Objective-aware Traffic Simulation via Inverse Reinforcement Learning

Guanjie Zheng[1,4*] , **Hanyang Liu**[2*] , **Kai Xu**[3] , **Zhenhui Li**[4]

[1]Shanghai Jiao Tong University
[2]Washington University in St. Louis
[3]Shanghai Tianrang Intelligent Technology Co., Ltd
[4]The Pennsylvania State University
gjzheng@sjtu.edu.cn, hanyang.liu@wustl.edu, kai.xu@tianrang-inc.com, jessieli@ist.psu.edu

Abstract

Traffic simulators act as an essential component in the operating and planning of transportation systems. Conventional traffic simulators usually employ a calibrated physical car-following model to describe vehicles' behaviors and their interactions with traffic environment. However, there is no universal physical model that can accurately predict the pattern of vehicle's behaviors in different situations. A fixed physical model tends to be less effective in a complicated environment given the non-stationary nature of traffic dynamics. In this paper, we formulate traffic simulation as an inverse reinforcement learning problem, and propose a parameter sharing adversarial inverse reinforcement learning model for dynamics-robust simulation learning. Our proposed model is able to imitate a vehicle's trajectories in the real world while simultaneously recovering the reward function that reveals the vehicle's true objective which is invariant to different dynamics. Extensive experiments on synthetic and real-world datasets show the superior performance of our approach compared to state-of-the-art methods and its robustness to variant dynamics of traffic.

1 Introduction

Traffic simulation has long been one of the significant topics in transportation. Microscopic traffic simulation plays an important role in planing, designing and operating of transportation systems. For instance, an elaborately designed traffic simulator allows the city operators and planers to test policies of urban road planning, traffic restrictions, and the optimization of traffic congestion, by accurately deducing possible effects of the applied policies to the urban traffic environment [Toledo *et al.*, 2003]. The prior work [Wei *et al.*, 2018] employs a traffic simulator to train and test policies of intelligent traffic signal control, as it can generate a large number of simulated data for the training of the signal controller.

Current transportation approaches used in the state-of-the-art traffic simulators such as [Yu and Fan, 2017; Osorio and Punzo, 2019] employ several physical and empirical equations to describe the kinetic movement of individual vehicles, referred to as the car-following model (CFM). The parameters of CFMs such as maximum acceleration and driver reaction time must be carefully calibrated using traffic data. A calibrated CFM can be exploited as a policy providing the vehicle optimal actions given the state of the environment, as shown in Figure 1. The optimality of this policy is enabled by the parameter calibration that obtains a close match between the observed and simulated traffic measurements.

An effective traffic simulator should produce accurate simulations despite variant dynamics in different traffic environments. This can be factorized into two specific objectives. The *first objective* is to accurately imitate expert vehicle behaviors given a certain environment with stationary dynamics. The movement of real-world vehicles depends on many factors including speed, distance to neighbors, road networks, traffic lights, and also driver's psychological factors. A CFM usually aims to imitate the car-following behavior by applying physical laws and human knowledge in the prediction of vehicle movement. Faced with sophisticated environment, models with emphasis on fitting different factors are continuously added to the CFM family. For example, Krauss model [Krauß, 1998] focuses on safety distance and speed, while Fritzsche model sets thresholds for vehicle's movement according to driver's psychological tendency. However, there is no universal model that can fully uncover the truth of vehicle-behavior patterns under comprehensive situations. Relying on inaccurate prior knowledge, despite calibrated, CFMs often fail to exhibit realistic simulations.

The *second objective* is to make the model robust to variant dynamics in different traffic environments. However, it is challenging due to the non-stationary nature of real-world traffic dynamics. For instance, weather shifts and variances of road conditions may change a vehicle's mechanical property and friction coefficient against the road surface, and eventually lead to variances of its acceleration and braking performance. In a real-world scenario, a vehicle would accordingly adjust its driving policy and behave differently (e.g., use different acceleration or speed given the same observation) under these dynamics changes. However, given a fixed policy (i.e., CFM), current simulators in general, fail to adapt poli-

[*] Equal contribution. Part of the work was done when Guanjie Zheng and Hanyang Liu were interns at Tianrang Inc. and when Guanjie Zheng was completing his PhD at the Pennsylvania State University.

cies to different dynamics. To simulate a different traffic environment with significantly different dynamics, the CFM must be re-calibrated using new trajectory data with respect to that environment, which is inefficient and sometimes unpractical. This makes these simulation models less effective when generalized to changed dynamics.

