TrafficSim: Learning to Simulate Realistic Multi-Agent Behaviors

Simon Suo[1,2], Sebastian Regalado[3], Sergio Casas[1,2], Raquel Urtasun[1,2]
[1]Uber ATG, [2]University of Toronto, [3]University of Waterloo
{suo, sergio, urtasun}@cs.toronto.edu, sdregala@edu.uwaterloo.ca

Abstract

Simulation has the potential to massively scale evaluation of self-driving systems, enabling rapid development as well as safe deployment. Bridging the gap between simulation and the real world requires realistic multi-agent behaviors. Existing simulation environments rely on heuristic-based models that directly encode traffic rules, which cannot capture irregular maneuvers (e.g., nudging, U-turns) and complex interactions (e.g., yielding, merging). In contrast, we leverage real-world data to learn directly from human demonstration, and thus capture more naturalistic driving behaviors. To this end, we propose TRAFFICSIM*, a multi-agent behavior model for realistic traffic simulation. In particular, we parameterize the policy with an implicit latent variable model that generates socially-consistent plans for all actors in the scene jointly. To learn a robust policy amenable for long horizon simulation, we unroll the policy in training and optimize through the fully differentiable simulation across time. Our learning objective incorporates both human demonstrations as well as common sense. We show* TRAFFICSIM *generates significantly more realistic traffic scenarios as compared to a diverse set of baselines. Notably, we can exploit trajectories generated by* TRAFFICSIM *as effective data augmentation for training better motion planner.*

1. Introduction

Self-driving has the potential to make drastic impact on our society. One of the key remaining challenges is how to measure progress. There are three main approaches for measuring the performance of a self-driving vehicle (SDV): 1) structured testing in the real world, 2) virtual replay of pre-recorded scenarios, and 3) simulation. These approaches are complementary, and each has its key advantages and shortcomings. The use of a test track enables structured and repeatable evaluation in the physical world. While this approach is perceptually realistic, testing is often limited to a few scenarios due to the long setup time and high cost for each test. Moreover it is hard and often impossible to test safety critical situations, such as unavoidable accidents. Virtual replay allows us to leverage diverse scenarios collected from the real world, but it is still limited to what we observe. Furthermore, since the replay is immutable, actors in the environment do not react when the SDV plan diverges from what happened and the sensor data does not reflect the new viewpoint. These challenges make simulation a particularly attractive alternative: in a virtual environment we can evaluate against a large number of diverse and dynamic scenarios in a safe, controllable, and cost-efficient manner.

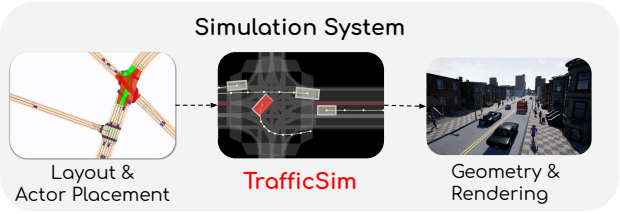

Figure 1. Generating realistic multi-agent behaviors is a key component in self-driving simulation

Simulation systems typically consist of three steps: 1) specifying the scene layout which includes the road topology and actor placement, 2) simulating the motion of dynamic agents forward, and 3) rendering the generated scenario with realistic geometry and appearance, as shown in Figure 1. In this paper, we focus on the second step: generating realistic multi-agent behaviors automatically. This can aid simulation design in several important ways: it can expedite scenario creation by automating background actors, increase scenario coverage by generating variants with emergent behaviors, and facilitate interactive scenario design by generating preview of potential interactions.

However, bridging the behavior gap between the simulated world and the real world remains an open challenge. Manually specifying each actor's trajectory is not scalable and results in unrealistic simulations since the actors do not react to the SDV actions. Heuristic-based models [39, 20, 25] capture basic reactive behavior, but rely on directly encoding traffic rules such as "vehicles follow the road and do not collide". While this approach generates plausible traffic flow, the generated behaviors lack the diversity and

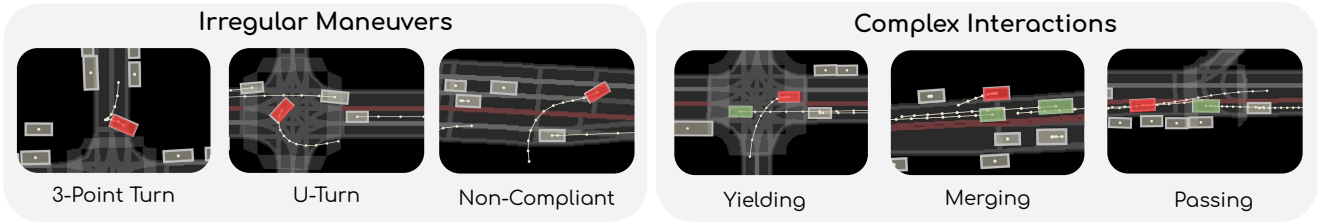

Figure 2. Diversity and nuance of human driving behaviors observed in the real world: red is actor of interest, green are interacting actors

nuance of human behaviors and interactions present in real-world urban traffic scenes. For instance, they cannot capture irregular maneuvers that do not follow the lane graph such as U-turns, or complex multi-agent interplays such as nudging past a vehicle stopped in a driving lane, or negotiations at an unprotected left turn. In contrast, learning-based approaches [11, 38, 35] are flexible and can capture a diverse set of behaviors. However, they often lack common sense and are generally brittle to distributional shift. Furthermore, they can also be computationally expensive if not optimized for simulating large numbers of actors over long horizon.

