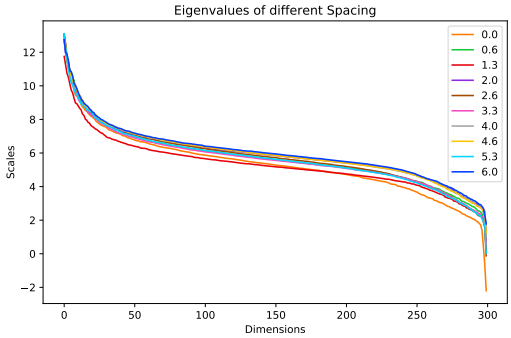
Extra Training Details By choosing an environment denoted by $M_{p,q}$, we randomly generate a p' and q' from $\mathcal{N}(p, 0.1)$, $\mathcal{N}(q, 0.1)$ to increase the robustness of the training for all settings. This is also the default setup in Romac et al. (2021).

Table 5.1: Training on different environment parameters. Each row represents a training scenario, where the first two columns are the range of sampled parameters. The mastered tasks are out of 121 evaluated tasks with the standard deviation calculated from ten independent runs.

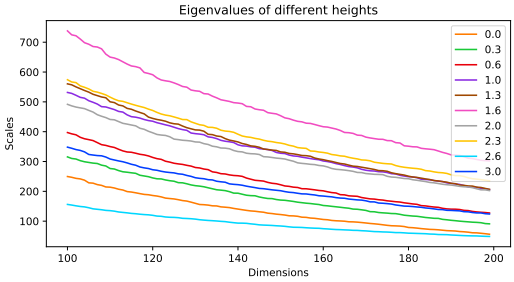
Obstacle spacing	Stump height	Mastered task
[2, 4]	[0.0, 0.3]	28.1 ± 6.1
[2, 4]	[1.3, 1.6]	41.6 ± 9.8
[2, 4]	[2.6, 3.0]	11.5 ± 10.9



(a) Controlling heights



(b) Controlling spacing



(c) Controlling heights on the original scale

Figure 5.4: **(b-c)** Log-scaled eigenvalues of sample covariance matrices of the trained embeddings generated by the near optimal policies for different environments.

CHAPTER 6

Summary and Future Work

In this thesis, we study the theoretical benefits of multitask learning in two major setups: supervised learning and reinforcement learning. We emphasize that task diversity plays an important role in all the settings discussed in this thesis. For supervised learning, we study a multitask representation learning, where a shared representation function is learned from source tasks and we show that a diversity condition on the source tasks guarantees a worst-case generalization performance on a downstream task. We further ask if we are given a potentially large dataset, whether a high diversity can be achieved when the algorithm is allowed to adaptively choose tasks. We study this problem under the framework called curriculum learning, where we show that diversity can be asymptotically achieved for linear multitask representation learning problem. To bridge supervised learning and reinforcement learning, we study a contextual bandit problem with funnel structure. We show that a high diversity leads to smaller discrepancies between tasks in different layers of the funnel structure, which allows us to show a smaller regret bound. In the reinforcement learning setting, we, for the first time, connects diversity to the exploration of RL. We show that a diverse task enables efficient myopic exploration, when policies are shared across different tasks. We provide a more concrete diversity condition for linear MDPs.

6.1 Future Work

There is still a significant gap between the practice of MTL and the current theoretical understanding. We highlight some promising directions.

Beyond ERM analysis. Most of current work on multitask representation learning are based on empirical risk minimizer (ERM), which assumes a global minimizer can be found. However, the optimization regime for even simple linear representation learning is non-convex. It worth exploring whether there is an efficient optimization algorithm that enjoys the similar theoretical guarantees ERM does.

Beyond representation learning. Though representation learning is heavily studies in both empirical and theoretical works, it does not cover all the interesting cases such as covariate shift and more general joint function class setting. For instance, in causal inference, the distribution may shift across tasks that is reflected on the causal graph. We may use a joint function class to model such changes, where the benefits of multitask learning is not well-undertood.

Diverse Reinforcement Learning. Our analysis are limited to problem with strong structural assumptions, i.e., linear MDPs. It is not clear what is a task set that achieves diversity in a general function approximation setting. Similar to the supervised learning setting, one may seek an adaptive task scheduler for multitask RL. How to adaptively learning a curriculum for Reinforcement Learning is still an open problem.