We aim to achieve both objectives. For the first objective, a natural consideration is to learn patterns of vehicles' behaviors directly from real-world observations, instead of relying on sometimes unreliable prior knowledge. Recently, imitation learning (IL) has shown promise for learning from demonstrations [Ho and Ermon, 2016; Fu *et al.*, 2018]. However, direct IL methods such as behavioral cloning [Michie *et al.*, 1990] that aim to directly extract an expert policy from data, would still fail in the second objective, as the learned policy may lose optimality when traffic dynamics change. A variant of IL, inverse reinforcement learning (IRL), different from direct IL, not only learns an expert's policy, but also infers the reward function (i.e., cost function or objective) from demonstrations. The learned reward function can help explain an expert's behavior and give the agent an objective to take actions imitating the expert, which enables IRL to be more interpretable than direct IL. To this end, we propose to use an IRL-based method that build off the adversarial IRL (AIRL) [Fu *et al.*, 2018], to train traffic simulating agents that generate accurate trajectories while simultaneously recovering the invariant underlying reward of the agents, which facilitates the robustness to variant dynamics.

With disentangled reward of an agent recovered from real-world demonstrations, we can infer the vehicle's true objective. Similar to the human driver's intention (e.g., drive efficiently under safe conditions), the estimated objective can be invariant to changing dynamics (e.g., maximum acceleration or deceleration). Considering the intricate real-world traffic with multiple vehicles interacting with each other, we extend AIRL to the multi-agent context of traffic simulation. We incorporate a scalable decentralized parameter sharing mechanism in [Gupta *et al.*, 2017] with AIRL, yielding a new algorithm called Parameter Sharing AIRL (PS-AIRL), as a dynamics-robust traffic simulation model. In addition, we propose an online updating procedure, using the learned reward to optimize new policies to adapt to different dynamics in the complex environment, without any new trajectory data. Specifically, our contributions are as threefold:

- We propose an IRL-based model that is able to infer real-world vehicle's true objective. It enables us to optimize policies adaptive and robust to different traffic dynamics.
- We extend the proposed model with the parameter sharing mechanism to a multi-agent context, enabling our model good scalable capacity in large traffic.
- Extensive experiments on both synthetic and real-world datasets show the superior performance in trajectory simulation, reward recovery and dynamics-robustness of PS-AIRL over state-of-the-art methods.

2 Related Work

Conventional CFM Calibration. Traditional traffic simulation methods such as SUMO [Krajzewicz *et al.*, 2012], AIMSUN [Barceló and Casas, 2005] and MITSIM [Yang and Koutsopoulos, 1996], are based on calibrated CFM models, which are used to simulate the interactions between vehicles. In general, CFM considers a set of features to determine the speed or acceleration, such as safety distance, and velocity difference between two adjacent vehicles in the same lane.

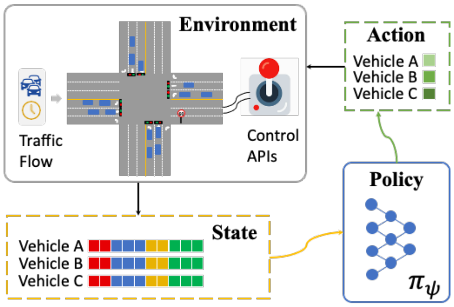

Figure 1: Interactions between the simulation environment and policy (e.g., car-following model).

Learning to Simulate. Recently, several approaches proposed to apply reinforcement learning in simulation learning problems such as autonomous driving [Wang *et al.*, 2018]. For these works, the reward design is necessary but complicated because it is difficult to mathematically interpret the vehicle's true objective. The prior work [Bhattacharyya *et al.*, 2018] proposed to frame the driving simulation problem in multi-agent imitation learning, and followed the framework of [Ho and Ermon, 2016] to learn optimal driving policy from demonstrations, which has a similar formulation to our paper. But different from [Bhattacharyya *et al.*, 2018], our traffic simulation problem has totally different concentrations. Our work mainly focus on the fidelity of simulated traffic flow and car-following behavior, as well as correct reactions to traffic light switchover, while [Bhattacharyya *et al.*, 2018] ignores traffic signals, and underlines safe driving when interacting with neighboring vehicles. The work [Zheng *et al.*, 2020] uses a similar framework for traffic simulation but is unable to achieve dynamic-robustness like our proposed model.

