Supplementary Material for RLFTSim: Realistic and Controllable Multi-Agent Traffic Simulation via Reinforcement Learning Fine-Tuning

Ehsan Ahmadi[1,2] Hunter Schofield[2,3] Behzad Khamidehi[2] Fazel Arasteh[2]
Jinjun Shan[3] Lili Mou[1,4] Dongfeng Bai[2] Kasra Rezaee[2]
[1]University of Alberta [2]Huawei, Noah's Ark Lab [3]York University [4]Canada CIFAR AI Chair, Amii
eahmadi@ualberta.ca {firstname.lastname}@huawei.com {hunterls, jjshan}@yorku.ca doublepower.mou@gmail.com

A. Methodological Details

A.1. Detailed RMM Formulation

We provide the complete derivation of the WOSAC Realism Meta-Metric (RMM) [1] that was summarized in § 3.

To compute the WOSAC meta-metric, 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 $t \in \{1, \dots, T\}$, we extract a D-dimensional feature vector:

$$\mathbf{f}_t^{(i)} = \big(f_{1,t}^{(i)},\, f_{2,t}^{(i)},\, \dots,\, f_{D,t}^{(i)}\big), \tag{S1}$$

whose components include:

- **Kinematic features:** linear/angular speed and acceleration
- **Interactive features:** closest distance to other agents, time-to-collision (TTC), and accident indication
- **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.

Given the seed scenario and its group of simulated rollouts, we first form **time-dependent** empirical distributions:

$$\hat{P}_{d,a,t}(k) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\big\{f_{d,a,t}^{(i)} \in \mathcal{B}_{d,a,k}\big\}, \tag{S2}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function.

Then, we marginalize time to obtain a **time-independent** histogram:

$$\hat{P}_{d,a}(k) = \frac{1}{T}\sum_{t=1}^{T} \hat{P}_{d,a,t}(k). \tag{S3}$$

Finally, letting $f_{d,a,t}^*$ fall into bin $k_{d,a,t}^*$ when observed on the ground truth, the WOSAC realism meta-metric is defined as a weighted geometric mean over all feature dimensions:

$$\mathrm{RMM} = \sum_{d=1}^{D} w_d \left[\prod_{(a,t_a)\in V} \hat{P}_{d,a}(k_{d,a,t_a}^*)\right]^{\frac{1}{|V|}}, \tag{S4}$$

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

A.2. Proofs and Mathematical Derivations

Proof of Proposition 1. Starting from the definition of g,

$$\begin{aligned}\mathbb{E}[g] = \sum_{i=1}^{N} \mathbb{E}\Bigg[&\nabla_\theta \log \pi_\theta(\tau_i) \\ &\times \left(\frac{1}{N}\sum_{j=1}^{N} \mathrm{RMM}_{-j} - \mathrm{RMM}_{-i}\right)\Bigg].\end{aligned} \tag{S5}$$

Expanding the two terms gives

$$\begin{aligned}\mathbb{E}[g] = \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}\left[\left(\sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(\tau_i)\right) \mathrm{RMM}_{-j}\right] \\ - \sum_{i=1}^{N} \mathbb{E}[\nabla_\theta \log \pi_\theta(\tau_i)\, \mathrm{RMM}_{-i}]\,.\end{aligned} \tag{S6}$$

Step 1: The leave-one-out subtraction term has zero expectation. Fix any $i \in \{1, \dots, N\}$. Since RMM_{-i} is computed from τ_{-i}, it depends only on the rollout set excluding τ_i. Because the rollouts are sampled i.i.d., τ_i is independent of τ_{-i}. Therefore, by the tower property,

$$\begin{aligned}\mathbb{E}[\nabla_\theta \log \pi_\theta(\tau_i)\, \mathrm{RMM}_{-i}] = \\ \mathbb{E}_{\tau_{-i}}[\mathrm{RMM}_{-i}\, \mathbb{E}_{\tau_i}[\nabla_\theta \log \pi_\theta(\tau_i)]]\,.\end{aligned} \tag{S7}$$

Using the score-function identity

$$\mathbb{E}_{\tau_i\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(\tau_i)] = \int \pi_\theta(\tau_i)\nabla_\theta \log \pi_\theta(\tau_i)\, d\tau_i \\ = \nabla_\theta \int \pi_\theta(\tau_i)\, d\tau_i = 0, \tag{S8}$$

we obtain

$$\mathbb{E}[\nabla_\theta \log \pi_\theta(\tau_i)\, \mathrm{RMM}_{-i}] = 0. \tag{S9}$$

Since this holds for every i,

$$\sum_{i=1}^{N} \mathbb{E}[\nabla_\theta \log \pi_\theta(\tau_i)\, \mathrm{RMM}_{-i}] = 0. \tag{S10}$$

Hence,

$$\mathbb{E}[g] = \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}\left[\left(\sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(\tau_i)\right) \mathrm{RMM}_{-j}\right]. \tag{S11}$$

Step 2: Apply the score-function identity to the joint distribution. Because the joint rollout distribution factorizes as $\pi_\theta(\tau_{1:N}) = \prod_{i=1}^{N} \pi_\theta(\tau_i)$, we have

$$\sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(\tau_i) = \nabla_\theta \log \pi_\theta(\tau_{1:N}). \tag{S12}$$

Therefore,

$$\mathbb{E}[g] = \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}[\nabla_\theta \log \pi_\theta(\tau_{1:N})\, \mathrm{RMM}_{-j}]\,. \tag{S13}$$