BIBLIOGRAPHY

Agarwal, A., Song, Y., Sun, W., Wang, K., Wang, M., and Zhang, X. (2022). Provable benefits of representational transfer in reinforcement learning. *arXiv preprint arXiv:2205.14571*.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al. (2019). Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*.

Andreas, J., Klein, D., and Levine, S. (2017). Modular multitask reinforcement learning with policy sketches. In *International Conference on Machine Learning*, pages 166–175. PMLR.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. (2017). Hindsight experience replay. *Advances in neural information processing systems*, 30.

Andrychowicz, O. M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al. (2020). Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20.

Aoki, R., Tung, F., and Oliveira, G. L. (2022). Heterogeneous multi-task learning with expert diversity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(6):3093–3102.

Argyriou, A., Evgeniou, T., and Pontil, M. (2006). Multi-task feature learning. *Advances in neural information processing systems*, 19.

Argyriou, A., Evgeniou, T., and Pontil, M. (2008). Convex multi-task feature learning. *Machine learning*, 73:243–272.

Auer, P., Jaksch, T., and Ortner, R. (2008). Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21.

Barry, T. E. (1987). The development of the hierarchy of effects: An historical perspective. *Current issues and Research in Advertising*, 10(1-2):251–295.

Bartlett, P. and Tewari, A. (2009). Regal: a regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Uncertainty in Artificial Intelligence: Proceedings of the 25th Conference*, pages 35–42. AUAI Press.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*, 79(1-2):151–175.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 19–26.

Bhatt, H. S., Rajkumar, A., and Roy, S. (2016). Multi-source iterative adaptation for cross-domain classification. In *IJCAI*, pages 3691–3697.

Bonilla, E. V., Chai, K. M., and Williams, C. (2008). Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*, pages 153–160.

Borrella, I., Caballero-Caballero, S., and Ponce-Cueto, E. (2019). Predict and intervene: Addressing the dropout problem in a mooc-based program. In *Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale*, pages 1–9.

Brunskill, E. and Li, L. (2013). Sample complexity of multi-task reinforcement learning. *arXiv preprint arXiv:1309.6821*.

Calandriello, D., Lazaric, A., and Restelli, M. (2014). Sparse multi-task reinforcement learning. *Advances in neural information processing systems*, 27.

Caruana, R. (1998). *Multitask learning*. Springer.

Caruana, R., Baluja, S., and Mitchell, T. (1995). Using the future to" sort out" the present: Rankprop and multitask learning for medical risk evaluation. *Advances in neural information processing systems*, 8.

Chane-Sane, E., Schmid, C., and Laptev, I. (2021). Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning*, pages 1430–1440. PMLR.

Chen, J., Modi, A., Krishnamurthy, A., Jiang, N., and Agarwal, A. (2022a). On the statistical efficiency of reward-free exploration in non-linear rl. *arXiv preprint arXiv:2206.10770*.

Chen, L., Jain, R., and Luo, H. (2022b). Improved no-regret algorithms for stochastic shortest path with linear mdp. In *International Conference on Machine Learning*, pages 3204–3245. PMLR.

Chen, X., Hu, J., Jin, C., Li, L., and Wang, L. (2021). Understanding domain randomization for sim-to-real transfer. *arXiv preprint arXiv:2110.03239*.

Cheng, Y., Feng, S., Yang, J., Zhang, H., and Liang, Y. (2022). Provable benefit of multitask representation learning in reinforcement learning. *arXiv preprint arXiv:2206.05900*.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214.

Cioba, A., Bromberg, M., Wang, Q., Niyogi, R., Batzolis, G., Shiu, D.-s., and Bernacchia, A. (2021). How to distribute data across tasks for meta-learning? *arXiv preprint arXiv:2103.08463*.

Clow, D. (2013). Moocs and the funnel of participation. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 185–189.

Cortes, C., Mohri, M., and Medina, A. M. (2019). Adaptation based on generalized discrepancy. *The Journal of Machine Learning Research*, 20(1):1–30.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016). Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865.