3 Problem Definition

We frame traffic simulation as a multi-agent control problem, regarding the complicated interaction between vehicles in traffic. Formally, we formulate our model in a decentralized partially observable Markov decision process (DPOMDP) defined by the tuple $(\mathcal{M}, \{\mathcal{S}_m\}, \{\mathcal{A}_m\}, \mathcal{T}, r, \gamma, \rho_0)$. Here $\mathcal{M}$ denotes a finite set of agents, $\{\mathcal{S}_m\}$ and $\{\mathcal{A}_m\}$ the sets of states and actions for each agent m, respectively. We have $\mathcal{S}_m \in \mathbb{R}^S$ and $\mathcal{A}_m \in \mathbb{R}^A$ for all agents $m \in \mathcal{M}$. ρ_0 denotes the initial state distribution. $r(s, a)$ is the reward function, and γ denotes the long-term reward discount factor. We assume for a set of given expert demonstrations, the environment dynamics remain unchanged. The deterministic state transition function is defined by $\mathcal{T}(s'|s, a)$. We formulate the microscopic traffic simulation problem in a inverse reinforcement learning (IRL) fashion. Given trajectories of expert vehicles, our goal is to learn the reward function of the vehicle agent. Formally, the problem is defined as follows:

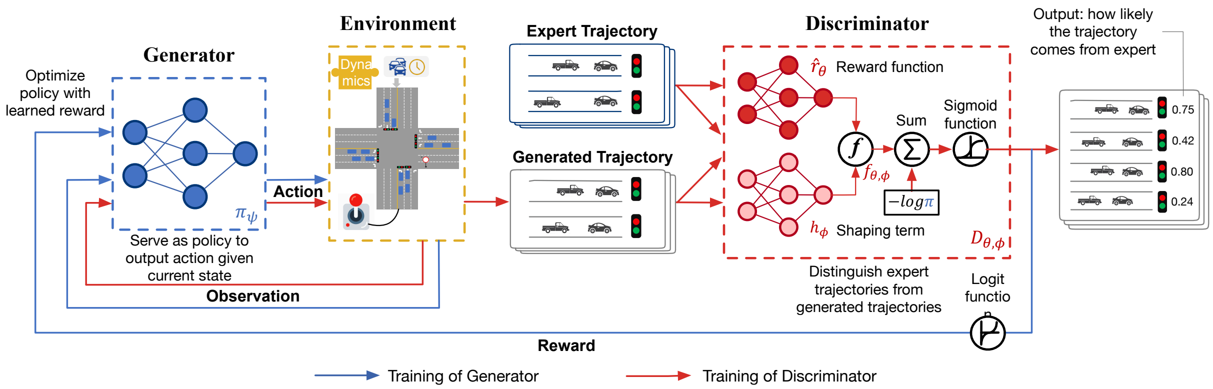
Generator
Optimize policy with learned reward
π_ψ
Serve as policy to output action given current state
Environment
Dyna mics
Action
Observation
Expert Trajectory
Generated Trajectory
Discriminator
$\hat{r}_\theta$ Reward function
Sum
Sigmoid function
f
$f_{\theta,\phi}$
$-log\pi$
h_ϕ Shaping term
$D_{\theta,\phi}$
Distinguish expert trajectories from generated trajectories
Logit functio n
Output: how likely the trajectory comes from expert
0.75
0.42
0.80
0.24
Reward
Training of Generator
Training of Discriminator

Figure 2: Overview of PS-AIRL. The discriminator and the generator are trained alternatively. When training the discriminator, the generator serves as a fixed policy that rollouts trajectories. The discriminator takes both generated and expert trajectories as input and distinguishes between them. When training the generator, the learned reward function is used as the cost function to update the optimal policy.

Problem 1 *Given a set of expert trajectories as demonstrations $\mathcal{D} = \{\tau_1, \tau_2, ..., \tau_M\}$ generated by an optimal policy $\pi^*(a|s)$ where $\tau = \{s_1, a_1, ..., s_T, a_T\}$, the goal is to recover the reward function $r_\theta(s, a)$, so that the log likelihood of the expert trajectories are maximized, i.e.,*

$$\max_\theta E_{\tau\sim\mathcal{D}}[\log p_\theta(\tau)] \tag{1}$$

Here, $p_\theta(\tau) \propto \exp\{\sum_{t=0}^T \gamma^t r_\theta(s_t, a_t)\}$ is the distribution of trajectories parameterized by θ in the reward $r_\theta(s, a)$.

4 Method

4.1 Adversarial Inverse Reinforcement Learning

Along the line of IRL, [Finn *et al.*, 2016b] proposed to treat the optimization problem in Eq. (1) as a GAN-like [Goodfellow *et al.*, 2014] optimization problem. This adversarial training framework alternates between training the discriminator to classify each expert trajectory and updating the policy to *adversarially* confuse the discriminator, and optimize the loss:

$$\min_\theta \max_\psi \mathbb{E}_{\pi_\psi}\left[\log\left(D_\theta(s, a)\right)\right] + \mathbb{E}_{\pi_E}\left[\log\left(1 - D_\theta(s, a)\right)\right] \tag{2}$$

where D_θ is the discriminator parameterized by θ, π_E is the optimal policy for expert demonstration data $\mathcal{D}$, and π_ψ is the learned policy parameterized by ψ in the generator. Specifically, the discriminator scores each sample with its likelihood of being from the expert, and use this score as the learning cost to train a policy that generates fake samples. Thus, the objective of the generator is to generate samples that confuse the discriminator, where $\log D_\theta(s, a)$ is used as a surrogate reward to help guide the generated policy into a region close to the expert policy. Its value grows larger as samples generated from π_ψ look similar to those in expert data.