Now fix j. Writing $\nabla_\theta \log \pi_\theta(\tau_{1:N}) = \nabla_\theta \log \pi_\theta(\tau_j) + \nabla_\theta \log \pi_\theta(\tau_{-j})$, the τ_j component contributes zero by the same argument as Step 1 (independence of τ_j and RMM_{-j}). The remaining score $\nabla_\theta \log \pi_\theta(\tau_{-j})$ and RMM_{-j} both depend only on τ_{-j}, so the score-function identity on the marginal $\pi_\theta(\tau_{-j})$ gives

$$\mathbb{E}[\nabla_\theta \log \pi_\theta(\tau_{1:N})\, \mathrm{RMM}_{-j}] = \nabla_\theta \mathbb{E}[\mathrm{RMM}_{-j}]. \tag{S14}$$

Substituting this into the previous display yields

$$\mathbb{E}[g] = \frac{1}{N}\sum_{j=1}^{N} \nabla_\theta \mathbb{E}[\mathrm{RMM}_{-j}] \\ = \nabla_\theta \mathbb{E}\left[\frac{1}{N}\sum_{j=1}^{N} \mathrm{RMM}_{-j}\right]. \tag{S15}$$

Step 3: Relate the objective to expected RMM on $N-1$ rollouts. By exchangeability, each RMM_{-j} has the same distribution as the meta-metric computed on $N-1$ i.i.d. rollouts. Hence,

$$\mathbb{E}\left[\frac{1}{N}\sum_{j=1}^{N} \mathrm{RMM}_{-j}\right] = \mathbb{E}[\mathrm{RMM}(\tau_{1:N-1})]\,. \tag{S16}$$

This proves the claim. □

Proof of Proposition 2. The full RMM Eq. 1 is a finite weighted sum $\sum_{d=1}^{D} w_d\, \mathrm{RMM}_d$ over D feature dimensions. We first establish the $O((N_{\text{eff}} \cdot T)^{-1})$ bound for each per-feature component $\mathrm{RMM}_d = \prod_{k=1}^{K} \widetilde{P}_k^{\alpha_k}$, written as RMM for brevity, then lift to the aggregate (Step 6). We proceed in six steps.

Step 1: Direct multinomial sampling estimator. Fix feature d and drop subscripts for clarity. Let $S = NT$ be the total sample size from N rollouts of length T. The simulator generates S independent feature observations $\{X_\ell\}_{\ell=1}^{S}$ directly from the simulator distribution q, where each X_ℓ falls in bin k with probability q_k. The empirical histogram estimator is

$$\widetilde{P}_k = \frac{1}{S}\sum_{\ell=1}^{S} \mathbf{1}\{X_\ell = k\}. \tag{S17}$$

This estimator directly measures the empirical frequency of simulator features in bin k.

Step 2: Unbiasedness. Since samples are drawn from the simulator distribution q, we have $\mathbb{E}_{X_\ell\sim q}[\mathbf{1}\{X_\ell = k\}] = q_k$. By linearity of expectation,

$$\mathbb{E}_{X_\ell\sim q}\left[\widetilde{P}_k\right] = \frac{1}{S}\sum_{\ell=1}^{S} \mathbb{E}_{X_\ell\sim q}[\mathbf{1}\{X_\ell = k\}] = q_k. \tag{S18}$$

Step 3: Variance and covariance of the estimator. Since samples are independent and drawn from multinomial distribution with parameters $(S, q_1, \ldots, q_K)$, the variance is:

$$\mathrm{Var}_{X_\ell\sim q}[\widetilde{P}_k] = \frac{1}{S}\cdot \mathrm{Var}_{X_\ell\sim q}[\mathbf{1}\{X_\ell = k\}] \tag{S19}$$

$$= \frac{1}{S} q_k(1-q_k) \tag{S20}$$

$$= \frac{q_k(1-q_k)}{S}. \tag{S21}$$

For the covariance structure, consider $i \neq j$:

$$\mathrm{Cov}[\widetilde{P}_i, \widetilde{P}_j] = \mathrm{Cov}\left[\frac{1}{S}\sum_{\ell=1}^{S} \mathbf{1}\{X_\ell = i\}, \frac{1}{S}\sum_{\ell=1}^{S} \mathbf{1}\{X_\ell = j\}\right] \tag{S22}$$

$$= \frac{1}{S}\mathrm{Cov}[\mathbf{1}\{X = i\}, \mathbf{1}\{X = j\}] \tag{S23}$$

$$= \frac{1}{S}\mathbb{E}[\mathbf{1}\{X = i\}\mathbf{1}\{X = j\}]$$

$$-\frac{1}{S}\mathbb{E}[\mathbf{1}\{X=i\}]\mathbb{E}[\mathbf{1}\{X=j\}] \tag{S24}$$

$$=0-\frac{1}{S}q_iq_j \tag{S25}$$

$$=-\frac{q_iq_j}{S} \tag{S26}$$

where Eq. S25 uses $\mathbf{1}\{X=i\}\mathbf{1}\{X=j\}=0$ for $i\neq j$.