To tackle these challenges, we present TRAFFICSIM, a multi-agent behavior model for traffic simulation. We leverage recent advances in motion forecasting, and formulate the joint actor policy with an implicit latent variable model [11], which can generate multiple scene-consistent samples of actor trajectories in parallel. Importantly, we present a novel learning framework to train robust policy amenable for traffic simulation over long time horizon. In particular, we leverage:

1. closed-loop training with back-propagation through the fully differentiable simulation, and

2. time-adaptive multi-task loss to balance between learning from demonstration and common sense.

Our experiments show that TRAFFICSIM is able to simulate traffic scenarios that remain realistic over long time horizon, with minimal collisions and traffic rule violations. In particular, it achieves the lowest scenario reconstruction error in comparison to a diverse set of baselines including heuristic, motion forecasting, and imitation learning models. We also show that we can train better motion planners by exploiting trajectories generated by TRAFFICSIM. Lastly, we show experiments in trading off simulation quality and computation. In particular, we can achieve up to 4x speedup with multi-step updates, or further reduce collisions with additional optimization at simulation-time.

2. Related Work

Simulation Environments: Simulating traffic actors is a ubiquitous task with wide ranging applications in transportation research, video games, and now training and evaluating self-driving vehicles [15]. Microscopic traffic simulators [30] employ heuristic-based models [39, 20, 25] to simulate traffic flow. These models capture accurate high-level traffic characteristic by directly encoding traffic rules (e.g., staying in lane, avoiding collision). However due to rigid assumptions, they are not realistic at the street level even after calibrating with real world data [24, 26]. In particular, they can not capture irregular maneuvers (e.g., nudging, U-turns) and complex multi-agent interaction (e.g., yielding, merging) that occur in the real world, shown in Figure 2. Progress in game engines greatly advanced the realism of physical simulations. Researchers have leveraged racing games [41, 33] and developed higher fidelity simulators [19, 4] to train and evaluate self-driving systems. Real world data is leveraged by [32] for realistic sensor simulation. However, actor behaviors are still very simplistic: simulated actors in [19] are governed by a basic heuristic-based controller that can only follow the lane while respecting traffic rules and avoiding head-on collisions. This is insufficient to evaluate SDVs, since one of the main challenge in self-driving is accurately anticipating and safely planning around diverse and often irregular human maneuvers, Thus, this motivates us to learn from real world data to bridge this gap.

Motion Forecasting: Motion forecasting is the task of predicting actor's future motion based on past context, which also requires accurate actor behavior modelling. Traditional approaches track an object and propagate its state to predict its future motion (e.g., Unscented Kalman filter [40] with kinematic bicycle model [28]). More recently, deep-learning based models have been developed to capture increasingly more complex behaviors. [18] rasterizes an high-definition (HD) map and a history of actor bounding boxes to leverage a CNN to forecast actor behavior. Since the future is inherently uncertain, [17, 14] output multiple trajectories per actor. [12] shows that explicitly incorporating prior knowledge help learn better predictive distributions. Several works [1, 35, 38, 11, 29] go beyond actor-independent modeling and explicitly reason about interaction among actors as the future unfolds. To characterize the behavior of multiple actors jointly, [1, 35, 38] leverage auto-regressive generation with social mechanisms. In contrast, [11] employs spatially-aware graph neural networks to model interaction in the

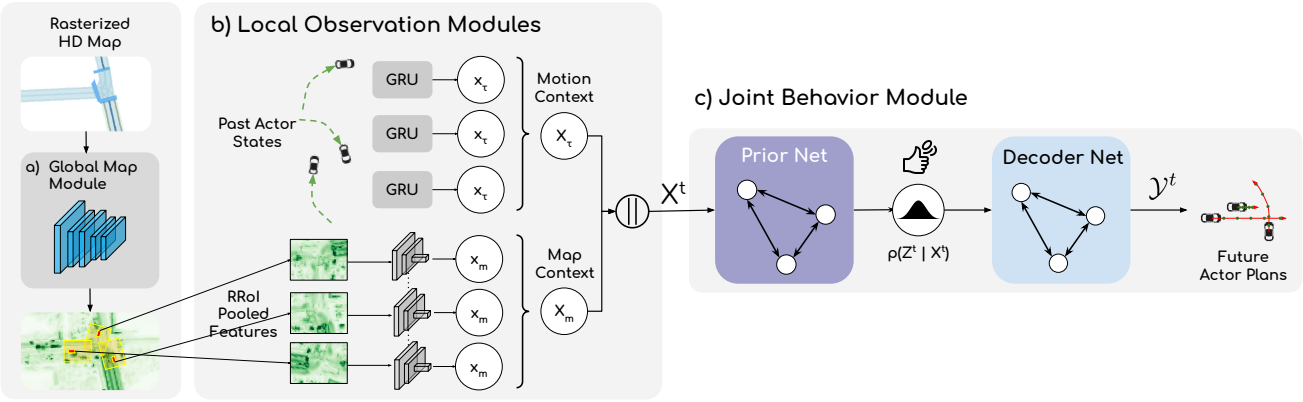

Figure 3. TRAFFICSIM architecture: (a) global map module is run once per map for repeated simulation runs. At each timestep, (b) local observation module extracts motion and map features, then (c) joint behavior module produces a multi-agent plan.

latent space, thereby capturing longer range interactions and avoiding slow sequence sampling. Importantly, these models can generate multiple socially-consistent samples, where each sample constitute a realistic traffic scenario. This enables modelling of complex multi-agent dynamics beyond simple pairwise interaction, and thus they are particularly amenable for simulating actor behaviors in virtual traffic. However, they cannot be directly used for simulation over long time horizon, since they are brittle to distributional shift and cannot recover from compounding error.