Dabney, W., Ostrovski, G., and Barreto, A. (2020). Temporally-extended \ϵ-greedy exploration. *arXiv preprint arXiv:2006.01782*.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2011). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42.

Dann, C., Lattimore, T., and Brunskill, E. (2017). Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 30.

Dann, C., Mansour, Y., Mohri, M., Sekhari, A., and Sridharan, K. (2022). Guarantees for epsilon-greedy reinforcement learning with function approximation. In *International Conference on Machine Learning*, pages 4666–4689. PMLR.

David, S. B., Lu, T., Luu, T., and Pál, D. (2010). Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136. JMLR Workshop and Conference Proceedings.

Deshmukh, A. A., Dogan, U., and Scott, C. (2017). Multi-task learning for contextual bandits. In *Advances in neural information processing systems*, pages 4848–4856.

Dong, K., Yang, J., and Ma, T. (2021). Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature. *arXiv preprint arXiv:2102.04168*.

Dragomir, S. S. and Gluscevic, V. (2000). Some inequalities for the Kullback-Leibler and x^2- distances in information theory and applications. *RGMIA research report collection*, 3(2):199–210.

Drugan, M. M. and Nowe, A. (2013). Designing multi-objective multi-armed bandits algorithms: A study. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. (2020). Few-shot learning via learning the representation, provably. In *International Conference on Learning Representations*.

Du, S. S., Koushik, J., Singh, A., and Póczos, B. (2017). Hypothesis transfer learning via transformation functions. In *Advances in Neural Information Processing Systems*, pages 574–584.

Evgeniou, T. and Pontil, M. (2004). Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117.

Farjadnasab, M. and Babazadeh, M. (2022). Model-free lqr design by q-function learning. *Automatica*, 137:110060.

Fazlyab, M., Robey, A., Hassani, H., Morari, M., and Pappas, G. (2019). Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in Neural Information Processing Systems*, 32:11427–11438.

Filippi, S., Cappe, O., Garivier, A., and Szepesvári, C. (2010). Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594.

Forman, E. M., Kerrigan, S. G., Butryn, M. L., Juarascio, A. S., Manasse, S. M., Ontañón, S., Dallal, D. H., Crochiere, R. J., and Moskow, D. (2019). Can the artificial intelligence technique of reinforcement learning use continuously-monitored digital data to optimize treatment for weight loss? *Journal of behavioral medicine*, 42:276–290.

Ghosh, S., Kim, R., Chhabria, P., Dwivedi, R., Klasjna, P., Liao, P., Zhang, K., and Murphy, S. (2023). Did we personalize? assessing personalization by an online reinforcement learning algorithm using resampling. *arXiv preprint arXiv:2304.05365*.

Golowich, N., Rakhlin, A., and Shamir, O. (2018). Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR.

Gong, C., Tao, D., Maybank, S. J., Liu, W., Kang, G., and Yang, J. (2016a). Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, 25(7):3249–3260.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. (2016b). Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848.

Graves, A., Bellemare, M. G., Menick, J., Munos, R., and Kavukcuoglu, K. (2017). Automated curriculum learning for neural networks. In *International Conference on Machine Learning*, pages 1311–1320. PMLR.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5.

Hessel, M., Soyer, H., Espeholt, L., Czarnecki, W., Schmitt, S., and van Hasselt, H. (2019). Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3796–3803.

Hoaglin, D. C. and Iglewicz, B. (1987). Fine-tuning some resistant rules for outlier labeling. *Journal of the American statistical Association*, 82(400):1147–1149.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Howard, J. Y. S. and Sheth, J. (1969). Jn (1969)" the theory of buyer behavior. *New York*.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. (2016). Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*.

Jaitly, N., Nguyen, P., Senior, A., and Vanhoucke, V. (2012). Application of pretrained deep neural networks to large vocabulary speech recognition. In *Proceedings of Interspeech*.

Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600.

Jiang, L., Meng, D., Zhao, Q., Shan, S., and Hauptmann, A. (2015). Self-paced curriculum learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Jiang, N. (2018). Pac reinforcement learning with an imperfect model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. (2020a). Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR.

Jin, C., Liu, Q., and Miryoosefi, S. (2021a). Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418.

Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. (2021b). On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29.

Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. (2021c). On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020b). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR.

Kalan, S. M. M., Fabian, Z., Avestimehr, A. S., and Soltanolkotabi, M. (2020). Minimax lower bounds for transfer learning with linear and one-hidden layer neural networks. *Advances in Neural Information Processing Systems*, 33:1959–1969.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., et al. (2018). Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR.

Konda, V. and Tsitsiklis, J. (1999). Actor-critic algorithms. *Advances in neural information processing systems*, 12.

Kpotufe, S. and Martinet, G. (2018). Marginal singularity, and the benefits of labels in covariate-shift. *arXiv preprint arXiv:1803.01833*.

Kumagai, W. (2016). Learning bound for parameter transfer learning. In *Advances in Neural Information Processing Systems*, pages 2721–2729.

Kuroki, S., Charoenphakdee, N., Bao, H., Honda, J., Sato, I., and Sugiyama, M. (2019). Unsupervised domain adaptation based on source-guided discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4122–4129.

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Lawrence, N. D. and Platt, J. C. (2004). Learning to learn with the informative vector machine. In *Proceedings of the twenty-first international conference on Machine learning*, page 65.

Lazaric, A., Brunskill, E., et al. (2013). Sequential transfer in multi-armed bandit with finite set of models. In *Advances in Neural Information Processing Systems*, pages 2220–2228.

Lazaric, A. and Ghavamzadeh, M. (2010). Bayesian multi-task reinforcement learning. In *ICML-27th international conference on machine learning*, pages 599–606. Omnipress.

Li, G., Kamath, P., Foster, D. J., and Srebro, N. (2021). Eluder dimension and generalized rank. *arXiv preprint arXiv:2104.06970*.

Li, H., Liao, X., and Carin, L. (2009). Multi-task reinforcement learning in partially observable stochastic environments. *Journal of Machine Learning Research*, 10(5).

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670.

Li, L., Chu, W., Langford, J., and Wang, X. (2011). Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306.

Li, M., Qin, J., Zheng, W. X., Wang, Y., and Kang, Y. (2022). Model-free design of stochastic lqr controller from a primal–dual optimization perspective. *Automatica*, 140:110253.

Liao, P., Greenewald, K., Klasnja, P., and Murphy, S. (2020). Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–22.

Lin, X., Baweja, H., Kantor, G., and Held, D. (2019). Adaptive auxiliary task weighting for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4772–4783.

Liu, C., Wang, Z., Sahoo, D., Fang, Y., Zhang, K., and Hoi, S. C. (2020). Adaptive task sampling for meta-learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 752–769. Springer.

Liu, M., Zhu, M., and Zhang, W. (2022). Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*.

Liu, Y. and Brunskill, E. (2018). When simple exploration is sample efficient: Identifying sufficient conditions for random exploration to yield pac rl algorithms. *arXiv preprint arXiv:1805.09045*.

Lozano, A. C. and Swirszcz, G. (2012). Multi-level lasso for sparse multi-task regression. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 595–602.

Lu, R., Huang, G., and Du, S. S. (2021). On the power of multitask representation learning in linear mdp. *arXiv preprint arXiv:2106.08053*.

Manikrao, U. S. and Prabhakar, T. (2005). Dynamic selection of web services with recommendation system. In *International conference on next generation web services practices (NWESP'05)*, pages 5–pp. IEEE.

Marmanis, D., Datcu, M., Esch, T., and Stilla, U. (2015). Deep learning earth observation classification using Imagenet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, 13(1):105–109.

Maurer, A., Pontil, M., and Romera-Paredes, B. (2016). The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 17(1):2853–2884.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Mulpuru, S. (2011). The purchase path of online buyers. *Forrester report*, 51:55.

Mulyar, A., Uzuner, O., and McInnes, B. (2021). Mt-clinical bert: scaling clinical information extraction with multitask learning. *Journal of the American Medical Informatics Association*, 28(10):2108–2115.

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P. (2020). Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21:1–50.

Nguyen, D.-K. and Okatani, T. (2019). Multi-task learning of hierarchical vision-language representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10492–10501.

Nguyen, T. D., Christoffel, M., and Sugiyama, M. (2016). Continuous target shift adaptation in supervised learning. In *Asian Conference on Machine Learning*, pages 285–300.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724.

