

BITS: Bi-level Imitation for Traffic Simulation

Danfei Xu^{1,2*}, Yuxiao Chen^{1*}, Boris Ivanovic¹, and Marco Pavone^{1,3}

Abstract—Simulation is the key to scaling up validation and verification for robotic systems such as autonomous vehicles. Despite advances in high-fidelity physics and sensor simulation, a critical gap remains in simulating realistic *behaviors* of road users. This is because devising first principle models for human-like behaviors is generally infeasible. In this work, we take a data-driven approach to generate traffic behaviors from real-world driving logs. The method achieves high sample efficiency and behavior diversity by exploiting the bi-level hierarchy of high-level intent inference and low-level driving behavior imitation. The method also incorporates a planning module to obtain stable long-horizon behaviors. We empirically validate our method with scenarios from two large-scale driving datasets and show our method achieves balanced traffic simulation performance in realism, diversity, and long-horizon stability. We also explore ways to evaluate behavior realism and introduce a suite of evaluation metrics for traffic simulation. Finally, as part of our core contributions, we develop and open source a software tool that unifies data formats across different driving datasets and converts scenes from existing datasets into interactive simulation environments. For video results and code release, see <https://bit.ly/3L9uzj3>.

I. INTRODUCTION

Simulation is an integral part of developing effective robotic systems. Simulators allow developers to rapidly verify changes and triage erroneous behaviors before deploying to physical systems. Realistic simulators are especially crucial for autonomous vehicles (AVs), because it is costly and potentially dangerous to test new features and changes directly on the road. Yet, despite advances in physics simulation and high-fidelity sensor simulation, AV developers still primarily rely on large-scale, real-world road testing for validation and verification [1]–[7]. One critical reason why is that existing simulation platforms do not generate realistic *behaviors* for simulated road users, such as cars and pedestrians, which is difficult because, unlike physics and graphics, it is challenging to design models that generate human-like behaviors from first principles.

Today’s mainstream driving simulators synthesize agent behaviors by either replaying recorded driving logs or implementing heuristics-based controllers. While log replay allows for scenario-specific triaging, it is difficult to validate new features as replayed agents do not react to counterfactual ego motions. On the other hand, heuristics-based controllers are often equipped with simple driving logic and can thus

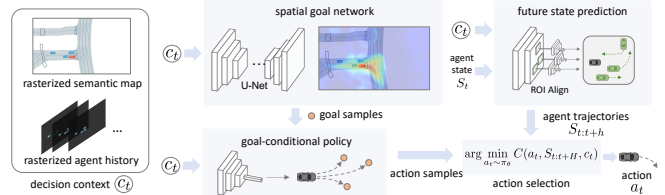


Fig. 1. **BITS framework overview:** Decision context c_t is a tensor containing the semantic map and rasterized agent history concatenated channel-wise. Given c_t as input, (1) the spatial goal network produces a 2D spatial distribution of short-horizon goals, (2) the goal-conditional policy generates a set of actions for each sampled goal, (3) a trajectory forecasting model predicts the future motion of the neighboring agents, and finally (4), based on the predicted future states, the framework selects the set of actions that minimizes a rule-based cost function.

respond to new ego behaviors in a closed-loop manner [8]–[10]. While these methods can produce plausible traffic flows, synthesizing diverse and complex driving behaviors such as yielding and cutting-in for a large number of real-world scenarios remains challenging.

On the other hand, learning-based approaches can learn reactive behaviors from real-world driving logs. For example, recent works show that trajectory forecasting models trained from large-scale driving logs can infer distributions of future agent trajectories in challenging scenarios [11]–[14]. While these methods excel at predicting realistic trajectories, they are brittle under domain shifts such as new scenes and unseen behaviors, and the multi-agent nature of traffic simulation may cause a combinatorial explosion in the number of agent states. This challenge is exacerbated when applying prediction approaches to closed-loop behavior simulation over long time horizons, as prediction errors at each step compound over time [15], leading to divergence and irreversible failures such as collisions and driving off-road.

In this work, we aim to develop a learning-based traffic simulation model that can generate diverse, stable, and realistic traffic behaviors. Our key insight is two-fold. First, while learning stable long-horizon driving behaviors requires large amounts of data, the problem has a natural bi-level hierarchy that can be exploited to improve learning efficiency. Specifically, we decouple the learning problem into high-level intent inference and low-level goal-conditioned control. Our model leverages a 2D birds-eye-view (BEV) of urban driving and learns to generate a spatial distribution of intended goal locations (Fig. 1). A low-level policy then generates short segments of goal-conditioned behaviors that move agents towards their inferred goals. In addition, the spatial goal distribution aids in disentangling multi-modal behaviors and generates diverse traffic simulations. While such a hierarchical policy greatly improves learning efficiency, agents

* equal contribution

²Danfei Xu is with Georgia Institute of Technology and NVIDIA research danfei@gatech.edu, danfeix@nvidia.com

¹Yuxiao Chen and Boris Ivanovic are with NVIDIA research, {yuxiaoc,bivanovic}@nvidia.com

³Marco Pavone is with Stanford University and with NVIDIA Research. {pavone@stanford.edu, mpavone@nvidia.com}

may still encounter unseen situations that the model alone struggles to handle. To stabilize the long-term behaviors of the model, we augment the policy with a prediction-and-planning module that samples likely actions from the hierarchical policy and selects actions with rule-following costs as regularization. This way, the overall framework balances between generating human-like behaviors and preventing divergences at out-of-distribution states.