Imitation Learning: Imitation learning (IL) aims to learn control policies from demonstration. Behavior cloning [34] treats state-action pairs as i.i.d examples to leverage supervised learning, but suffers from distributional shift due to compounding error [36]. Intuitively, during offline open-loop training, the policy only observes ground truth past states, but when unrolled in closed-loop at test time, it encounters novel states induced by its sequence of suboptimal past decisions and fails to recover. Many approaches have been proposed to mitigate the inevitable deviation from the observed distribution, but each has its drawbacks. Online supervision [36] require access to an interactive expert that captures the full distribution over human driving behaviors. Data augmentation [2, 8] depends on manually designed out-of-distribution states and corresponding desired actions, which is often brittle and adds bias. Uncertainty-based regularization [9, 21] leverages predictive uncertainty to avoid deviating from the observed distribution, but can be challenging and computationally expensive to estimate accurately. Adversarial IL approaches [23, 7] jointly learn the policy with a discriminator. However, they are empirically difficult to train (requiring careful reward augmentation [5] and curriculum design [3]), and are generally limited to simulating a small number of actors [6] in a specific map topology (e.g., NGSIM). In contrast, we aim to learn a joint actor policy that generalizes to diverse set of urban streets and simulates the behavior for large number of actors in parallel. Furthermore, while IL methods typically assume non-differentiable environment, we directly model differentiable state transitions instead. This allow us to directly optimize with back-propagation through the simulation.

3. Learning Multi-Agent Traffic Behaviors

In this section, we describe our approach for learning realistic multi-agent behaviors for traffic simulation. Given a HD map $\mathcal{M}$, traffic control $\mathcal{C}$, and initial dynamic states of N traffic actors, our goal is to simulate their motion forward. We use $Y^t = \{y_1^t, y_2^t, ..., y_N^t\}$ to denote a collection of N actor states at time t. More precisely, each actor state is parameterized as a bounding box $y_i^t = (b_x, b_y, b_w, b_h, b_\theta)$ with 2D position, width, height, and heading. In the following, we first describe how to extract rich context from the simulation environment. Then, we explain our joint actor policy that explicitly reasons about interaction and generates socially consistent plans. Lastly, we present a learning framework that leverages back-propagation through the differentiable simulation, and balances imitation and common sense. We illustrate the full architecture in Figure 3.

3.1. Extracting Rich Context from the Environment

Accurately modelling actor behaviors requires rich scene context from past motion and map topology. Towards this goal, we propose a differentiable observation module that takes as input the past actor states $Y^{:t}$, traffic control $\mathcal{C}$, and HD map $\mathcal{M}$, and processes them in two stages. First, we rasterize the HD map $\mathcal{M}$ and use a CNN-based perception backbone network inspired by [42, 13] to extract rich geometrical features $\tilde{\mathcal{M}}$, shown in Figure 3 (a). Since we are only interested in the region of interest defined by $\mathcal{M}$ and these spatial features are static across time, we can process each map once, and cached them for repeated simulation runs.

Then we leverage a local observation module with two components: a *map feature extractor* and a *past trajectory encoder* shown in Figure 3 (b). Unlike the global map module, these feature extractors are run once per simulation step, and are thus designed to be lightweight. To extract local context X_m^t around each actor, we apply Rotated Region of Interest Align [31] to the pre-processed map features $\tilde{\mathcal{M}}$. To encode the past trajectories of each actor in the scene, we employ a 4-layer GRU with 128 hidden states, yielding X_τ^t. Finally, we concatenate the map and past trajectory features to form the scene context $X^t = [X_m^t, X_\tau^t]$, which we use as input to the joint actor policy.

3.2. Implicit Latent Variable Model for Multi-Agent Reasoning

We use a joint actor policy to explicitly reasons about multi-agent interactions, shown in Figure 3 (c). This allows us to sample multiple socially consistent plans for all actors in the scene in parallel. Concretely, we aim to characterize the joint distribution over actors' future states $\mathcal{Y}^t = \{Y^{t+1}, Y^{t+2}, ..., Y^{t+T_{\text{plan}}}\}$. This formulation allows us to leverage supervision over the full planning horizon T_{plan} to learn better long-term interaction. To simplify notation, we use $\mathcal{Y}^t$ in subsequent discussions.

It is difficult to represent this joint distribution over actors in an explicit form due to uncertainty over each actor's goal and complex interactions between actors as the future unfolds. A natural solution is to implicitly characterize this distribution via a latent variable model [37, 11]:

$$P(\mathcal{Y}^t|X^t) = \int_Z P(\mathcal{Y}^t|X^t, Z^t)P(Z^t|X^t) \qquad (1)$$

where Z_t is a latent variable that encodes future scene dynamics. To sample from the joint distribution, we first draw scene latent samples $Z_{(k)}^t \sim P(Z^t|X^t)$, then decode actor plans $\mathcal{Y}_{(k)}^t = f(X^t, Z_{(k)}^t)$ with a deterministic decoder [11]. Thus, we can generate K scene-consistent samples of actor plans efficiently in one stage of parallel sampling. Furthermore, to learn this latent variable model, we introduce a posterior latent distribution $q(Z^t, |X^t, \mathcal{Y}^t)$ to leverage variational inference [27, 37]. Intuitively, it learns to map ground truth future $\mathcal{Y}_{GT}^t$ to the scene latent space for best reconstruction.

