LASIL: Learner-Aware Supervised Imitation Learning For Long-term Microscopic Traffic Simulation

Ke Guo[1,2,3], Zhenwei Miao[1]*, Wei Jing[1], Weiwei Liu[1], Weizi Li[4], Dayang Hao[1], Jia Pan[2,3]
[1]Alibaba Group [2]The University of Hong Kong
[3]Centre for Transformative Garment Production, Hong Kong [4]University of Tennessee, Knoxville
u3006612@connect.hku.hk, zhenwei.mzw@alibaba-inc.com
21wjing@gmail.com, 11932061@zju.edu.cn, weizili@utk.edu, jpan@cs.hku.hk

Abstract

Microscopic traffic simulation plays a crucial role in transportation engineering by providing insights into individual vehicle behavior and overall traffic flow. However, creating a realistic simulator that accurately replicates human driving behaviors in various traffic conditions presents significant challenges. Traditional simulators relying on heuristic models often fail to deliver accurate simulations due to the complexity of real-world traffic environments. Due to the covariate shift issue, existing imitation learning-based simulators often fail to generate stable long-term simulations. In this paper, we propose a novel approach called learner-aware supervised imitation learning to address the covariate shift problem in multi-agent imitation learning. By leveraging a variational autoencoder simultaneously modeling the expert and learner state distribution, our approach augments expert states such that the augmented state is aware of learner state distribution. Our method, applied to urban traffic simulation, demonstrates significant improvements over existing state-of-the-art baselines in both short-term microscopic and long-term macroscopic realism when evaluated on the real-world dataset pNEUMA.

1. Introduction

Microscopic traffic simulation is a cornerstone in transportation engineering. It enables engineers to predict and analyze individual vehicle behavior, providing crucial insights into how alterations in road structures or traffic management strategies might influence overall traffic flow; it allows for the testing of diverse scenarios without disrupting real-world traffic; and it enhances safety by pinpointing potential hazards and devising strategies to mitigate risks. By leveraging simulations, engineers can optimize traffic flow, reduce congestion, and enhance overall efficiency, proving particularly advantageous in designing or enhancing road networks. Using the simulation data, policymakers and urban planners can make informed decisions aligned with community needs regarding transportation infrastructure.

However, generating realistic and accurate simulations that can simultaneously replicate the microscopic response of human drivers in various traffic conditions and long-term macroscopic traffic statistics is challenging. In recent years, significant efforts have been invested in developing realistic traffic simulators with the goal to accurately model human driving behaviors. Traditional traffic simulators, such as SUMO [24], AIMSUN [2], and MITSIM [44], typically rely on heuristic car-following models like the Intelligent Driver Model (IDM) [42]. However, despite carefully calibrating parameters, the rule-based models often fail to deliver accurate simulations due to the complexity of real-world traffic environments [12, 30]. Factors such as the road structure, neighboring vehicles, and even driver psychology can influence the decisions of human drivers, making it challenging to achieve accurate simulations [7, 27, 43].

In pursuit of realistic traffic simulations, researchers have turned to neural networks to represent the driving model through imitation learning (IL) from human demonstrations. Most traffic simulation approaches [4, 34, 41] leverage behavior cloning (BC) [32] to learn a driving policy by minimizing the disparity between the model output and the human demonstrations in training data. However, BC is hindered by *covariate shift* [37], where the state induced by the learner's policy progressively diverges from the expert's distribution. Existing BC-based simulators have succeeded in short-term (less than 20 seconds) simulation applications like autonomous driving tests but often fail to generate stable long-term simulations.

To address covariate shift, existing methods such as DAgger [37], DART [25], and ADAPS [28] incorporate supervisor (humans or principled simulations) corrections

*Zhenwei Miao is the corresponding author.

at the learner's or perturbed expert's states. However, human supervision can be problematic due to intensive labor and judgment errors, and principled simulations may not account for heterogeneous driving behaviors. Recent traffic simulators [40, 46] propose using inverse reinforcement learning (IRL). These methods, such as generative adversarial imitation learning (GAIL) [17] and adversarial inverse reinforcement learning (AIRL) [13], learn a reward function using a discriminator neural network within Generative Adversarial Networks (GANs) [14]. The policy network is trained to maximize the learned reward through online reinforcement learning (RL), enabling agents to handle out-of-distribution states. However, directly applying GAIL to traffic simulation can be problematic [5]. The dynamic nature of the environment in the multi-agent system can lead to noises during policy learning, resulting in highly biased estimated gradients. Furthermore, training the discriminator in GAIL is challenging due to the instability and sensitivity of hyperparameters during min-max optimization.

To address the issue of *covariate shift* in multi-agent imitation learning without depending on costly expert supervision or unstable discriminators and multi-agent RL, we propose **Learner-Aware Supervised Imitation Learning** (LASIL) for tackling the covariate shift problem during policy learning. We mitigate the distribution shift between expert and learner state distribution by augmenting the expert state distribution. However, there is no expert supervision at any augmented state, so we ensure that an augmented state is close to an expert state such that the future trajectory of the expert state can serve as the target trajectory for the augmented state to constrain the learner within the expert state distribution. Hence, our goal is to augment the expert state to cover the learner's state distribution while remaining close to the original expert state distribution. To achieve this, we use a variational autoencoder (VAE) [21] to simultaneously model the distributions of both the expert and learner states. By minimizing the VAE's latent space regularization loss of modeling both distributions, we project the expert and learner states into a unified latent space. And by minimizing the VAE's reconstruction loss, the resulting reconstruction leveraging such learnt latent space will resemble both expert and learner state distribution. As a result, when inputting expert states into the trained VAE, we obtain a **learner-aware augmented expert state**.

In practice, we divide an agent's state into two parts: past trajectory and context, i.e., other features in the state including vehicle type, waypoints, and destination. We observe that the distribution of the context is consistent regardless of a learnt policy due to the static features of traffic conditions embedded in the context, leading to less covariate shift. Therefore, we propose a **context-conditioned VAE** to model the context-conditioned trajectory distribution of both expert and learner states. The decoder of the context-conditioned VAE will receive both the latent variable and the context, and yield only the trajectory information.