Step 4: Variance inflation and effective sample size. The variance of each bin estimator depends on how the simulator distribution q matches the ground truth frequencies α. Define the variance inflation factor $\kappa=\sum_{k=1}^{K}\frac{\alpha_k^2}{q_k}$, which measures the mismatch between ground truth frequencies α and simulator distribution q. Here we can derive the following bounds on κ:

Lower bound: By Cauchy-Schwarz inequality with vectors $\mathbf{u}=\left(\frac{\alpha_1}{\sqrt{q_1}},\ldots,\frac{\alpha_K}{\sqrt{q_K}}\right)$ and $\mathbf{v}=(\sqrt{q_1},\ldots,\sqrt{q_K})$:

$$\left(\sum_{k=1}^{K}\frac{\alpha_k}{\sqrt{q_k}}\cdot\sqrt{q_k}\right)^2\leq\left(\sum_{k=1}^{K}\frac{\alpha_k^2}{q_k}\right)\left(\sum_{k=1}^{K}q_k\right). \tag{S27}$$

Since $\sum_{k=1}^{K}\alpha_k=\sum_{k=1}^{K}q_k=1$, we get:

$$1^2\leq\kappa\cdot 1\quad\Rightarrow\quad\kappa\geq 1. \tag{S28}$$

Equality holds when $\alpha_k=q_k$ for all k (perfect simulator). When the simulator is biased, $\kappa>1$.

Upper bound: Since $\alpha_k\leq 1$ and $q_k>0$ for all k in the support, we have:

$$\kappa=\sum_{k=1}^{K}\frac{\alpha_k^2}{q_k}\leq\frac{1}{\min_{k:q_k>0}q_k}. \tag{S29}$$

The effective sample size is defined as $N_{\text{eff}}=\frac{N}{\kappa}$, giving us the bounds:

$$N\cdot\min_{k:q_k>0}q_k\leq N_{\text{eff}}\leq N. \tag{S30}$$

Step 5: Meta-metric variance. The meta-metric has the form $\text{RMM}=\prod_{k=1}^{K}\widetilde{P}_k^{\alpha_k}$ where α_k are fixed ground truth frequencies and $\sum_{k=1}^{K}\alpha_k=1$. Taking logarithms:

$$\log(\text{RMM})=\sum_{k=1}^{K}\alpha_k\log(\widetilde{P}_k). \tag{S31}$$

By the first-order delta method, since $\text{Var}[\log(\widetilde{P}_k)]\approx\frac{\text{Var}[\widetilde{P}_k]}{(\mathbb{E}[\widetilde{P}_k])^2}=\frac{\text{Var}[\widetilde{P}_k]}{q_k^2}$ and $\text{Cov}[\log(\widetilde{P}_i),\log(\widetilde{P}_j)]\approx\frac{\text{Cov}[\widetilde{P}_i,\widetilde{P}_j]}{q_iq_j}$, we have:

$$\text{Var}[\log(\text{RMM})]=\text{Var}\left[\sum_{k=1}^{K}\alpha_k\log(\widetilde{P}_k)\right] \tag{S32}$$

$$=\sum_{k=1}^{K}\alpha_k^2\text{Var}[\log(\widetilde{P}_k)]+\sum_{i\neq j}\alpha_i\alpha_j\text{Cov}[\log(\widetilde{P}_i),\log(\widetilde{P}_j)] \tag{S33}$$

$$=\sum_{k=1}^{K}\alpha_k^2\frac{\text{Var}[\widetilde{P}_k]}{q_k^2}+\sum_{i\neq j}\alpha_i\alpha_j\frac{\text{Cov}[\widetilde{P}_i,\widetilde{P}_j]}{q_iq_j}. \tag{S34}$$

Substituting the variance and covariance formulas from Step 3:

$$\text{Var}[\log(\text{RMM})]=\sum_{k=1}^{K}\alpha_k^2\frac{q_k(1-q_k)/S}{q_k^2}+\sum_{i\neq j}\alpha_i\alpha_j\frac{-q_iq_j/S}{q_iq_j} \tag{S35}$$

$$=\sum_{k=1}^{K}\alpha_k^2\frac{1-q_k}{Sq_k}-\sum_{i\neq j}\alpha_i\alpha_j\frac{1}{S} \tag{S36}$$

$$=\frac{1}{S}\left[\sum_{k=1}^{K}\alpha_k^2\frac{1-q_k}{q_k}-\sum_{i\neq j}\alpha_i\alpha_j\right] \tag{S37}$$

$$=\frac{1}{S}\left[\sum_{k=1}^{K}\alpha_k^2\frac{1-q_k}{q_k}-\left(\left(\sum_{k=1}^{K}\alpha_k\right)^2-\sum_{k=1}^{K}\alpha_k^2\right)\right] \tag{S38}$$

$$=\frac{1}{S}\left[\sum_{k=1}^{K}\alpha_k^2\frac{1-q_k}{q_k}-\left(1-\sum_{k=1}^{K}\alpha_k^2\right)\right] \tag{S39}$$

$$=\frac{1}{S}\left[\sum_{k=1}^{K}\alpha_k^2\left(\frac{1-q_k}{q_k}+1\right)-1\right] \tag{S40}$$

$$=\frac{1}{S}\left[\sum_{k=1}^{K}\frac{\alpha_k^2}{q_k}-1\right] \tag{S41}$$

$$=\frac{\kappa-1}{S}=O\left(\frac{\kappa}{NT}\right)=O\left(\frac{1}{N_{\text{eff}}T}\right). \tag{S42}$$

Finally, applying the delta method to $\text{RMM}=\exp(\log(\text{RMM}))$:

$$\text{Var}[\text{RMM}]\approx(\text{RMM})^2\cdot\text{Var}[\log(\text{RMM})]=O\left(\frac{1}{N_{\text{eff}}T}\right). \tag{S43}$$

