Inverse Reinforcement Learning and its application on calibration of human driver models

Yuanxin Zhong, Xinpeng Wang

Outline

- Motivation
- Review of Inverse Reinforcement Learning (IRL) based on Max-Entropy
- Problem Formulation
- Data Processing and Experiment Setup
- Preliminary Results
- Ongoing and Future Work

Motivation

- For autonomous vehicles (AVs), how to model the interactive driving behaviors?
  - Rule based model
    - Interpretable, but not representative
  - Learning-based model
    - Powerful, but not interpretable
  - Utility based model ✓
    - Interpretable, flexible, data-efficient
    - E.g.: Game theory, optimal control
- However, how to determine the utility function of human drivers realistically?

Motivation

- IRL: Learn the cost/reward functions from (human) demonstrations

"Forward" reinforcement learning

Given:

- State $x \in \mathcal{X}$, action $u \in \mathcal{U}$
- (sometimes) transitions $p(x'|x,u)$
- Reward function $r(x,u)$

Goal:

- To learn $\pi^*(u|x)$

Inverse reinforcement learning

Given:

- State $x \in \mathcal{X}$, action $u \in \mathcal{U}$
- (sometimes) transitions $p(x'|x,u)$
- Samples $\{\xi_i\}$ sampled from $\pi^*(\xi)$

Goal:

- To learn $r_\theta(x,u)$

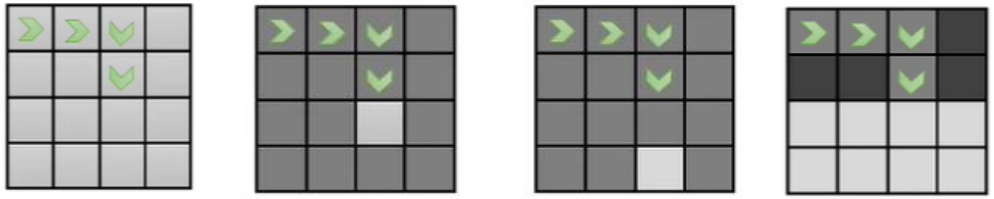
Reward parameters

Notation

- **State** x, **Action/Input** u
- **Demonstration**: $\xi_i \in D, i \in 1 \dots |D|$
- **Policy** π
- **Reward function**: $r(x, u|\theta)\big(= \theta^T f(x, u)\big)$, θ as the parameter to learn
  - θ can be Θ in some fomulation
- **Cumulative reward function**: $R(x, u)$
- **Social value orientation** (SVO) angle: ϕ
- **Time**: t
- **Optimization step**: k

Inverse reinforcement learning

- Demonstration: $D = \{\xi_1, \ldots \xi_n\}$
  - Where $\xi = \{x_0, u_0, x_1, u_1, \ldots, x_T, u_T\}$
- To find a reward function $R_\theta(\xi) = \sum_t r_\theta(x_t, u_t)$ $(= \theta^T \boldsymbol{f}(\xi))$
- Ill-defined problem: many rewards can explain the same behavior



- One valid goal: feature matching
  - Let π^{r_θ} be the optimal policy for r_θ
  - Pick θ such that $E_{\pi^{t_\theta}}[\boldsymbol{f}(x,u)] = \mathrm{E}_{\pi^*}[\boldsymbol{f}(x,u)] \approx \sum_{i=1}^{n} \boldsymbol{f}(\xi_i)$

Still ambiguous!

Expected feature value under π^{r_θ}

Unknown optimal policy approximate using expert samples

Max entropy IRL

- Out of all the feature-matching θ, pick the one s.t. π^{R_θ} has max entropy
- Solution: $P_{R_\theta}(\xi) \propto e^{\beta R_\theta(\xi)}$
  - $R(\xi) = \sum_t r(x_t, u_t)$
  - $\beta \uparrow$, more greedy
- Max entropy IRL: $P_{R_\theta}(\xi) = \frac{1}{Z} e^{\beta R_\theta(\xi)}$
  - Where normalization term $Z = \int \exp(\beta R_\theta(\xi) d\xi)$ Hard to compute
- Goal: $\max_\theta L = \sum_{\xi \in D} \log P_{R_\theta}(\xi)$

How to compute Z

1. Direct computation with DP
2. Sampling-based method
3. Guided cost learning
4. Laplace approximation

Max entropy IRL

- Goal:

$$\max_{\theta} L(\theta) = \sum_{\xi \in D} \log P_{R_\theta}(\xi)$$
$$= \sum_{\xi \in D} \log \frac{1}{Z} \exp\big(R_\theta(\xi)\big)$$
$$= \sum_{\xi \in D} R_\theta(\xi) - |D| \log Z$$
$$= \sum_{\xi \in D} R_\theta(\xi) - |D| \log \sum_{\xi} \exp\big(R_\theta(\xi)\big)$$

$$\nabla_\theta L(\theta) = \sum_{\xi \in D} \frac{dR_\theta(\xi)}{d\theta} - \frac{|D|}{\sum_\xi \exp\big(R_\theta(\xi)\big)} \sum_{\xi} \exp\big(R_\theta(\xi)\big) \frac{dR_\theta(\xi)}{d\theta}$$

Empirical feature value: $\sum_{i=1}^{M} f(\tau_i)$

Expected feature value

$$\sum_{\xi} P(\xi|\theta) \frac{dR_\theta(\xi)}{d\theta}$$

$$\sum_{s} P(s|\theta) \frac{dr_\theta(\xi)}{d\theta}$$

Method [1]: compute gradient directly

Algorithm:

1. Initialize θ, gather demonstrations D
2. Solve for optimal policy $\pi(a|s)$ w.r.t. reward r_θ
3. Solve for state visitation frequencies $P(s|\theta)$
4. Compute gradient $\nabla_\theta L = -\frac{1}{|D|}\sum_{\xi \in D} \frac{dr_\theta}{d\theta}(\xi) - \sum_s p(s|\theta)\frac{dr_\theta}{d\theta}(s)$
5. Update θ with one gradient step using $\nabla_\theta L$

- Computationally expensive
- Cannot handle continuous state/action & high-dimension problems.