We use the same strategy to train a discriminator-generator network. However, one limitation of [Finn *et al.*, 2016b] is that, the use of full trajectories usually cause high variance. Instead, we follow [Finn *et al.*, 2016a] to use state-action pairs as input with a more straightforward conversion for the discriminator:

$$D(s, a) = \frac{\exp\{f(s, a)\}}{\exp\{f(s, a)\} + \pi(a|s)} \tag{3}$$

where f is a learned function, and policy π is trained to maximize reward $r(s, a) = \log(1 - D(s, a)) - \log D(s, a)$. In each iteration, the extracted reward serves as the learning cost for training the generator to update the policy. Updating the discriminator is equivalent to updating the reward function, and in turn updating the policy can be viewed as improving the sampling distribution used to estimate the discriminator. The structure of the discriminator in Eq. (3) can be viewed as a sigmoid function with input $f(s, a) - \log\pi$, as shown in Fig. 2. Similar to GAN, the discriminator is trained to reach the optimality when the expected $D(s, a)$ over all state-action pairs is $1/2$. At the optimality, $f^*(s, a) = \log\pi^*(a|s)$ and the reward can be extracted from the optimal discriminator by $r(s, a) = f(s, a) + Const.$

4.2 Dynamics-robust Reward Learning

We seek to learn the optimal policy for the vehicle agent from real-world demonstrations. So the learned policy should be close to, in the real world, the driver's policy to control the movement of vehicles given the state of traffic environment. Despite the non-stationary nature of traffic dynamics, the vehicle's objective during driving should constantly remain the same. This means that although the optimal policy under different system dynamics may vary, the vehicle constantly has the same reward function. Therefore, it is highly appealing to learn an underlying dynamic-robust reward function beneath the expert policy. By defining the traffic simulation learning as an IRL problem, we adapt the adversarial IRL (AIRL) as in Section 4.1 on the traffic simulation problem and propose a novel simulation model, which serves to not only imitate the expert trajectories (i.e., driving behaviors), but simultaneously recover the underlying reward function.

We aim to utilize the reward-invariant characteristics of AIRL to train an agent that simulates a vehicle's behavior in traffic while being robust to dynamics with changing traffic environment. The reward ambiguity is an essential issue in IRL approaches. [Ng *et al.*, 1999] proposes a class of reward transformations that preserve the optimal policy, for any function $\Phi : \mathcal{S} \longrightarrow \mathbb{R}$, we have

$$\hat{r}(s, a, s') \in \{\hat{r}|\hat{r} = r(s, a, s') + \gamma\Phi(s') - \Phi(s)\} \tag{4}$$

Algorithm 1: Training procedure of PS-AIRL

```
Input: Expert trajectories $\tau_E \sim \pi_E$
Output: Policy $\pi_\psi$, reward $r_{\theta,\phi}$
Initialize policy parameters $\psi$, and discriminator
  parameters $\theta$ and $\phi$
for $k \longleftarrow$ 0, 1, ... do
    Rollout trajectories for all $M$ agents
      $\vec{\tau} = \{\tau_1, \tau_2, ..., \tau_M\} \sim \pi_{\psi_k}$
    Update $\theta$, $\phi$ in discriminator given $\vec{\tau}$ via
      minimizing $\mathbb{E}_{\pi_\psi}[\log(D_{\theta,\phi}(s,a,s'))] +$
      $\mathbb{E}_{\pi_E}[\log(1 - D_{\theta,\phi}(s,a,s'))]$
    Update reward using discriminator output
      $r_{\theta,\phi} \longleftarrow \log D_{\theta,\phi} - \log(1 - D_{\theta,\phi})$
    Update policy $\pi_\psi$ with a TRPO step by solving the
      optimization in Eq. (7). Then $\pi_{\psi_{k+1}} \longleftarrow \pi_\psi$.
```

If the true reward is solely a function of state, we are able to extract a reward that is fully disentangled from dynamics (i.e., invariant to changing dynamics) and recover the ground truth reward up to a constant [Fu *et al.*, 2018].