In summary, our contributions are as follows:

- We propose learner-aware supervised imitation learning (LASIL) to achieve stable learning and alleviate covariate shift in multi-agent imitation learning.
- We propose a learner-aware data augmentation method based on a context-conditioned VAE that generates learner-aware augmented expert states.
- Our approach is tailored for urban traffic simulation. To the best of our knowledge, it is the first imitation learning-based traffic simulator that can reproduce long-term (more than 10 minutes) stable microscopic simulation, achieving 40x simulation length improvements over previous state-of-the-art [4, 41, 46].
- We evaluate our method on the real-world dataset pNEUMA [3] with over half a million trajectories. Our approach outperforms state-of-the-art baselines in both short-term microscopic and long-term macroscopic simulations. The code is available at https://github.com/Kguo-cs/LSAIL.

2. Related Work

2.1. Imitation learning

Existing imitation learning (IL) methods can be broadly categorized into behavior cloning (BC) and inverse reinforcement learning (IRL) approaches. BC [32] learns a policy in a supervised fashion by minimizing the discrepancy between the learner's actions and those of an expert. However, BC suffers from the issue of covariate shift, where the state distribution induced by the learner's policy gradually deviates from that of the expert. To address this, methods like DAgger [37] and DART [25] request supervisor corrections at the learner's or perturbed expert's states. Our method follows a similar supervised learning approach to DAgger and DART, but does not require access to an expert policy.

Due to the challenges in obtaining expert supervision, recent IRL-based methods utilize feedback from a neural network-based discriminator to handle covariate shift. Typically, these methods involve an iterative process alternating between reward estimation and reinforcement learning. Earlier IRL-based methods [15, 35, 39, 47] require frequent dynamic programming processes, while recent adversarial IL approaches integrate reward function learning with policy learning using a GAN formulation. However, both GAN and RL training processes are known to be unstable, sensitive to hyperparameters, and have poor sample efficiency [9]. Moreover, the discriminator can easily exploit insignificant differences between expert and policy samples, leading to undesirable performance [45]. In contrast, our method avoids the min-max optimization problem and the sample-inefficient RL process, requiring minimal

fine-tuning. Instead, we utilize the real future trajectory as the target state to learn corrective behavior.

2.2. Imitation Learning-based Traffic simulation

Recent traffic simulators focus on enhancing realism by leveraging imitation learning (IL) from human driving demonstrations, which extend traditional BC and IRL methods to tackle the challenging multi-agent IL problem.

BC-based traffic simulators like TrafficSim [41] and SimNet [4] typically begin by training a prediction model and subsequently adjust the predicted trajectories to prevent collisions and adhere to traffic regulations during simulation. However, BC-based methods face challenges in achieving long-term simulation due to the covariate shift problem. To mitigate this issue, we augment expert state based on the policy distribution, enabling stable long-term simulation. Additionally, we enhance performance by modifying the predicted trajectory during simulation through road projection and ensuring smoothness from the current state. Notably, we skip the computationally intensive collision removal operation, prioritizing the capture of macroscopic long-term influence over micro-level details.

IRL-based simulators learn the underlying reward function of human driving behavior and derive the driving policy by maximizing the learned reward. While adversarial IRL methods theoretically address the covariate shift of BC in a single-agent context through online interaction, its performance deteriorates in the multi-agent IL domain due to the dynamic environment, complicating the training process. To tackle this challenge, approaches like PS-GAIL [40] and PS-ARIL [46] adopt two-stage learning and gradually introduce vehicles to the environment. Nonetheless, these methods still exhibit significant undesirable traffic phenomena, such as off-road driving, collisions, and abrupt braking. Building upon PS-GAIL, the reward-augmented imitation learning (RAIL) method [5, 22] penalizes undesirable phenomena by introducing a hand-crafted reward. However, maximizing the new reward does not guarantee the recovery of human-like trajectories. Despite numerous enhancements to the original GAIL framework, these IRL-based methods often struggle to produce stable long-term traffic flow, as evidenced in our experiments. In contrast, our method is a supervised learning approach, leading to faster, simpler, and more stable learning of driving policies.

3. Background

3.1. Markov Decision Process

We model the human driving process using a Markov decision process (MDP) denoted as $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, r\}$, incorporating a time horizon of T, where $\mathcal{S}$ and $\mathcal{A}$ represent the continuous state and action spaces respectively. A stochastic function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ describes the system dynamics, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function. The policy $\pi(\boldsymbol{a}|\boldsymbol{s})$ determines the probability of selecting an action a at a state s, and a trajectory $\tau = (\boldsymbol{s}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{s}_T, \boldsymbol{a}_T)$ represents a sequence of state-action pairs. The marginal state distribution of a policy π is computed as $\rho^\pi(\boldsymbol{s}) = \frac{1}{D}\sum_{\boldsymbol{s}\in\mathcal{D}_\pi} \delta_{\boldsymbol{s}}$, where $\mathcal{D}_\pi$ denotes the set of states of size D in trajectories induced by the policy π, and $\delta_{\boldsymbol{s}}$ signifies a Dirac distribution centered on $\boldsymbol{s}$. Similarly, the marginal state-action distribution $\rho^\pi(\boldsymbol{s}, \boldsymbol{a})$ is computed.

