RLFTSim: Multi-Agent Traffic Simulation via Reinforcement Learning Fine-Tuning (Technical Report for Waymo Open Sim Agents Challenge)

Ehsan Ahmadi
University of Alberta
eahmadi@ualberta.ca

Hunter Schofield
York University
hunterls@yorku.ca

Supervised open-loop training has been widely adopted to train traffic simulation models, but it fails to provide direct supervision about physical constraints of the environment, traffic rules, and reactive behavior of agents, which are essential for a grounded simulation model. To address this, we propose a reinforcement learning fine-tuning framework for traffic simulation models that directly optimizes for realism metrics. We instantiate RLFTSim atop a state-of-the-art pretrained simulator and design a reward signal based on Reinforcement Learning Leave-One-Out. Our experimental evaluation on the Waymo Open Motion Dataset (WOMD) demonstrates that RLFTSim achieves comparable realism enhancement to existing closed-loop fine-tuning methods while requiring significantly less training data, as it only requires a single epoch of training on 30% of WOMD's training split. This data efficiency, combined with the framework's ability to directly optimize for realism metrics, positions RLFTSim as a practical approach for improving traffic simulation fidelity.

1. Introduction

One of the main challenges in developing autonomous vehicles (AVs) is the scarcity of data. This becomes more significant when we consider crash scenarios that an AV should be capable of handling. To verify the safety capability of an AV, based on a statistical estimate discussed in [7], an AV should be capable of driving two million kilometers without fatal accidents to claim that it is as safe as humans. Considering the costs of such verification, the use of simulation is justified to make the verification and development of AVs feasible.

Here, we are interested in the problem of realistic behavioral multi-agent traffic simulation in the agent's state space, as we elaborate in Section 3. The problem of traffic simulation has previously been addressed with rule-based simulators, which can provide various feedback signals from kinematics and physics simulations [1, 5]. When it comes to behavior modeling of agents, rule-based simulators either replay logged trajectories or use rule-based models, such as the constant-speed actor model or the Intelligent Driver Model [14]. We argue that there is a significant sim-to-real gap when using these approaches. Even in the case of log-replay, since the agent's behavior is not reactive, it becomes unrealistic when facing closed-loop inference.

Recently, learning-based simulation models have been introduced to bridge the realism gap and generate reactive multi-agent traffic simulation rollouts [15, 17]. However, previous research works are mostly trained in an open-loop setting with imitation learning objectives [15, 17]. We argue that to make simulation models capable of overcoming distribution shifts caused by error accumulation in closed-loop simulation and avoiding the causal confusion problem [3], the simulation needs to be trained in a closed-loop setting. Moreover, the simulation model needs to be grounded in traffic rules and physical constraints of the environment. Meanwhile, the simulation model needs to be a realistic representative of daily driving scenarios while still imitating human agents to make its adoption worthwhile. In the past, a combination of open-loop supervision via imitation learning and closed-loop training via reinforcement learning has been used for ego-centric motion planning for AVs [10], but effective scaling of this approach for multi-agent simulation cases is still an open area of research.

In this work, inspired by the success of reinforcement learning with verifiable rewards for new skill learning in foundation models [12], we present a reinforcement learning-based fine-tuning approach for traffic simulation (RLFTSim) to address the physics and traffic-rules alignment problem for pre-trained models with imitation learning. Our method addresses the alignment problem for open-loop trained models with closed-loop objectives. Specifically, we focus on the Waymo Open Simulation Challenge (WOSAC) and use its definition of realism metametric (RMM) as the optimization objective function for RLFT. However, RMM in its original form is a sparse signal calculated per group of 32 rollouts. To make it suitable for RL training, we adapt the metametric computation using the Reinforce Leave-One-Out (RLOO) definition of the objective function [8], to provide denser reward signals while maintaining low variance for the policy gradient estimates in RL post-training.