To learn this *dynamic-robust reward* and ensure it remains in the set of rewards that correspond to the same optimal policy, AIRL follows the above reward transformation, and replaces the learned function $f(s,a)$ in Eq. (3) with

$$f_{\theta,\phi}(s,a,s') = \hat{r}_\theta(s) + \gamma h_\phi(s') - h_\phi(s) \tag{5}$$

where $\hat{r}_\theta(s)$ is the state-only reward approximator, h_ϕ is a shaping term, and $s' = \pi(s,a)$ is the next state. The transformation in Eq. (5) ensures that the learned function $f_{\theta,\phi}$ corresponds to the same optimal policy as the reward approximator $\hat{r}_\theta$ does. Correspondingly, $D_\theta(s,a)$ in Eq. (3) becomes

$$D_{\theta,\phi}(s,a,s') = \frac{\exp\{f_{\theta\phi}(s,a,s')\}}{\exp\{f_{\theta\phi}(s,a,s')\} + \pi(a|s)} \tag{6}$$

As a result, the discriminator can be represented by the combination of the two functions $\hat{r}_\theta$ and h_ϕ as shown in Fig. 2.

4.3 Parameter-sharing Agents

As illustrated in Section 3, we formulate the traffic simulation as multi-agent system problem by taking every vehicle in the traffic system as an agent interacting with the environment and each other. Inspired by parameter sharing trust region policy optimization (PS-TRPO) [Gupta *et al.*, 2017], we incorporate its decentralized parameter sharing training protocol with AIRL in Section 4.1, and propose the parameter sharing AIRL (PS-AIRL) that learns policies capable of simultaneously controlling multiple vehicles in complex traffic environment. In our formulation, the control is decentralized while the learning is not. Accordingly, we make some simple assumptions for the decentralized parameter sharing agents. See Appendix for details[1].

Under the decentralized parameter sharing training protocol as in [Gupta *et al.*, 2017], our proposed PS-AIRL can be highly sample-efficient since it reduces the number of parameters by a factor M, and shares experience across all agents in the environment. We use TRPO [Schulman *et al.*, 2015] as our policy optimizer, which allows precise control of the expected policy improvement during the optimization. For a policy π_ψ, at each iteration k, we perform an update to the policy parameters ψ by solving the following problem:

$$\begin{aligned} \min_{\psi} \quad & \mathbb{E}_{s,a\sim\pi_{\psi_k}, m\in\mathcal{M}}\left[\frac{\pi_\psi(a|s,m)}{\pi_{\psi_k}(a|s,m)} A_{\psi_k}(s,m,a)\right] \\ s.t., \quad & \mathbb{E}_{s\sim\pi_{\psi_k}}\left[D_{\mathrm{KL}}\left(\pi_{\psi_k} \| \pi_\psi\right)\right] \le \delta \end{aligned} \tag{7}$$

where π_{ψ_k} is the policy obtained in the previous iteration, $A_{\psi_k}(s,m,a)$ is an advantage function that can be estimated by the difference between the empirically predicted value of action and the baseline value. The training procedure of PS-AIRL is shown in Algorithm 1. The training pipeline details can be found in the appendix.

5 Experiment

We generally aim to answer the following three questions.

- **Q1**: *Trajectory simulation.* Is PS-AIRL better at imitating the vehicle movement in expert trajectories?
- **Q2**: *Reward recovery.* Can PS-AIRL learn the vehicle's objective by recovering the reward function?
- **Q3**: *Capability to generalize.* Is the recovered reward function robust to dynamics changes in the traffic environment?

5.1 Data and Experiment Setting

We evaluate our proposed model on 3 different real-world traffic trajectory datasets with distinct road network structures collected from Hangzhou of China, and Los Angeles of US, including 3 typical 4-way intersections, a 4×4 network, and a 1×4 arterial network. See Appendix for details.

We compared PS-AIRL to two traditional CFM-based methods, CFM-RS and CRF-TS [Krauß, 1998], and three imitation learning-based models, BC [Michie *et al.*, 1990], DeepIRL [Wulfmeier *et al.*, 2015] and MA-GAIL [Zheng *et al.*, 2020]. See Appendix for details.

We use root mean square error (RMSE) to evaluate the mismatch between ground truth traffic trajectories and the trajectories generated by the simulation model. In addition, to verify the capacity of PS-AIRL in recovering the underlying reward that can generalize to different dynamics, we compare the reward values each model reaches given a hand-crafted reward function. See Appendix for its definition.

5.2 Traffic Simulation

To answer Q1, we evaluate each model by using them to recover traffic trajectories, as shown in Table 1 and Figure 3.

We first compare our proposed algorithm with state-of-the-art simulation algorithms CFM-RS and CFM-TS (as in Table 1). We can observe that our proposed PS-AIRL outperforms the baseline methods on all datasets. The two traditional methods CFM-RS and CFM-TS perform poorly due to the generally inferior fitting ability of physical CFM models against sophisticated traffic environment. We further compare PS-AIRL with state-of-art imitation learning methods. The results are shown in Figure 3. We can observe that PS-AIRL performs consistently better than other baselines. Specifically, as expected, PS-AIRL and MA-GAIL perform better

[1] The appendix will be released on the authors' website and arxiv.

