# Combining Model-Based Controllers and Generative Adversarial Imitation Learning for Traffic Simulation

Haonan Chen, Tianchen Ji, Shuijing Liu, and Katherine Driggs-Campbell

***Abstract*— An accurate model of human drivers is essential to validate the performance of autonomous vehicles in multiagent and interactive scenarios. Previous works on human driver modeling either use model-based controllers that are not adaptive and need laborious parameter-tuning or learn an end-to-end black box model that has few safety guarantees. We propose a two-stage hybrid driver model, where a high-level neural network generates driver traits that are used as the parameters of the low-level model-based controllers for simulated drivers. We train our model using generative adversarial imitation learning with reward augmentation and parameter sharing from real-world vehicle trajectory data. By combining data-driven and model-based approaches, our method simulates traffic agents with expressive, safe, and human-like behaviors. We demonstrate that our method outperforms state-of-the-art baselines in terms of imitation performance and safety in a multi-agent highway driving scenario.**

## I. Introduction

The ability to safely interact with human traffic participants in various scenarios is important for autonomous vehicles yet challenging to validate [2], [3]. Although testing the autonomous driving algorithms in the real-world reflects their performance accurately, real-world testing is usually expensive and can be dangerous [4]. A cheaper and safer alternative is simulation, which has the potential to generate large-scale traffic scenarios for validation purposes. However, simulating human-like behaviors in multiagent traffic scenarios remains an open challenge because of the stochasticity and complexity of human behaviors [5].

Building human driver models for simulation is a challenging task for two main reasons. First, human behaviors are highly variable since they frequently interact and negotiate with each other and occasionally bend the traffic rules [6]. As a result, parametric equations have difficulty capturing the uncertainty of human decisions. Second, extracting a model from human driving data is difficult [7]. Many existing works assume that the humans have a utility function and attempt to recover the function through learning [8]–[10]. However, humans may be irrational and such a utility function may not even exist under specific circumstances [11].

Despite these challenges, driver modeling has been the subject of extensive research [12]–[15]. Model based methods, such as Intelligent Driver Model (IDM) and Minimizing Overall Braking Induced by Lane Changes (MOBIL), use a set of equations with fixed parameters to describe human driving behaviors. Thus, IDM and MOBIL are unable to synthesize complex, and evolving human behaviors in an adaptive manner [16]–[19]. To overcome this problem, Generative Adversarial Imitation Learning (GAIL) has been used to learn expressive and adaptable driver models from datasets of human drivers [7], [20], [21]. Additional techniques, such as parameter sharing and reward augmentation, have also been explored to improve the scalability and performance of GAIL [22], [23]. Despite improving the imitation performance, these black box models have few safety guarantees and are computationally expensive to train.

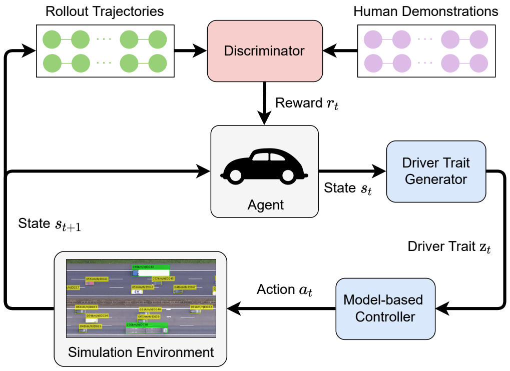

Fig. 1: **Overview of our method.** To simulate natural traffic flows, we propose a hybrid model that trains a neural network with multi-agent reward augmented GAIL to generate driver traits $z$, which are passed to model-based controllers to output accelerations and steering angles.

To address the above problems of model-based and learning-based approaches, we develop human driver models that can exhibit expressive, adaptive, and safe behaviors in complex driving scenarios. We propose a hybrid two-stage driver model that combines model-based methods with imitation learning based methods. In particular, an IDM [24] and a proportional-derivative (PD) controller are used for low-level longitude and lateral control of the vehicle, respectively. As opposed to using fixed hyper-parameters in the IDM and PD controller, we train a high-level auto-tuning network to adjust the parameters in real time so that the vehicle can exhibit adaptive behavior in constantly changing driving situations. By adopting the idea of parameter sharing GAIL (PS-GAIL) [22], the trained network parameters are shared by all the agents to improve the efficiency and scalability. In addition, we use reward augmentation to modify the objective to minimize undesirable behaviors. To stabilize the training, we gradually increase the number of simulated agents with a prescribed learning curriculum.