3.2. Imitation Learning

In the application of IL for learning a driving policy, we assume that all agents adopt the same policy. The objective is to learn a policy π that minimizes the f-divergence between the marginal state-action distribution of the expert's demonstrations $\rho^{\exp}(\boldsymbol{s}, \boldsymbol{a})$ and the learner policy's distribution $\rho^\pi(\boldsymbol{s}, \boldsymbol{a})$. For example, BC optimizes the policy to minimize the Kullback-Leibler (KL) divergence at the expert state distribution $\mathbb{E}_{\rho^{\exp}(\boldsymbol{s})}[\mathrm{KL}(\pi^{\exp}(\boldsymbol{a}|\boldsymbol{s})\|\pi(\boldsymbol{a}|\boldsymbol{s}))]$, while DAgger minimizes $\mathbb{E}_{\rho^\pi(\boldsymbol{s})}[\mathrm{KL}(\pi^{\exp}(\boldsymbol{a}|\boldsymbol{s})\|\pi(\boldsymbol{a}|\boldsymbol{s}))]$. On the other hand, GAIL minimizes the Jensen-Shannon divergence $D_{\mathrm{JS}}(\rho^{\exp}(\boldsymbol{s}, \boldsymbol{a})\|\rho^\pi(\boldsymbol{s}, \boldsymbol{a}))$, while AIRL minimizes the KL divergence $\mathrm{KL}(\rho^\pi(\boldsymbol{s}, \boldsymbol{a})\|\rho^{\exp}(\boldsymbol{s}, \boldsymbol{a}))$.

However, GAIL and AIRL often exhibit unsatisfactory practical performance due to their optimization processes' instability and sample inefficiency, which involve GANs and RL. In contrast, BC only requires simple and stable supervised learning but suffers from the covariate shift issue, as it only minimizes the policy difference at the expert state distribution without guaranteeing performance at the learner state distribution. DAgger addresses the covariate shift problem but requires access to the expert policy. To mitigate the covariate shift problem without depending on the expert policy, we propose maximizing the transition probability $\mathbb{E}_{\boldsymbol{s}\sim\rho^{\exp}(\boldsymbol{s}),\boldsymbol{s}+\boldsymbol{\epsilon}\sim\rho^\pi(\boldsymbol{s})}[\mathcal{T}(\boldsymbol{s}+\boldsymbol{\epsilon}, \pi(\boldsymbol{s}+\boldsymbol{\epsilon}), \boldsymbol{s}')]$, where $\boldsymbol{\epsilon}$ is a small augmentation term. Note that our policy learns to predict the distribution of the next state $\boldsymbol{s}'$ instead of the action. Our approach assumes that when the expert's original future trajectory can serve as supervision to guide the agent back towards the expert distribution.

3.3. Variational Autoencoder

The VAE defines a generative model given by $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p_\theta(\boldsymbol{x}|\boldsymbol{z})$, where $\boldsymbol{z}$ is the latent variable with prior distribution $p(\boldsymbol{z})$, and $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ represents the conditional distribution modeling the likelihood of data $\boldsymbol{x}$ given $\boldsymbol{z}$. The learning objective is to maximize the training samples' marginal log-likelihood $\log p_\theta(\boldsymbol{x})$. However, due to the intractability of marginalization, VAE maximizes the variational lower

bound using $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ as the approximate posterior:

$$\begin{aligned}&\log p_\theta(\boldsymbol{x})\\&\geq \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] - \mathrm{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})\right) \quad (1)\\&:= -\mathcal{L}_{\mathrm{rec}}(\boldsymbol{x}) - \mathcal{L}_{\mathrm{KL}}(\boldsymbol{x}),\end{aligned}$$

where $\mathcal{L}_{\mathrm{rec}}(\boldsymbol{x})$ represents the reconstruction loss, which penalizes the network for creating outputs different from the input. $\mathcal{L}_{\mathrm{KL}}(\boldsymbol{x})$ represents the KL divergence loss to make a continuous and smooth latent space, allowing easy random sampling and interpolation. Intuitively, this KL loss encourages the encoder to distribute all encodings evenly around the center of the latent space.

4. Method

In Fig. 1, we present an overview of our method. Our model comprises three modules: a VAE-based data augmentation module, a policy network, and a post-processing (LQR and on-road projection) module. During training, we augment each expert data by generating a learner-aware augmented expert state through the VAE's reconstruction of its past trajectory. Using this augmented state along with the original future trajectory, we train the learner's policy network through supervised learning. During simulation, we roll out the policy network with several post-processing steps including sampling, projecting the trajectories onto the road, and smoothing the projected trajectories.

4.1. State Representation

As human drivers make decisions mainly depending on their surrounding information, we build a graph to model the traffic system with each driving agent as its node.

Node: The state of each agent can be divided into two components: the past trajectory $\boldsymbol{s}_p$ and the context $\boldsymbol{s}_c$. In practice, the past trajectory is composed of the agent's positions at several past time steps, including the current time step. The context includes its type, nearby waypoints' positions and corresponding road width, the traffic light status of the road it is traveling on, and destination. The waypoints are composed of the center points of the routing roads with a fixed interval. We transform each agent's state into its individual coordinate system to learn a policy with transformation invariance. To reduce the implicit covariate shift, we set each agent's current position adding a Gaussian perturbation as the origin and the x-axis directions pointing towards the agent's destination like [16].

Edge: When each agent's state coordinates are transformed from the global coordinate system to their individual coordinate system, information about the relative positions among agents is lost. However, a traffic model needs the relative information of agents to understand how they interact with each other. To preserve these relationships, we introduce directed edges between neighboring agents. The edge feature is the relative position of the destination node's coordinate origin in the source node's coordinate system.

4.2. Learner-aware Data Augmentation

To address the covariate shift issue between the expert and learner state distributions, we propose to minimize the expert's and learner's embedded state distribution difference. In practice, we propose utilizing the same VAE to model the expert and learner state distributions simultaneously. As both the expert and learner state are projected to the same latent space, the distribution of the state reconstructed from the joint latent space can resemble both distributions.