We name our method **BITS (Bi-level Imitation for Traffic Simulation)**. We evaluate BITS on two popular driving log datasets, Lyft Level 5 [16] and nuScenes [17]. The Lyft dataset contains 1000 hours of driving data collected along a 6.8 mile route in Palo Alto. While densely covered, the trajectory annotations are auto-labeled using a perception stack, resulting in abundant labeling errors [18]. In contrast, the nuScenes dataset contains 5.5 hours of manually-annotated trajectories with more diverse scenarios. Through these two datasets, we demonstrate the capability of our method under different types of learning challenges. Beyond generating realistic traffic behaviors, we also explore ways to evaluate the generated behaviors through a suite of analytical and learned metrics, since conventional trajectory generation metrics such as ADE and FDE are ill-fitted for evaluating closed-loop traffic simulation. Finally, as part of our core contributions, we develop and open source a software tool `trajdata` that unifies data formats across different AV datasets and allows users to transform scenarios from existing datasets into interactive simulation environments. We hope that our novel traffic behavior simulation method, along with the evaluation protocol and data interface software, can serve as a foundation for further research on this topic.

II. RELATED WORK

Traffic simulation. Approaches for traffic simulation can broadly be categorized into two groups: *macroscopic*, emphasizing on large scale traffic flows over individual agent states, and *microscopic*, studying individual road user behaviors [19], [20], which is the focus of our work. Existing systems for microscopic simulation such as SUMO [8], Aimsun [9], and VISSIM [10] use analytical models to control agents in a scene, including cellular automata, the intelligent driver model (IDM), and the optimal velocity model [21]. These analytic models typically have fixed routes for vehicles to follow and separate longitudinal and lateral motions of agents, with some models omitting lateral motion entirely. Accordingly, these analytical microscopic traffic simulation tools lack sufficient complexity and expressiveness for developing or evaluating autonomous driving features.

Recent works have started to explore more expressive models [22]–[24] and learn from real-world trajectory datasets [16], [17], [25]. Notably, STRIVE [24] proposes to generate near-collision scenarios by searching in the latent space of a trajectory prediction model. TrafficSim [23] adapts a graph-based trajectory prediction model to perform scene-level traffic simulation. Symphony [26] applies a similar hierarchical structure and rely on a parallel beam search with a trained discriminator to obtain the actions for all the agents

in the scene. Compared to STRIVE, which focuses on generating worst-case scenarios and short open-loop trajectories, we aim to synthesize a broad range of long-horizon closed-loop traffic behaviors. Compared to Symphony, which uses an online learning setting, our method is trained completely offline, and is thus amenable to scaling to larger and more diverse datasets. Compared to TrafficSim and Symphony, where both methods rely on scene-level control to ensure consistent interaction among agents (e.g., coordinated collision avoidance), our method enables each agent to act without coordination with others at the model level. Such an agent-centric setup allows our method to be deployed in more practical simulation use cases, where different types of agents (analytical or learned) are mixed together to interact in a scene. Moreover, we also show that our method outperforms TrafficSim in simulation diversity and stability.

Trajectory prediction. A separate set of research aims to predict the trajectories of agents in traffic by learning from driving logs. Several early examples include Social GAN [27], GAIL [28], MFP [29], and DESIRE [30]. More recent trajectory prediction models can be further categorized into *agent-centric* models, which generate independent predictions for each agent in a scene [30]–[32], and *scene-centric* models, which generate joint predictions for all (or a subset of) agents in a scene [33], [34]. While these methods perform well over short time-horizons (up to 5s), their performance generally degrades over longer time-horizons. To combat this, many recent state-of-the-art methods adopt a multi-stage approach which first predicts agents’ goal locations and subsequently links agents’ current positions to their inferred goals [35]–[43]. While these methods solely focus on open-loop prediction, we will show in this work that inferring and conditioning on an agent’s goal also improves the stability of long-horizon closed-loop simulation.

Imitation learning. Our bi-level imitation learning method is heavily inspired by literature in hierarchical decision making and multimodal imitation learning. A hierarchical policy consists of a high-level planner that sets abstract goals and a low-level policy that learns to achieve the goals [44]–[50]. Methods along this vein have favorable properties such as compositionality [49] and interpretability [51], and they have achieved superior performance in long-horizon tasks especially [47]–[50]. At the same time, we seek behavior diversity in traffic simulation, which most prior works neglect. Multimodal imitation learning was recently studied in the manipulation domain [52]–[55]. Notably, GTI [54] trains a CVAE-based planner to set multi-modal subgoals for a low-level controller to achieve. Our method adopts a similar hierarchical structure but instead exploits the domain structure of driving to efficiently represent goal distributions as 2D BEV spatial maps (shown in Fig. 1). We empirically show that our method generates realistic, diverse, and stable long-horizon traffic simulations.

III. BI-LEVEL IMITATION FOR TRAFFIC SIMULATION

In this section, we dive into the details of the traffic simulation problem and our primary technical contributions.