Our hybrid approach combines the advantages and ad-

H. Chen, T. Ji, S. Liu, and K. Driggs-Campbell are with the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. emails: {haonan2, tj12, sliu105, krdc}@illinois.edu

This material is based upon work supported by the National Science Foundation under Grant No. 2143435. This work utilizes resources supported by the National Science Foundation's Major Research Instrumentation program, grant #1725729, as well as the University of Illinois at Urbana-Champaign [1].

dresses the problems of both types of methods. The high-level network has better adaptability than model-based controllers, allowing our model to exhibit human-like behaviors. Compared with pure data-driven methods, the proposed model is safer due to the safety guarantees of the low-level controllers.

Our contributions can be summarized as follows:

1) We propose a novel two-stage driver model which combines model-based and data-driven approaches to generate expressive and safe human driving behaviors using reward augmented GAIL (Figure 1).
2) Using parameter-sharing, reward augmentation, and curriculum learning, our model is able to control multiple vehicles simultaneously for a realistic traffic simulation.
3) We demonstrate that our method achieves higher driving safety and maintains competitive performance compared to previous works in experiments using real-world driving trajectories.

This paper is organized as follows: We review previous related works in Section II. We introduce the preliminaries including model-based controllers and multi-agent GAIL in Section III. We formalize the problem of human driver modeling and propose our model architecture in Section IV. Experiments and results are discussed in Section V. We conclude the paper in Section VI.

## II. Related Works

We review previous works in model-based, learning-based, and hybrid methods for human driver modeling.

### A. *Model-Based Controllers*

Model based methods such as Intelligent Driver Model (IDM) and its variants have been widely used for longitude driving behavior modeling [16], [25], [26]. IDM uses a parametric model to maintain a safe headway distance with front vehicles. Minimizing Overall Braking Induced by Lane Changes (MOBIL) extends these car-following models to handle lane-changing behaviors [19]. Depending on neighboring vehicles, MOBIL makes lane-changing decisions if certain criteria are satisfied. Both IDM and MOBIL use a set of equations to describe human driving behaviors and thus are unable to synthesize complex and constantly changing human behaviors in an adaptive manner.

### B. *Imitation Learning for Driver Modeling*

Generative adversarial imitation learning (GAIL) and its variants have been widely used to learn expressive and adaptable driver models from datasets of human drivers [7], [20], [21]. PS-GAIL uses parameter sharing to allow a GAIL policy to control multiple agents simultaneously [22]. Reward Augmented Imitation Learning (RAIL) combines imitation learning with reinforcement learning reward, which allows designers to use prior knowledge to improve the imitation performance of drivers [23]. However, these black-box imitation learning models lack safety guarantees, which makes them prone to undesirable traffic phenomenon such as collisions and off-roads. In addition, training end-to-end policies is computationally expensive and not data efficient.

### C. *Hybrid Methods for Driver Modeling*

There are extensions of IDM to model realistic driving behaviors. Monteil *et al.* [27] use Kalman filter with physical inequality constraints to estimate the IDM parameters from the sensor data. Morton *et al.* [28] use the Levenberg-Marquardt algorithm to learn the behavioral parameters as a nonlinear least-squares problems. Buyer [29] uses a particle filter for online parameter estimation based on conventional Sequential Importance Resampling (SIR). Bhattacharyya *et al.* [30] estimate a distribution over the parameters of IDM model using particle filtering. However, all of the above methods use *non-adaptive parameters* of the underlying rule-based models. The parameters for these approaches are not updated over time as traffic conditions change, resulting in static driver behaviors.

Beyond traffic simulation, hybrid approaches have been employed for safe autonomous driving. Pulver *et al.* [31] use DAgger [32] to imitate an expensive-to-run optimizer and train an efficient optimizer for motion planning. Yurtsever *et al.* [33] include the waypoints of A* planner as part of the state and reward for deep Q-learning based driving agents. These methods typically do not aim to imitate driver-like behaviors extracted from driver data. Both of these models only use the model-based approach for training and ultimately produce a neural network for controlling the vehicle. Thus, the resulting controllers do not have the performance guarantees of the underlying model-based method.

In this work, we train a neural network to dynamically adjust the parameters for the model-based controllers to generate more realistic trajectories. Moreover, many previous works for traffic simulation only consider longitudinal control of the vehicles [7], [20]–[23], [30], while our work also includes the lateral control to enable lane-changing behaviors, leading to more realistic traffic simulation.

## III. Preliminaries

We introduce the model-based controllers used to generate low-level actions for vehicle control in human driver modeling, the formulation of driving as a Markov Decision Process (MDP), and general ideas of Generative Adversarial Imitation Learning in multi-agent systems.

