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

Supplementary Material

8. Model details

The hyper-parameters of our model architecture are listed in Tab. 3. All models are trained using the Adam optimizer with a learning rate of 0.0003, and batch size 32 with 8 Geforce 3090. The VAE are trained simultaneously with the policy network using the online simulated learned data and offline expert data.

Table 3. Hyper-parameters.

Hyper-parameter	**Value**
History time steps	10
Future time steps T	10
Route point number	30
Neighbor number	6
Neighbor maximum distance	20 m
Origin perturbation std	2 m
EGAT hidden size	512
VAE latent dim	8
VAE encoder layer number	1
VAE decoder layer number	1
Policy Network layer number	1
LQR acceleration weight $\eta_{\boldsymbol{a}}$	1
Learner VAE loss weight λ	1
Training simulation interval N	50
Training simulation length S	50

9. Baseline details

SUMO uses the mobil model and IDM with various tuned parameters (including desired speed, acceleration, deceleration, minimum gap, time headway) for 6 types (including motorcycle, car, taxi, bus, medium and heavy vehicle) of vehicles. These parameters are shown in 4.

BC: our model without the VAE, LQR and the on-road projection module.

All RL baselines are trained using IPPO with default parameters in the Ray library. **MARL**: the reward is the sum of a displacement reward (weight 0.01), an off-road penalty (weight 1), and a terminal reward (weight 0.01).

MARL+BC: adds a BC term (weight 1) to the loss function of the MARL policy.

PS-GAIL: learns a policy using reward functions from a discriminator network, which is also trained using Adam with a learning rate of 0.0003.

RAIL: PS-GAIL with additional rewards from MARL.

10. Dataset Preparation

The data preparation process of pNEUMA dataset is introduced in this section.

10.1. Trajectory Data

The trajectory data is downloaded from the official website (https://open-traffic.epfl.ch), specifically the data recorded by ALL Drones during all periods except for the first period (8.30-9.00 at 2018/10/24) due to a large position error caused by wind gusts.

10.2. Routing

To determine the route for each vehicle trajectory, we used the method in https://github.com/wannesm/LeuvenMapMatching. However, this method generated many circular routes that are rarely observed in real data. To address this issue, we skipped the intermediate routing node points if they were far away from the actual trajectory, which helped reduce the number of unrealistic circular routes.

10.3. Road Network

The map information is downloaded from OpenStreetMap, and then we import it into SUMO to generate the road network. We only include highways for vehicles in the road network while excluding other road types, such as sidewalks and railways. However, we find that the map data is not always accurate, so we manually adjust the road shapes to reduce the number of off-road driving cases in the recorded trajectory data. Additionally, we modify the lane connection relations in junctions to alleviate traffic jams during SUMO simulations.

10.4. Traffic Light

Because the traffic light information is not provided by the dataset, we design an algorithm to estimate traffic light information from the recorded trajectory data.

Firstly, we filter all vehicle starting and stopping points near all signaled intersections from the trajectory data.

Secondly, we cluster these points based on their located edge, as we assume that all lanes on one edge are controlled by the same traffic light.

Thirdly, we obtain all time steps when each traffic light turns green by identifying its corresponding clustered points

Table 4. Sumo hyper-parameters.

Hyper-parameter	**Motorcycle**	**Car**	**taxi**	**bus**	**medium vehicle**	**heavy vehicle**
desired speed (m/s)	30	30	30	11.70	30	17.38
acceleration (m/s^2)	2.5	2.5	2.5	2.5	2.5	2.5
deceleration (m/s^2)	10.0	10.0	10.0	10.0	10.0	10.0
minimum gap (s)	0.1	0.1	0.1	0.1	0.1	0.1
time headway (s)	0.1	0.1	0.1	0.1	0.1	0.1

whose time gap to the previous point is larger than seven seconds. Similarly, we can obtain all time steps when it turns red.

Fourthly, based on all the time steps when the traffic light turns green, we need to calculate the traffic light's first turning green time step, green time, and cycle length. We assume that all traffic lights in the same junction have the same cycle length, which can only be 45 or 90 seconds. To estimate the first green time and cycle length, we use a cost function, where a negative cost is given if the filtered turning green time steps match the estimated turning green time step, and a positive cost is given if there is no filtered turning green time step matching the estimated turning green time step. By enumerating all first turning green time steps with an interval of 0.01 seconds, and cycle length of 45 or 90 seconds, we output the result with the minimal costs. Based on the other traffic light turning green time steps in the same junction, we can obtain its turning red time. If there is no other traffic light in the same junction, we need to estimate its turning red time as we do in the estimation of green time. Based on the turning red time and the turning green time, we can obtain the green time of each traffic light.

