TrafficBots V1.5: Traffic Simulation via Conditional VAE and Transformer with Relative Pose Encoding

Zhejun Zhang
ETH Zürich
zhejun.zhang@vision.ee.ethz.ch

Christos Sakaridis
ETH Zürich
csakarid@vision.ee.ethz.ch

Luc Van Gool
ETH Zürich, INSAIT Sofia University
vangool@vision.ee.ethz.ch

Abstract

In this technical report we present TrafficBots V1.5, a baseline method for the closed-loop simulation of traffic agents. TrafficBots V1.5 achieves baseline-level performance on the Waymo Open Sim Agents Challenge (WOSAC) 2024. It is a simple baseline that combines TrafficBots, a CVAE-based multi-agent policy conditioned on each agent's individual destination and personality, and HPTR, the heterogeneous polyline transformer with relative pose encoding. To improve the performance on the WOSAC leaderboard, we apply scheduled teacher-forcing at the training time and we filter the sampled scenarios at the inference time. The code is available at: https://github.com/zhejz/TrafficBotsV1.5

1. Introduction

The problem of closed-loop multi-agent traffic simulation can be addressed by learning a policy for each traffic participant. Specifically, at each time step, the policy predicts the actions of all agents, given the historical observations form previous time steps, including the map, traffic lights and agent trajectories. Based on TrafficBots [17], the TrafficBots V1.5 policy is shared by all agents. Different behaviors are generated by conditioning the policy on the individual destination and personality of each agent. In contrast to the SceneTransformer [7] architecture and representation, which are not rotation and translation invariant, TrafficBots V1.5 uses the pairwise-relative representation and the HPTR [18] architecture. This greatly improves the accuracy of TrafficBots without sacrificing its efficiency and scalability. Moreover, instead using a recurrent neural network (RNN) to encode the temporal axis, TrafficBots V1.5 uses stacked historical observation as input, such that its architecture is solely based on Transformers [11].

1.1. TrafficBots

TrafficBots [17] is a multi-agent policy built upon motion prediction and end-to-end (E2E) driving. Compared to previous data-driven traffic simulators, TrafficBots demonstrates superior configurability and scalability. To generate configurable behaviors, for each agent TrafficBots introduces a destination as navigational information, and a time-invariant latent personality that specifies the behavioral style. Unlike the goal which depends on the prediction horizon and hence leads to causal confusions, the destination approximates the output of a navigator which is available in the problem formulation of E2E driving [16]. Importantly, the destination indicates where the agent wants to reach eventually, i.e. not necessarily at a specific time step in the future. In order to capture the diverse behaviors from human demonstrations, the personality is learned using the conditional variational autoencoder (CVAE) [9] following prior works on multi-modal motion prediction [10]. To ensure the scalability, TrafficBots uses the scene-centric representation [7] and presents a new scheme of positional encoding for angles, allowing all agents to share the same vectorized context and the use of an architecture based on dot-product attention. However, due to the lack of rotation and translation invariance induced by the scene-centric representation, TrafficBots does not achieve superior performance compared methods using agent-centric representation.

1.2. HPTR

Depending on how the coordinate system of the vectorized representation is selected, motion prediction methods fall into three categories: agent-centric [6], scene-centric [7] and pairwise-relative [2]. While agent-centric methods achieve top accuracy but suffer from poor scal-
ability, and scene-centric methods show superior scalability but suffer from poor accuracy, the pairwise-relative methods get the best from both worlds. However, previous pairwise-relative methods are mostly using graph neural networks [2], which are often less efficiently implemented on the graphics processing unit (GPU) compared to Transformers with dot-product attention. To address this problem, a novel attention module called Nearest Neighbor Attention with Relative Pose Encoding (KNARPE) is introduced [18], which allows the pairwise-relative representation to be used by Transformers. Based on KNARPE, a pure Transformer-based framework called Heterogeneous Polyline Transformer with Relative pose encoding (HPTR) [18] is presented, which uses a hierarchical architecture to enable asynchronous token update and avoid redundant computations. By sharing contexts among traffic agents and reusing the unchanged contexts in driving scenarios, HPTR is as efficient as scene-centric methods, while performing on par with state-of-the-art agent-centric methods on marginal motion prediction tasks.

2. Method

TrafficBots V1.5 updates TrafficBots with the HPTR input representation and network architecture. This section describes the changes we have made while combining TrafficBots and HPTR.