### A. *Model-Based Controllers for Traffic Simulation*

*1) Longitudinal Control:* To control the longitudinal behavior of a vehicle, we use IDM to generate acceleration predictions based on the following equations:

$$d^* = d_0 + Tv + \frac{v\Delta v}{2\sqrt{ab}}, \tag{1}$$

$$\dot{v} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{d^*}{d}\right)^2\right], \tag{2}$$

where $v$ is the current velocity, $\Delta v$ is the current relative velocity with respect to the front vehicle, and $d$ is the current distance to the front vehicle. The vehicle dynamics are parameterized by the minimum distance to the front vehicle that the driver will tolerate $d_0$, the desired time gap to the front vehicle $T$, the maximum acceleration ability $a$, the maximum deceleration ability $b$, the driver's desired velocity $v_0$, and the velocity exponent $\delta$ which describes how the acceleration decreases when the velocity of the vehicle approaches the desired velocity. By making use of an explicit driver model, the IDM often produces collision-free trajectories [28].

*2) Lateral Control:* Following previous works [7], [20]–[23], [30], we use kinematic bicycle model to describe vehicle dynamics. We use a PD controller for the lateral control of the vehicles. The model-based PD controller generates steering angle commands according to the following equations:

$$\Delta\psi_r = \arcsin\left(-\frac{K_p\,\Delta_{\text{lat}}}{v}\right), \tag{3}$$

$$\dot{\psi}_r = K_h\left((\psi_L + \Delta\psi_r) - \psi\right), \tag{4}$$

$$\delta = \arcsin\left(\frac{l}{2v}\dot{\psi}_r\right), \tag{5}$$

where $\Delta_{\text{lat}}$ is the lateral position of the vehicle with respect to the center line of the target lane, $\psi_L$ is the current lane heading, $\psi$ is the current vehicle heading, $l$ is the wheelbase, $K_p$ is the position control gain, and $K_h$ is the heading control gain. Given $K_p$, $K_h$, and a target lane index $p$, the PD controller generates a steering angle based on the current state of the vehicle.

### *B. Markov Decision Process*

We model driving as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the transition probability distribution, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in (0, 1)$ is the discount factor, and $\rho_0$ is the distribution of the initial state $s_0$.

A stochastic policy, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, defines the probability of taking each action from each state. At each time step $t$, the agent takes an action $a_t \in \mathcal{A}$ according to its policy $\pi(\cdot|s_t)$, receives a reward $r_t$, and transits to the next state $s_{t+1}$ according to the unknown state transition model $P(\cdot|s_t, a_t)$. The total accumulated return from time step $t$ is defined as $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$. The goal of the agent is to find a policy that maximizes the expected return from each state, defined as $V_\pi(s) = \mathbb{E}_\pi[R_t|s_t = s]$.

### *C. Generative Adversarial Imitation Learning*

The goal of generative adversarial imitation learning is to learn a policy $\pi$ that imitates an expert policy $\pi_E$ given demonstrations from the expert. A demonstration is defined as a sequence of state-action pairs $\tau = \{s_0, a_0, s_1, a_1, \dots\}$ obtained from the interactions of the expert policy $\pi_E$ with the environment.

In GAIL, a discriminator $D$ learns to distinguish expert behaviors from non-expert ones while a policy $\pi$ attempts to emulate expert behaviors by minimizing the Jensen-Shannon divergence between the two state-action occupancy distributions. The objective of GAIL is

$$\min_{\pi} \max_{D} \; \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))], \tag{6}$$

where $D$ is the probability that the state-action pair $(s, a)$ is generated by the non-expert policy $\pi$. The optimization in GAIL is performed by alternating between a gradient step to increase the objective (6) with respect to the discriminator parameters and a policy update step to decrease the objective (6) with respect to the policy parameters.

To realize simultaneous control over multiple agents for traffic flow simulation, we adopt the idea of parameter sharing GAIL, where a policy with shared parameters is used by each agent to generate trajectories respectively [22]. We also penalize undesirable traffic phenomena through reward augmentation [23]. Although using the same policy parameters, the agents can still exhibit different safe behaviors as each agent receives unique observations. Furthermore, all the agents receive rewards from the same discriminator with augmented reward to update the policy at each iteration of the optimization.

## IV. Methodology

Next, we present the problem formulation of the driving behavior imitation and our proposed method.

### *A. Problem Formulation*

To model the human driver behavior, we formulate the problem as parameter estimation of IDM and PD controllers with stochasticity. We aim to recover the distribution over the parameters of model-based controllers by mimicking human driving data. We assume that the state-action samples are independent and identically distributed (i.i.d.) and drawn from the demonstration distribution $\tau_E = \{(s_i, a_i)\}_{i=1}^{N} \overset{i.i.d}{\sim} \rho_E(s, a)$. The occupancy distribution is defined by $\rho_\pi(s, a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t p(s_t = s|\pi)$, where $\pi(a|s)$ is the probability of taking action $a$ at state $s$ following policy $\pi$, $p(s_t = s|\pi)$ is the probability that the agent reaches state $s$ at time $t$ executing policy $\pi$ starting from initial state distribution $s_0 \sim \rho_0$. Our main assumption is that human drivers can be described by their own driver traits[1] $z$. Each driver trait corresponds to a configuration for model-based controllers. In this work, we focus on highway driving scenarios; however, we note that our model can easily extend to other driving scenarios in which IDM and PD controllers can be applied (e.g., merge, roundabout, intersection, etc.).