2. Related Works

2.1. Learning-based Traffic Simulation Approaches

Data-driven models have emerged as a cornerstone for realistic traffic simulation, moving beyond heuristic rule-based simulators that struggled with complex maneuvers and interactions [14]. Early learning-based frameworks like [13] modeled multi-agent driving behavior with implicit latent variables and differentiable closed-loop training, significantly improving realism over rule-based baselines. Recently, large-scale sequence modeling approaches inspired by language models have been introduced [11, 15]. The Waymo Open Motion Dataset (WOMD) [4], nuPlan [2], and similar datasets of millions of real trajectories have enabled training transformer-based models that treat multi-agent traffic motion as a *next-token prediction* problem. For example, [15] is trained to predict the next discrete motion token for each agent, given vectorized states and map context. By learning from over a billion tokens from multiple datasets, SMART achieved state-of-the-art realism on WOMD's Sim Agents challenge, even generalizing zero-shot across datasets.

2.2. Model Alignment and Fine-Tuning

A core challenge in learned simulators is *model alignment*: ensuring the distribution of simulated behaviors matches real driving data when the model is unrolled in a closed-loop setting. Pure behavior cloning suffers from error accumulation during closed-loop rollouts, where small prediction errors compound over time and lead agents into unrealistic states.

One approach to address this is *closed-loop supervised fine-tuning*, which avoids reward design altogether. [16] introduces a strategy called *Closest Among Top-K* (CAT-K) rollouts, which perturbs the policy by sampling multiple plausible next actions and then selects the one closest to the ground-truth trajectory for training feedback. This method achieves the benefits of closed-loop training using only offline demonstrations, with no need for environment interaction or adversarial discriminators.

Alternatively, researchers have explored reinforcement learning (RL) and other closed-loop fine-tuning techniques to directly optimize for desired behaviors. For instance, generative adversarial imitation learning and inverse RL were applied in early works to encourage realism in long rollouts. [6] combined adversarial training with a discriminator to prune unrealistic trajectories and beam search decoding, significantly reducing collisions and off-road events compared to vanilla imitation. However, the true objective reward of human agents in traffic scenes is unknown, and designing effective reward functions for RL-based approaches remains challenging, as simple metrics often fail to capture the nuanced behaviors required for realistic human-like driving.

A recent single-agent planning work demonstrates that integrating imitation learning and RL can yield robust policies: [10] shows that an imitation-learned policy fine-tuned with simple safety rewards can reduce collision rates by over 38% in rare challenging scenarios, outperforming imitation alone.

In this work, we extend the application of RL for fine-tuning traffic simulation by using a well-defined realism metric from the Waymo Open Simulation Challenge as our reward signal, avoiding the challenge of designing custom reward functions while leveraging the benefits of RL-based closed-loop post-training.

3. Reinforcement Learning Fine-Tuning for Traffic Simulation

In this section, we present our reinforcement learning fine-tuning framework for traffic simulation alignment. We begin by formulating the multi-agent traffic simulation problem as a contextual MDP (Section 3.1). We then describe the WOSAC realism metametric that serves as our evaluation criterion (Section 3.2). Finally, we present our approach for generating dense reward signals using the RLOO technique to enable effective RL-based fine-tuning (Section 3.3).

3.1. Problem Formulation

Multi-agent traffic behavior modeling can be formulated as a Contextual Markov Decision Process defined by the tuple $(S_t, A_t, S_{t+1}, R_{t+1}, C)$ with discrete time steps $t \in [1, T]$. The state S_t includes finite-horizon past states of up to N agents: $S_t = \{S_{t'}^j; j \leq N, t' \leq t\}$. The action space consists of tokenized actions for all agents: $A_t = \{A_t^j \in \mathcal{V}\}$, where $\mathcal{V}$ is the vocabulary of discrete action tokens. The reward $R_t \in \mathbb{R}$ is a scalar value. The context C includes vectorized static and dynamic map features: $C = \{M \in \mathbb{R}^{N_m \times D_m}, L \in \mathbb{R}^{N_{tl} \times D_{tl}}\}$, where M represents static map information with up to N_m downsampled vector points of dimension D_m, and L represents dynamic map information including past temporal states of up to N_{tl} traffic lights with feature dimension D_{tl}.