Method [2]: inverse optimal control

- Iteratively, approximate Z with the current set of optimal trajectories
  - $Z_j = e^{\beta R_{\theta_j}(\xi_j^*)}$

$$Z = \int \exp(\beta R_\theta(\xi) d\xi)$$

- Algorithm
  1. Compute the empirical feature vector over all demos $\tilde{\boldsymbol{f}} = \sum_{i=1}^{n} \boldsymbol{f}(\xi_i)$
  2. Initialize θ
  3. For each demo, find an optimized trajectory w.r.t current θ: $\xi_1^\theta \ldots \xi_n^\theta$, and $\boldsymbol{f^\theta} = \frac{1}{n}\sum_{i=1}^{n} f(\xi_i^\theta)$
  4. Compute the gradient of likelihood:$\nabla_\theta L(\theta) \approx \boldsymbol{f^\theta} - \tilde{\boldsymbol{f}}$ and, use it to update θ
  5. Repeat from 3.

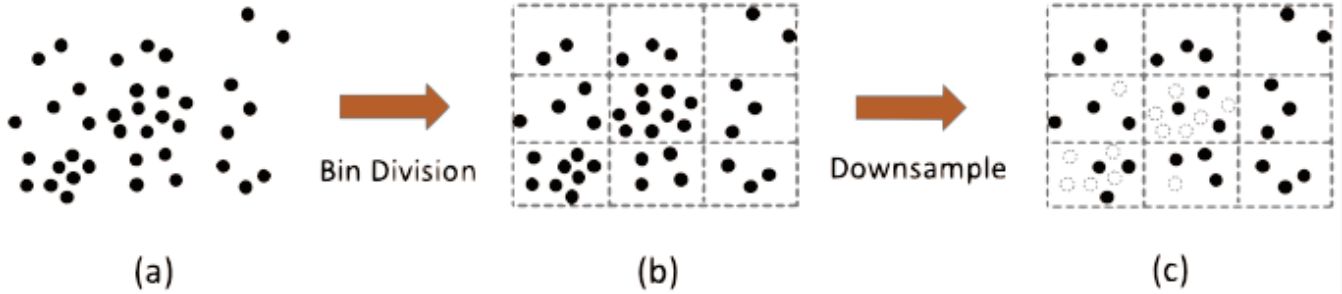
- Efficient, but the computation of Z might be biased.

Method [3]: approximate Z by sampling

- $Z_{\xi_i} \approx \sum_{m=1}^{K} e^{\beta R(\tau_m^i, \theta)}$
- In a sampling setting, the gradient becomes
  - $$\nabla_\theta L = \frac{\beta}{M} \sum_{i=1}^{M} \left(\boldsymbol{f}(\xi_i) - \tilde{\boldsymbol{f}}(\xi_i) \right)$$

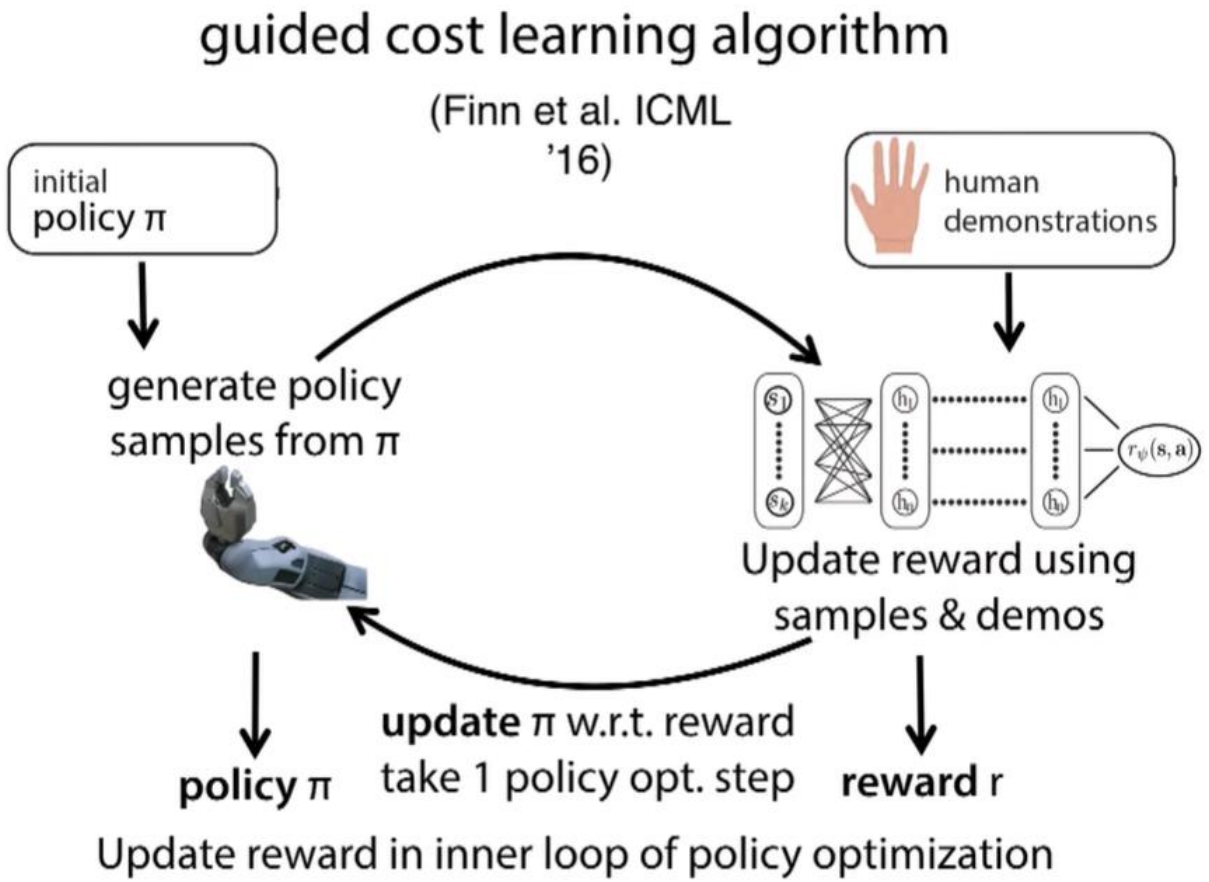
$$\tilde{\boldsymbol{f}}(\xi_i) = \sum_{m=1}^{K} \frac{e^{\beta R(\tau_m^i, \theta)}}{\sum_{m=1}^{K} e^{\beta R(\tau_m^i, \theta)}} \boldsymbol{f}(\tau_m^i)$$

- A key step: after sampling with some scheme, redistribute samples s.t. they are (almost) uniform in the feature space