Our policy $\pi$ composes a driver trait generator $\pi_\theta$ parameterized by $\theta$ and model-based controllers $MC$ to output agents' actions $a \in \mathcal{A}$, $\pi(a|s) := MC\left(a|\pi_\theta(z|s)\right)$. Driver trait generator $\pi_\theta : \mathcal{S} \to \mathcal{Z}$ maps the state to driver traits of human drivers. $MC : \mathcal{Z} \to \mathcal{A}$ takes in the driver traits and outputs the driving actions.

[1] We refer to driver traits as the individual driver's preference parameters that characterize how maneuvers are performed. For the IDM, traits are represented as the tunable parameters including jam distance, desired time gap, desired velocity during the driving, and so on.

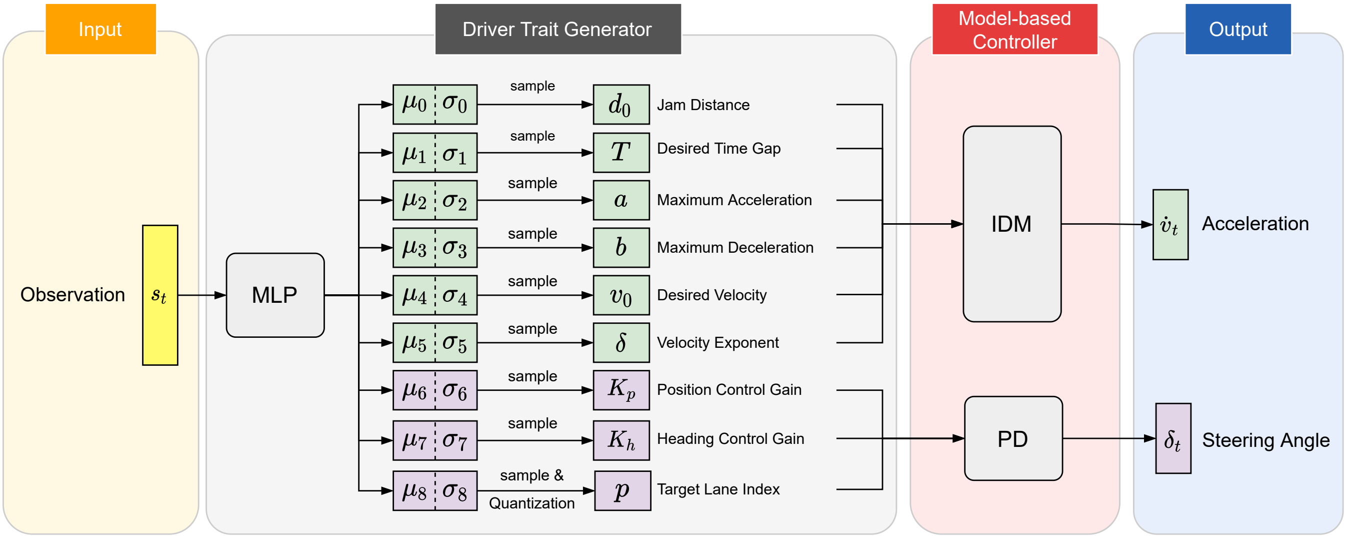

Fig. 2: **An illustration of our model architecture, explicitly showing the two-stage policy.** The model takes in the observation, processes it with multiple layer perceptron, and outputs the driver traits. Model-based controllers dictate how the vehicles accelerate and steer on the road.

**Algorithm 1** Our algorithm

```
**Input**: Expert demonstrations $\tau_E = \{(s_i, a_i)\}_{i=0}^N$, shared policy parameter $\theta_0$ of driver trait generator, discriminator parameters $w_0$, and batch size $B$.
**Output**: Learned policies $\pi_\theta$, reward function $D_w$
for $i = 0, 1, 2, \ldots$ do
    Sample the driver traits $z \leftarrow \pi_\theta$
    Rollout trajectories for all agents $\{(s_t, a_t)\}_{t=0}^T \sim MC(z)$
    Score trajectories $\{(s_t, a_t)\}_{t=0}^T$ and calculate augmented rewards from the discriminator
    Update the discriminator parameters from $w_i$ to $w_{i+1}$ by taking the gradient of Eq.7 w.r.t. $w$
    Take a policy step from $\theta_i$ to $\theta_{i+1}$ using PPO [34] to decrease the objective function Eq.7
```

Our objective is to minimize the distance between our learned two-stage policy and the expert policy (i.e., human demonstrations). We use a discriminator $D$ to estimate the distance between the expert occupancy distribution $\rho_E$ and non-expert occupancy distribution $\rho_\pi$. The augmented distance $d$ is used as the reward signal for MDP. The agent is expected to minimize the cumulative distances $\sum_{k=0}^{\infty} \gamma^k d_{t+k+1}$ measured by the discriminator.