We propose (1) a hierarchical imitation learning framework that generates diverse and realistic traffic behaviors, (2) a prediction-and-planning module that stabilizes long-horizon simulation, and (3) a suite of analytical and learned metrics for traffic simulation. Specific implementation details (network architectures, etc.) can be found in Sec. IV-A.

A. Traffic simulation as imitation learning

We take an agent-centric approach to traffic simulation, i.e., each agent makes decisions in a decentralized manner without explicit coordination. As mentioned previously, this allows for flexible integration with existing simulation frameworks containing other types of simulated agents and encourages the emergence of new interactive behaviors. We focus on simulating vehicle traffic in this work, but can easily extend to other agent types (e.g., cyclists, pedestrians). We use s and c to denote the dynamic state and decision-relevant context for an agent, respectively. Specifically, state s includes the position, heading, and velocity of an agent. Context $c = (I, S)$ includes a local semantic map I and the h previous states of an agent and its N neighboring agents $S_{t-h:t} = \{s_{t-h:t}^{(0)}, s_{t-h:t}^{(1)}, \dots, s_{t-h:t}^{(N)}\}$. Given the decision context information c_t and the current state s_t , the goal of a traffic simulation model π_θ is to generate the next state of the agent $s_{t+1} = \mathcal{T}(\pi_\theta(c_t), s_t)$ subject to a dynamics transition function $\mathcal{T}(\cdot)$. We use a simple unicycle model with dynamics constraints as \mathcal{T} and defer more realistic vehicle dynamics models to future works.

We leverage driving logs captured in the real world [16], [17] to train our traffic model. Since log data readily includes semantic maps and the trajectories of all observed agents, we can treat driving logs as a set of multi-agent expert demonstration sequences $\tau = \{c_0^{(i)}, s_0^{(i)}, c_1^{(i)}, s_1^{(i)}, \dots, c_T^{(i)}, s_T^{(i)}\}_{i=0}^N$ and formulate traffic simulation as a supervised imitation learning problem. However, the nature of urban driving poses significant technical challenges. First, the decision process is partially-observed as the model does not have access to the underlying intent of the demonstrator and other decision-relevant cues such as the turn signals of other vehicles. Accordingly, action supervisions are inherently ambiguous and are usually modeled with probabilistic distributions [30]–[32]. Although this ambiguity complicates training, effectively modeling action distributions also enables generating diverse counterfactual traffic simulations. Second, since each agent acts without explicit coordination, their joint behaviors create a combinatorial space of possible future states. Such uncertainty makes generating stable traffic simulations extremely challenging. Below, we describe how our approach can generate multimodal simulations with a stochastic hierarchical policy and mitigate uncertainty in state evolution with a prediction-and-planning module.

B. Bi-level imitation learning for multi-modal simulation

The goal of our traffic simulation model is to produce diverse and plausible behaviors by learning from real-world driving logs as demonstrations. Most existing methods in trajectory prediction use deep latent variable models (e.g.,

VAEs) to capture the behavior distributions. However, findings in the imitation learning literature [56] suggest that learning to generate stable long-horizon behaviors requires a large amount of training data. Our method instead decomposes the learning problem into (1) training a high-level goal network that captures the spatial distribution of possible short-term goals, and (2) training a goal-conditional policy that learns to reach the predicted goal. The spatial goal network exploits the 2D BEV of driving motion and represents the spatial goal distribution efficiently with a 2D grid. This decomposition additionally moves the burden of modeling multi-modal trajectories to the high-level goal predictor, enabling the low-level goal-conditioned policy to reuse goal-reaching skills to improve sample efficiency.

Spatial goal network. The spatial goal network is trained to predict the distribution of the short-term goal pose (2D position and heading) $p(\tilde{s}_{t+H}|c_t)$ of an agent given its decision context c_t . Following prior works [22], [57], we first encode the decision context into a rasterized semantic map, which includes the semantic map I and past trajectories rasterized as 2D bounding boxes in additional channels. The model takes as input the rasterized semantic map and outputs a 2D grid of goal likelihood as well as residual to refine the predicted goal location. The output takes the shape of a 4-channel tensor with the same spatial size as the input rasterized map. Channel 0 is the likelihood of the coarse goal, channel 1 and 2 are the (x, y) scalar residual relative to the grid location, and channel 3 is the heading prediction at each grid location. Once a location is selected based on the probability map, the location is corrected by the residual and transformed into a goal pose \tilde{s}_{t+H} in an agent local frame. We treat the 2D location map as a joint distribution and train via cross-entropy loss across locations. The other channels are trained with masked regression losses (e.g., squared error).

Goal-conditional policy. The goal-conditional policy takes the form of a deterministic trajectory generator $s_{t:t+H} = \pi_\theta(c_t, \tilde{s}_{t+H})$. Although we may further augment the policy with stochastic components, we empirically found it unnecessary as the short-term goal largely reduces the uncertainty in prediction. Inspired by prior works [23], [31], instead of directly regressing each state in an agent’s trajectory the model predicts controls (velocity, change of heading) at each future time step and forward integrates them through an agent’s dynamics model (e.g., extended unicycle dynamics [58] for vehicles). The errors between the predicted and reference trajectory are then backpropagated directly through the dynamics model. Overall, this strategy provides a strong dynamically-grounded learning signal which corrects predictions at earlier steps to reduce errors at later steps.