We leverage the graph neural network (GNN) based scene interaction module introduced by [10] to parameterize the prior network $p_\gamma(Z^t|X^t)$, posterior network $q_\phi(Z^t, |X^t, \mathcal{Y}^t)$, and the deterministic decoder $\mathcal{Y}^t = f_\theta(X^t, Z^t)$, for encoding to and decoding from the scene-level latent variable Z^t. By propagating messages across a fully connected interaction graph with actors as nodes, the latent space learns to capture not only individual actor goals and style, but also multi-agent interactions. More concretely, we partition the latent space to learn a distributed representation $Z^t = \{z_1, z_2, ..., z_N\}$ of the scene, where z_n is spatially anchored to actor n and captures unobserved dynamics most relevant to that actor. This choice enables effective relational reasoning across a large and variable number of actors and diverse map topologies (i.e., to deal with the complexity of urban traffic). Additional implementation details can be found in the supplementary.

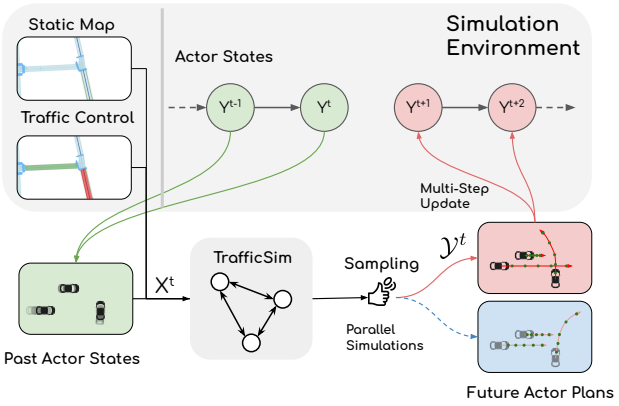

Figure 4. TRAFFICSIM models all actors jointly to simulate realistic traffic scenarios through time. We can sample at each timestep to obtain parallel simulations.

3.3. Simulating Traffic Scenarios

We model each traffic scenario as a sequential process where traffic actors interact and plan their behaviors at each timestep. Leveraging the differentiable observation module and joint actor policy, we can generate traffic scenarios by starting with an initial history of the actors $Y^{-H:0}$ and simulating their motion forward for T steps. Concretely, at each timestep t, we first extract scene context X^t, then sample actor plans $\mathcal{Y}^t \sim P_{\theta,\gamma}(\mathcal{Y}^t|X^t)$ from our joint actor policy, shown in Figure 4. Since our policy produces a T_{plan}-timestep plan of the future $\mathcal{Y}^t = \{Y^{t+1}, ..., Y^{t+T_{\text{plan}}}\}$, we can either use the first timestep of the joint plan Y^{t+1} to update the simulation environment at the highest frequency, or take multiple steps $\{Y^{t+1}, ..., Y^{t+\kappa}\}$ for faster simulation with minimal loss in simulation quality:

$$P(Y^{1:T}|Y^{-H:0}, \mathcal{M}, \mathcal{C}) = \prod_{t \in \mathcal{T}} P(Y^{t+1:t+\kappa}|X^t) \qquad (2)$$

We provide further discussion on trading off simulation quality and computation in Section 4.

3.4. Learning from Examples and Common Sense

In this section, we describe our approach for learning multi-agent behaviors by leveraging large-scale datasets of human driving behaviors. We train by unrolling our policy (i.e., in closed-loop) and exploiting our fully differentiable formulation to directly optimize with back-propagation through the simulation across time. Furthermore, we pro-

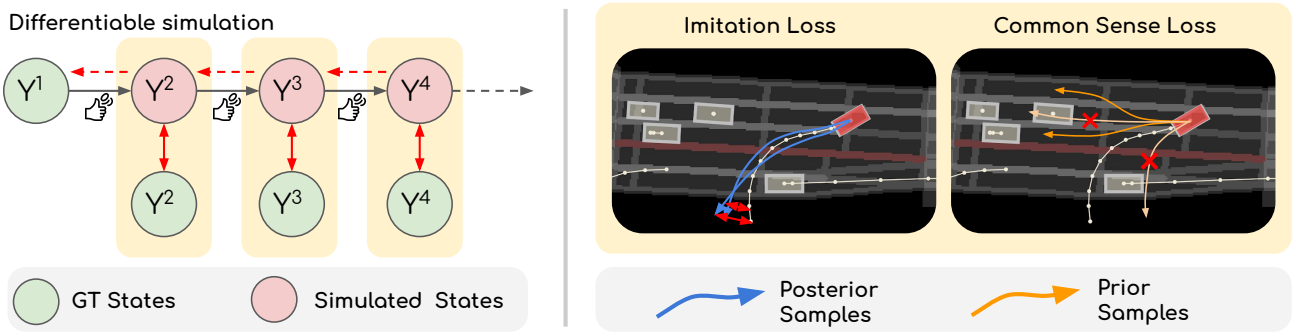

Figure 5. We optimize our policy with back-propagation through the differentiable simulation (**left**), and apply imitation and common sense loss at each simulated state (**right**).

pose a multi-task loss that balances between learning from demonstration and injecting common sense.

Back-Propagation through Differentiable Simulation: Learning from demonstration via behavior cloning yields good open-loop behaviors (i.e., accurate $P(\mathcal{Y}^t|X^t)$ when X^t comes from the observation distribution), but can suffer from compounding error in closed-loop execution [36] (i.e., when X^t is induced by the policy). To bridge this gap, we propose to unroll the policy for closed-loop training and compute the loss $\mathcal{L}^t$ at each timestep t, as shown in Figure 5 (left). Since we model state transitions in a fully differentiable manner, we can directly optimize the total loss with back-propagation through the simulation across time. In particular, the gradient is back-propagated through action sampled from the policy at each timestep via reparameterization. This gives a direct signal for how current decision influences future states.