In contrast to the past trajectory, the context distribution is more challenging to model but exhibits less covariate shift. Therefore, we propose using a context-conditioned VAE to specifically model the context-conditioned trajectory distribution rather than the state distribution. For each expert or learner state represented as $\boldsymbol{s} = (\boldsymbol{s}_p, \boldsymbol{s}_c)$, we employ an encoder $q_\phi(\boldsymbol{z}|\boldsymbol{s}_p, \boldsymbol{s}_c)$ to obtain the latent variable distribution. We can sample a latent variable $\boldsymbol{z}$ from this distribution by applying the reparameterization trick. Subsequently, a decoder $p_\theta(\boldsymbol{s}_p|\boldsymbol{z}, \boldsymbol{s}_c)$ is used to reconstruct the distribution of the past trajectory $\boldsymbol{s}_p$ given $\boldsymbol{z}$ and $\boldsymbol{s}_c$. The VAE is trained to maximize the variational lower bound, which incorporates the context-conditioned log-likelihood of both the expert and learner past trajectories:

$$\mathcal{L}_{VAE} = \mathbb{E}_{\boldsymbol{s}\sim\rho^{\exp}}\left[\mathcal{L}(\boldsymbol{s}_p|\boldsymbol{s}_c)\right] + \lambda\mathbb{E}_{\boldsymbol{s}\sim\rho^{\pi}}\left[\mathcal{L}(\boldsymbol{s}_p|\boldsymbol{s}_c)\right]. \quad (2)$$

Here, λ is a hyperparameter that controls the degree to which the augmented context-conditioned past trajectory distribution aligns with the learner's context-conditioned past trajectory distribution. The term $\mathcal{L}(\boldsymbol{s}_p|\boldsymbol{s}_c)$ is given by:

$$\begin{aligned}\mathcal{L}(\boldsymbol{s}_p|\boldsymbol{s}_c) =& \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{s}_p,\boldsymbol{s}_c)}\left[\log p_\theta(\boldsymbol{s}_p|\boldsymbol{z},\boldsymbol{s}_c)\right]\\&- \mathrm{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{s}_p,\boldsymbol{s}_c)\|p(\boldsymbol{z})\right).\end{aligned} \quad (3)$$

For simplicity, we assume p_θ and q_ϕ as multivariate normal distributions with diagonal variance matrices. The prior distribution $p(\boldsymbol{z})$ is set as an isotropic unit Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

4.3. Edge-enhanced Graph Attention Network

We build all models in our approach including the VAE encoder, decoder, and learner's policy network based on an edge-enhanced graph attention (EGAT) network [11, 33], which model interactions by aggregating neighboring node and edge information using an attention mechanism. The node features in the first layer are obtained by embedding the node input with a fully connected layer and the node features in the other layers are calculated by:

$$\boldsymbol{h}_i^l = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^l\boldsymbol{W}^l\left[\boldsymbol{h}_i^{(l-1)}\|\boldsymbol{e}_{ij}\|\boldsymbol{h}_j^{(l-1)}\right]\right), \quad (4)$$

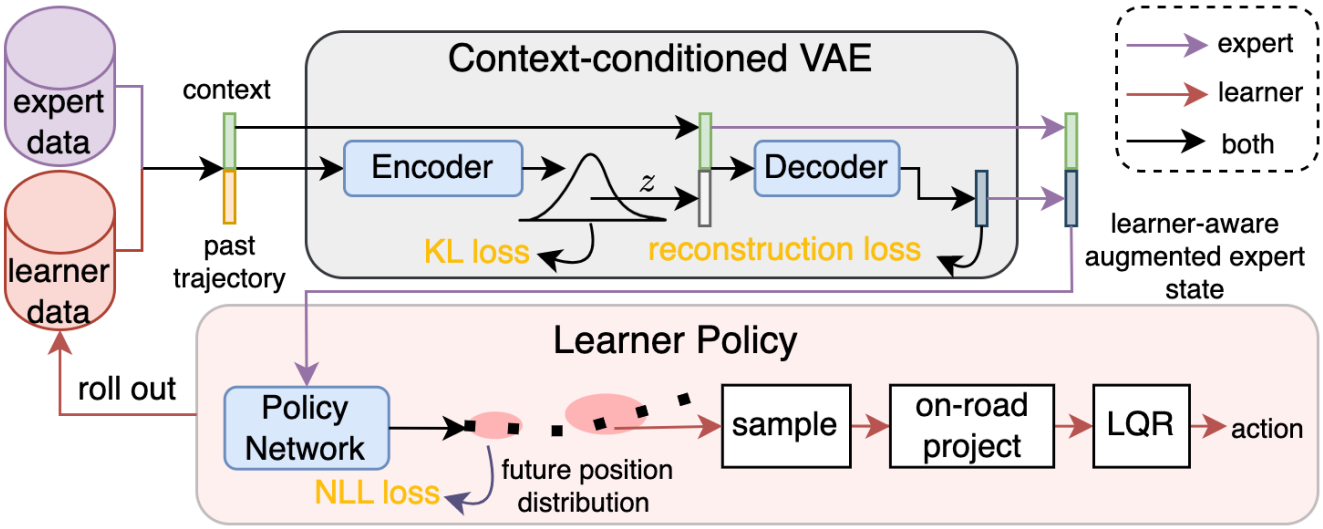

Figure 1. Overview of our approach. The processes with purple and red arrows handle only expert or learner data, respectively, while the processes represented by black arrows apply to both data. Each state is a multi-agent state represented by a graph. Each model, including the VAE encoder, decoder, and policy network, is implemented using an EGAT network.

where $\boldsymbol{h}_i^l$ is the feature of node i in the l th layer, $\boldsymbol{e}_{ij}$ is the coordinate origin's position of node j relative to node i, $\|\|$ means the concatenation, $\boldsymbol{W}^l$ is the learnable weight matrix, $\mathcal{N}_i$ is the set of the first-order neighbors of node i (including the node itself), and σ is a non-linear activation function. The attention coefficient α_{ij}^l indicates the importance of node j to node i, considering both node and edge features, and is computed as:

$$\alpha_{ij}^l = \mathrm{softmax}_j\left(\sigma\left((\boldsymbol{w}^l)^T\left[\boldsymbol{h}_i^{(l-1)}\|\boldsymbol{e}_{ij}\|\boldsymbol{h}_j^{(l-1)}\right]\right)\right), \quad (5)$$

where $\boldsymbol{w}^l$ is a learnable weight vector, and normalization is performed on the weights across all neighbors of node i using a $\mathrm{softmax}$ function. After passing through multiple EGAT layers, the node features at the last layer is fed into a fully connected layer to obtain the outputs of the network.