3.2. WOSAC Realism Metametric

To compute the WOSAC metametric, let $\{\tau_i\}_{i=1}^N$ be $N = 32$ simulator rollouts sharing the same history and map context, each of length T time steps, and let τ^* denote the corresponding ground-truth trajectory. For each rollout τ_i and each timestep $t \in \{1, \ldots, T\}$, we extract a D-dimensional feature vector $\mathbf{f}_t^{(i)} = \left(f_{1,t}^{(i)}, f_{2,t}^{(i)}, \ldots, f_{D,t}^{(i)}\right)$, whose components are: (i) Kinematic features: linear/angular speed and acceleration, (ii) interactive features: closest distance to other agents, time-to-collision (TTC), and accident indication, and (iii) map-based features: distance to road boundary, off-road indication, and traffic light violation. We compute the same per-timestep features $\mathbf{f}_t^*$ for the ground-truth trajectory. Each feature dimension d is discretized into bins $\{\mathcal{B}_{d,a,k}\}_{k=1}^K$, where a is the agent index, and $K = 20$ is the number of bins.

We first form **time-dependent** empirical distributions:

$$\hat{P}_{d,a,t}(k) \;=\; \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{f_{d,a,t}^{(i)} \in \mathcal{B}_{d,a,k}\}, \qquad (1)$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. Then, we marginalize out time to obtain a **time-independent** histogram $\hat{P}_{d,a}(k) = \frac{1}{T}\sum_{t=1}^{T}\hat{P}_{d,a,t}(k)$.

Finally, letting $f_{d,a,t}^{*}$ fall into bin k_d^* when observed on the ground truth, the WOSAC realism metametric is defined as a weighted perplexity over all feature dimensions:

$$M_{\text{WOSAC}} \;=\; \sum_{d=1}^{D} w_d \left[\prod_{(a,t_a)\in V} \hat{P}_{d,a}(k^*_{d,a,t_a})\right]^{\frac{1}{|V|}}, \qquad (2)$$

where each weight $w_d \geq 0$ reflects the relative importance of feature d, $V = \{(a,t_a); a \in \text{eval. agents}, t_a \in \text{valid time steps}\}$. Larger values of M_{WOSAC} indicate that the simulator's distribution of kinematic, interactive, and map-based features more closely aligns with real-world behavior.

3.3. Dense Reward Signal via RLOO

The metametric evaluates the realism of generated traffic rollouts from various aspects, including map compliance, interactivity, and kinematic features. A question that arises is: why not use the metametric as a reward signal for RL-based alignment of simulation models? In fact, the realism metametric could be used in its native form for RL-based realism alignment; however, it would be highly sparse andsample inefficient as it is defined as a mapping from 32 rollouts to a scalar value.

An approach to overcome this issue would be to use a small number of rollouts for metametric calculation; however, this also comes with the cost of a high-variance reward signal, and thus a high-variance policy gradient estimate in policy gradient-based methods.

To overcome the sparsity of the metametric as a reward signal, we adapt the RLOO technique from reinforcement learning literature. The RLOO technique uses multiple Monte Carlo rollouts for each seed context and history data to obtain the baseline [8]. Its advantage for a specific rollout is given by averaging the rewards for the rest of the rollouts. The objective function $\mathcal{J}_{\text{RLOO}}(\theta)$ is defined as:

$$\mathbb{E}_{\substack{(S_{<t},C)\sim\mathcal{D}\\ \{S_{i,t:T}\}_{i=1}^{k}\sim\pi_\theta}}\left[\frac{1}{k}\sum_{i=1}^{k}\left(R_i - \frac{1}{k-1}\sum_{j\neq i} R_j\right)\right], \qquad (3)$$

where $\mathcal{D}$ is the dataset, and the trajectories are sampled from the current policy $\pi_\theta(\cdot \mid S_{<t}, C)$.