C. Prediction and planning for long-horizon stability

So far, we have described a bi-level imitation learning method that can generate plausible traffic simulations from limited data. The policy can synthesize diverse behaviors by sampling from the multi-modal spatial goal predictor. However, the performance of the policy remains bounded by the size and coverage of training data. Driving logs are biased

towards nominal behaviors and contain almost no safety-critical situations such as collisions or driving off-road. The objective of generating diverse behaviors further amplifies this challenge, as agents are encouraged to enter previously-unseen regions of the map and create new interactions. As a result, to achieve stable long-horizon simulations, agents must generate reasonable behaviors even at states where guidance from training data is lacking.

To this end, we propose to augment the policy with a *prediction-and-planning* module to stabilize long-horizon rollouts. The module draws action samples a_t from the stochastic bi-level policy π_θ described above and selects the action that minimizes a rule-based cost function C given the predicted future states of the environment $S_{t:t+H}$, that is, $\arg \min_{a_t \sim \pi_\theta} C(a_t, S_{t:t+H}, c_t)$. This approach is similar to a typical AV planning algorithm, with the important difference that we use our learned policy to generate human-like motion trajectory candidates. The key idea is that the policy π_θ can directly follow the data likelihood at in-distribution states, where most action samples are rule-following, and receive corrective guidance at states where action samples may lead to bad consequences. In addition, the sampling module allows for agile adjustment of the simulator (e.g. level of diversity, emphasis on multiple objectives) without retraining. Below we describe the model for future state prediction and the cost function for action selection.

Future state prediction. Since we assume a known vehicle model and static map, the main task of the model is to predict the future motion trajectories of nearby agents. We follow a typical trajectory prediction pipeline and featurize each agent by its local and global scene context. Specifically, we use RoIAlign [59] to crop the features extracted by the intermediate layer of a deep CNN. The per-agent features are then concatenated with a global scene context feature (extracted by the final layer of the same CNN) to make the final trajectory prediction $s_{t:t+H}^{(i)}$ for each neighboring agent i . The model is illustrated in Fig. 1. We use deterministic prediction in this work and defer more sophisticated probabilistic prediction and planning to future work.

Cost-based trajectory selection. We consider two rule-based costs: collision and road departure. The collision cost is computed by approximating the minimum distance between two bounding rectangles. To calculate the road departure cost, we first generate a distance map that records the Manhattan distance to the drivable area in pixel space (obtained efficiently via D convolution steps). The resulting distance map assigns zero to points within the drivable area, with values increasing outside the drivable area until saturating at D . We directly use this Manhattan distance value as a penalty. Note that both cost terms are zero for nominal trajectories, i.e., trajectories that do not result in collision and road departure, minimizing their effect on selecting among rule-following action samples.

D. Evaluation metrics for traffic simulation

Designing metrics for simulation is particularly difficult due to the lack of ground truth. As a result, metrics such

as average displacement error (ADE), commonly used for prediction, do not suit the task of simulation. To address this gap, we propose three types of simulation metrics: (i) metrics related to common traffic rules violation, such as driving off-road or colliding with other agents; (ii) metrics measuring the statistics of simulation rollouts such as speed profile, coverage of the driving area, and behavior diversity between different simulation trials; (iii) data-driven metrics learned from real-world driving logs that measure trajectory likelihood. Here we describe (ii) and (iii) in details.

Coverage and diversity. To calculate how much of a scene is covered by the agents, we first compute a simulator’s trajectory distribution via Kernel Density Estimation on a 2D spatial grid over all time steps. We then count the number of grid points where the density is above a threshold for drivable areas. To measure the diversity of simulations, we run multiple trials with the same initial condition and record the density profile ρ_i for each trial i . Given two different trials, we compute the Wasserstein distance between their density profiles with [60]. For n trials of the same scene, we calculate the pairwise Wasserstein distance among density profiles and take the mean as the diversity score: Diversity = $\frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \text{Wass}(\rho_i, \rho_j)$, where $\text{Wass}(\cdot, \cdot)$ is the Wasserstein distance.

Learned metrics. There are many potential ways to learn a metric that evaluates whether simulated behavior is human-like. One such way is to evaluate the trajectory likelihoods using a model trained from real-world driving logs. To evaluate this possibility, we develop an occupancy-based prediction model that predicts where an agent will be in future time steps. The model uses a similar structure to our spatial goal network and discretizes the position space into bins. The model is trained to minimize a cross-entropy loss function and its predictions are then used to compute the trajectory likelihoods of simulated trajectories.

IV. EVALUATION

Our experiments seek to validate the primary claims that (1) BITS can generate plausible behaviors by learning from real-world driving logs, (2) compared to flat policies, our hierarchical policy achieves better sample efficiency and behavior diversity, (3) our proposed prediction-and-planning module is effective at providing guidance for out-of-distribution states. We conduct evaluations with two large-scale real-world driving datasets, Lyft Level 5 [16] and nuScenes [17]. Since learning-based traffic simulation is a new topic and lacks a standardized benchmark, we also develop and open source a software that unifies data formats across different AV datasets (starting with the two used in this work) and can transform static scenes from datasets to interactive simulation environments. We use this framework to run closed-loop simulations and report performance based on the metrics described in Sec. III-D.