4.4. Policy Loss

To train the learner's policy network, we first sample a **learner-aware augmented expert state** from the **context-conditioned VAE** for each expert state. Then, the policy network predicts its future position distribution over T time steps, denoted as $p(\hat{\boldsymbol{p}}_1^m, \hat{\boldsymbol{p}}_2^m, ..., \hat{\boldsymbol{p}}_T^m)$, which is assumed to be a product of multi-variable Gaussian distributions:

$$p(\hat{\boldsymbol{p}}_1^m, \hat{\boldsymbol{p}}_2^m, ..., \hat{\boldsymbol{p}}_T^m) = \prod_{t=1}^{T} \mathcal{N}(\hat{\boldsymbol{\mu}}_t^m, \hat{\boldsymbol{\Sigma}}_t^m), \quad (6)$$

where $\hat{\boldsymbol{\mu}}_t^m$ and $\hat{\boldsymbol{\Sigma}}_t^m$ represent the mean and covariance matrix of the predicted position $\hat{\boldsymbol{p}}_t^m$ at future time step t. We assume that there is no correlation between the position distributions at different future time steps. To learn the policy network, we minimize the negative log-likelihood (NLL) loss of all agents' ground-truth future trajectories:

$$\mathcal{L}_{NLL} = -\sum_{m=1}^{M}\sum_{t=1}^{T} \log(\mathcal{N}(\boldsymbol{p}_t^m - \hat{\boldsymbol{\mu}}_t^m, \hat{\boldsymbol{\Sigma}}_t^m)), \quad (7)$$

where $\boldsymbol{p}_t^m$ denotes the ground truth position of agent m at future time step t, and M is the total number of agents.

4.5. Simulation Process

During training, we simultaneously simulate (roll out) the learned policy network iteratively to get learner state samples. Instead of directly updating each agent to its predicted position, we apply several post-processing steps to the prediction for better realism. Firstly, we sample from the distribution, and then project each sampled position onto the nearest on-road point. Then, we smooth the projected trajectory with a linear-quadratic regulator (LQR) [1] by minimizing the total commutative quadratic cost of a linear dynamic system described by:

$$\begin{bmatrix} \tilde{\boldsymbol{p}}_{t+1}^m \\ \tilde{\boldsymbol{v}}_{t+1}^m \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{D} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{p}}_t^m \\ \tilde{\boldsymbol{v}}_t^m \end{bmatrix} + \begin{bmatrix} \boldsymbol{D}^2 \\ \boldsymbol{D} \end{bmatrix} \tilde{\boldsymbol{a}}_t^m, \quad (8)$$

where $\boldsymbol{D}$ is a diagonal matrix with the interval of each time step as diagonal entries, and $\tilde{\boldsymbol{p}}_t^m$, $\tilde{\boldsymbol{v}}_t^m, \tilde{\boldsymbol{a}}_t^m$ represent the LQR-planned position, velocity, and acceleration. The system is subject to a quadratic cost function given by:

$$\mathcal{J} = \sum_{m=1}^{M}\sum_{t=1}^{T} \|\tilde{\boldsymbol{p}}_t^m - \bar{\boldsymbol{p}}_t^m\|^2 + \eta_{\boldsymbol{a}}\|\tilde{\boldsymbol{a}}_t^m\|^2, \quad (9)$$

where the projected predicted position $\bar{\boldsymbol{p}}_m^t$ is considered as the target pose, and the hyper-parameter $\eta_{\boldsymbol{a}}$ is used to penalize high acceleration. Finally, each agent is updated to the first position of the planned trajectory.

4.6. Training Process

At each training step, the policy network is trained to maximize the probability of the expert's future trajectory using its context state and history trajectory augmented by the VAE as input. Simultaneously, the VAE is iteratively trained to reconstruct the expert and learner data. The expert data are the same as the policy network's training data before augmentation. The learner data is sampled from a replay buffer. Every N training steps, we empty the replay buffer and roll out the current policy with post-processing for S time steps to generate the learner data and store it in the replay buffer. During the roll-out, we start at a random time step in the training dataset and apply our model in a closed-loop manner. At each time step during the roll-out, we sample a future trajectory for each vehicle from the Gaussian distribution predicted by our policy network, project the predicted trajectory onto the road, smooth the on-road trajectory with LQR, and finally update the vehicle to the first position of the smoothed trajectory.

5. Experiment

5.1. Dataset

We utilize a real-world urban dataset called pNEUMA [3], which was collected by 10 drones in Athens over a span of 4 days. The dataset encompasses over half a million vehicle trajectories within a large area encompassing over 100 km of lanes and approximately 100 intersections. The recordings were conducted at 5 intervals each day, with each period lasting about 15 minutes and a data collection time interval of 0.04 seconds. To enhance computational efficiency, we adopt a time step of 0.4 seconds. The dataset is divided into a training set, comprising recordings from the first 3 days, and a validation/test set, consisting of recordings from the final day. Notably, we choose not to utilize other popular traffic datasets like NGSIM [8] or HighD [23] or autonomous driving datasets like Lyft [18] or nuPlan [6], as they only encompass samll-scale scenarios. This limitation makes it inadequate for evaluate macroscopic realism.

5.2. Metrics

We evaluate the realism of our simulator by measuring the similarity between the simulation result and real data. During evaluation, each vehicle enters the simulator at its first recorded time and position, and is controlled by our simulator to complete its recorded route. When an agent reaches its final recorded position, it is removed from the simulator.