RLOO provides variance reduction by using a baseline computed from other rollouts in the same batch, similar to how baselines are used in standard policy gradient methods to reduce variance. Importantly, the leave-one-out construction ensures that the baseline remains unbiased, as it is computed independently of each sample's reward [8]. This approach enables effective RL-based post-training by providing a reward signal for each rollout rather than requiring full batches of rollouts, while maintaining reasonable variance properties through the baseline subtraction mechanism. To further reduce the variance, similar to [12], we normalize the reward signal across the rollouts. Moreover, we add a Kullback-Leibler divergence term to the objective function to encourage the policy to stay close to the pretrained policy. The final objective function $\mathcal{J}_{\text{RLFTSim}}(\theta)$ is given by:

$$\mathbb{E}_{\substack{(S_{<t},C)\sim\mathcal{D}\\ \{S_{i,t:T}\}_{i=1}^{k}\sim\pi_\theta}}\left[\frac{1}{k}\sum_{i=1}^{k}\left(\tilde{R}_i - \frac{1}{k-1}\sum_{j\neq i}\tilde{R}_j\right)\right] - \beta\mathbb{E}_{(S_{<t},C)\sim\mathcal{D}}[D_{KL}(\pi_{\theta_0}||\pi_\theta)], \qquad (4)$$

where $\tilde{R}_i = \frac{R_i}{\sigma_R}$ is the normalized reward with σ_R being the standard deviation of rewards across the batch, π_{θ_0} is the pretrained policy, and β is the KL regularization coefficient that controls the trade-off between reward maximization and staying close to the original policy.

4. Experiments

In this section, we present the results of our experiments to answer the following research question: **RQ:** Is reinforcement learning-based post-training with dense reward signals effective in enhancing the realism of simulated traffic rollouts?

4.1. Simulation Realism Enhancement with RLFT

The evaluation results for the simulation challenge are provided in Tab. 1. We use SMART-tiny, a state-of-the-art simulation model, as the base model [15]. After RL-based fine-tuning for 18K steps, we obtain the SMART-RLFTSim model. Compared with the base model, we can see that the realism metametric performance across all dimensions has improved (bottom row). Moreover, our model outperforms state-of-the-art models in two dimensions: interactive and map-based. We note that SMART-tiny CLSFT [16] is another fine-tuning method that, similar to our case, uses SMART-tiny as the base model. Our model achieves comparable fine-tuning performance using a single epoch with 30% of the training data, while SMART-tiny CLSFT is fine-tuned using 10 epochs of training data. We note that the ideal realism score is bounded by the value that we can obtain for the ground truth rollouts. The oracle's metametric is lower than one due to metric definition properties.

4.2. Implementation Details

The SMART-tiny base model is trained for 32 epochs on WOMD following [16].

Table 1. Traffic simulation benchmarking results. Results are based on WOSAC 2025 leaderboard evaluation on the private test split. (↓)/(↑) indicate lower/higher values are better.

Model	Metametric↑	Kinematic↑	Interactive↑	Map-based↑	minADE↓	Offroad Likelihood↑	Collision Likelihood↑
SMART-R1	**0.7858**	**0.4944**	0.8110	**0.9201**	**1.2885**	0.9554	0.9709
TrajTok	0.7852	0.4887	0.8116	0.9207	1.3179	**0.9555**	0.9742
unimotion	0.7851	0.4943	0.8105	0.9187	1.3036	0.9535	0.9693
SMART-tiny CLSFT[16]	0.7846	0.4931	0.8106	0.9177	1.3065	0.9524	0.9702
SMART-RLFTSim (ours)	0.7844	0.4893	**0.8128**	0.9164	1.3470	0.9512	**0.9755**
comBOT	0.7837	0.4899	0.8102	0.9175	1.3687	0.9525	0.9703
AgentFormer	0.7836	0.4906	0.8103	0.9167	1.3422	0.9515	0.9706
UniMM[9]	0.7829	0.4914	0.8089	0.9161	1.2949	0.9505	0.9680
SMART-topk32	0.7814	0.4854	0.8089	0.9153	1.3931	0.9500	0.9693