Based on $MC$ and our discriminator, we can reformulate the MDP as $(\mathcal{S}, \mathcal{Z}, P, d, \gamma, \mathcal{S}_0)$, where $\mathcal{Z}$ is the driver trait space and $d$ is the augmented distance between the occupancy distribution generated by our model and expert occupancy distribution. The driver trait generation policy is trained to solve reformulated MDP and maximizes the probability that trajectories generated by $MC$ is expert trajectories.

### B. Imitation Learning with Driver Trait Generation

Our proposed approach combines model-based controller and model-free imitation learning to generate explainable and expressive driver behaviors. Algorithm 1 provides the pseudo code of our proposed approach. The discriminator is a classifier to encourage behavior mimicking and traffic flow generation. The agent infers the driving parameters which are used for model-based controllers. Our learning objective is

$$\min_{\pi_\theta} \max_{D} \sum_{\tau} \rho_{MC(\pi_\theta)}(\tau) \log(D(\tau)) + \rho_E(\tau) \log(1 - D(\tau)) \tag{7}$$

We use the driver trait generator to estimate driver traits $z = (d_0, T, a, b, v_0, \delta, K_p, K_h, p)$, where $d_0$ denotes the jam distance, $T$ denotes the desired time gap, $a$ denotes the maximum acceleration, $b$ denotes the maximum deceleration, $v_0$ denotes the desired velocity, $\delta$ denotes the velocity exponent in IDM, $K_p$ denotes the position control gain, $K_h$ denotes the heading control gail, $p$ denotes the target lane index for PD controller. Our driver trait generation policy, $\pi_\theta$, takes in the positions, velocities, accelerations, and the distance to the lane markings of the ego vehicle and surrounding vehicles as the state. Our policy outputs the mean $\mu$ and standard variation $\sigma$ for driver traits (Figure 2). The driver traits are sampled from normal distribution $N(\mu_i, \sigma_i)$. We perform additional quantization for target lane index to generate discrete variable. We use the similar reparameterization trick [35].

### C. Curriculum Learning for Multi-Agent Settings

When training multi-agent systems, one major challenge is that the problem is non-stationarity, due to the fact that each agent is updating their policy during training. The change in each agent's policies affects other agents' goals and objectives, and vice-versa. This non-stationarity typically leads to unstable training procedures [36]. We use curriculum learning to increase the stability of the training process. We

TABLE I: The ego vehicle and surrounding vehicles that we use as the observation. The position, the velocity, the acceleration, and the distances to lane markings are included for each vehicle.

| Name | Description |
|---|---|
| Ego | The agent controlled by our policy. |
| Front | The preceding vehicle in the same lane |
| Rear | The following vehicle in the same lane. |
| leftFront | The preceding vehicle in the left lane. |
| leftAlongside | The adjacent vehicle in the right lane. |
| leftRear | The following vehicle in the left lane. |
| rightFront | The preceding vehicle in the right lane. |
| rightAlongside | The adjacent vehicle in the right lane. |
| rightRear | The following vehicle in the right lane. |

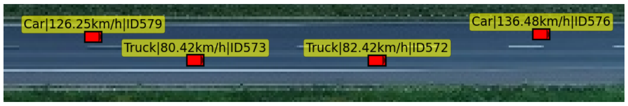

Fig. 3: **Visualisation of our simulation environment** adapted from the HighD dataset. The driving simulator has exactly the same configuration as the real-world dataset.

start out with only a small percentage of agents controlled by our policy and then gradually increase the percentage of agents to be controlled. Ultimately, all of the driving agents are controlled by our policy and a traffic flow is generated.

## V. Experiments

In this section, we describe the simulation environment for driver modeling, the dataset we use for training, and present our experimental results.

### A. Policy Representation

We represent our driver trait generator $\pi_{DT}$ and discriminator $D$ using MLP. We use the Actor-Critic approach for our PPO-based driver trait generator. We implement the actor and critic with two hidden layers and 128 hidden units. ELU activation functions are applied in actor. LeakyReLU activation functions are applied in critic. The discriminator consists of one hidden layer and 128 hidden units, with tanh activation functions applied. Adam optimizers with a constant learning rate of $7 \times 10^{-7}$ is used to train the network.

### B. Dataset

We use the HighD dataset as the expert demonstrations that contains naturalistic vehicle trajectories in German highway [37]. HighD dataset records the traffic flow from bird's-eye view. The positions, the speeds, and the accelerations of the agents are initialized according to the dataset.