2.1. Architecture

We make minimal changes while applying HPTR to TrafficBots. We remove the temporal RNN from TrafficBots, and follow HPTR to use stacked historical observations as input. The policy network, the personality encoder, and the destination predictor of TrafficBots are now all based on the pairwise-relative representation and KNARPE attention module. We keep the intra-map, enhance-traffic-light and enhance-agent Transformers of HPTR. Following TrafficBots, multi-modal outputs are generated by conditioning the policy on each agent’s individual destination and personality. Therefore, the anchors and the anchor-to-all Transformer of HPTR are discarded in TrafficBots V1.5. Instead of a learnable prior personality of TrafficBots, we use a standard Gaussian for the prior personality. We add a traffic light state predictor, but its accuracy is not good enough to improve the overall simulation performance on the leaderboard. Details and hyperparameters of the network architecture can be found in the open-source repository of TrafficBots V1.5, as well as in the TrafficBots [17] and HPTR [18] papers. Our model does not take the \( z \) axis, i.e. the altitude dimension, into account. At the inference time we assume the \( z \) dimension of agent trajectories is constant and is equal to its last observed value.

2.2. Training

We found two techniques that improve the performance of TrafficBots V1.5. Firstly, we adopt a larger free nats [3] equals to 1.0 for the KL-divergence between the posterior and the prior personality. This allows the posterior personality to encode more information. Secondly, we use scheduled sampling [1] and apply teacher-forcing to 30% agents at the beginning of the training. The percentage of teacher-forcing decreases linearly to 0 during the training. Due to limited computational resources, we train our models for 5 days on 4 NVIDIA RTX 4090 GPUs. Training for longer time or using more GPUs should improve the performance further. Following TrafficBots, we use back-propagation through time and L2 imitation loss to training the policy, cross-entropy loss to train the destination and traffic light state predictor, and KL-divergence loss to train the personality CVAE. We do not apply a loss that encourages collision avoidance [2, 4, 15] because this will bias the model, even though it could improve the performance on the leaderboard.

2.3. Inference

Instead of bias the model itself, we apply a milder approach to bias the model outputs towards safer behavior and hence improve the collision-based metrics on the WOSAC leaderboard. Specifically, we sample 128 scenarios at the inference time and select 32 scenarios that contain the least collision events. Except the episode filtering, we do not apply any other post-processing techniques.

3. Results

The performance of our method on the WOSAC leaderboard [13] is shown in Table 1. More details about the challenge and the metrics can be found in the WOSAC paper [5]. Our method achieves baseline-level performance in terms of the realism meta metric, which is a weighted sum of other metrics except the minADE. We apply a unicycle model to approximate the dynamics of all types of agents and select the parameters heuristically. This allows our simulation to generate smooth trajectories, but it also affects the WOSAC kinematic metrics negatively. Our TrafficBots V1.5 is outperformed by other methods in terms of interactive metrics which involve collision avoidance. We believe adding a loss that encourages collision avoidance would help, but it is controversial if a collision-free simulation would be useful for the development of autonomous driving algorithms. In terms of the map-based metrics which consider off-road driving, our model is slightly worse than other methods. The minADE of TrafficBots V1.5 is significantly larger than other methods on the leaderboard, which is a problem inherited from TrafficBots. Overall, from Table 1 we observe that GPT-based [8] architectures that apply to-
kenization and next-token prediction, such as SMART, BehaviorGPT and GUMP, achieve top performance. Interestingly, none of these GPT-based architectures uses goal or personality conditioning, but they are still able to generate multi-modal outputs. It seems the multi-modality in traffic simulation can be addressed using tokenization and the cross-entropy loss. This indicates that the poor performance of TrafficBots might be caused by the CVAE and regression losses on continuous states and actions. Another interesting thing is that the GPT-based methods achieve the best performance without considering the traffic lights. This indicates that the dataset might be imbalanced and the evaluation metrics might be flawed.

4. Conclusion

In this technical report we present TrafficBots V1.5, which is a baseline method that combines TrafficBots, a prior work on the closed-loop traffic simulation using CVAE, and HPTR, a prior work on the network architecture of motion prediction using the pairwise-relative representation. Our method is the only CVAE-based method on the WOSAC leaderboard 2024. The performance of our method is slightly worse than the GPT-based methods. However, as a baseline method that involves minor novelty, it achieves the performance we expected, and there are many possibilities to improve this simple baseline.

References