Step 6: Lifting to the aggregate. Each per-feature component satisfies $\text{Var}(\text{RMM}_d)=O((N_{\text{eff},d}\cdot T)^{-1})$ where $N_{\text{eff},d}=N/\kappa_d$. By the sub-additivity of standard deviation and $\sum_d w_d=1$,

$$\text{Var}(\textstyle\sum_d w_d\,\text{RMM}_d)\leq(\sum_d w_d)^2\max_d\text{Var}(\text{RMM}_d)$$

$$= \max_d \mathrm{Var}(\mathrm{RMM}_d)$$
$$= O\left(\frac{1}{\hat{N}_{\text{eff}} T}\right), \tag{S44}$$

where $\hat{N}_{\text{eff}} = N/\max_d \kappa_d$. The bounds from Step 4 give $1 \le \kappa_d \le 1/\min_{\substack{k: \\ q_{k,d}>0}} q_{k,d}$ for each d, so $\max_d \kappa_d \le 1/\min_{\substack{k,d: \\ q_{k,d}>0}} q_{k,d}$ and hence $\hat{N}_{\text{eff}} \in \left[N \cdot \min_{\substack{k,d: \\ q_{k,d}>0}} q_{k,d},\ N\right]$. □

Proof of Proposition 3. We establish the variance bounds for both estimators using approximations suitable for the leave-one-out setting.

Step 1: Variance of MLOO. From Proposition 2, each RMM_{-i} has variance $\mathrm{Var}(\mathrm{RMM}_{-i}) = O(1/((N-1)T))$.

The MLOO estimator can be written as:

$$\mathrm{RMM}_i^{\text{MLOO}} = \frac{1}{N}\sum_{j=1}^{N} \mathrm{RMM}_{-j} - \mathrm{RMM}_{-i} \tag{S45}$$
$$= \frac{1}{N}\sum_{j\neq i} \mathrm{RMM}_{-j} + \frac{1}{N}\mathrm{RMM}_{-i} - \mathrm{RMM}_{-i} \tag{S46}$$
$$= \frac{1}{N}\sum_{j\neq i} \mathrm{RMM}_{-j} - \frac{N-1}{N}\mathrm{RMM}_{-i}. \tag{S47}$$

Since the leave-one-out estimates are correlated (they share $N-2$ common rollouts), we need to account for covariances. Let $\sigma^2 = \mathrm{Var}(\mathrm{RMM}_{-i}) = \frac{C_{\text{var}}}{(N-1)T}$ for some constant $C_{\text{var}} > 0$. For $i \neq j$, we approximate the covariance between RMM_{-i} and RMM_{-j} as proportional to the fraction of shared rollouts:

$$\mathrm{Cov}(\mathrm{RMM}_{-i}, \mathrm{RMM}_{-j}) \approx \frac{N-2}{N-1}\cdot\sigma^2 = \frac{N-2}{N-1}\cdot\frac{C_{\text{var}}}{(N-1)T}. \tag{S48}$$

Therefore:

$$\mathrm{Var}(\mathrm{RMM}_i^{\text{MLOO}})$$
$$= \mathrm{Var}\left(\frac{1}{N}\sum_{j\neq i}\mathrm{RMM}_{-j} - \frac{N-1}{N}\mathrm{RMM}_{-i}\right) \tag{S49}$$
$$= \frac{1}{N^2}\mathrm{Var}\left(\sum_{j\neq i}\mathrm{RMM}_{-j}\right) + \frac{(N-1)^2}{N^2}\mathrm{Var}(\mathrm{RMM}_{-i}) - 2\cdot\frac{N-1}{N^2}\mathrm{Cov}\left(\mathrm{RMM}_{-i}, \sum_{j\neq i}\mathrm{RMM}_{-j}\right) \tag{S50}$$
$$= \frac{1}{N^2}\left[(N-1)\sigma^2 + (N-1)(N-2)\cdot\frac{N-2}{N-1}\sigma^2\right] + \frac{(N-1)^2}{N^2}\sigma^2 - 2\cdot\frac{N-1}{N^2}\cdot(N-1)\cdot\frac{N-2}{N-1}\sigma^2 \tag{S51}$$
$$= \frac{1}{N^2}\left[(N-1) + (N-1)(N-2)^2/(N-1)\right]\sigma^2 + \frac{(N-1)^2}{N^2}\sigma^2 - 2\cdot\frac{(N-1)(N-2)}{N^2}\sigma^2 \tag{S52}$$
$$= \frac{1}{N^2}\left[(N-1) + (N-2)^2\right]\sigma^2 + \frac{(N-1)^2}{N^2}\sigma^2 - 2\cdot\frac{(N-1)(N-2)}{N^2}\sigma^2 \tag{S53}$$
$$= \frac{\sigma^2}{N^2}\left[N-1+(N-2)^2+(N-1)^2 - 2(N-1)(N-2)\right] \tag{S54}$$
$$= \frac{\sigma^2}{N^2}\left[N-1+((N-2)-(N-1))^2\right] \tag{S55}$$
$$= \frac{\sigma^2}{N^2}\left[N-1+1\right] = \frac{N\sigma^2}{N^2} = \frac{\sigma^2}{N} = \frac{C_{\text{var}}}{N(N-1)T}. \tag{S56}$$

Step 2: Variance of RLOO. The RLOO estimator is:

$$\mathrm{RMM}_i^{\text{RLOO}} = \mathrm{RMM}_i - \frac{1}{N-1}\sum_{j\neq i}\mathrm{RMM}_j.$$