Short-term microscopic: Following [5, 40], we conduct a short-term microscopic evaluation by simulating for 20 seconds from a random time step in the test dataset. We first measure the similarity between the simulated and real data using **position and velocity RMSE** metrics given by:

$$\text{RMSE} = \frac{1}{T_s}\sum_{t=1}^{T_s}\sqrt{\frac{1}{M}\sum_{m=1}^{M}\|\boldsymbol{s}_t^m - \hat{\boldsymbol{s}}_t^m\|^2}, \tag{10}$$

where $\boldsymbol{s}_t^m$ and $\hat{\boldsymbol{s}}_t^m$ were the real and simulated value of the position or velocity of the agent m at time step t, respectively. T_s was the total simulated time steps, and M was the total simulated agent number. Besides, we measure the **minimum Average Displace Error (minADE)** to not penalize reasonable trajectory unlike the real one:

$$\text{minADE}_N = \frac{1}{M}\sum_{m=1}^{M}\min_{\hat{\boldsymbol{s}}_n}\frac{1}{T_s}\sum_{t=1}^{T_s}\|\boldsymbol{s}_t^m - \hat{\boldsymbol{s}}_{t,n}^m\|^2, \tag{11}$$

where $N = 20$ is roll-out times. We also calculate the **off-road rate** (the avarage proportion of vehicles that deviate more than 1.5m from the road over all time steps). The common collision rate metric is not used because we focus on the long-term impact and the dataset does not provide accurate vehicle size and heading information.

Long-term macroscopic: we also evaluate our model's long-term macroscopic accuracy on five periods for 800 seconds from its initial recording time. To measure the long-term performance, we use two standard macroscopic metrics for traffic flow data [29, 36, 38], namely **road density and speed RMSE**, in addition to the **off-road rate**. The density of a road at a time step is calculated by dividing the number of vehicles on the road by its total lane length, assuming that all lanes have the same width. Meanwhile, the road speed is computed as the mean speed of all vehicles on the road. To quantify the similarity between the simulated and ground truth values, we still use RMSE, where the variable M becomes the total number of roads.

5.3. Performance

We compare our method against state-of-the-art baselines:

SUMO [24]: we use the IDM model [42] as the car-following model and mobil [19] as the lane-changing model. We tune the IDM's parameters for 6 types of vehicle by minimizing the MSE between the IDM calculated acceleration and real acceleration using an Adam optimizer [20].

BC [32]: we learn our model directly by the BC method.

MARL [26]: we train our model using IPPO [10] as the Multi-Agent RL algorithm, where the reward function is composed of three parts: a displacement reward (average distance between agent trajectory with GT trajectory), an off-road penalty and a terminal reward.

MARL+BC [31]: we add a behavior cloning term in the loss function while learning the policy with MARL.

PS-GAIL [40]: we let all vehicles share the same policy parameter and critic parameter and learn the policy using reward functions computed by GAIL.

Table 1. Comparison with baselines and ablated models on microscopic metrics for 20 seconds.

Model	Position RMSE(m)	Velocity RMSE(m/s)	minADE_{20}(m)	Off-road(%)
SUMO [24]	41.25	7.00	24.20	**0**
BC [32]	40.08±1.61	6.74±0.37	22.02±3.98	33.43±3.10
MARL [26]	48.66±1.54	8.92±0.67	40.07±1.26	3.70±0.40
MARL+BC [31]	29.78±0.42	4.54±0.14	22.07±0.21	11.14±1.17
PS-GAIL [40]	34.78±0.43	5.33±0.12	27.86±0.33	7.65±0.32
RAIL [5]	31.62±0.65	5.51±0.09	24.51±0.41	2.57±0.47
LASIL (ours)	**19.21±0.44**	**3.02±0.07**	**12.79±0.32**	0.28±0.01
w/o Augmentation	21.62±0.54	3.34±0.14	13.81±0.37	1.31±0.18
w/o Context-conditioned	23.03±0.87	3.73±0.12	14.94±0.43	0.51±0.04
w/o On-road Projection	22.53±0.42	3.26±0.10	14.42±0.33	1.74±0.60
w/o LQR	22.62±0.43	4.05±0.15	14.02±0.28	0.22±0.02

Table 2. Comparison with baselines and ablated models on macroscopic metrics for 800 seconds.

Model	Road Density RMSE(veh/km)	Road Speed RMSE(m/s)	Off-road(%)
SUMO [24]	52.70	5.52	**0**
BC [32]	61.51±1.53	5.38±0.21	42.15±5.25
MARL [26]	90.59±2.65	5.27±0.48	4.63±0.64
MARL+BC [31]	43.60±2.80	4.01±0.13	22.73±5.62
PS-GAIL [40]	54.06±1.23	4.03±0.05	13.24±3.20
RAIL [5]	54.45±1.89	3.89±0.11	2.92±0.38
LASIL (ours)	**45.13±0.25**	**3.17±0.14**	0.34±0.03
w/o Augmentation	55.65±0.44	3.73±0.23	9.62±1.40
w/o Context-conditioned	55.98±0.51	4.48±0.19	0.64±0.08
w/o On-road Projection	52.90±0.54	3.59±0.20	11.33±2.37
w/o LQR	61.68±0.96	3.54±0.16	0.47±0.07

RAIL [5]: we learn the model as **PS-GAIL** but with additional displacement, off-road and terminal rewards.

We train and evaluate each model five times to obtain the mean and standard deviation (std) of various metrics. Note that we do not apply on-road projection and LQR for baselines. We evaluate both short-term and long-term performance, as shown in Tab. 1 and Tab. 2, respectively. Our method achieves better results than all baselines in terms of position and velocity RMSE, road density and speed RMSE, with minor off-road rate.

5.4. Ablation Study

We conducted a series of ablation experiments to assess the individual contributions of crucial components in our approach, whose results are presented in Tab. 1 and Tab. 2.

Augmentation: By removing our VAE module and directly learning the original expert state and action, we analyze the importance of our data augmentation technique. The results demonstrate that augmentation plays a vital role in improving both short-term and long-term simulation performance by mitigating covariate shift.

Context-conditioned VAE: To emphasize the significance of modeling the context-conditioned trajectory distribution rather than the whole state distribution, we replaced our context-conditioned VAE with a naive VAE that directly models the expert and learner state distribution. The observed drop in performance demonstrates the challenges in reconstructing the context distribution.

On-road projection: Our ablation study on the on-road projection module aims to show its impact on reducing the off-road rate. The results show that the on-road projection module leads to a notable decrease in the off-road rate and moderate improvements in other performance metrics.

LQR: Removing LQR module allows us to evaluate its effectiveness. While LQR led to a higher short-term off-road rate due to more constrained movement (driving off-road due to inertia), its removal deteriorates other short-term and long-term metrics, because LQR can smooth the trajectory, thus leads to more realistic simulations.