### C. Simulator

Figure 3 shows a screenshot of our simulation environment that replays the dataset and trains our policy. Table I is a summary of all vehicles' state that are used as observations. We include the position, velocity, acceleration, and the distances to the lane markings of the agents and neighbors as the state $s \in \mathcal{S}$. During the training, a percentage of the agents are chosen to be controlled by a driving model while the remaining agents are controlled by the IDM.

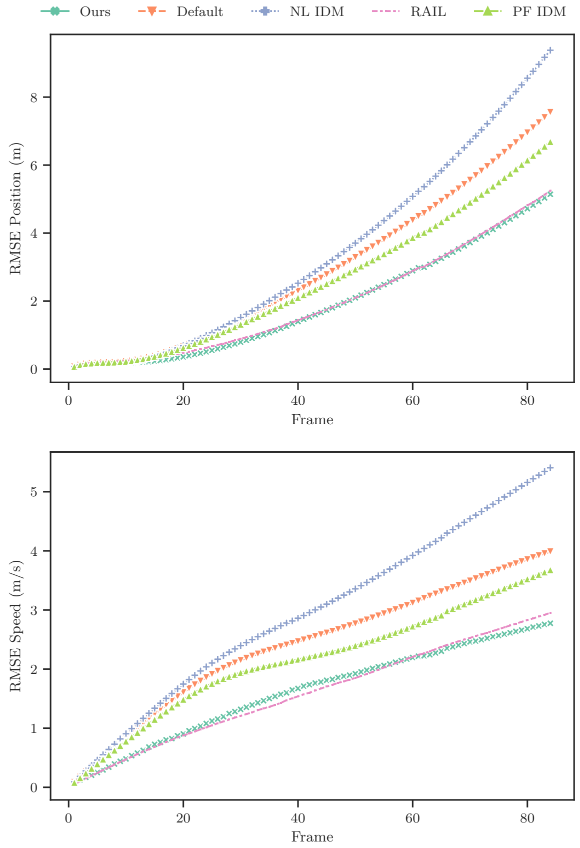

Fig. 4: **Root mean square error in position and speed** averaged over testing vehicles in all driving scenarios to benchmark our model against other driver models. The x-axis is the frame number. The y-axis is RMSE.

### D. Experiment setup

*1) Baselines:* The baselines that we used to compare against our proposed policy are IDM with the default parameter values recommended in [38] (Default), NL IDM with parameters estimated by nonlinear least squares from the data [28], reward augmentation integrated into the multi-agent imitation learning (RAIL [23]), PF IDM with parameters learned from data using particle filtering [30].

*2) Evaluation:* We test all models with 40 randomly selected driving scenarios until the end of each episode. Half of the scenarios are lane changing scenarios while the other half are lane keeping scenarios. To evaluate the imitation performance of all methods, we measure the root mean square error (RMSE) of the position and velocity between trajectories generated by driver models and the dataset to show the local imitation performance for 85 frames with frame rate equal to 25 Hz. We compare the speed distributions to show how the agents imitate the human drivers globally. To measure the safety of all methods, we measure the off-road ratio and collision rate of all driving agents. The off-road ratio is the ratio between the frames that the vehicles are operating away from roads and the total number of frames. The collision rate is the number of collisions divided by the total number of driving agents.

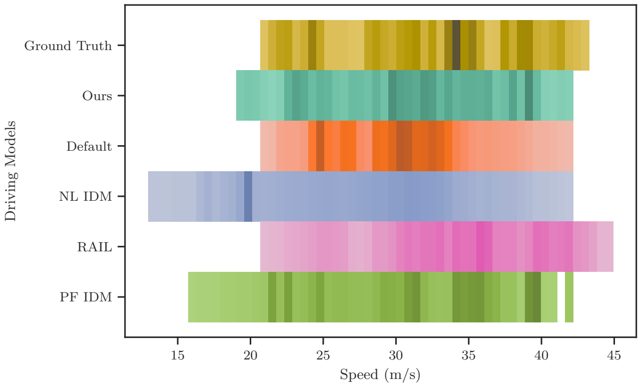

Fig. 5: **Speed distribution in all driving scenarios to benchmark our model against other driver models.** The x-axis is the speed. The y-axis is the driving models. The darkness of the color indicates the frequency of the speed.

### E. Results

We assess the local imitation performance of our model against the baselines in terms of RMSE to see the similarity of driving models and the ground truth (Figure 4). While our proposed method outperforms other baselines in terms of the RMSE speed, our method is comparable to RAIL in terms of the RMSE position. Leveraging on deep neural networks, our method and RAIL utilize the expert data to better recover the human driving patterns than the model-based baselines. The agents driven by our method produce more variable behaviors than RAIL and the other hybrid methods, which makes our agents behave more similar to the ground truth overall.