11. Runtime

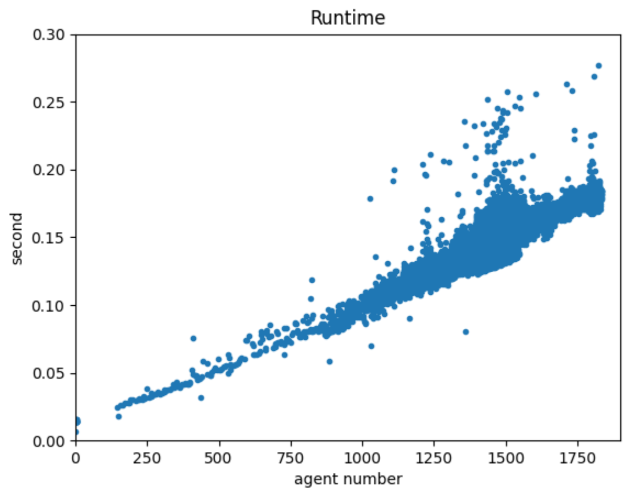
We perform runtime experiments using a single Nvidia GeForce GTX 1080 GPU and an Intel i7-8700@3.2GHz CPU. These experiments take into account all components of our traffic model, including input preparation, trajectory prediction, and action generation. The runtime results for all time step are recorded during the long-term evaluation, as shown in Fig. 3. We can see that the runtime increases almost linearly with the number of agents. Besides, our method can finish one simulation step of thousands of agents within an acceptable time limit (smaller than 1 second).

12. Qualitative results

12.1. Prediction and Planned Trajectories

In Fig. 4, we present the trajectories produced by our context-conditioned VAE during training. Based on the augmented past trajectory and context information, our policy network predicts a future trajectory, which is subsequently refined by the LQR module, in different scenarios such as lane keeping, turning, and lane changing. The results illustrate that our context-conditioned VAE is capable of generating a wide range of past trajectories that encompass the distribution of possible policies, while remaining reasonable and closely resembling the actual past trajectory. Moreover, our method accurately predicts future trajectories that closely align with the actual path, based on the augmented history and context. Additionally, the incorporation of the LQR module enhances the smoothness of the trajectory. Importantly, our approach also demonstrates the ability to generate diverse behaviors that comply with the surrounding environment.

Figure 3. Runtime of each time step during the long-term evaluation.



12.2. Statistical Distribution

In Fig. 5, we illustrate the distribution of speed and distance to the leading vehicle during long-term simulations. Our method produces more similar speed distributions to the ground truth than SUMO since the Intelligent Driver Model (IDM) always aims to move at the highest speed. Furthermore, our method generates leader distance distri-