Since RMM_i evaluates the meta-metric on a single rollout (N=1, S=T), Proposition 2 gives $\tau^2 = \mathrm{Var}(\mathrm{RMM}_i) = \frac{D}{T}$ for some constant $D > 0$. Because rollouts are i.i.d., the RMM_i are independent, hence:

$$\mathrm{Var}(\mathrm{RMM}_i^{\text{RLOO}}) = \mathrm{Var}\left(\mathrm{RMM}_i - \frac{1}{N-1}\sum_{j\neq i}\mathrm{RMM}_j\right) \tag{S57}$$
$$= \mathrm{Var}(\mathrm{RMM}_i) + \mathrm{Var}\left(\frac{1}{N-1}\sum_{j\neq i}\mathrm{RMM}_j\right) \tag{S58}$$
$$= \tau^2 + \frac{1}{(N-1)^2}\mathrm{Var}\left(\sum_{j\neq i}\mathrm{RMM}_j\right) \tag{S59}$$
$$= \tau^2 + \frac{1}{(N-1)^2}\cdot(N-1)\tau^2 \tag{S60}$$
$$= \tau^2 + \frac{\tau^2}{N-1} \tag{S61}$$
$$= \frac{N\tau^2}{N-1} \tag{S62}$$
$$= \frac{ND}{(N-1)T}. \tag{S63}$$

Since $\frac{N}{N-1}$ is bounded (specifically, $1 < \frac{N}{N-1} \le 2$ for $N \ge 2$), we have:

$$\mathrm{Var}(\mathrm{RMM}_i^{\text{RLOO}}) = O\left(\frac{1}{T}\right). \tag{S64}$$

Table S1. Hyperparameter sweep ranges explored for RLFTSim. Final values are highlighted in **bold**.

Hyperparameter	Values swept
Learning rate	1e-6, **3e-6**, 1e-5, 3e-5
Adam β_1	**0.9**, 0.95
Warmup steps	**100,** 500
Entropy bonus	**0.0**, 0.01
Training rollouts per update	2, **4**, 6, 8
Weight decay	0.0, **0.01**
Batch size	**8**,16,32
Gradient clipping	**1.0**
KL controller: horizon	**5**
KL controller: min	**1e-3**
KL controller: max	**1e+3**
KL controller: target	0.005,**0.01**
Sync. ref. model: steps	500
Sync. ref. model: alpha	0.005
GCFT goal reward weight λ	**0.1**, 0.5

□

A.3. Goal Conditioning Architectures

Here we provide further implementation details on the goal conditioning methods discussed in § 4.2.

Agent Token Embedding Concatenation. An intuitive method of including goal information in the observation is to directly include the goal coordinate, $\mathbf{x}_g^i$, or the relative goal position, $\mathbf{r}_g^i = (r_g^i, \phi_g^i)$, in the observed state for each agent, $S_t = \{S_{t'}^i || \mathbf{x}_g^i \mid i \leq N_a, t' \leq t\}$, where $||$ denotes concatenation. However, since the domain of goals is expansive, it can be difficult for a model to learn an appropriate feature vector. This is because goal coordinates are continuous, and the concatenation process further increases the dimensionality of the embedding vector, which increases the fine-tuning iterations required to generalize.

Positional Encoding Indication. Instead of including goal coordinates in the input of the agent token embedding encoder, an alternative method is to extend the relative positional encoding (RPE) to include a binary categorical embedding that indicates whether the relationship between agent i and road token j is a goal relationship, which occurs when $j = P_g^i$. Although this also requires introducing new parameters similar to extending the agent token embeddings, since the input domain is binary, it is easier for a model to learn during the fine-tuning stage. Furthermore, given that goal indication is binary and tied to individual polylines, an agent can be unconditioned by providing no goal indication for any polyline. This allows the simulation for that agent to solely focus on maintaining realism. Thus, this method enables a hybrid style simulation where some agents can be conditioned on particular goals while others remain unconditioned.

B. Implementation Details

Model Training. We train SMART-tiny models on the Waymo Open Motion Dataset (WOMD) for 32 epochs following the implementation and hyperparameters in [2]. For the base model training, we use standard supervised learning with cross-entropy loss for next-token prediction. For the RLFT post-training stage, we use the configuration in Tab. S1. We use the adaptive KL controller from [5] to control the KL divergence between the model's output distribution and the pre-trained model's output distribution, and sustain mode training stability; its hyperparameters are set as in Tab. S1. For the GCFT post-training stage, the hyperparameter configuration is kept the same; however, to ensure that realism is maintained while improving controllability, we use a reward weight of λ=0.1 (Eq. 10). More aggressive goal conditioning can be achieved with higher lambda values. Experiments for both the base model pre-training and fine-tuning are conducted on a server with Intel Xeon Platinum 8180 CPU @2.50GHz, 728GB RAM, and 8x NVIDIA V100 GPUs each with 32GB GPU memory.

Dataset. We use the Waymo Open Motion Dataset (WOMD) for training and evaluation. WOMD has 486,995/44,097/44,920 scenarios in the training/validation/test splits, respectively. Each scenario contains up to 128 agents, including agents of type vehicle, pedestrian, and cyclist. Each scenario has a length of 9.1 seconds, consisting of 1.1 seconds for the history input length and 8 seconds for the future simulation horizon.