Augmenting Imitation with Common Sense: Pure imitation suffers from poor supervision when a stochastic policy inevitably deviates from the observed realization of the scenario. Furthermore, inherent bias in the collected data (e.g., lack of safety critical scenarios) means pure imitation can not reason about the danger of collision. Thus, we augment imitation with an auxiliary common sense objective, and use a time-adaptive multi-task loss to balance the supervision:

$$\mathcal{L} = \sum_t \lambda(t)\mathcal{L}^t_{\text{imitation}} + (1-\lambda(t))\mathcal{L}^t_{\text{collision}} \tag{3}$$

Through the simulation horizon, we anneal $\lambda(t)$ to favor supervision from common sense over imitation. In this work, we focus on using collision avoidance as the common sense objective. Other forms (e.g., map-based losses [12]) are left to future work. We now describe both objectives in details, also shown in Figure 5 (right).

Imitation Objective: To learn from demonstrations, we adapt the variational learning objective of the CVAE framework [37] and optimize the evidence-based lower bound (ELBO) of the log likelihood $\log P(\mathcal{Y}^t|X^t)$ at each timestep t. Concretely, the imitation loss consists of a reconstruction component and a KL divergence component:

$$\mathcal{L}^t_{\text{imitation}}(\mathcal{Y}^t_{\text{post}}, \mathcal{Y}^t_{\text{GT}}) = \mathcal{L}^t_{\text{recon}} + \beta \cdot \mathcal{L}^t_{\text{KL}} \tag{4}$$

$$L^t_{recon} = \sum_a^N \sum_{\tau=t+1}^{t+P} L_\delta(y^\tau_a - y^\tau_{a,GT}) \tag{5}$$

$$L^t_{KL} = \text{KL}\left(q_\phi\left(Z^t|X^t, \mathcal{Y}^t_{GT}\right)||p_\gamma\left(Z^t|X^t\right)\right) \tag{6}$$

We apply reconstruction loss to the posterior samples $\mathcal{Y}^t_{\text{post}} = f_\theta(X^t, Z^t_{\text{post}})$, where Z^t_{post} is conditioned on ground truth future $\mathcal{Y}^t_{\text{GT}}$. We use Huber loss L_δ for reconstruction and reweight the KL term with β as proposed by [22].

Collision Avoidance Objective: We apply a pair-wise collision loss to prior samples $\mathcal{Y}^t_{prior}$ to directly regularize $P(\mathcal{Y}^t|X^t)$:

$$\mathcal{L}^t_{\text{collision}}(\mathcal{Y}^t_{\text{prior}}) = \frac{1}{N^2}\sum_{i\neq j}\max(1, \sum_{\tau=t+1}^{t+P}\mathcal{L}_{\text{pair}}(y^\tau_i, y^\tau_j)) \tag{7}$$

Importantly, we design an efficient differentiable relaxation to ease optimization. In particular, we approximate each vehicle with 5 circles, and compute L2 distance between centroids of the closest circles of each pair of actors:

$$\mathcal{L}_{\text{pair}}(y^\tau_i, y^\tau_j) = \begin{cases} 1 - \frac{d_{\text{closest}}}{r_i + r_j}, & \text{if } d_{\text{closest}} \leq r_i + r_j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

4. Experimental Evaluation

In this section, we first describe the simulation setup and propose a suite of metrics for measuring simulation quality. We show our approach generates more realistic traffic scenarios as compared to a diverse set of baselines. Notably, training an imitation-based motion planner on synthetic data generated by TRAFFICSIM outperforms in planning L2 as compared to using same amount of real data. This shows

Model		SCR_{12s} (%)	TRV_{12s} (%)	minSFDE (m)	minSADE (m)	meanSFDE (m)	meanSADE (m)	$MASD_{12s}$ (m)
Heuristic	IDM [39]	1.19	**0.25**	4.97	3.03	5.39	3.48	4.01
Motion Forecasting	MTP [17]	11.00	9.67	2.19	1.47	2.77	2.19	**7.15**
	ESP [35]	4.08	4.79	3.42	1.56	3.52	1.60	0.29
	ILVM [11]	2.90	4.37	2.56	1.33	2.92	1.50	1.17
Imitation Learning	AdversarialIL	10.05	8.34	2.89	1.19	3.87	1.51	4.89
	DataAug	3.78	8.23	2.04	1.22	2.62	1.56	2.29
Ours	TRAFFICSIM	**0.50**	2.77	**1.13**	**0.57**	**1.75**	**0.85**	2.50

Table 1. **[ATG4D]** Comparison against existing approaches ($S = 15$ samples, $T = 12$ seconds, $T_{\text{label}} = 8$ seconds)

there's minimal behavior gap between TRAFFICSIM and the real world. Lastly, we study how to tradeoff between simulation quality and computation.

Dataset: We benchmark our approach on a large-scale self-driving dataset ATG4D, which contains more than one million frames collected over several cities in North America with a 64-beam, roof-mounted LiDAR. Our labels are very precise 3D bounding box tracks. There are 6500 snippets in total, each 25 seconds long. In each city, we have access to high definition maps capturing the geometry and the topology of each road network. We consider a rectangular region of interest centered around the self-driving vehicle that spans 140 meters along the direction of its heading and 80 meters across. The region is fixed across time for each simulation.

Simulation Setup: In this work, we use real traffic states from ATG4D as initialization for the simulations. This give us realistic actor placement and dynamic state, thus controlling for domain gap that might arise from initialization. We subdivide full snippets into 11s chunks, using the first 3s as the initial states $Y^{-H:0}$, and the subsequent $T_{\text{label}} = 8s$ as expert demonstration for training. We run the simulation forward for $T = 12s$ for both training and evaluation, and use the full training episode for back-propagation learning. We use $\delta_t = 0.5s$ as the duration for a simulation tick (i.e., simulation frequency of $2Hz$). We use observed traffic light states from the log snippets for simulation.