Method	HZ-1		HZ-2		HZ-3		GD		LA	
	Pos	Speed	Pos	Speed	Pos	Speed	Pos	Speed	Pos	Speed
CFM-RS	173.0	5.3	153.0	5.6	129.0	5.7	286.0	5.3	1280.9	10.3
CFM-TS	188.3	5.8	147.0	6.1	149.0	6.1	310.0	5.5	1294.7	10.8
PS-AIRL	**120.9**	**4.5**	**41.0**	**2.1**	**10.7**	**1.1**	**45.6**	**1.0**	**681.5**	**5.1**
Improvement	30.1%	15.1%	72.1%	47.6%	91.7%	80.7%	84.1%	81.1%	46.8%	50.5%

Table 1: Performance comparison of PS-AIRL with state-of-art methods in terms of RMSE of recovered vehicle position (m) and speed (m/s).

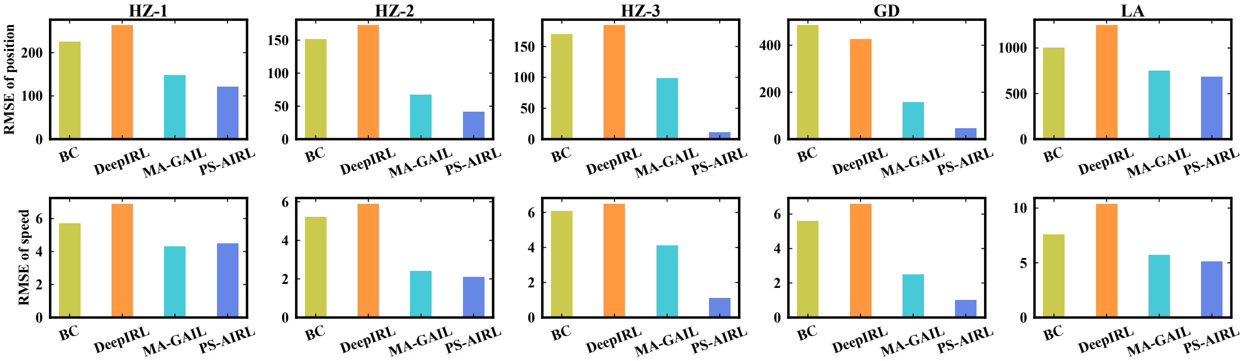

Figure 3: Performance comparison of variants of learning methods in terms of RMSE of position. PS-AIRL beats all alternatives.

than the other two, for they can not only model the sequential decision process (compared to BC) and utilize the adversarial process to improve the imitation (compared to DeepIRL). More importantly, PS-AIRL achieves superior performance over MA-GAIL (except in HZ-1 these two perform similarly in terms of RMSE of speed). It indicates that our proposed model benefits from the learned invariant reward. For instance, if one vehicle enter the system right after another, we may not expect it to accelerate a lot. In contrast, if one vehicle enter the system with no preceding vehicle, it may speed up freely within the speed limit. Behaviors may seem different under these two cases, but the reward function still remains the same. Therefore, armed with explicit reward learning, PS-AIRL is able to outperform.

Simulation Case Study. To have a closer look at the simulation of vehicle behavior, we extract the learned trajectory of 5 vehicles (vehicles with id 20, 40, 60, 80, 100) from HZ-3 as examples. Comparison between trajectories generated by different methods is shown in Fig. 4. The trajectories recovered by PS-AIRL are most similar to the expert ones.

Method	HZ-1	HZ-2	HZ-3	GD
BC	-0.345	-0.277	-0.248	-0.349
DeepIRL	-0.586	-0.238	-0.311	-0.360
MA-GAIL	-0.524	-0.210	-0.228	-0.181
PS-AIRL	**-0.173**	**-0.203**	**-0.201**	**-0.161**
Expert	-0.060	-0.052	-0.062	-0.065

Table 2: Achieved reward of different imitation learning methods. Higher reward indicates better imitation of the expert.

5.3 Objective Recovery

To answer Q2, we leverage a hand-designed reward function as the ground truth and let the algorithms recover it. The reward increases when the vehicle approaches its desired speed and succeed in keeping safe distance from its preceding vehicles (see Appendix for details). Specifically, we use this hand-designed reward function to train an expert policy and use the generated expert trajectories to train each method. Note that, CFM-based methods are not included here for the following reasons. CFM-based methods can yield very unreal parameters (e.g., acceleration of 10 m/s^2) to match the generated trajectory. These will cause collision of vehicles or even one vehicle directly pass through another. This is an unreal setting. Hence, these CFM-based methods can not recover the true policy that vehicles follow in the simulation.