Osband, I. and Roy, B. V. (2014). Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, pages 1466–1474.

Osband, I. and Van Roy, B. (2017). Why is posterior sampling better than optimism for reinforcement learning? In *International conference on machine learning*, pages 2701–2710. PMLR.

Osband, I., Van Roy, B., Russo, D. J., Wen, Z., et al. (2019). Deep exploration via randomized value functions. *J. Mach. Learn. Res.*, 20(124):1–62.

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2010). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. (2018). Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE.

Pentina, A., Sharmanska, V., and Lampert, C. H. (2015). Curriculum learning of multiple tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5492–5500.

Persello, C. and Bruzzone, L. (2012). Active learning for domain adaptation in the supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11):4468–4483.

Pescher, C., Reichhart, P., and Spann, M. (2014). Consumer decision-making processes in mobile viral marketing campaigns. *Journal of interactive marketing*, 28(1):43–54.

Portelas, R., Colas, C., Hofmann, K., and Oudeyer, P.-Y. (2020). Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *Conference on Robot Learning*, pages 835–853. PMLR.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Redko, I., Habrard, A., and Sebban, M. (2017). Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 737–753. Springer.

Romac, C., Portelas, R., Hofmann, K., and Oudeyer, P.-Y. (2021). Teachmyagent: a benchmark for automatic curriculum learning in deep rl. In *International Conference on Machine Learning*, pages 9052–9063. PMLR.

Rubino, G. and Tuffin, B. (2009). *Rare event simulation using Monte Carlo methods*. John Wiley & Sons.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Russo, D. and Van Roy, B. (2013). Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, pages 2256–2264.

Russo, D. and Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243.

Sachan, M. and Xing, E. (2016). Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 453–463.

Sadeghi, F. and Levine, S. (2016). Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*.

Scaman, K. and Virmaux, A. (2018). Lipschitz regularity of deep neural networks: Analysis and efficient estimation. *arXiv preprint arXiv:1805.10965*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schwaighofer, A., Tresp, V., and Yu, K. (2005). Learning gaussian process kernels via hierarchical bayes. In *Advances in Neural Information Processing Systems*, pages 1209–1216.

Schwartz, E. M., Bradlow, E. T., and Fader, P. S. (2017). Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522.

Settles, B. (2009). Active learning literature survey.

Settles, B. (2011). From theories to queries: Active learning in practice. In *Active learning and experimental design workshop in conjunction with AISTATS 2010*, pages 1–18. JMLR Workshop and Conference Proceedings.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shui, C., Abbasi, M., Robitaille, L.-É., Wang, B., and Gagné, C. (2019). A principled approach for learning task similarity in multitask learning. *arXiv preprint arXiv:1903.09109*.

Simchowitz, M. and Foster, D. (2020). Naive exploration is optimal for online lqr. In *International Conference on Machine Learning*, pages 8937–8948. PMLR.

Soare, M., Alsharif, O., Lazaric, A., and Pineau, J. (2014). Multi-task linear bandits. In *NIPS2014 Workshop on Transfer and Multi-task Learning: Theory meets Practice*.

Srebro, N., Sridharan, K., and Tewari, A. (2010). Optimistic rates for learning with a smooth loss. *arXiv preprint arXiv:1009.3896*.

Subramanian, S., Trischler, A., Bengio, Y., and Pal, C. J. (2018). Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*.

Sugiyama, M., Krauledat, M., and MÃžller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May):985–1005.

Sugiyama, M. and Storkey, A. J. (2007). Mixture regression for covariate shift. In *Advances in Neural Information Processing Systems*, pages 1337–1344.

Sun, L., Wang, T., Hui, B., Li, Y., Tian, L., et al. (2022). Explainable and personalized medical cost prediction based on multitask learning over mobile devices. *Mobile Information Systems*, 2022.

Sun, S., Shi, H., and Wu, Y. (2015). A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92.

Tai, Y., Yang, J., and Liu, X. (2017). Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279.

Tang, Y., Wang, X., Harrison, A. P., Lu, L., Xiao, J., and Summers, R. M. (2018). Attention-guided curriculum learning for weakly supervised classification and localization of thoracic diseases on chest radiographs. In *International Workshop on Machine Learning in Medical Imaging*, pages 249–258. Springer.