Evaluation Protocol. The results in Tab. 1 are based on the private test split of the WOSAC leaderboard. Unless otherwise specified, all ablation studies and analyses are conducted on a randomly selected 20% subset of the WOMD validation split (8,800 scenarios). For realism evaluation, we generate 32 rollouts per scenario following the WOSAC protocol[1]. The agents that only appear in future time steps are excluded from the simulation and evaluation. The evaluation metrics by default are only based on the specified evaluation agent IDs (the ego vehicle and up to 8 agents tagged as tracks_to_predict in the WOMD). Although the other agents are not evaluated, they are included in the simulation and indirectly affect the evaluation metrics for the selected agents.

GCFT Reward Design. For both soft and hard goals, we assign a binary reward to each agent, indicating whether the agent successfully passes (soft target) or reaches (hard target) its designated goal. The final goal-reaching reward for a scenario is computed by averaging this binary signal over the ego agent and all agents labeled as tracks_to_predict.

[1]More details can be found in https://waymo.com/open/challenges/2025/sim-agents/

Table S2. Extended Benchmarking. **Top:** Performance scaling comparison of our RLFTSim vs. CAT-K [4] with the number of fine-tuning epochs. † indicates a weaker reference model with only 1 epoch of pre-training. **Middle:** Stronger realism enhancement with a weaker reference model. **Bottom:** Max realism meta-metric for the ground truth trajectories.

Model	Epoch	RMM↑	Kinematic↑	Interactive↑	Map-based↑
SMART-tiny [2] (ref. model)	0.00	0.77692	0.48329	0.80288	0.91135
SMART-tiny RLFTSim (Ours)	0.25	0.78137	0.49008	0.80807	0.91348
	0.50	0.78183	0.48953	0.80922	0.91364
	1.00	0.78166	0.49001	0.80897	0.91321
	1.50	0.78113	0.48926	0.80873	0.91242
SMART-tiny CAT-K [4]	0.25	0.78093	0.49107	0.80435	0.91644
	0.50	0.78101	0.49093	0.80492	0.91603
	1.00	0.78091	0.49124	0.80499	0.91547
	2.00	0.78086	0.49087	0.80498	0.91556
	5.00	0.77983	0.48991	0.80534	0.91270
†SMART-tiny [2] (ref. model)	0.00	0.75073	0.46441	0.76797	0.89220
†SMART-tiny RLFTSim (Ours)	0.50	0.76368	0.46924	0.78618	0.90301
	1.00	0.76360	0.46851	0.79099	0.89701
	1.50	0.76421	0.46721	0.79045	0.90018
Oracle	NA	0.82925	0.54976	0.85227	0.95935

C. Additional Experimental Results

C.1. Extended Realism Benchmarking

In Tab. S2, we provide a more detailed analysis of RLFTSim's fine-tuning performance. The results show that RLFTSim achieves a higher peak RMM score (0.7818) compared to our re-trained SMART-tiny CAT-K model (0.7810) using their public implementation on the same reference model. While the margin of improvement over the strong baseline may seem modest, it is important to contextualize this within the performance ceiling. The base SMART-tiny model already achieves an RMM of 0.7769, which is approaching the oracle score of 0.8293, defined as the RMM computed when ground-truth trajectories are used as rollouts. As a model's performance nears this upper bound, further gains become increasingly challenging to achieve. We observe that both RLFTSim and the re-trained CAT-K baseline reach their peak RMM within the first epoch of fine-tuning, after which performance plateaus.

The effectiveness of RLFTSim is more prominently illustrated when applied to a weaker starting model, as a diagnostic experiment. As shown in the middle section of Tab. S2, when fine-tuning a less-optimized SMART-tiny model (†SMART-tiny), which is only pre-trained for 1 epoch, with a starting RMM of 0.7507, RLFTSim delivers a substantial performance boost, increasing the RMM to 0.7642 (+1.8%). This demonstrates our method's capability to enhance the realism of the base model, while the margin of improvement is dependent on the starting performance of the base model.

C.2. Heuristic Rewards

Tab. S3 presents a comprehensive comparison of different reward formulations for RL-based traffic simulation alignment. The results demonstrate the effectiveness of our proposed RMM^{MLOO} reward signal compared to heuristic alternatives.

Table S3. Heuristic rewards for the realism meta-metric. All metrics are evaluated on the ego vehicle and agents tagged as `tracks_to_predict` (up to 9 agents). (↑) indicates that larger values are better, and (↓) indicates smaller values are better. Miss rate is computed with the passing goal criterion. **Bold** and underline indicate the best and second best values, respectively.

Reward	RMM ↑	Collision (%) ↓	Offroad (%) ↓	ADE (m) ↓	minADE (m) ↓
SMART-tiny [2] (ref. model)	0.7769	5.67	15.14	2.59	**1.30**
RMM^{MLOO}	**0.7818**	4.53	14.71	2.55	1.31
Collision-offroad-ADE	0.7788	4.93	14.73	**2.39**	1.32
Collision-offroad	0.7769	**4.51**	**13.95**	2.62	1.36

Table S4. Analysis on maneuver controllability benchmark. Targets are only chosen from the set that does not contain the ground-truth maneuver. Only the ego vehicle is evaluated. Note that lower absolute goal completion rates are expected as GT maneuvers are excluded from the benchmark.