Speed distribution over testing vehicles in all driving scenarios is shown in Figure 5. We also include the speed distribution from the ground truth dataset. The ground truth speed distribution is multimodal with peaks at 24, 34, and 39 m/s. NL IDM and RAIL generate relatively uniform speed distributions that do not capture this multimodality, and they only produce slight peaks at 19.5 m/s and 35.5 m/s, respectively. The default method does capture modal behavior, but only shows two peaks at 24.5 and 30 m/s. Only our method and the PF IDM approach are able to capture similar multimodality to that of the ground truth. Our method produces speeds at 23, 29.5, 32.5, and 39 m/s, while the PF IDM produces likely speeds at 22.5, 31, 34 and 39.5 m/s. While both methods capture similar variability, we observe that our method more closely captures the true distribution, likely contributing to the better overall imitation performance compared to the PF IDM and other methods.

Figure 6 shows the frequency of undesirable traffic phenomena through the metrics of off-road ratio and collision rate. Pure model-based controllers explicitly constrain the agents to drive within the road, resulting in zero off-road ratios. The Default method has lowest collision rate because the default parameters of headway distance and headway time is sufficiently large to avoid collisions. RAIL exhibits poor performance in terms of off-road ratio and collision rate, which confirms that RAIL (and other typical end-to-end learning methods) do not guarantee safe driving. Our method and other model-based methods achieve zero off-road ratios since these methods leverage PD controllers for lateral control. However, since our method adapts other safety parameters to match typical human behavior, we exhibit a higher collision rate than the default method, while still outperforming the end-to-end data-driven method. This result demonstrates the trade-off between human-like imitation performance and safety guarantees, and that our hybrid method effectively balances objectives of generating realistic traffic flow.

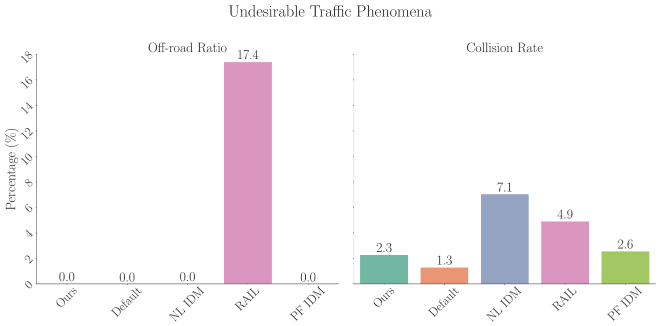

Fig. 6: **Percentage of the undesirable driving behavior averaged over all testing vehicles in all driving scenarios.** The off-road ratio shows that model based methods eliminate drift that data-driven methods tend to produce. The collision ratio highlights the lack of safety guarantees of data-driven methods and, to a lesser extent, hybrid models.

## VI. Conclusion and Future Work

In this work, we propose a hybrid two-stage framework, which combines model-based controllers and data-driven methods, for human driver modeling. In our method, driver traits are generated in real time through a deep neural network and are used by model-based controllers to synthesize low-level actions. We use curriculum learning and parameter sharing to train a driver trait generator with reward augmented imitation learning that effectively controls multiple agents to generate realistic traffic flow. Our experiments demonstrate that the proposed method can effectively balance imitation performance and safety by generating more realistic trajectories than model-based methods and safer behaviors than pure data-driven approaches.

Our work has some limitations, which lead to potential directions for future work. In practice, a driver is able to monitor the surrounding vehicles, but full state observations can be redundant (e.g., a human driver takes actions without accurate measurements of acceleration). As a result, matching agent observations in simulation and real world by incorporating partial observability into our model can be another step towards better imitation performance. Moreover, the discriminator in our method, which is used to generate reward signals for GAIL, is trained to evaluate the similarity between expert and synthesized behaviors of a *single agent*. To generate a human-like *traffic flow*, however, an additional global discriminator which captures the dynamics of the multiagent system can potentially generate more accurate rewards and further improve the performance of our model.

# References

[1] V. Kindratenko, D. Mu, Y. Zhan, J. Maloney, S. H. Hashemi, B. Rabe, K. Xu, R. Campbell, J. Peng, and W. Gropp, *HAL: Computer System for Scalable Deep Learning*. Association for Computing Machinery, 2020, p. 41–48.

[2] P. Du and K. Driggs-Campbell, "Finding diverse failure scenarios in autonomous systems using adaptive stress testing," *SAE International Journal of Connected and Automated Vehicles*, vol. 2, 12 2019.

[3] P. Du and K. Driggs-Campbell, "Adaptive failure search using critical states from domain experts," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 38–44.

[4] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, p. 15–24, Apr 2016.

[5] K. Brown, K. Driggs-Campbell, and M. J. Kochenderfer, "Modeling and prediction of human driver behavior: A survey," *ArXiv*, vol. abs/2006.08832, 2020.