Teshima, T., Sato, I., and Sugiyama, M. (2020). Few-shot domain adaptation by causal mechanism transfer. *arXiv preprint arXiv:2002.03497*.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE.

Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.

Tripuraneni, N., Jordan, M., and Jin, C. (2020). On the theory of transfer learning: The importance of task diversity. *Advances in neural information processing systems*, 33:7852–7862.

Turğay, E., Öner, D., and Tekin, C. (2018). Multi-objective contextual bandit problem with similarity information. *arXiv preprint arXiv:1803.04015*.

Uehara, M., Zhang, X., and Sun, W. (2021). Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*.

Vanschoren, J. (2019). Meta-learning. *Automated machine learning: methods, systems, challenges*, pages 35–61.

Vilalta, R. and Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial intelligence review*, 18:77–95.

Wang, B., Mendez, J., Cai, M., and Eaton, E. (2019a). Transfer learning via minimizing the performance gap between domains. In *Advances in Neural Information Processing Systems*, pages 10645–10655.

Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153.

Wang, R., Du, S. S., Yang, L., and Salakhutdinov, R. R. (2020). On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826.

Wang, X., Chen, Y., and Zhu, W. (2021). A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576.

Wang, X. and Schneider, J. G. (2015). Generalization bounds for transfer learning under model shift. In *UAI*, pages 922–931.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. (2019b). Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*.

Weinshall, D. and Amir, D. (2020). Theory of curriculum learning, with convex loss functions. *Journal of Machine Learning Research*, 21(222):1–19.

Weng, R., Yu, H., Huang, S., Cheng, S., and Luo, W. (2020). Acquiring knowledge from pre-trained model to neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9266–9273.

Wilson, A., Fern, A., Ray, S., and Tadepalli, P. (2007). Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*, pages 1015–1022.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., and Agarwal, A. (2021). Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694.

Xu, Z., Meisami, A., and Tewari, A. (2020). Decision making problems with funnel structure: A multi-task learning approach with application to email marketing campaigns. *arXiv preprint arXiv:2010.08048*.

Xu, Z., Meisami, A., and Tewari, A. (2021). Decision making problems with funnel structure: A multi-task learning approach with application to email marketing campaigns. In *International Conference on Artificial Intelligence and Statistics*, pages 127–135. PMLR.

Xu, Z. and Tewari, A. (2021). Representation learning beyond linear prediction functions. *Advances in Neural Information Processing Systems*, 34:4792–4804.

Xu, Z. and Tewari, A. (2022). On the statistical benefits of curriculum learning. In *International Conference on Machine Learning*, pages 24663–24682. PMLR.

Yang, J., Lei, Q., Lee, J. D., and Du, S. S. (2022). Nearly minimax algorithms for linear bandits with shared representation. *arXiv preprint arXiv:2203.15664*.

Yang, R., Xu, H., Wu, Y., and Wang, X. (2020). Multi-task reinforcement learning with soft modularization. *Advances in Neural Information Processing Systems*, 33:4767–4777.

Yao, J., Killian, T., Konidaris, G., and Doshi-Velez, F. (2018). Direct policy transfer via hidden parameter markov decision processes. In *LLARLA Workshop, FAIM*, volume 2018.

Yom-Tov, E., Feraru, G., Kozdoba, M., Mannor, S., Tennenholtz, M., and Hochberg, I. (2017). Encouraging physical activity in patients with diabetes: intervention using a reinforcement learning system. *Journal of medical Internet research*, 19(10):e338.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645.

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. (2022). Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

Zhang, C. and Wang, Z. (2021). Provably efficient multi-task reinforcement learning with model transfer. *Advances in Neural Information Processing Systems*, 34:19771–19783.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. (2013). Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827.

Zhang, Y., Liu, T., Long, M., and Jordan, M. I. (2019). Bridging theory and algorithm for domain adaptation. *arXiv preprint arXiv:1904.05801*.

Zhang, Y. and Yang, Q. (2017). A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*.

Zhang, Z., Ji, X., and Du, S. (2021). Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon. In *Conference on Learning Theory*, pages 4528–4531. PMLR.