5.5. Qualitative Result

In Fig. 2, we present the mean road density and speed for real-world data, SUMO simulation, and our proposed method over all time steps. The results demonstrate that our proposed method surpasses the capabilities of the SUMO simulator at replicating long-term macroscopic traffic patterns due to our model's enhanced ability to replicate the typical microscopic driving behaviors over a long period.

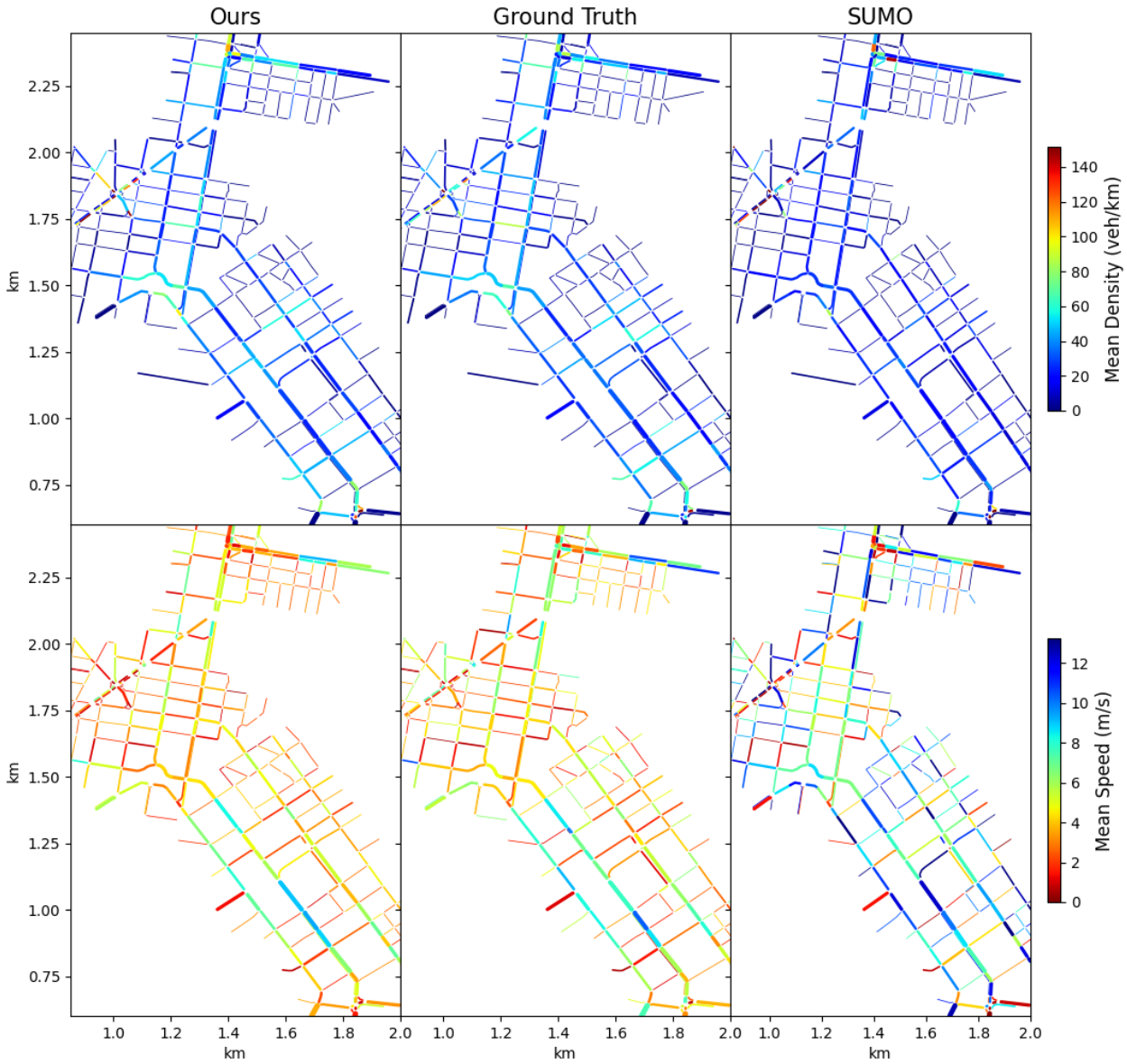

Figure 2. Mean density and speed on each road over all time steps in the long-term evaluation. Our method's density and speed hot-maps have a more similar color to the ground-truth one compared with the SUMO's.

6. Conclusion

In conclusion, we have addressed the challenge of creating a realistic traffic simulator that accurately models human driving behaviors in various traffic conditions. Traditional imitation learning-based simulators often fail to deliver accurate long-term simulations due to the covariate shift problem in multi-agent imitation learning. To tackle this issue, we proposed a learner-aware supervised imitation learning method, which leverages a context-conditioned VAE to generate learner-aware augmented expert states. We leverage a context-conditioned VAE to simultaneously reconstruct the expert and learner state. This approach enables the reproduction of long-term stable microscopic traffic simulations, marking a significant advancement in the field of urban traffic simulation. Our method has demonstrated superior performance over existing state-of-the-art simulators when evaluated on the real-world dataset pNEUMA, achieving better short-term microscopic and long-term macroscopic similarity to real-world data than state-of-the-art baselines.

7. Acknowledgements

This research project is partially supported by the Innovation and Technology Commission of the HKSAR Government under the InnoHK initiative, Hong Kong General Research Fund (11202119 and 11208718), Innovation and Technology Commission (GHP/126/21GD), and Guangdong, Hong Kong and Macao Joint Innovation Project (2023A0505010016).