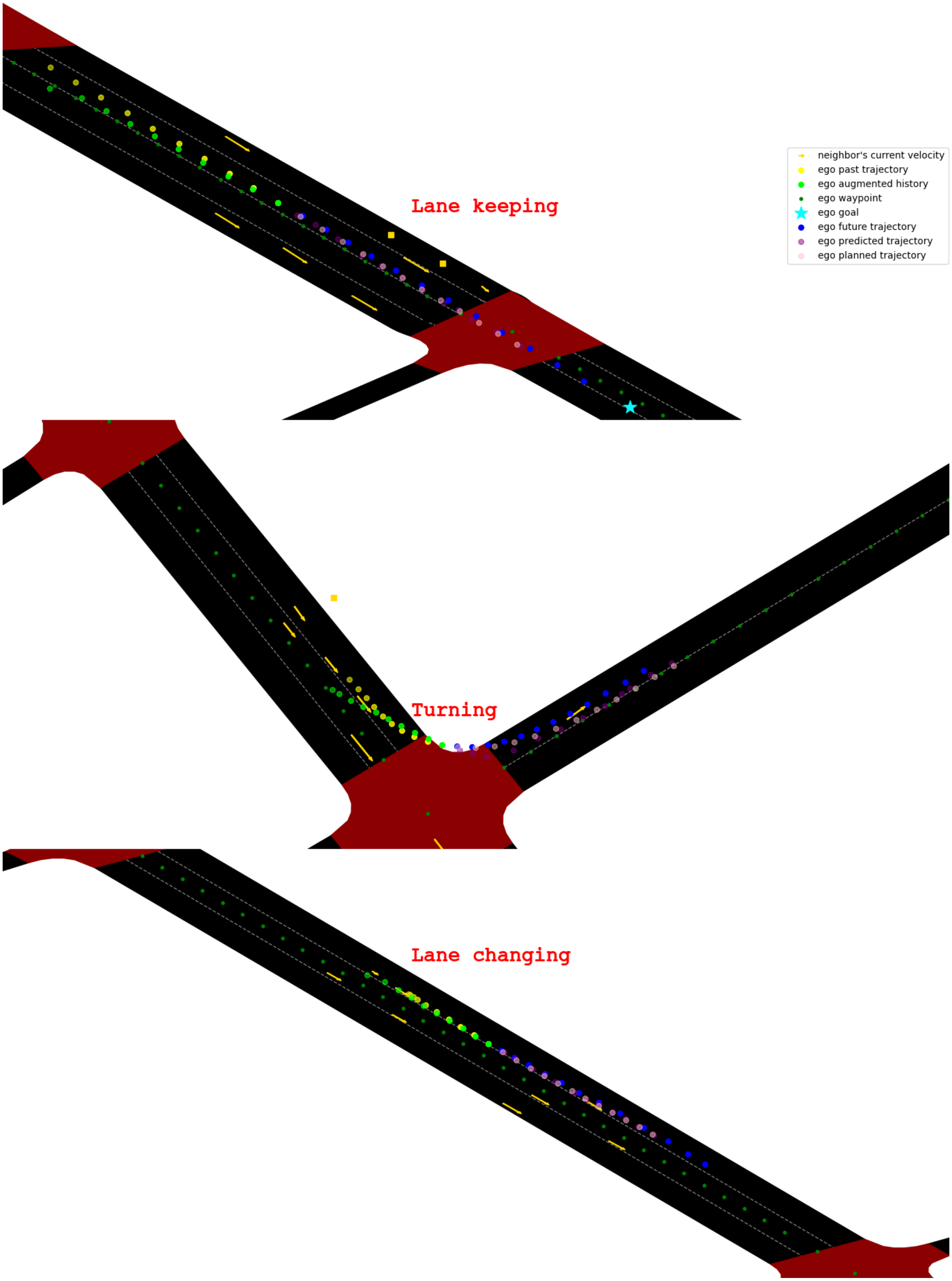


Figure 4. Trajectories augmented by our context-conditioned VAE, predicted by our policy network and subsequent planned trajectory by the LQR module in lane keeping, turning and lane changing scenarios.

butions that closely match the ground truth.

13. Road Shape Change Experiment

Our microscopic long-term traffic simulators can help transportation engineers and planners to analyze and predict the impact of microscopic adjustments on traffic patterns without disrupting real-world traffic. For example, it can help analyze how changing road shape affects traffic patterns. In Fig. 6, we present the mean road density and speed changes in our simulator after modifying several roads' shapes. We can see that a local microscopic modification in road network can causes traffic congestion or alleviation in distant areas.