Another sampling method: Guided cost learning [4]

- Sample adaptively to estimate Z by constructing a policy



The method we pick: Continuous inverse optimal control (CIOC):

- Assumption:
  - Expert demonstrations are locally optimal
- $R_\Theta(\tilde{\xi}) \approx R_\Theta(\xi_D) + (\tilde{\xi} - \xi_D)^T \frac{\partial R}{\partial \xi_D} + (\tilde{\xi} - \xi_D)^T \frac{\partial^2 R}{\partial \xi_D^2} (\tilde{\xi} - \xi_D)$ – Taylor expansion
- Then, we can approximate the likelihood of any trajectory:
  - $P_{R_\Theta}(\tilde{\xi}) = \frac{1}{Z} e^{\beta R_\Theta(\tilde{\xi})} = e^{\beta R_\Theta(\tilde{\xi})} \left(\int \exp(R_\Theta(\tau) d\tau)\right)^{-1}$

$$= \left[\int e^{-\frac{1}{2}\mathbf{g}^T\mathbf{H}^{-1}\mathbf{g} + \frac{1}{2}(\mathbf{H}(\tilde{u}-u)+\mathbf{g})^T\mathbf{H}^{-1}(\mathbf{H}(\tilde{u}-u)+\mathbf{g})} d\tilde{u}\right]^{-1}$$

$$= e^{\frac{1}{2}\mathbf{g}^T\mathbf{H}^{-1}\mathbf{g}} |-\mathbf{H}|^{\frac{1}{2}} (2\pi)^{-\frac{d_u}{2}}$$

Z becomes Gaussian Integral

- $\mathcal{L} = \frac{1}{2}\mathbf{g}^T\mathbf{H}^{-1}\mathbf{g} + \frac{1}{2}\log\left(\det(-\mathbf{H})\right) - \frac{d_\mathbf{u}}{2}\log 2\pi$ (1)
  - Where $\mathbf{g} = \frac{\partial R}{\partial \xi_D}, \mathbf{H} = \frac{\partial^2 R}{\partial \xi_D^2}$

The algorithm of continuous inverse optimal control (CIOC)[5]:

- Algorithm*:

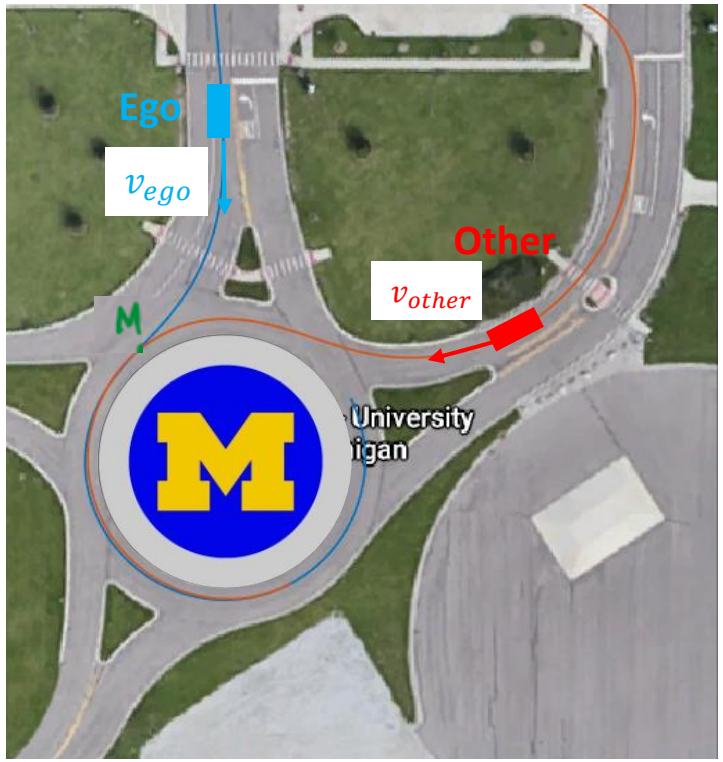
1. Gather demonstration D, compute feature counts $\boldsymbol{f}(\xi)$
   - An extra regularizing feature f_r and weight θ_r to secure a positive $\det(-\mathbf{H})$
2. Compute $\mathcal{L}(\Theta)$ & $\frac{\partial \mathcal{L}}{\partial \Theta}$ approximately according to (1)
3. Solve $\max_{\theta} \mathcal{L}(\theta)$ using augmented Lagrangian methods
   - Drive $\theta_r \to 0$ while maximizing $\mathcal{L}(\theta)$ with gradient-based optimization

- Pros:
  - Compute the likelihood of demos analytically: no need to sample/ solving an MDP
  - Efficient for higher-dimensional problems & continuous problems
  - Demos only need to be “locally optimal”

Problem Formulation: Learn a human driver model

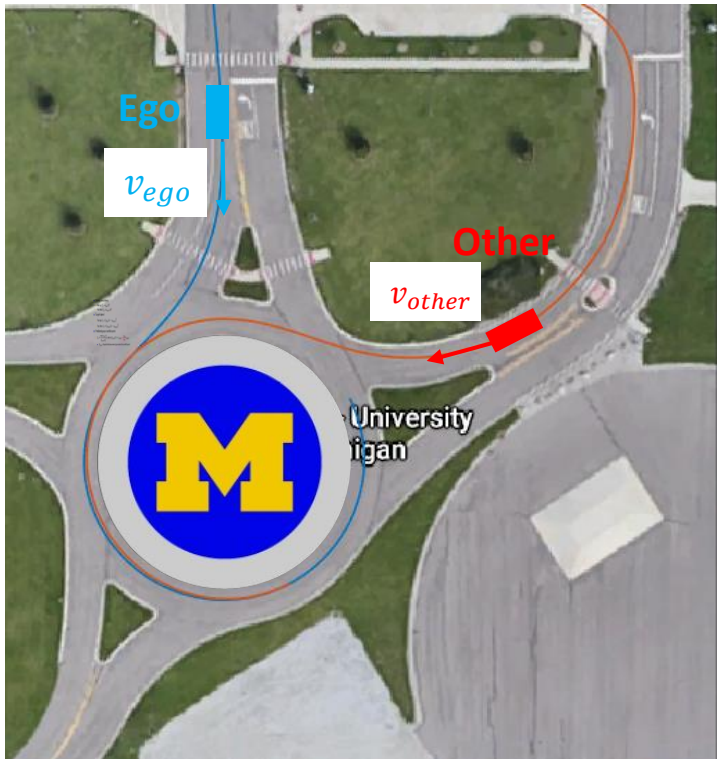
- Goal: Using IRL, learn a universal utility function for driving, that also incorporate individual driving styles of human drivers
  - $r(x, u|\ \Theta,\ \phi) \leftarrow \sum_{agents} g(x, u|\Theta) g_{\phi}(\phi)$