A. Evaluation setup

Datasets. The Lyft dataset [16] contains 1000 hours of driving data collected along a 6.8 mile route in Palo Alto.

TABLE I
QUANTITATIVE RESULTS ON THE LYFT DATASET [16]

	FR↓	coll↓	cov↑	div↑	speed	jerk	sADE
SimNet [22]	38.35	35.57	460.44	0.00	1.65	3.85	3.19
SocialGAN [27]	64.96	42.86	189.98	9.02	1.48	5.17	13.22
SocialGAN+p	69.41	42.96	131.47	7.64	1.51	4.78	13.90
TPP [31]	15.62	14.65	495.69	3.23	1.14	2.60	5.50
TPP+p	16.03	15.12	508.16	2.75	1.23	2.32	5.44
TrafficSim [23]	26.98	15.98	566.35	7.68	1.50	3.82	5.89
TrafficSim+p	22.97	13.58	617.50	7.96	1.73	3.09	6.04
BITS (max)	20.71	18.75	443.93	0.00	0.76	4.44	7.77
BITS (sample)	25.37	22.37	780.52	16.84	1.86	4.29	7.78
BITS	9.97	8.66	1014.43	22.94	1.96	3.75	11.21
Dataset	18.36	17.49	327.25	0.00	0.00	0.00	0.00

TABLE II
QUANTITATIVE RESULTS ON THE nuSCENES DATASET [17]

	FR↓	coll↓	cov↑	div↑	speed	jerk	sADE
SimNet [22]	24.58	15.80	395.21	0.00	6.81	18.61	7.01
SocialGAN [27]	71.33	33.81	154.21	2.29	38.44	42.51	19.89
SocialGAN+p	71.86	35.84	151.61	2.32	37.07	42.48	19.76
TPP [31]	49.76	33.23	489.36	7.06	8.37	4.61	12.13
TPP+p	9.84	9.52	661.07	7.73	8.17	5.96	13.09
TrafficSim [23]	27.08	20.39	861.00	4.28	10.03	10.01	10.51
TrafficSim+p	25.97	18.45	933.53	4.43	10.44	11.93	11.06
BITS (max)	13.24	11.51	559.58	0.00	6.35	13.87	6.41
BITS (sample)	14.72	13.79	888.12	6.36	6.54	13.44	6.70
BITS	6.63	5.67	1122.94	8.40	6.91	13.20	8.39
Dataset	11.99	11.39	397.92	0.00	0.00	0.00	0.00

The dataset contains many repeated trips along each road segment. Since the annotations are auto-generated using a perception stack, there are many labeling errors including inaccurate agent positions, headings, and semantic types [18]. In contrast, nuScenes [17] contains 5.5 hours of accurate manually-labeled trajectories spanning two cities (Boston and Singapore) with more diverse scenarios and denser traffic. Through these datasets, we compare our method to baselines under different learning challenges such as noisy labels and small training sets. For both datasets, we train all models on trajectories from the train split and conduct evaluation on 100 scenes randomly sampled from the validation split. We consider only vehicle simulation in this paper and defer simulating other types of agents (e.g., pedestrians, cyclists) to future works.

Simulation environments. As stated above, we initialize our simulation environments from real driving data, leading to realistic agent placements and dynamic states. Note that all scenes are drawn from the validation split previously unseen to the trained models. Each agent in the scene is independently controlled by replicas of the same model. The simulation runs at a frequency of 10 Hz and results are reported on 20-seconds simulation episodes.

Baselines. We consider methods from both the traffic simulation and trajectory prediction literature. **SimNet** [22] is a deterministic behavior-cloning model for traffic simulation. **TrafficSim** is an agent-centric adaptation of the original traffic simulation method in [23] that features an isotropic Gaussian CVAE. We remove the scene consistency loss in training since we do not assume control over all agents. **SocialGAN** [27] learns to generate trajectories through adversarial imitation. **TPP** is adapted from Trajectron++ [31], comprised of a discrete CVAE with Gaussian trajectory decoder for each discrete mode. We also consider variants of these methods augmented with our planning-and-control

module (marked with “+p”), i.e., selecting future action samples with a cost function. We also evaluate ablations of our method, **BITS (max)** takes the maximum-likelihood action instead of sampling and **BITS (sample)** samples actions for rollouts without the prediction-and-planning module described in Sec. III-C. All methods share the same rasterized input format, ResNet-18 encoder backbone, and MLP-based trajectory decoder networks.