Achieved Reward Value. We conduct experiments on HZ-1, HZ-2, HZ-3 and GD, and the results are shown in Table 2. As expected, PS-AIRL achieves higher reward than other baselines and approach much closer to the upper bound reward value provided by the Expert. This indicates the better imitation (i.e. simulation) power of PS-AIRL.

Recovered Reward Function. We are also interested in whether PS-AIRL can recover the groundtruth reward function. Hence, we visualize the reward function learned by PS-AIRL and the surrogate reward of MA-GAIL (the best baseline). Due to the space limit, results from other methods are omitted. They generally show even worse results than MA-GAIL. We enumerate the reward values for different speed and gap values. (Other state feature values are set to default values.) As in Fig. 5, compared to MA-GAIL, PS-AIRL recovers a reward closer to the groundtruth yet with a smoother shape. Our designed reward function applies penalties on slow speed and unsafe small gap. So the expert policy trained with this groundtruth reward tends to drive the vehicle on a faster speed and keep it against the front vehicle out of the

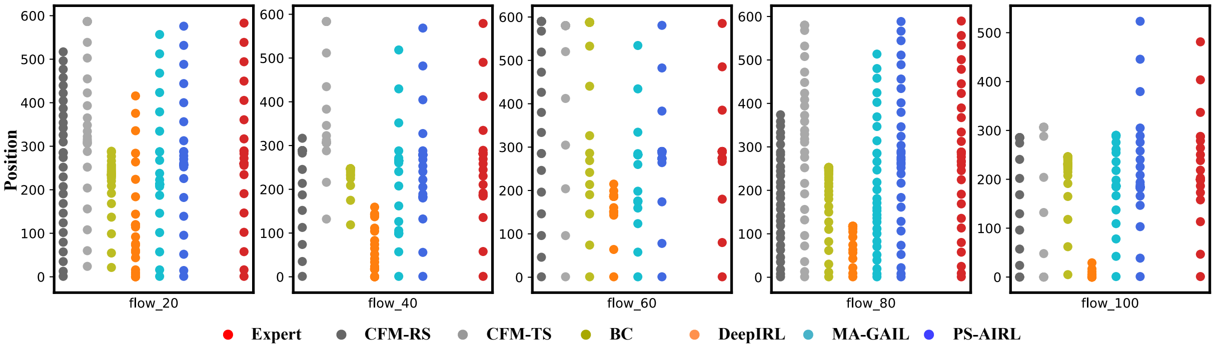

Figure 4: Trajectories of vehicles recovered by different methods on HZ-3. We arbitrarily select the vehicles with id 20, 40, 60, 80 and 100 and show their position w.r.t time. The position of vehicles recovered by PS-AIRL (blue) is most similar to the Expert (red).

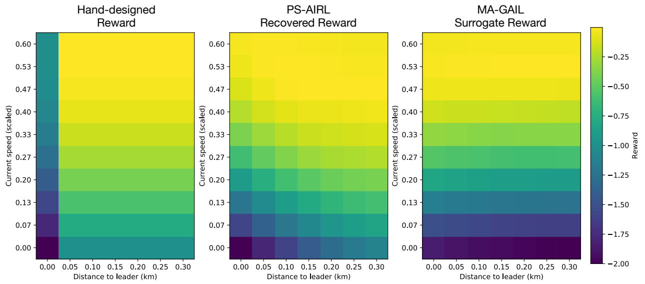

Figure 5: Hand-designed reward (groundtruth), recovered reward of PS-AIRL and surrogate reward (output of discriminator) of MA-GAIL w.r.t. current speed and gap, on HZ-1.

safety gap. It shows that PS-AIRL is able to learn this objective of the expert in choosing optimal policies from demonstrations, assigning higher reward values to states with faster speed and safe gap, while applying penalties on the opposite. In contrast, the surrogate reward learned by the MA-GAIL totally ignores the safe gap. In addition, when vehicles are out of the safe gap, PS-AIRL recovered a reward value closer to the groudtruth.

5.4 Robustness to Dynamics

To answer Q3, we transfer the model learned in Section 5.3 to a new system dynamics. Specifically, compared with Section 5.3, we change the maximum acceleration and deceleration of vehicles from initially 2 m/s^2 and 4 m/s^2 into 5 m/s^2 and 5 m/s^2 respectively. Other settings remain unchanged. Because the reward function remains unchanged (i.e., drivers still want to drive as fast as possible but keep a safe gap), reward-learning methods should be able to generalize well.

For DeepIRL, MA-GAIL and PS-AIRL, we use the learned reward function (surrogate reward function for MA-GAIL) to train a new policy. (BC policy is directly transferred because it do not learn reward function.) We also include a direct transfer version of PS-AIRL as comparison to demonstrate the necessity of re-training policy under new dynamics. Similar to the reward recovery experiment, CFM models also do not apply to this comparison.