References

[1] Karl Johan Åström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton University Press, 2021. 5

[2] Jaime Barceló and Jordi Casas. Dynamic network simulation with aimsun. *Simulation approaches in transportation analysis: Recent advances and challenges*, pages 57–98, 2005. 1

[3] Emmanouil Barmpounakis and Nikolas Geroliminis. On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment. *Transportation research part C: emerging technologies*, 111:50–71, 2020. 2, 6

[4] Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Blazej Osinski, Hugo Grimmett, and Peter Ondruska. SimNet: Learning reactive self-driving simulations from real-world observations. *IEEE International Conference on Robotics and Automation*, pages 5119–5125, 2021. 1, 2, 3

[5] Raunak P. Bhattacharyya, Derek J. Phillips, Changliu Liu, Jayesh K. Gupta, K. Driggs-Campbell, and Mykel J. Kochenderfer. Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning. *International Conference on Robotics and Automation*, pages 789–795, 2019. 2, 3, 6, 7

[6] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *IEEE/CVF Computer Vision and Pattern Recognition Conference ADP3 workshop*, 2021. 6

[7] Qianwen Chao, Huikun Bi, Weizi Li, Tianlu Mao, Zhaoqi Wang, Ming C Lin, and Zhigang Deng. A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. In *Computer Graphics Forum*, pages 287–308, 2020. 1

[8] James Colyar and John Halkias. Us highway 101 dataset. *Federal Highway Administration, Tech. Rep.*, 2007. 6

[9] Robert Dadashi, L'eonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal wasserstein imitation learning. In *International Conference of Learning Representation*, 2021. 2

[10] Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020. 6

[11] Frederik Diehl, Thomas Brunner, Michael Truong Le, and Alois Knoll. Graph neural networks for modelling traffic participant interaction. In *IEEE Intelligent Vehicles Symposium*, pages 695–701, 2019. 4

[12] Shuo Feng, Xintao Yan, Haowei Sun, Yiheng Feng, and Henry X. Liu. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature Communications*, 12, 2021. 1

[13] Justin Fu, Katie Luo Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference of Learning Representation*, 2018. 2

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63:139 – 144, 2014. 2

[15] Ke Guo, Wenxi Liu, and Jia Pan. End-to-end trajectory distribution prediction based on occupancy grid maps. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2232–2241, 2022. 2

[16] Ke Guo, Wei Jing, Junbo Chen, and Jia Pan. CCIL: Context-conditioned imitation learning for urban driving. In *Robotics: Science and Systems*, 2023. 4

[17] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016. 2

[18] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference of Robot Learning*, 2020. 6

[19] Arne Kesting, Martin Treiber, and Dirk Helbing. General lane-changing model MOBIL for car-following models. *Transportation Research Record*, 1999(1):86–94, 2007. 6

[20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 6

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[22] Yann Koeberle, Stefano Sabatini, Dzmitry V. Tsishkou, and Christophe Sabourin. Exploring the trade off between human driving imitation and safety for traffic simulation. *IEEE International Conference on Intelligent Transportation Systems*, pages 779–786, 2022. 3

[23] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highD dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *International Conference on Intelligent Transportation Systems*, pages 2118–2125, 2018. 6

[24] 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), 2012. 1, 6, 7

[25] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on Robot Learning*, pages 143–156, 2017. 1, 2

[26] Quanyi Li, Zhenghao Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *Advances in Neural Information Processing Systems*, 2023. 6, 7

[27] Weizi Li, David Wolinski, and Ming C. Lin. City-scale traffic animation using statistical learning and metamodel-based optimization. *ACM Trans. Graph.*, 36(6):200:1–200:12, 2017. 1

[28] Weizi Li, David Wolinski, and Ming C. Lin. ADAPS: Autonomous driving via principled simulations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7625–7631, 2019. 1

[29] Michael James Lighthill and Gerald Beresford Whitham. On kinematic waves ii. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178):317–345, 1955. 6

[30] Weiwei Liu, Wei Jing, Ke Guo, Gang Xu, Yong Liu, et al. Traco: Learning virtual traffic coordinator for cooperation with multi-agent reinforcement learning. In *Conference on Robot Learning*, pages 2465–2477, 2023. 1

[31] Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Becca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. *arXiv preprint arXiv:2212.11419*, 2022. 6, 7

[32] Donald Michie, Michael Bain, and Jean Hayes-Michie. Cognitive models from subcognitive skills. *IEEE control engineering series*, 44:71–99, 1990. 1, 2, 6, 7

[33] Xiaoyu Mo, Zhiyu Huang, Yang Xing, and Chen Lv. Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network. *IEEE Transactions on Intelligent Transportation Systems*, 23:9554–9567, 2022. 4

[34] Jeremy Morton, Tim A Wheeler, and Mykel J Kochenderfer. Analysis of recurrent neural networks for probabilistic modeling of driver behavior. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1289–1298, 2016. 1

[35] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *International Conference of Machine Learning*, page 2, 2000. 2

[36] Paul I Richards. Shock waves on the highway. *Operations research*, 4(1):42–51, 1956. 6

[37] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011. 1, 2

[38] Jason Sewall, David Wilkie, Paul Merrell, and Ming C Lin. Continuum traffic simulation. In *Computer Graphics Forum*, number 2, pages 439–448, 2010. 6

[39] Yu Shen, Weizi Li, and Ming C. Lin. Inverse reinforcement learning with hybrid-weight trust-region optimization and curriculum learning for autonomous maneuvering. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7421–7428, 2022. 2

[40] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. *Advances in neural information processing systems*, 31, 2018. 2, 3, 6, 7

[41] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. TrafficSim: Learning to simulate realistic multi-agent behaviors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10400–10409, 2021. 1, 2, 3

[42] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000. 1, 6

[43] David Wilkie, Jason Sewall, Weizi Li, and Ming C. Lin. Virtualized traffic at metropolitan scales. *Frontiers in Robotics and AI*, 2:11, 2015. 1

[44] 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. 1

[45] Kaifeng Zhang, Rui Zhao, Ziming Zhang, and Yang Gao. Auto-encoding adversarial imitation learning. *arXiv preprint arXiv:2206.11004*, abs/2206.11004, 2022. 2

[46] Guanjie Zheng, Hanyang Liu, Kai Xu, and Zhenhui Jessie Li. Objective-aware traffic simulation via inverse reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2021. 2, 3

[47] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 1433–1438, 2008. 2