Baselines: We use a wide variety of baselines. The Intelligent Driver Model (IDM) [39] is a heuristic car-following model that explicitly encode traffic rules. We adapt three state-of-the-art motion forecasting models for traffic simulation. MTP [17] models multi-modal futures, but assume independence across actors. ESP [35] models interaction at the output level, via social auto-regressive formulation. ILVM [11] models interaction using a scene-level latent variable model. Finally, we consider imitation learning techniques that have been applied to driving behavior modelling. Following [8, 16, 2], DataAug adds perturbed trajectories to help the policy learn to recover from mistakes. Inspired by [23, 6, 3], AdversarialIL learns a discriminator as supervision for the policy. We defer implementation details to the supplementary.

4.1. Metrics

Evaluating traffic simulation is challenging since there is no singular metric that can fully capture the quality of the generated traffic scenarios. Thus we propose a suite of metrics for measuring the diversity and realism, with a particular focus on *coverage* of real world scenarios. For all evaluations, we sample $K = 15$ scenarios from the model given each initial condition. More concretely, we create batches of K scenarios with the same initialization. Then at each timestep, we sample a single $\mathcal{Y}^t_{(k)}$ from $P(\mathcal{Y}^t_{(k)}|X^t_{(k)})$ for each scenario (k), all in parallel. After unrolling for $\frac{T}{\delta_t}$ steps, we obtain the full scenarios. We provide implementation details in the supplementary.

Interaction Reasoning: To evaluate the consistency of the actors' behaviors, we propose to measure the scenario collision rate (SCR): the average percentage of actors in collision in each sampled scenario (thus lower being better). Two actors are considered in collision if the overlap between their bounding boxes at any time step is higher than a small IOU threshold.

Traffic Rule Compliance: Traffic actors should comply with traffic rules. Thus, we propose to measure traffic rule violation (TRV) rate, and focus on two specific traffic rules: 1) staying within drivable areas, and 2) obey traffic light signals.

Scenario Reconstruction: We use distance-based scenario reconstruction metric to evaluate the model's ability to sample a scenario close to the ground truth. (i.e., recovering irregular maneuvers and complex interactions collected from the real world). For each scenario sample, we calculate average distance error (ADE) across time, and final distance error (FDE) at the last labeled timestep. We calculate min-

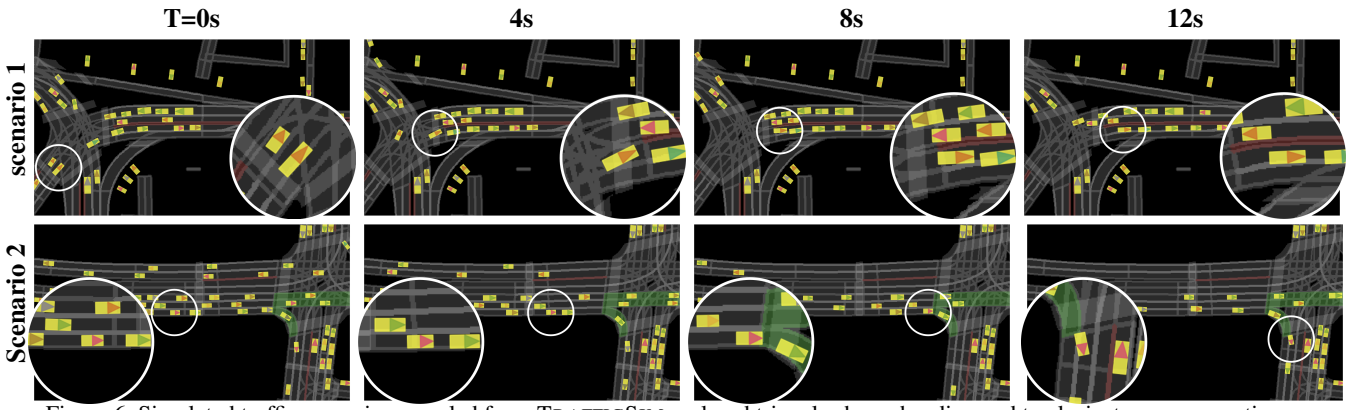

Figure 6. Simulated traffic scenarios sampled from TRAFFICSIM: colored triangle shows heading and tracks instances across time

Model	T_{plan} (timesteps)	Unroll in Training	Common Sense	SCR_{12s} (%)	TRV_{12s} (%)	minSFDE (m)	minSADE (m)	meanSFDE (m)	meanSADE (m)	MASD_{12s} (m)
$\mathcal{M}_0$	1			5.92	10.19	2.04	0.88	2.50	1.04	0.80
$\mathcal{M}_1$	10			2.32	3.43	1.72	0.99	2.09	1.29	2.40
$\mathcal{M}_2$	1	✓		1.28	3.30	**1.02**	**0.54**	**1.70**	0.88	**3.57**
$\mathcal{M}_3$	10	✓		0.60	3.02	1.21	0.58	1.70	**0.84**	2.16
$\mathcal{M}^*$	10	✓	✓	**0.50**	**2.77**	1.13	0.57	1.75	0.85	2.50

Table 2. **[ATG4D]** Ablation study ($S = 15$ samples, $T = 12$ seconds, $T_{\text{label}} = 8$ seconds)

SADE/minSFDE by selecting the best matching scenario sample, and meanSADE/meanSFDE by averaging over all scenario samples.

Diversity: Following [43], we propose a map-aware average self-distance (MASD) metric to measure how different sampled simulations are, when conditioned on a specific scenario initialization. Concretely, we measure the average distance between the two *most distinct* samples that do not violate traffic rules. Higher MASD is desired, but it is only meaningful when the diverse samples have similar level of realism (e.g., SCR, TRV).