    Shared: Θ — Personal: ϕ
  - $g(x, u|\Theta) \leftarrow \theta^T f(x, u)$
- Target scenario: Roundabout entering
  - Start with 2-vehicle interaction
  - $x = [x_{ego}, x_{other}, v_{ego}, v_{other}]$
  - $u = [a_{ego}, a_{other}]$



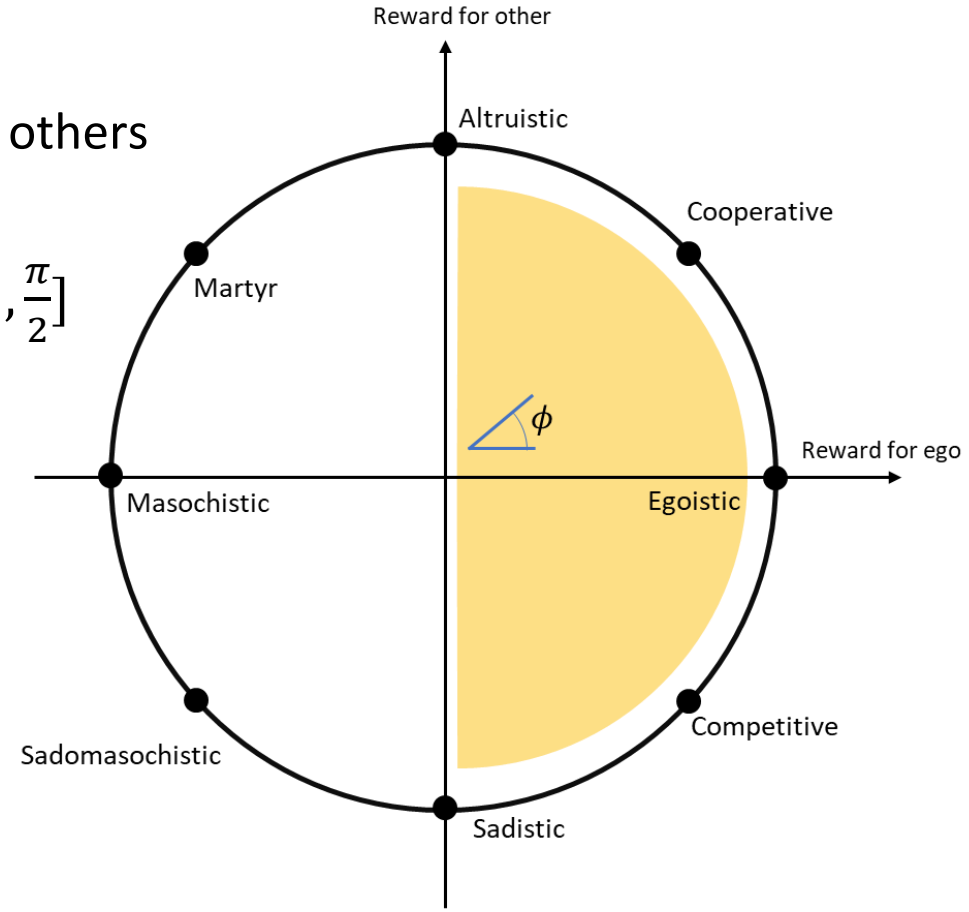
Problem Formulation: Feature Design

- Driving smoothness
  - For ego: $f_1 = a_{ego}(t)^2$
  - For other: $f_2 = a_{other}(t)^2$
- Target speed
  - For ego: $f_3 = \left(v_{ego}(t) - v_{target}\right)^2$
  - For other: $f_4 = \left(v_{other}(t) - v_{target}\right)^2$
- Predicted gap at conflict point
  - $f_5 = \frac{\exp(\alpha x_{ego}(t))}{\left(D_{pred}(t)\right)^2}$, where $D_{pred}(t) = x_{other} - \frac{x_{ego}}{v_{ego}} v_{other}$
    - D_{pred}: Projected distance when Ego reaches the conflict point



Social Value Orientation[6]

- Characterize the value preference on benefits for ego versus others
- Reward formulation:
  - $r_{ego} = g_{ego}\cos(\phi) + g_{other}\sin(\phi) + g_{shared}, \phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$
- $\Theta = \begin{bmatrix} \Theta_{ego} \\ \Theta_{other} \\ \Theta_{shared} \end{bmatrix}$
- $g_{ego} = \Theta_{ego}^T [f_1, f_3]^T$
- $g_{other} = \Theta_{other}^T [f_2, f_4]^T$
- $g_{shared} = \Theta_{shared}^T f_5$



Interpretation of SVO angle

- Goal: Use IRL to learn a universal Θ, while estimating a personalized ϕ for each demonstration

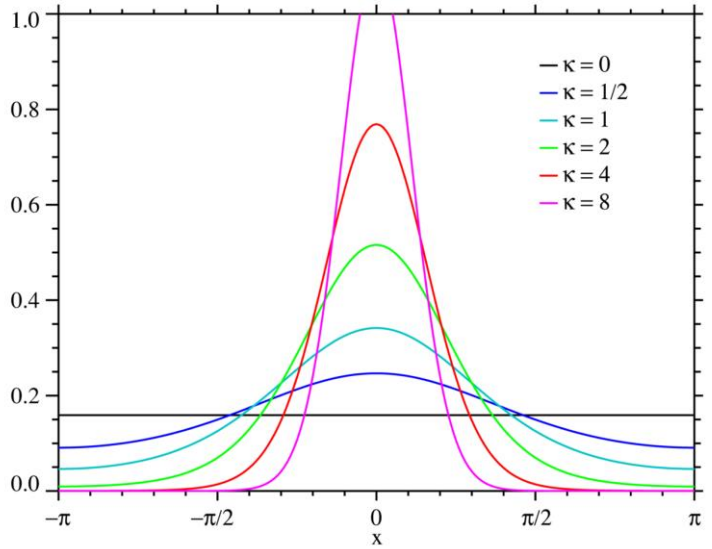
[6] Schwarting, W., et al. (2019). Social behavior for autonomous vehicles. Proceedings of the National Academy of Sciences of the United States of America, 116(50), 2492–24978.