Table 3 shows the achieved reward in the new environment.

Method	HZ-1	HZ-2	HZ-3	GD
BC ($\mathcal{D}$)	-0.406	-0.322	-0.336	-0.681
DeepIRL	-0.586	-0.288	-0.362	-0.369
MA-GAIL	-0.483	-0.243	-0.337	-0.175
PS-AIRL ($\mathcal{D}$)	-0.448	-0.254	-0.321	-0.598
PS-AIRL	**-0.384**	**-0.192**	**-0.248**	**-0.161**

Table 3: Reward under new dynamics. BC ($\mathcal{D}$) and PS-AIRL ($\mathcal{D}$) means directly applying the learned policy. The other methods use the learned reward to retrain a policy under new dynamics.

PS-AIRL achieves highest reward, because with the invariant reward function learned from the training environment, PS-AIRL can learn a new policy that adapts to the changed environment. DeepIRL perform poorly in transferred environment because of its incapability of dealing with large number of states. PS-AIRL ($\mathcal{D}$) fails to generalize to the new environment, which indicates that the policy learned in the old dynamics are not applicable anymore. MA-GAIL, though retrained with the surrogate reward function, can not transfer well, due to the dependency on the system dynamics.

6 Conclusion

In this paper, we formulated the traffic simulation problem as a multi-agent inverse reinforcement learning problem, and proposed PS-AIRL that directly learns the policy and reward function from demonstrations. Different from traditional methods, PS-AIRL does not need any prior knowledge in advance. It infers the vehicle's true objective and a new policy under new traffic dynamics, which enables us to build a dynamics-robust traffic simulation framework with PS-AIRL. Extensive experiments on both synthetic and real-world datasets show the superior performances of PS-AIRL on imitation learning tasks over the baselines and its better generalization ability to variant system dynamics.

Acknowledgments

This work was supported in part by NSF awards #1652525, #1618448 and #1639150. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[Barceló and Casas, 2005] Jaime Barceló and Jordi Casas. Dynamic network simulation with aimsun. In *Simulation approaches in transportation analysis*, pages 57–98. Springer, 2005.

[Bhattacharyya *et al.*, 2018] Raunak P Bhattacharyya, Derek J Phillips, Blake Wulfe, Jeremy Morton, Alex Kuefler, and Mykel J Kochenderfer. Multi-agent imitation learning for driving simulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1534–1539. IEEE, 2018.

[Finn *et al.*, 2016a] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[Finn *et al.*, 2016b] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning (ICML)*, pages 49–58, 2016.

[Fu *et al.*, 2018] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2018.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[Gupta *et al.*, 2017] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer, 2017.

[Ho and Ermon, 2016] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4565–4573, 2016.

[Krajzewicz *et al.*, 2012] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[Krauß, 1998] Stefan Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, Dt. Zentrum für Luft-und Raumfahrt eV, Abt., 1998.

[Michie *et al.*, 1990] D Michie, M Bain, and J Hayes-Miches. Cognitive models from subcognitive skills. *IEEE control engineering series*, 44:71–99, 1990.

[Ng *et al.*, 1999] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[Osorio and Punzo, 2019] Carolina Osorio and Vincenzo Punzo. Efficient calibration of microscopic car-following models for large-scale stochastic network simulators. *Transportation Research Part B: Methodological*, 119:156–173, 2019.

[Schulman *et al.*, 2015] John Schulman, Sergey Levine, Philipp Moritz, Michael I Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the International Conference on Machine learning (ICML)*, 2015.

[Toledo *et al.*, 2003] Tomer Toledo, Haris N Koutsopoulos, Angus Davol, Moshe E Ben-Akiva, Wilco Burghout, Ingmar Andréasson, Tobias Johansson, and Christen Lundin. Calibration and validation of microscopic traffic simulation tools: Stockholm case study. *Transportation Research Record*, 1831(1):65–75, 2003.

[Wang *et al.*, 2018] Sen Wang, Daoyuan Jia, and Xinshuo Weng. Deep reinforcement learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[Wei *et al.*, 2018] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2496–2505. ACM, 2018.

[Wulfmeier *et al.*, 2015] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

[Yang and Koutsopoulos, 1996] Qi Yang and Haris N Koutsopoulos. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C: Emerging Technologies*, 4(3):113–129, 1996.

[Yu and Fan, 2017] Miao Yu and Wei David Fan. Calibration of microscopic traffic simulation models using metaheuristic algorithms. *International Journal of Transportation Science and Technology*, 6(1):63–77, 2017.

[Zheng *et al.*, 2020] G. Zheng, H. Liu, K. Xu, and Z. Li. Learning to simulate vehicle trajectories from demonstrations. In *ICDE*, pages 1822–1825, 2020.