Method	Goal Completion Rate (%) ↑	
	Reached	Passed
SMART-tiny [2] (ref. model)	35.374	42.891
RLFTSim (cat, soft)	37.585	**75.340**
RLFTSim (cat, hard)	**50.000**	68.367
RLFTSim (ind, soft)	38.782	64.456
RLFTSim (ind, hard)	44.218	50.680

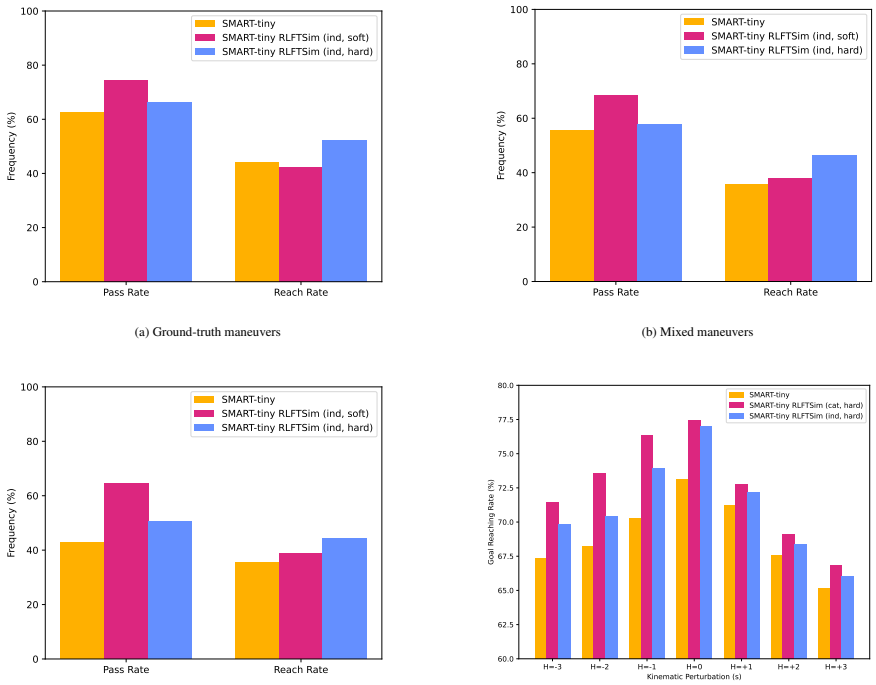
While the collision-offroad reward achieves the lowest collision (4.51%) and offroad (13.95%) rates, it sacrifices overall realism, as evidenced by its lower RMM score (0.7769), which is tied with the base model. In contrast, RMM^{MLOO} achieves the best RMM (0.7818), demonstrating superior alignment with realistic driving behaviors while maintaining competitive safety metrics. The collision-offroad-ade reward, which combines safety metrics with trajectory accuracy, achieves the best ADE (2.39m), but still underperforms RMM^{MLOO} in overall realism. Notably, the base model achieves the best minADE (1.30m), suggesting that pre-trained imitation learning models excel at trajectory accuracy but may not fully capture the distribution of realistic behaviors measured by RMM. These results validate our design choice of using MLOO as the primary reward signal, as it optimizes the official benchmark metric (RMM) while maintaining reasonable performance across all other metrics, including collision rates, offroad rates, and trajectory errors.

C.3. Extended Controllability Analysis

Here we provide more analysis on the controllability benchmarking discussed in § 5.4 and present detailed results.

A key motivation for a controllable simulator is the ability to provide externally supplied behaviors to individual agents, especially those diverging from the agent's original trajectory. To probe this capability, we introduce two benchmarks that assess how effectively a simulation can be conditioned on novel, goal-directed behaviors.

Kinematic Controllability. This benchmark perturbs agents according to their ground-truth kinematics. This benchmark consists of two sections. The first section is a slow kinematic



(a) Ground-truth maneuvers

(b) Mixed maneuvers

(c) Alternative maneuvers

(d) Kinematic controllability

Figure S1. Controllability benchmark performance across various experimental conditions: (a) all goals are set to ground-truth maneuvers, (b) goals are randomly sampled from all maneuvers, (c) goals are exclusively sampled from alternative maneuvers, and (d) simulation controllability with kinematic perturbations. GCFT models consistently outperform the baseline across all conditions, demonstrating effective controllability distillation.

benchmark, where agent goal states are sampled from the ground-truth trajectories along the temporal axis on $(T - H, T]$ with $H \in [1, 3]$ seconds. The second section is a fast kinematic benchmark, where the final kinematic state at time T is propagated using a bicycle model for up to H seconds.

We evaluate the effect of training with hard and soft goals on these scenarios to identify how well different GCFT methods are suited for controllability distillation. Tab. S4 compares the baseline SMART-tiny model and GCFT models in their ability to reach goals for different driving maneuvers from the ground-truth trajectory. After 20K GCFT steps, models trained with either hard or soft goal rewards significantly improve their ability to generate rollouts with new maneuvers.

Maneuver Controllability. Here, we select a subset of 100 scenarios from the WOMD evaluation dataset where the ego vehicle has multiple valid maneuvers. These maneuvers include: drive straight, left turn, right turn, left U-turn, lane change left, and lane change right. For each scenario, we manually selected alternative goal coordinates (§ 4.2), each corresponding to a different driving maneuver from the ground-truth. The resulting benchmark includes both ground-truth and alternative maneuvers.