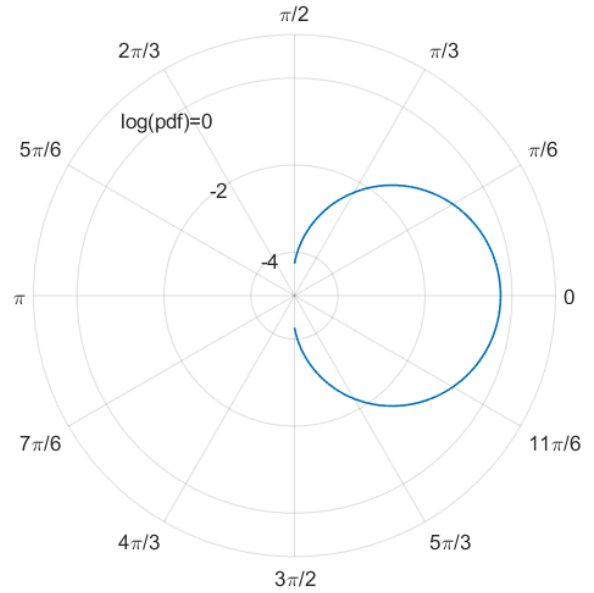
Combining SVO and IRL

- Previous methods either learned a universal $r(x, u)$[5,6], or used part of demos to learn different $r(x, u)$[7].
- Our goal: to learn personalized reward functions while fully utilizing all demos.
- Key objective: maximize the likelihood of given demonstration over both $\Theta, \Phi = \{\phi_1, \dots, \phi_{|D|}\}$

$$\max_{\Theta,\Phi} P(D|\Theta, \Phi) = \prod_{i=1}^{|D|} P(\xi_i|\Theta, \phi_i)$$

- High-level ideas:
  - Based on Algorithm* (CIOC)
  - Maintain a probabilistic estimate for each ϕ_i
  - Update Θ and ϕ_i iteratively





Prior for ϕ_i: von-Mises distribution

Combining SVO and IRL: the workflow

1. Assume a prior distribution of all ϕ_i : $\phi_i \sim vM(0, \kappa)$, pick $\phi_i^0 = 0$
2. At step k, compute optimal Θ^k using Algorithm*
   - $\Theta^k = \arg\max_{\Theta} \prod_{i=1}^{|D|} P(\xi|\phi_i^{k-1}, \Theta)$
3. Based on Θ^k, update the posterior of ϕ_i:
   - $P(\phi_i|\xi_i, \Theta^k) \propto P(\xi_i|\phi_i, \Theta^k) P(\phi_i|\Theta^k)$ *(Bayes update)*
   - Assuming $P(\phi_i|\Theta^k) = P(\phi_i|\Theta^{k-1}, \xi_i)$ *(use posterior of last step as prior of this step)*
4. Select $\phi_i^k = \arg\max_{\phi_i} P(\phi_i|\Theta^k, \xi_i)$
5. Repeat 2,3 until convergence

Proof of improvement: a sketch

- Let $l_i(\phi_i, \Theta) = P(\xi_i|\phi_i, \Theta)$; $L(\Phi, \Theta) = \prod_{i=1}^{|D|} P(\xi_i|\phi_i, \Theta)$

Goal: to prove that $L(\phi^k, \Theta^k) \geq L(\phi^{k-1}, \Theta^{k-1})$

On one hand (from step2):

- $\Theta^k = \arg\max_{\Theta} L(\Phi^{k-1}, \Theta)$, so we have $L(\Phi^{k-1}, \Theta^k) \geq L(\Phi^{k-1}, \Theta^{k-1})$

On the other hand (from step4):

- $\phi_i^k = \arg\max_{\phi} P(\phi|\Theta^k, \xi_i) = \arg\max_{\phi} l_i(\phi, \Theta^k) P(\phi|\Theta^{k-1}, \xi_i)$
- While $\phi_i^{k-1} = \arg\max_{\phi} P(\phi|\Theta^{k-1}, \xi_i)$
- Therefore,
  - if $\phi_i^k \neq \phi_i^{k-1}$, then $l_i(\phi_i^k, \Theta^k) > l_i(\phi_i^{k-1}, \Theta^k)$
  - If $\phi_i^k = \phi_i^{k-1}$, then $l_i(\phi_i^k, \Theta^k) = l_i(\phi_i^{k-1}, \Theta^k)$
- Taking product of all l_i, we have: $L(\Phi^k, \Theta^k) \geq L(\Phi^{k-1}, \Theta^k)$

Therefore, We have $L(\phi^k, \Theta^k) \geq L(\phi^{k-1}, \Theta^{k-1})$. ■

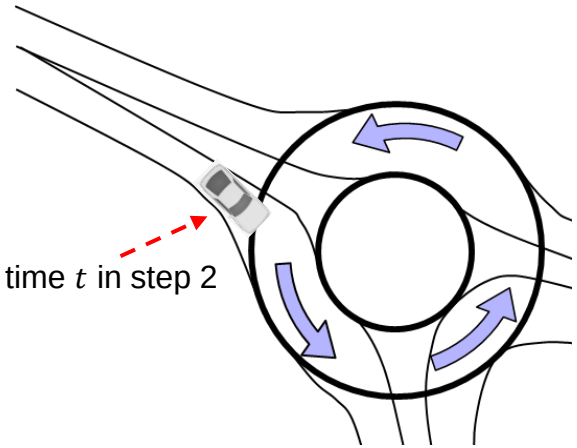
Datasets

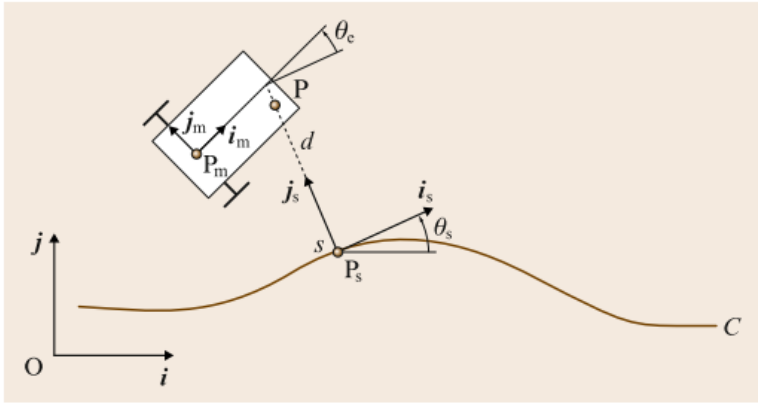
Traffic scenario datasets:

- INTERnational, Adversarial and Cooperative moTION Dataset (INTERACTION)
  - Naturalistic motions of various traffic participants
  - HD-Map is provided
- HighD
  - Traffic data recorded by drones at German highways
- RounD
  - Traffic data recorded by drones at German roundabouts

Trajectory Separation

- Select roundabout scenarios for finding interactions between two road users
- Procedure to generate each demonstration
  1. Select ego vehicle
  2. Find the timestamp t when ego vehicle is entering the roundabout
  3. Find the other vehicle currently inside roundabout with proper angle difference
  4. Collect the trajectory of the two vehicles between $t - 4s$ and $t + 2s$
  5. The trajectory will be finally converted to Frenet coordinate along reference path

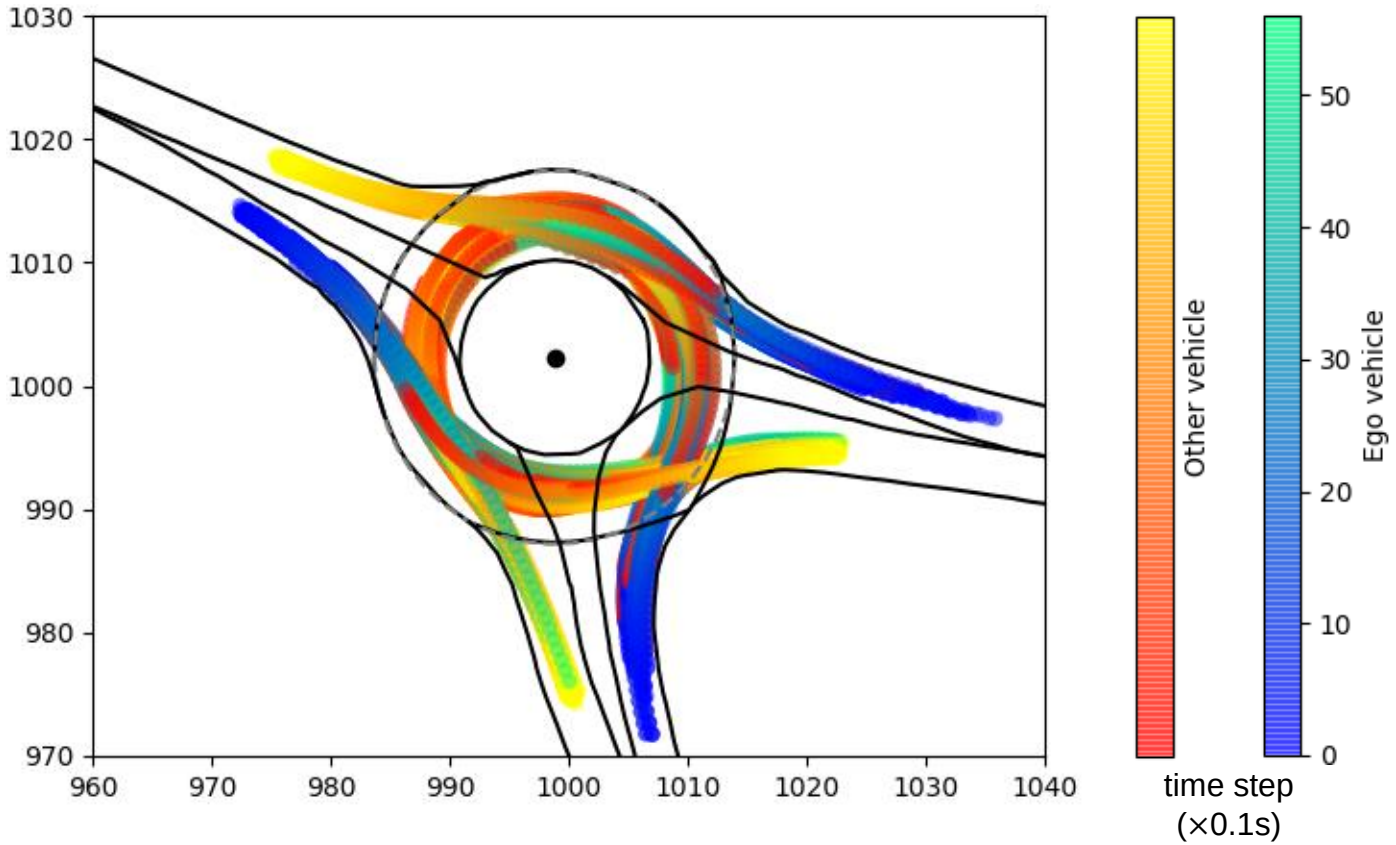




Frenet coordinate system

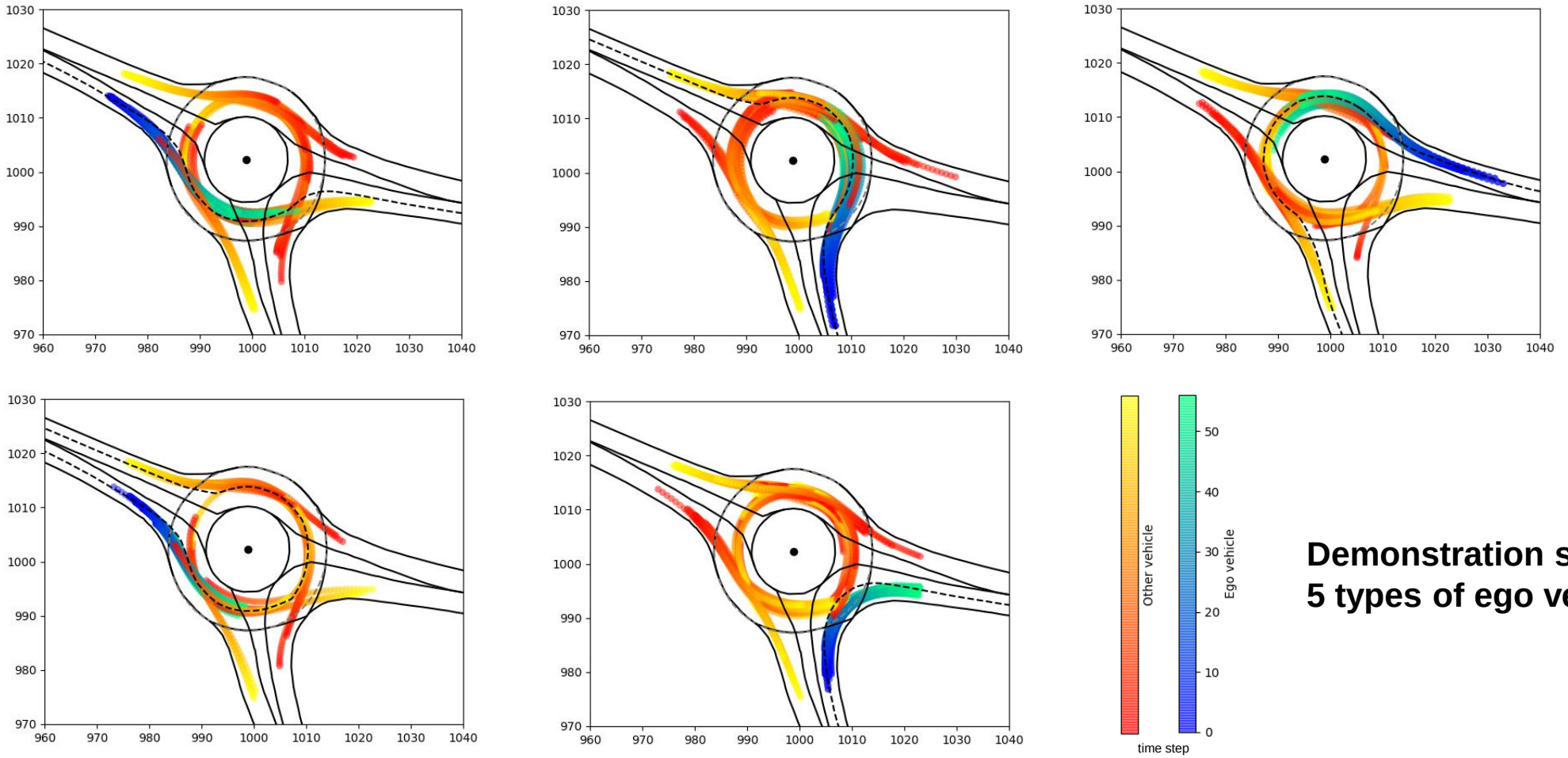
Demonstrations Visualization

- 500 extracted demonstrations from DR_DEU_Roundabout_OF sequence in INTERACTION dataset



Interaction Generation

- Each demonstration is associated with reference paths for ego and the other vehicle

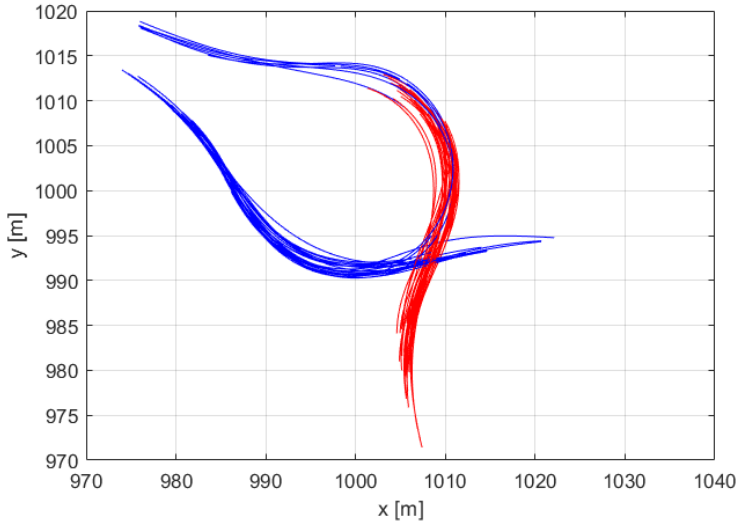


Demonstration samples from 5 types of ego vehicle routes

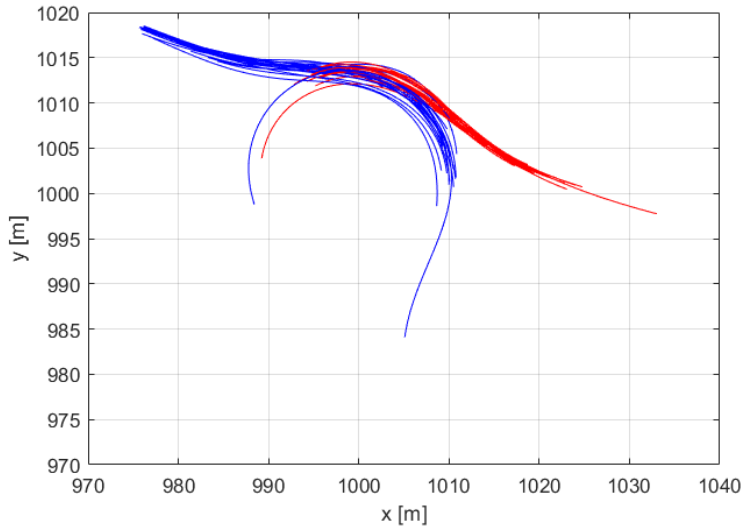
Demonstration for Experiment

- Time horizon is truncated to 3s (Ts = 0.1s)
- Only trajectories where the other vehicle enters just before ego vehicle are selected