Metrics. As mentioned in Sec. III-D, designing evaluation metrics for traffic simulation is challenging as no single metric alone can measure the quality of simulation, and we cannot easily compare with ground truth trajectories as our goal is to generate new and diverse behaviors. To address this problem, we consider three types of evaluation metrics. *Environment metrics* measure rule violations, environment coverage, and trajectory diversity. Both **coverage** and **diversity** as described in Sec. III-D are calculated from 5 simulation trials with different seeds per scene. We define a critical failure as an agent colliding with other agents or driving off-road for more than 1s. Failure Rate (**FR**) is the average fraction of agents experiencing a critical failure in a scene. We also report raw collision rate (**coll**). *Dataset metrics* compare simulation and ground truth data statistics. They are computed using a normalized Wasserstein distance between the histograms of the driving profiles of the simulated and recorded trajectories. We focus on **speed** and **jerk**, commonly used as driver comfort metrics. We also report scene Average Distance Error (**sADE**) which measures the average position differences between simulated and recorded trajectories. Note that sADE is not suitable for measuring simulation realism and is included only as a reference since it heavily penalizes alternative simulations (e.g., turning left vs. going straight). Finally, *learned metrics* as described in Sec. III-D measure simulation realism based on a likelihood model trained from real-world driving log.

B. Main results

Table I and Table II show quantitative results of closed-loop simulation on Lyft and nuScenes datasets, respectively. Fig. 2 qualitatively visualizes trajectories generated by selected methods and Fig. 3 shows a more detailed analysis on time-to-failure caused by road departure error. We make the following core observations from these results.

Recorded driving data is noisy. As stated above, both datasets contain certain levels of labeling noise indicated by the non-zero failure rates for ground truth data (labeled “Dataset”), with higher noise in Lyft than nuScenes and a majority of errors stemming from vehicle-vehicle collisions due to imprecise bounding box labels. Our method BITS is able to achieve lower failure rates than even the recorded trajectories in both datasets.

Sample efficiency. nuScenes contains far fewer training samples than Lyft, calling for high sample efficiency for a policy to manage compounding errors over long simulations. As shown in Table II, even without the prediction-and-planning module, both variants of BITS achieve low failure rates compared to baselines. For a more direct analysis, we

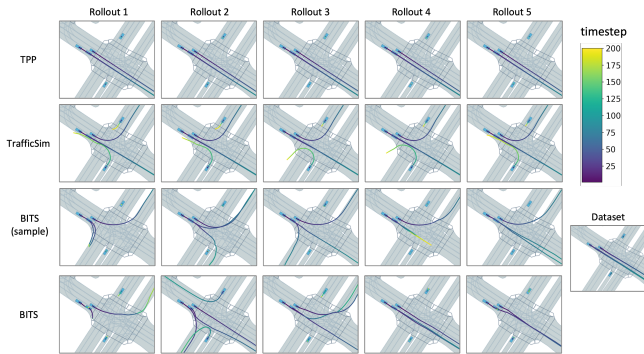


Fig. 2. Trajectories generated by each stochastic method over 5 trials in the Lyft dataset [16]. Our method (BITS, last row) generates diverse and stable long-horizon simulation rollouts (visualized as colored lines emanating from agents). Other methods suffer from a lack of diversity (e.g., TPP [31] on top) or high collision and off-road rates (e.g., TrafficSim [23] in the second row). Agents are represented by blue bounding boxes and trajectory line color denotes simulation timestep.

also report the mean time-to-failure in Fig. 3, observing that failure rates in nuScenes increase significantly over the course of simulation for the non-hierarchical policy baselines and remain low for BITS and its ablations.

BITS generates diverse and stable simulations. We observe that the baseline methods exhibit trade-offs between generating diverse rollouts and overfitting to a single mode of behaviors. For example, in Lyft, TPP suffers from mode collapse which yields low failure rates at the cost of low diversity. TrafficSim achieves relatively high diversity and coverage, but also high failure rates. This observation is corroborated by the visualizations shown in Fig. 2, where all simulation trials by TPP are visually identical and resembles the ground truth (on right, titled “Dataset”), and while the trajectories generated by TrafficSim are diverse, some suffer from collisions and road departures. In contrast, BITS simultaneously attains high diversity and coverage with a low failure rate. This contrast is more pronounced in nuScenes where the training set is small. BITS achieves a balanced performance even without the prediction-and-planning module thanks to its high sample efficiency.

Prediction-and-planning is not always effective. The prediction-and-planning module is generally effective in reducing failure rates, with two important exceptions: (1) when action samples are not diverse, and (2) when all action samples lead to failure. Case (1) is exemplified by TPP in the Lyft environment (Fig. 2), where the model’s predictions overfit to a single behavior mode. Case (2) is exemplified by SocialGAN in both datasets. While the simulations are relatively diverse, they have high failure rates, entailing poor action sample quality. The prediction-and-planning module has negligible effects on the simulation in both cases.

Quantifying behavioral realism. As discussed above, evaluating simulation realism remains a challenging open problem because there is no single correct answer for traffic simulation. Here we consider both dataset statistics and learned metrics as a proxy for quantifying behavioral realism. For Lyft, we see that all methods achieve comparable

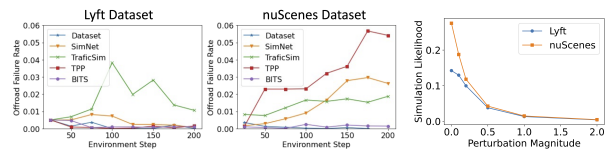


Fig. 3. **Left and Center:** Time-to-failure rates caused by road departure (offroad) errors. **Right:** Learned likelihood score of recorded trajectories under different levels of perturbations.

speed and jerk statistical distances relative to the recorded trajectories. As expected, SimNet has the lowest sADE due to its behavior cloning objective. In nuScenes, BITS achieves comparable performance to SimNet in dataset metrics, showing that our method does not have to sacrifice behavioral realism for diversity and stability.