For the RLFT step, we employ the following hyperparameters: learning rate of 3e-6 with linear warmup for 500 initial steps starting from 3e-7, KL regularization coefficient $\beta = 0.001$, 6 training rollouts per batch, and batch size of 16. We use the AdamW optimizer with weight decay of 0.01 and default values for other hyperparameters.

5. Conclusion

We present **RLFTSim**, a reinforcement learning fine-tuning framework that addresses the alignment problem in traffic simulation by directly optimizing for realism metrics. Our key contribution is adapting the RLOO technique to overcome the sparsity of the WOSAC realism metametric, enabling dense reward signals while maintaining unbiased gradient estimates for effective RL-based post-training.

Experimental evaluation on the Waymo Open Motion Dataset demonstrates that RLFTSim achieves comparable realism enhancement to existing closed-loop fine-tuning methods while requiring significantly less training data—using only a single epoch on 30% of WOMD's training split compared to 10 epochs for supervised approaches. This data efficiency stems from our dense reward formulation that provides per-rollout feedback rather than requiring full batches of 32 rollouts.

Our approach successfully bridges the gap between open-loop pretrained models and closed-loop deployment by leveraging well-defined realism metrics as reward signals, avoiding the challenge of designing custom reward functions. The RLOO adaptation with reward normalization and KL regularization provides a practical solution for aligning traffic simulation models with real-world driving behavior.

References

[1] Montgomery Alban, Ehsan Ahmadi, Randy Goebel, and Amir Rasouli. Getting smarter for motion planning in autonomous driving systems. *arXiv:2502.15824*, 2025.

[2] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv:2106.11810*, 2021.

[3] Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *NeurIPS*, 2019.

[4] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles Qi, Yin Zhou, Zoey Yang, Aurelien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. *arXiv:2104.10133*, 2021.

[5] Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. In *NeurIPS*, 2024.

[6] Maximilian Igl, Daewoo Kim, Alex Kuefler, Paul Mougin, Punit Shah, Kyriacos Shiarlis, Dragomir Anguelov, Mark Palatucci, Brandyn White, and Shimon Whiteson. Symphony: Learning realistic and diverse agents for autonomous driving simulation. In *ICRA*, 2022.

[7] Nidhi Kalra and Susan M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 2016.

[8] Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! In *Deep Reinforcement Learning Meets Structured Prediction Workshop at ICLR*, 2019.

[9] Longzhong Lin, Xuewu Lin, Kechun Xu, Haojian Lu, Lichao Huang, Rong Xiong, and Yue Wang. Revisit mixture models for multi-agent simulation: Experimental study within a unified framework. *arXiv:2501.17015*, 2025.

[10] Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Rebecca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, Dragomir Anguelov, and Sergey Levine. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *IROS*, 2023.

[11] Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *ICML*, 2024.

[12] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*, 2024.

[13] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel

Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *CVPR*, 2021.

[14] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E*, 2000.

[15] Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng KAN. SMART: Scalable multi-agent real-time motion generation via next-token prediction. In *NeurIPS*, 2024.

[16] Zhejun Zhang, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. In *CVPR*, 2025.

[17] Zikang Zhou, Haibo Hu, Xinhong Chen, Jianping Wang, Nan Guan, Kui Wu, Yung-Hui Li, Yu-Kai Huang, and Chun Jason Xue. Behaviorgpt: Smart agent simulation for autonomous driving with next-patch prediction. In *NeurIPS*, 2024.