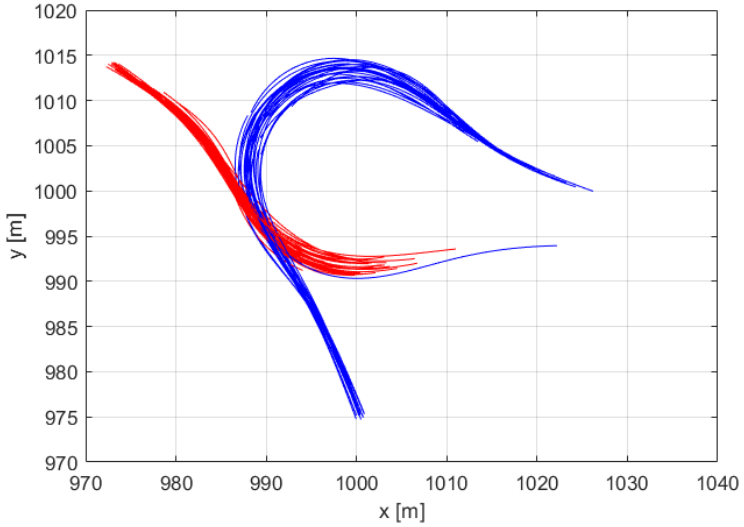
Ego from south: 35 demos



Ego from west: 25 demos



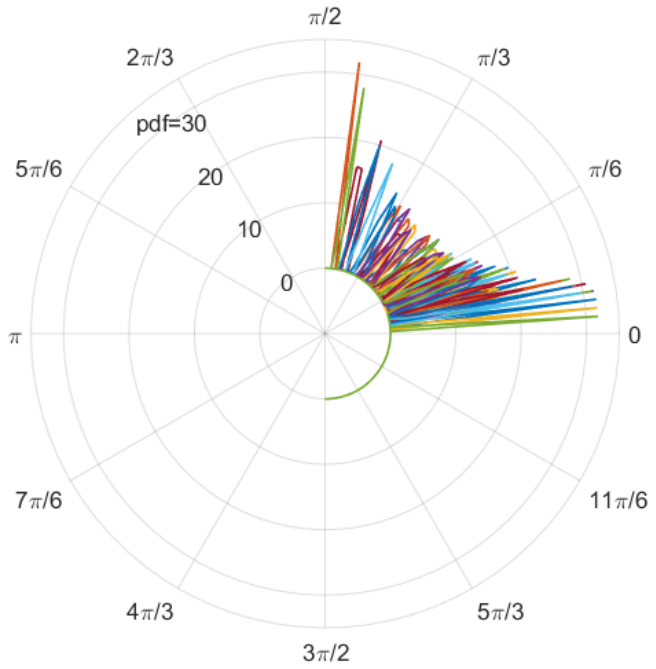
Ego from west: 50 demos



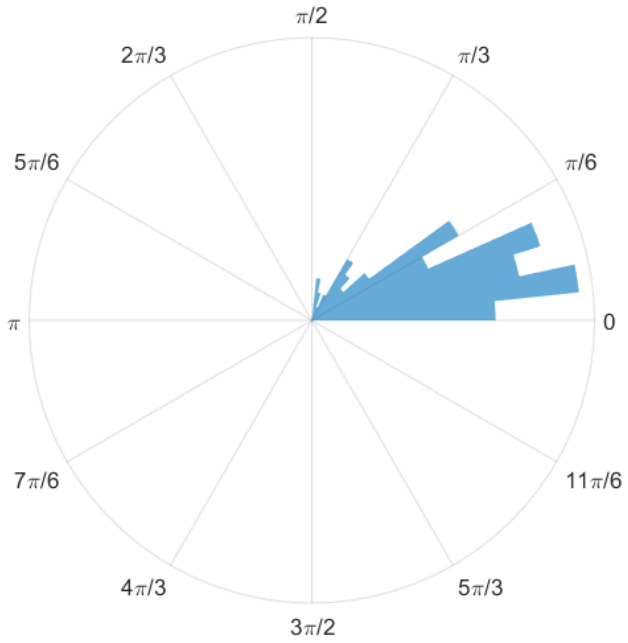
3 types of trajectories used for IRL

Preliminary Results

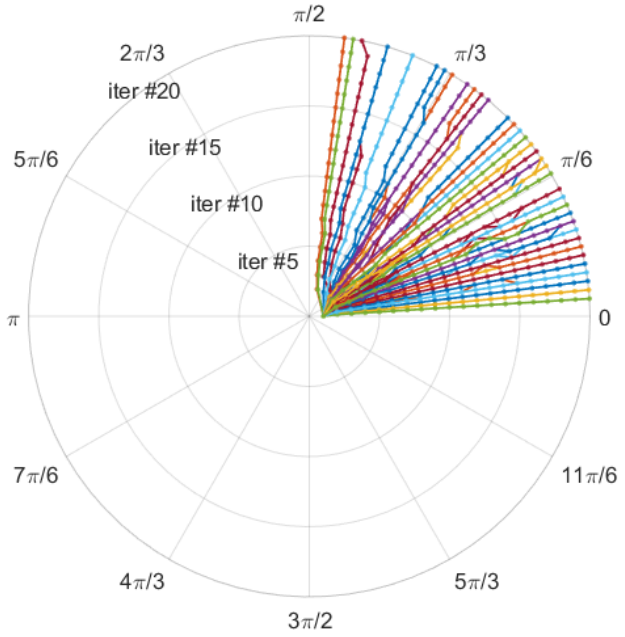
- Results are plotted in polar coordinate based on ϕ



Final posterior distribution of ϕ_i



Histogram of ϕ_i



Evolution of ϕ_i

Future work

- Use estimated θ and ϕ to predict the vehicle trajectory
- Test the algorithm in other maps of INTERACTION

References

1. Ziebart, B. D., Maas, A., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Proceedings of the National Conference on Artificial Intelligence* (Vol. 3, pp. 1433–1438). Retrieved from www.aaai.org
2. Kuderer, M., Gulati, S., & Burgard, W. (2015). Learning driving styles for autonomous vehicles from demonstration. *Proceedings - IEEE International Conference on Robotics and Automation*, *2015-June*(June), 2641–2646. https://doi.org/10.1109/ICRA.2015.7139555
3. Wu, Z., Sun, L., Zhan, W., Yang, C., & Tomizuka, M. (2020). Efficient Sampling-Based Maximum Entropy Inverse Reinforcement Learning with Application to Autonomous Driving. *ArXiv*, *5*(4), 5355–5362.
4. Finn, Chelsea, Sergey Levine, and Pieter Abbeel. "Guided cost learning: Deep inverse optimal control via policy optimization." International conference on machine learning. PMLR, 2016.
5. Levine, S., & Koltun, V. (2012). Continuous inverse optimal control with locally optimal examples. *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, *1*, 41–48.
6. Schwarting, W., Pierson, A., Alonso-Mora, J., Karaman, S., & Rus, D. (2019). Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences of the United States of America*, *116*(50), 2492–24978.
7. Sun, L., Wu, Z., Ma, H., & Tomizuka, M. (2020). Expressing diverse human driving behavior with probabilistic rewards and online inference. *IEEE International Conference on Intelligent Robots and Systems*, 2020–2026.

Thank you

Any question is welcome!