TABLE III

LEARNED LIKELIHOOD SCORES FOR TRAFFIC SIMULATION METHODS.

	Lyft	nuScenes
SimNet [22]	0.112	0.167
TPP [31]	0.135	0.155
TrafficSim [23]	0.126	0.176
BITS	0.131	0.276
Dataset	0.142	0.275

Finally, we consider the learned metric described in Sec. III-D. To show that this occupancy likelihood-based metric indeed captures meaningful data likelihoods, we roll out ground truth trajectories with different levels of Ornstein-Uhlenbeck noise [61] and measure the predicted likelihood score. As shown in Fig. 3, the likelihood score decreases smoothly as the noise intensity grows, indicating that the learned metric captures the effect of disturbances well. We report the learned likelihood scores for representative baselines in Table. III. Sensibly, we see that in both datasets the ground truth dataset trajectories have the highest likelihood scores. BITS yields comparable or higher likelihood scores than other baselines, with scores on par with ground truth dataset trajectories on nuScenes data.

V. DISCUSSION AND CONCLUSIONS

Limitations. First, despite our efforts, devising evaluation metrics for traffic simulation remains an open research problem. In particular, the learned metric is likely biased by the model choice and the training data. Second, we do not consider traffic rules (e.g., driving on the correct side of the road, obeying traffic lights) due to the limited annotation in the existing dataset, while they can be incorporated via the cost-based action formulation (Sec. III-C) when available.

Conclusions. In this work, we present BITS, a novel data-driven traffic simulation model. BITS achieves high sample efficiency and behavioral diversity through a bi-level imitation learning formulation and generates stable long-horizon rollouts aided by a prediction-and-planning module. To facilitate future studies in the field, we develop and open source a software tool that unifies data formats from different AV datasets and transforms scenes from existing datasets into interactive simulation environments. We compare BITS against a number of competitive baselines on two large-scale real-world AV datasets and find that BITS can generate diverse, realistic, and stable traffic simulations.