[6] R. P. Bhattacharyya, D. J. Phillips, C. Liu, J. K. Gupta, K. Driggs-Campbell, and M. J. Kochenderfer, "Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 789–795.

[7] R. P. Bhattacharyya, B. Wulfe, D. J. Phillips, A. Kuefler, J. Morton, R. Senanayake, and M. J. Kochenderfer, "Modeling human driving behavior through generative adversarial imitation learning," *Computing Research Repository (CoRR)*, 2020.

[8] A. Bobu, D. R. Scobee, J. F. Fisac, S. S. Sastry, and A. D. Dragan, "Less is more: Rethinking probabilistic models of human behavior," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2020, pp. 429–437.

[9] C. L. Baker, R. Saxe, and J. B. Tenenbaum, "Action understanding as inverse planning," *Cognition*, vol. 113, no. 3, pp. 329–349, 2009.

[10] O. Morgenstern and J. Von Neumann, *Theory of games and economic behavior*. Princeton university press, 1953.

[11] P. Dasgupta, D. Gale, O. Hart, and E. Maskin, "Irrationality in game theory," in *Economic Analysis of Markets and Games: Essays in Honor of Frank Hahn*. The MIT Press, 03 1992.

[12] P. A. Seddon, "A program for simulating the dispersion of platoons of road traffic," *Simulation*, vol. 18, no. 3, pp. 81–90, 1972.

[13] P. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, p. 105–111, 1981.

[14] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, p. 1805–1824, 2000.

[15] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, K. Takeda, and F. Itakura, "Driver modeling based on driving behavior and its evaluation in driver identification," *Proceedings of the IEEE*, vol. 95, no. 2, p. 427–437, 2007.

[16] R. M. Malinauskas, "The intelligent driver model: Analysis and application to adaptive cruise control," Ph.D. dissertation, Clemson University, 2014.

[17] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, p. 4585–4605, 2010.

[18] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, p. 296–305, 2014.

[19] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.

[20] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 204–211.

[21] A. Kuefler and M. J. Kochenderfer, "Burn-in demonstrations for multimodal imitation learning," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2018, p. 1071–1078.

[22] R. P. Bhattacharyya, D. J. Phillips, B. Wulfe, J. Morton, A. Kuefler, and M. J. Kochenderfer, "Multi-agent imitation learning for driving simulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1534–1539.

[23] R. P. Bhattacharyya, D. J. Phillips, C. Liu, J. K. Gupta, K. Driggs-Campbell, and M. J. Kochenderfer, "Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 789–795.

[24] C. Little, "The intelligent vehicle initiative: Advancing "human-centered" smart vehicles," *Public Roads*, vol. 61, no. 2, 1997.

[25] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Cooperation-aware reinforcement learning for merging in dense traffic," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3441–3447, 2019.

[26] M. N. Sharath and N. R. Velaga, "Enhanced intelligent driver model for two-dimensional motion planning in mixed traffic," *Transportation Research Part C-emerging Technologies*, vol. 120, p. 102780, 2020.

[27] J. Monteil, N. O'Hara, V. Cahill, and M. Bouroche, "Real-time estimation of drivers' behaviour," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015, pp. 2046–2052.

[28] J. Morton, T. A. Wheeler, and M. J. Kochenderfer, "Analysis of recurrent neural networks for probabilistic modeling of driver behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1289–1298, 2017.

[29] J. Buyer, D. Waldenmayer, N. Sußmann, R. Zöllner, and J. M. Zöllner, "Interaction-aware approach for online parameter estimation of a multi-lane intelligent driver model," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019, pp. 3967–3973.

[30] R. Bhattacharyya, S. Jung, L. A. Kruse, R. Senanayake, and M. J. Kochenderfer, "A hybrid rule-based and data-driven approach to driver modeling through particle filtering," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021.

[31] H. Pulver, F. Eiras, L. Carozza, M. Hawasly, S. V. Albrecht, and S. Ramamoorthy, "Pilot: Efficient planning by imitation learning and optimisation for safe autonomous driving," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1442–1449, 2021.

[32] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 627–635. [Online]. Available: https://proceedings.mlr.press/v15/ross11a.html

[33] E. Yurtsever, L. Capito, K. A. Redmill, and Ü. Özgüner, "Integrating deep reinforcement learning with model-based path planners for automated driving," *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1311–1316, 2020.

[34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[35] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[36] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. de Cote, "A survey of learning in multiagent environments: Dealing with non-stationarity," *ArXiv*, vol. abs/1707.09183, 2017.

[37] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2118–2125.

[38] M. Treiber and A. Kesting, "The intelligent driver model with stochasticity -new insights into traffic flow oscillations," *Transportation Research Procedia*, vol. 23, pp. 174–187, 2017.