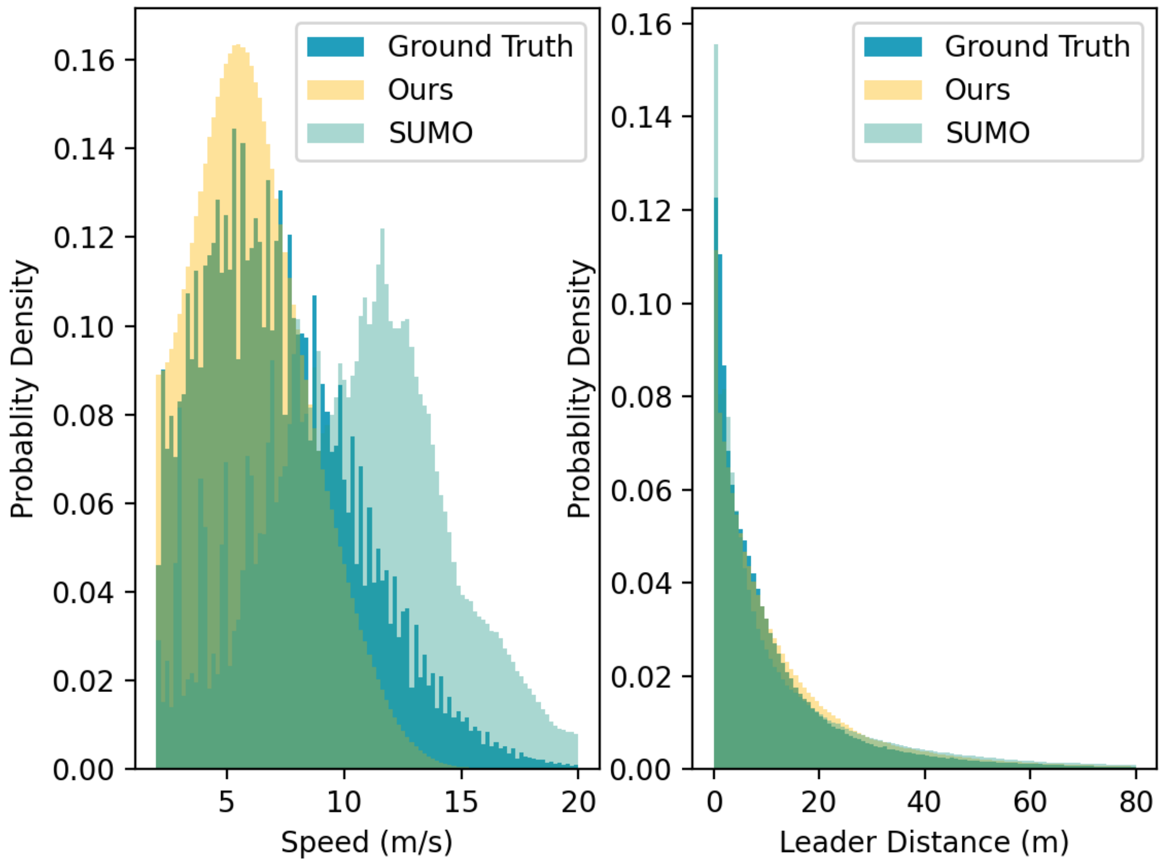


Figure 5. Distributions of speed and leader vehicle's distance in long-term evaluation.

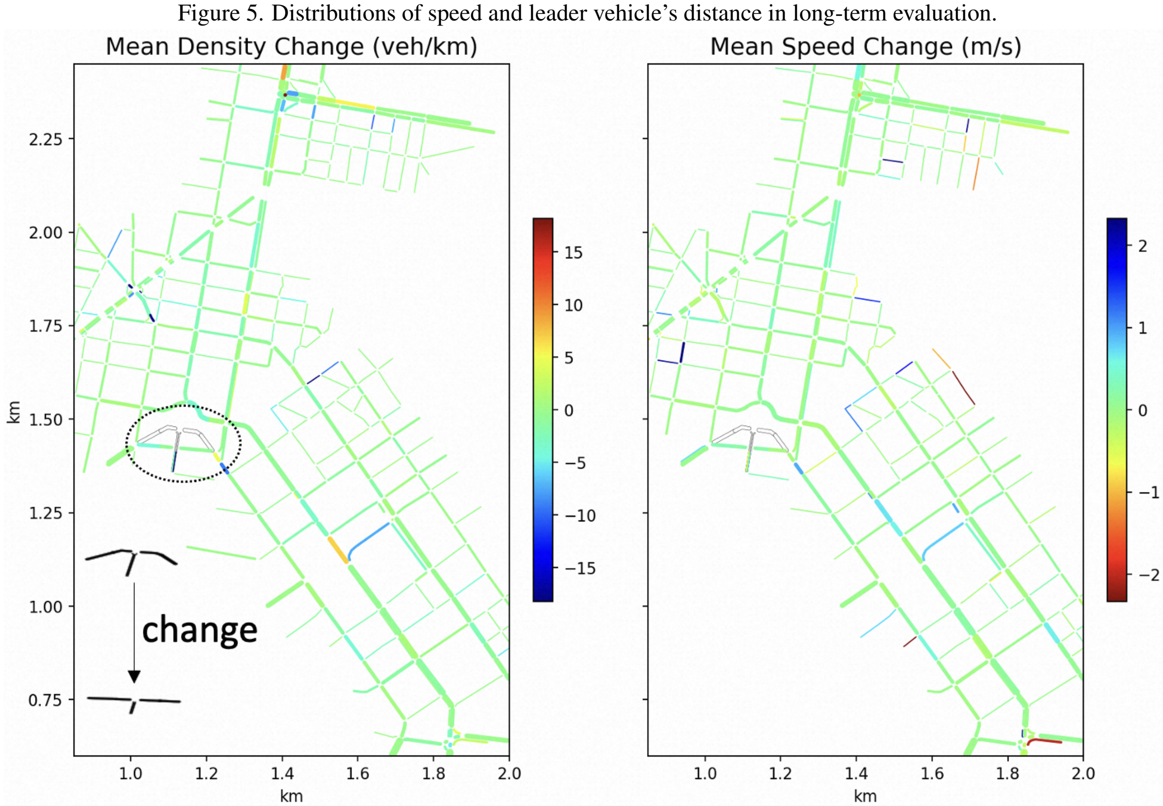


Figure 6. Mean density and speed changes after modifying road network.