REFERENCES

- [1] Waymo, "Safety report," Waymo LLC, 2021, Available at <https://waymo.com/safety/safety-report>. Retrieved on July 4, 2021.
- [2] Uber Advanced Technologies Group, "A principled approach to safety," 2020, Available at <https://uber.app.box.com/v/UberATGSafetyReport>.
- [3] NVIDIA, "Self-driving safety report," 2021, Available at <https://images.nvidia.com/content/self-driving-cars/safety-report/auto-print-self-driving-safety-report-2021-update.pdf>.
- [4] Argo AI, "Developing a self-driving system you can trust," Apr. 2021, Available at <https://www.argo.ai/wp-content/uploads/2021/04/ArgoSafetyReport.pdf>.
- [5] Zoox, "Safety report volume 2.0," 2021, Available at <https://zoox.com/safety/>.
- [6] Motional, "Voluntary safety self-assessment," 2021.
- [7] General Motors, "Self-driving safety report," 2018.
- [8] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.
- [9] J. Casas, J. L. Ferrer, D. Garcia, J. Perarnau, and A. Torday, "Traffic simulation with aimsun," in *Fundamentals of traffic simulation*. Springer, 2010, pp. 173–232.
- [10] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator vissim," in *Fundamentals of traffic simulation*. Springer, 2010, pp. 63–93.
- [11] Waymo, "Motion prediction leaderboard," 2021, Available at <https://waymo.com/open/challenges/2021/motion-prediction/>.
- [12] Argo AI, "Argoverse 2: Motion forecasting competition," 2022, Available at <https://eval.ai/challenge/1719/leaderboard/4098>.
- [13] Lyft, "Motion prediction for autonomous vehicles," 2021, Available at <https://www.kaggle.com/competitions/lyft-motion-prediction-autonomous-vehicles/leaderboard>.
- [14] Motional, "nuScenes prediction task leaderboard," 2021.
- [15] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 661–668.
- [16] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," in *Conf. on Robot Learning*, 2020.
- [17] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020.
- [18] B. Ivanovic, K.-H. Lee, P. Tokmakov, B. Wulfe, R. McAllister, A. Gaidon, and M. Pavone, "Heterogeneous-agent trajectory forecasting incorporating class uncertainty," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2022. [Online]. Available: <https://arxiv.org/abs/2104.12446>
- [19] G. Kotusevski and K. Hawick, "A review of traffic simulation software," *Research Letters in the Information and Mathematical Sciences*, vol. 13, pp. 35–54, 2009.
- [20] P. M. Ejercito, K. G. E. Nebrija, R. P. Feria, and L. L. Lara-Figueroa, "Traffic simulation software review," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2017, pp. 1–4.
- [21] E. Brockfeld, R. D. Kühne, A. Skabardonis, and P. Wagner, "Toward benchmarking of microscopic traffic flow models," *Transportation research record*, vol. 1852, no. 1, pp. 124–129, 2003.
- [22] L. Bergamini, Y. Ye, O. Scheel, L. Chen, C. Hu, L. Del Pero, B. Osinski, H. Grimmer, and P. Ondruska, "Simnet: Learning reactive self-driving simulations from real-world observations," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5119–5125.
- [23] S. Suo, S. Regalado, S. Casas, and R. Urtasun, "TrafficSim: Learning to simulate realistic multi-agent behaviors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 400–10 409.
- [24] D. Rempe, J. Phillion, L. J. Guibas, S. Fidler, and O. Litany, "Generating useful accident-prone driving scenarios via a learned traffic prior," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2022.
- [25] Waymo, "Waymo Open Dataset: An autonomous driving dataset," <https://waymo.com/open/>, 2019.
- [26] M. Igl, D. Kim, A. Kuefler, P. Mougin, P. Shah, K. Shiarlis, D. Anguelov, M. Palatucci, B. White, and S. Whiteson, "Symphony: Learning realistic and diverse agents for autonomous driving simulation," *arXiv preprint arXiv:2205.03195*, 2022.
- [27] A. Gupta, J. Johnson, F. Li, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.
- [28] A. Kuefler, J. Morton, T. A. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *IEEE Intelligent Vehicles Symposium*, 2017.
- [29] Y. C. Tang and R. Salakhutdinov, "Multiple futures prediction," in *Conf. on Neural Information Processing Systems*, 2019.
- [30] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker, "DESIRE: distant future prediction in dynamic scenes with interacting agents," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.
- [31] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conf. on Computer Vision*, 2020.
- [32] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.
- [33] S. Casas, C. Gulino, R. Liao, and R. Urtasun, "SpAGNN: Spatially-aware graph neural networks for relational behavior forecasting from sensor data," in *Proc. IEEE Conf. on Robotics and Automation*, 2020.
- [34] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. Urtasun, "Implicit latent variable model for scene-consistent motion forecasting," in *European Conf. on Computer Vision*, 2020.
- [35] H. Zhao and R. P. Wildes, "Where are you heading? dynamic trajectory prediction with expert goal examples," in *IEEE Int. Conf. on Computer Vision*, 2019.
- [36] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *European Conf. on Computer Vision*, 2020.
- [37] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "PRECOG: Prediction conditioned on goals in visual multi-agent settings," in *IEEE Int. Conf. on Computer Vision*, 2019.
- [38] C. Choi, S. Malla, A. Patil, and J. H. Choi, "DROGON: A trajectory prediction model based on intention-conditioned behavior reasoning," in *Conf. on Robot Learning*, 2020.
- [39] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov, "TNT: Target-driveN Trajectory Prediction," in *Conf. on Robot Learning*, 2020.
- [40] J. Gu, C. Sun, and H. Zhao, "DenseTNT: End-to-end trajectory prediction from dense goal sets," in *IEEE Int. Conf. on Computer Vision*, 2021.
- [41] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanculescu, and F. Moutarde, "HOME: Heatmap output for future motion estimation," in *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, 2021.
- [42] —, "GOHOME: Graph-oriented heatmap output for future motion estimation," in *Proc. IEEE Conf. on Robotics and Automation*, 2022.
- [43] —, "THOMAS: Trajectory heatmap output with learned multi-agent sampling," in *Int. Conf. on Learning Representations*, 2022.
- [44] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.
- [45] P. Dayan and G. E. Hinton, "Feudal reinforcement learning," *Advances in neural information processing systems*, vol. 5, 1992.
- [46] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.
- [47] A. S. Vechnets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "Feudal networks for hierarchical reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3540–3549.
- [48] H. Le, N. Jiang, A. Agarwal, M. Dudik, Y. Yue, and H. Daumé III, "Hierarchical imitation and reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 2917–2926.
- [49] K. Shiarlis, M. Wulfmeier, S. Salter, S. Whiteson, and I. Posner, "Taco: Learning task decomposition via temporal alignment for control," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4654–4663.
- [50] K. Pertsch, O. Rybkin, F. Ebert, S. Zhou, D. Jayaraman, C. Finn, and S. Levine, "Long-horizon visual planning with goal-conditioned hierarchical predictors," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 321–17 333, 2020.
- [51] T. Shu, C. Xiong, and R. Socher, "Hierarchical and interpretable skill acquisition in multi-task reinforcement learning," *arXiv preprint arXiv:1712.07294*, 2017.

- [52] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," in *Conference on robot learning*. PMLR, 2020, pp. 1113–1132.
- [53] A. Mandlekar, F. Ramos, B. Boots, S. Savarese, L. Fei-Fei, A. Garg, and D. Fox, "Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4414–4420.
- [54] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, "Learning to generalize across long-horizon tasks from human demonstrations," *arXiv preprint arXiv:2003.06085*, 2020.
- [55] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning*. PMLR, 2022, pp. 158–168.
- [56] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *AI & Statistics*, 2011.
- [57] S. Konev, K. Brodt, and A. Sanakoyeu, "Motioncnn: A strong baseline for motion prediction in autonomous driving," 2021.
- [58] S. M. LaValle, "Better unicycle models," in *Planning Algorithms*. Cambridge Univ. Press, 2006, pp. 743–743.
- [59] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [60] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, September 2009, pp. 460–467.
- [61] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical review*, vol. 36, no. 5, p. 823, 1930.