4.2. Experimental Results

Comparison Against Existing Approaches: Table 1 shows quantitative results. Car following models generate collision free behavior that strictly follows traffic rules. But they do not recover naturalistic driving, and thus score poorly on scenario reconstruction metrics. Motion forecasting models recover accurate traffic behavior, but exhibit unrealistic interactions and traffic rule violations when unrolled for a long simulation horizon. Imitation learning techniques attempt to bridge the gap between train and test, and thus results in marginally better scenario reconstruction as compared to motion forecasting baselines. However, they inject additional bias that results in worse collision rate and traffic rule violation. Our TRAFFICSIM achieves the best of both worlds: best results on scenario reconstruction and interaction, and similar to IDM in traffic rule violation, without directly encoding the rules. We note that the ground truth TRV rate is 1.26%, since human exhibit non-compliant behaviors. Figure 6 shows qualitative visualization of traffic scenarios generated from TrafficSim. Figure 7 shows that TRAFFICSIM can generate samples with irregular maneuvers and complex interactions, which cannot be captured by heuristic models like IDM.

TRAFFICSIM for Data Augmentation: TRAFFICSIM can be used to generate synthetic training data for learning better motion planners. More concretely, we generate $5s$ scenario snippets and train a planner to imitate behaviors of all actors in the scenario. As shown in Table 3, the planner trained with synthetic data generated from TRAFFICSIM significantly outperforms baselines in open-loop planning metrics when evaluated against real scenarios. Most notably, we achieve *lower* planning L2 error, while matching collision rate and progress of planner trained with the same amount of real data. This shows that the scenarios generated from TRAFFICSIM are realistic and have minimal gap from behaviors observed in the real world, and can be used as effective data augmentation. We show more details on this experiment in the supplementary.

Ablation Study: We show the importance of each component of our model and training methodology in Table 2. Open-loop training ($\mathcal{M}_0$ & $\mathcal{M}_1$) performs poorly due to compounding error at test-time. Closed-loop train-

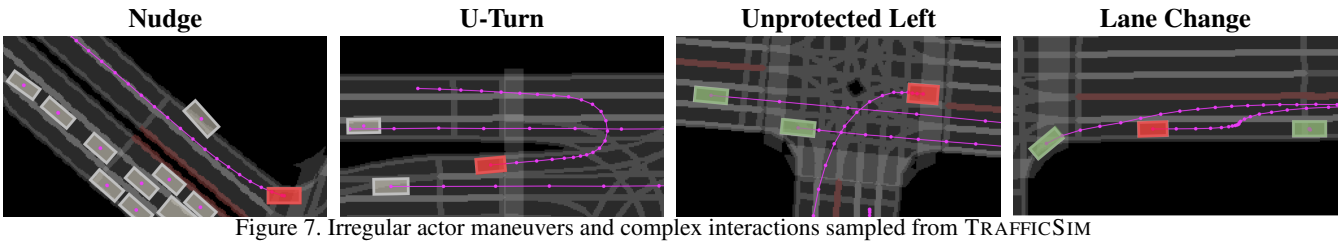

Figure 7. Irregular actor maneuvers and complex interactions sampled from TRAFFICSIM

Training Data	Collision Rate (%)	Planning L2 (m)	Progress (m)
Real	10.56	4.85	31.05
IDM [39]	22.05	10.49	29.17
MTP [17]	19.54	9.89	26.46
ESP [35]	19.38	8.76	24.73
ILVM [11]	14.82	7.16	26.39
AdversarialIL	13.83	5.19	29.02
DataAug	15.75	6.88	26.88
TRAFFICSIM	**10.73**	**4.52**	29.44

Table 3. Imitation planner trained with synthetic data from TRAFFICSIM outperforms real data in planning L2 error.

ing with back-propagation through simulation ($\mathcal{M}_2$) is the most important component in learning a robust policy. Explicitly modelling longer horizon plan (i.e., $\mathcal{Y}^t = \{Y^{t+1}, ..., Y^{t+T_{\text{plan}}}\}$ instead of Y^{t+1}) ($\mathcal{M}_3$) improves interaction reasoning. Augmenting imitation with common sense ($\mathcal{M}^*$) further reduces collision and traffic rule violation rates.

Multi-Step Update for Fast Simulation: We can achieve faster simulation by running model inference once per κ ticks of the simulation. This is possible since TRAFFICSIM explicitly models the actor plans $\mathcal{Y}^t = \{Y^{t+1}, Y^{t+2}, ..., Y^{t+T_{\text{plan}}}\}$ and accurately captures future interactions in the planning horizon T_{plan}, even without extracting scene context at the highest simulation frequency. In particular, we can choose the desired tradeoff between simulation quality and speed by modulating κ at simulation-time without retraining, as long as $\kappa \leq T_{\text{plan}}$. Table 4 shows we can effectively achieve 4x speedup with minimal degradation in simulation quality. Runtime is profiled on a single Nvidia GTX 1080 Ti.

Incorporating Constraints at Simulation-Time: Explicitly modelling actor plans $\mathcal{Y}^t$ at each timestep also makes it easy to incorporate additional constraints at simulation time. In particular, we can define constraints such as avoiding collision and obeying traffic rules over the actor plans $\mathcal{Y}^t$, to anticipate and prevent undesired behaviors in the future. Concretely, we evaluate two optimization methods for avoiding collision: 1) rejection sampling which discard actor plans that collide and re-sample, and 2) gradient-based optimization of the scene latent Z^t to minimize the differentiable

Inference Frequency	Runtime (s)	SCR_{12s} (%)	TRV_{12s} (%)	min SFDE (m)	mean SFDE (m)
2Hz	0.83	**0.50**	**2.77**	1.13	1.75
1Hz	0.45	0.85	3.17	**1.12**	**1.73**
0.5Hz	**0.24**	0.96	3.64	1.16	**1.73**

Table 4. Multi-step update achieves up to 4x speedup with minimal degradation in simulation quality.

Post-Processing	SCR_{12s} (%)	TRV_{12s} (%)	min SFDE (m)	mean SFDE (m)
None	0.50	**2.77**	**1.13**	**1.75**
Rejection Sampling	0.33	3.01	**1.13**	**1.75**
Gradient Optimization	**0.12**	3.00	**1.13**	**1.75**

Table 5. Additional optimization at simulation-time further reduces collision rate.

relaxation of collision. Table 5 shows that both methods are effective in reducing collision while keeping the simulation realistic. More details in supplementary.

TRAFFICSIM for interactive Simulation: We create an interactive simulation tool to showcase how simulation designers can leverage TRAFFICSIM to construct and preview interesting traffic scenarios. In particular, they can alter traffic light states and add, modify, or remove actors during simulation. In response, TRAFFICSIM generates realistic variants of the traffic scenario. Demo in supplementary video.

5. Conclusion

In this work, we have proposed a novel method for generating diverse and realistic traffic simulation. TRAFFICSIM is a multi-agent behavior model that generates socially-consistent plans for all actors in the scene jointly. It is learned using back-propagation through the fully differentiable simulation, by imitating trajectory observations from a real-world self-driving dataset and incorporating common sense. TRAFFICSIM enables exciting new possibilities in data augmentation, interactive scenario design, and safety evaluation. For future work, we aim to extend this work to learn controllable actors where we can specify attributes such as goal, route, and style.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE CVPR*, 2016.

[2] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

[3] F. Behbahani, K. Shiarlis, X. Chen, V. Kurin, S. Kasewa, C. Stirbu, J. Gomes, S. Paul, F. A. Oliehoek, J. Messias, and S. Whiteson. Learning from demonstration in the wild. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 775–781, 2019.

[4] A. Best, S. Narang, Lucas Pasqualin, D. Barber, and D. Manocha. Autonovi-sim: Autonomous vehicle simulation platform with weather, sensing, and traffic control. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1161–11618, 2018.

[5] 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 *2019 International Conference on Robotics and Automation (ICRA)*, pages 789–795, 2019.

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

[7] Raunak P. Bhattacharyya, Blake Wulfe, D. Phillips, Alex Kuefler, J. Morton, Ransalu Senanayake, and M. Kochenderfer. Modeling human driving behavior through generative adversarial imitation learning. *ArXiv*, abs/2006.06412, 2020.

[8] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.

[9] Kiante Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized imitation learning. In *International Conference on Learning Representations*, 2020.

[10] Sergio Casas, Cole Gulino, Renjie Liao, and R. Urtasun. Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9491–9497, 2020.

[11] Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020.

[12] Sergio Casas, Cole Gulino, Simon Suo, and Raquel Urtasun. The importance of prior knowledge in precise multimodal prediction. *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[13] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, 2018.

[14] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 86–99. PMLR, 30 Oct–01 Nov 2020.

[15] Qianwen Chao, Huikun Bi, Weizi Li, Tianlu Mao, Zhaoqi Wang, and Ming Lin. A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. *Computer Graphics Forum*, 07 2019.

[16] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700, 2018.

[17] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096, 2019.

[18] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, and Jeff Schneider. Motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1808.05819*, 2018.

[19] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 1–16. PMLR, 13–15 Nov 2017.

[20] P. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B-methodological*, 15:105–111, 1981.

[21] Mikael Henaff, Alfredo Canziani, and Yann LeCun. Model-predictive policy learning with uncertainty regularization for driving in dense traffic. In *International Conference on Learning Representations*, 2019.

[22] I. Higgins, Loïc Matthey, A. Pal, Christopher P. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[23] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4572–4580, 2016.

[24] Arne Kesting and Martin Treiber. Calibrating car-following models by using trajectory data: Methodological study. *Transportation Research Record*, 2088(1):148–156, 2008.

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

[26] Arne Kesting, Martin Treiber, and Dirk Helbing. Agents for traffic simulation. *Multi-agent systems: Simulation and applications*, pages 325–356, 2009.

[27] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[28] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099. IEEE, 2015.

[29] L. L. Li, B. Yang, M. Liang, W. Zeng, M. Ren, S. Segal, and R. Urtasun. End-to-end contextual perception and prediction with interaction transformer. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5784–5791, 2020.

[30] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[31] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 2018.

[32] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[33] Mark Martinez, Chawin Sitawarin, Kevin Finch, Lennart Meincke, Alex Yablonski, and Alain L. Kornhauser. Beyond grand theft auto V for training, testing and enhancing deep learning in self driving cars. *CoRR*, abs/1712.01397, 2017.

[34] Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems 1*, pages 305–313. Morgan-Kaufmann, 1989.

[35] N. Rhinehart, R. Mcallister, K. Kitani, and S. Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2821–2830, 2019.

[36] Stéphane Ross, Geoffrey Gordon, and Drew 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*, pages 627–635, 2011.

[37] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[38] Charlie Tang and Russ R Salakhutdinov. Multiple futures prediction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[39] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, Aug 2000.

[40] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158, 2000.

[41] Bernhard Wymann, Eric Espié, Christophe Guionneau, Christos Dimitrakakis, Rémi Coulom, and Andrew Sumner. Torcs, the open racing car simulator. *Software available at http://torcs. sourceforge. net*, 4(6):2, 2000.

[42] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE CVPR*, 2018.

[43] Ye Yuan and Kris M. Kitani. Diverse trajectory forecasting with determinantal point processes. In *International Conference on Learning Representations*, 2020.