The maneuver controllability benchmark results are presented across various experimental conditions in Fig. S1. Note that the selected scenarios for this benchmark are particularly difficult since the ego vehicle has a wide range of valid driving behaviours at the beginning of the rollout. When all goals match the ground-truth maneuver (Fig. S1a), GCFT models demonstrate that controllability training preserves the ability to reproduce original behaviors, better than the pre-trained model. The mixed maneuver condition (Fig. S1b) shows robust performance when presented with both original and alternative maneuvers, while the most challenging alternative maneuver condition (Fig. S1c) reveals the model's capability to generate realistic simulations for entirely different driving behaviors than those in the ground-truth data. Similarly, for both mixed and alternative maneuver conditions, the GCFT models outperform the pre-trained model, indicating that GCFT is effective in distilling controllability.

Fig. S1d demonstrates the kinematic controllability results for GCFT models compared to the baseline SMART-tiny model. The figure shows goal-reaching success rates across different temporal perturbation horizons ($H \in [-3, 3]$ seconds) for both forward and backward kinematic extrapolations. Models trained with goals concatenated in the embedding consistently outperform those trained with the goal polyline indication observation, with both GCFT variants significantly

Table S5. Model agnosticism ablation study. Experiments are done using 20% of the WOMD validation split.

Model	RMM↑	Kinematic↑	Interactive↑	Map-based↑
TrafficBots V1.5 [3]	0.71743	0.42712	0.73166	0.86502
TrafficBots V1.5 RLFTSim (Ours)	0.72305	0.43209	0.73773	0.87043

improving upon the baseline across all tested conditions.

C.4. Model Agnosticism

To demonstrate the capability of RLFTSim to be model-agnostic, we fine-tune the TrafficBots V1.5 [3] model using RLFTSim to enhance the performance further. Tab. S5 also compares the TrafficBots V1.5 model after 1 epoch of pre-training with the TrafficBots V1.5 RLFTSim model after a further fine-tuning for 12000 steps. Unlike the SMART baseline model, TrafficBots V1.5 uses a continuous action space and has a variational autoencoder architecture. The improved RMM from 0.7174 to 0.7231 demonstrates the ability of RLFTSim to support both continuous and discrete action spaces as well as a wide range of model architectures.

C.5. Extended Qualitative Examples

The collision and off-road examples (Fig. S2, Fig. S3, Fig. S4, Fig. S5) show how RLFTSim addresses safety violations present in the baseline SMART-tiny model. The pre-trained model generates vehicle-pedestrian collisions, rear-end crashes, and off-road excursions, while RLFTSim produces behaviors that respect traffic rules and adhere to drivable areas. These improvements correspond to the enhanced interactive and map-based metrics reported in Tab. 1.

The goal-conditioned examples (Fig. S6, Fig. S7, Fig. S8) showcase how GCFT distills controllability in the simulation, allowing for specific goals to be specified that the fine-tuned model is capable of reaching.

References

[1] Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nick Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, Brandyn White, and Dragomir Anguelov. The Waymo Open Sim Agents Challenge. In *NeurIPS*, 2023. 1

[2] Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng KAN. SMART: Scalable multi-agent real-time motion generation via next-token prediction. In *NeurIPS*, 2024. 5, 6

[3] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. TrafficBots: Towards world models for autonomous driving simulation and motion prediction. In *ICRA*, 2023. 8

[4] 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. 6

[5] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2019. arXiv:1909.08593. 5

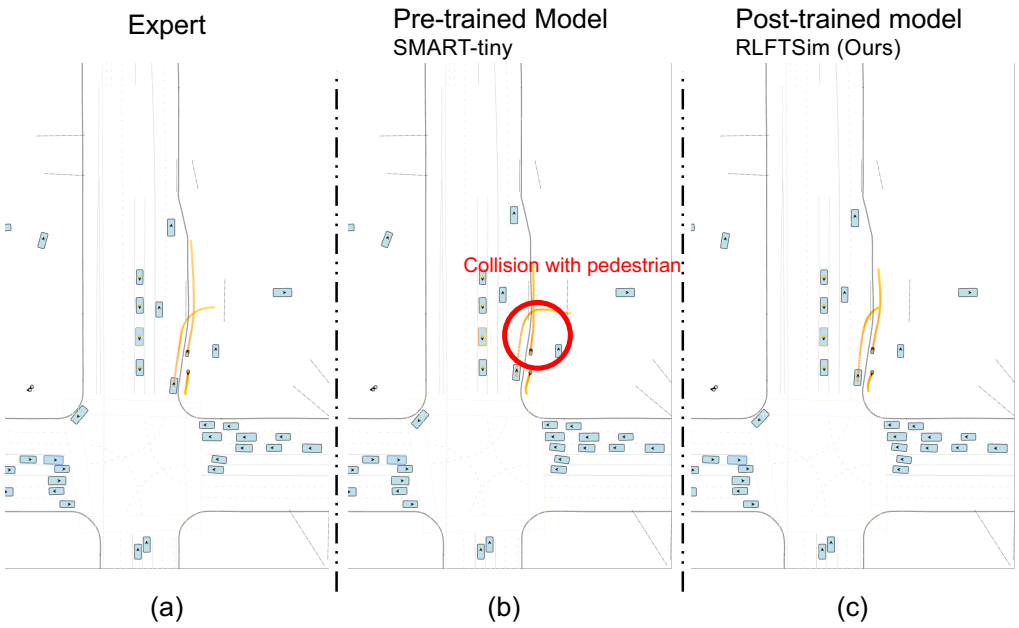


Figure S2. **Qualitative Sample - Collision 1.** (b) In the simulation for the pre-trained model, the vehicle entering the circle fails to yield to the pedestrian and collides with it. (c) There is no collision in the case of the expert data (a) and RLFTSim (c).

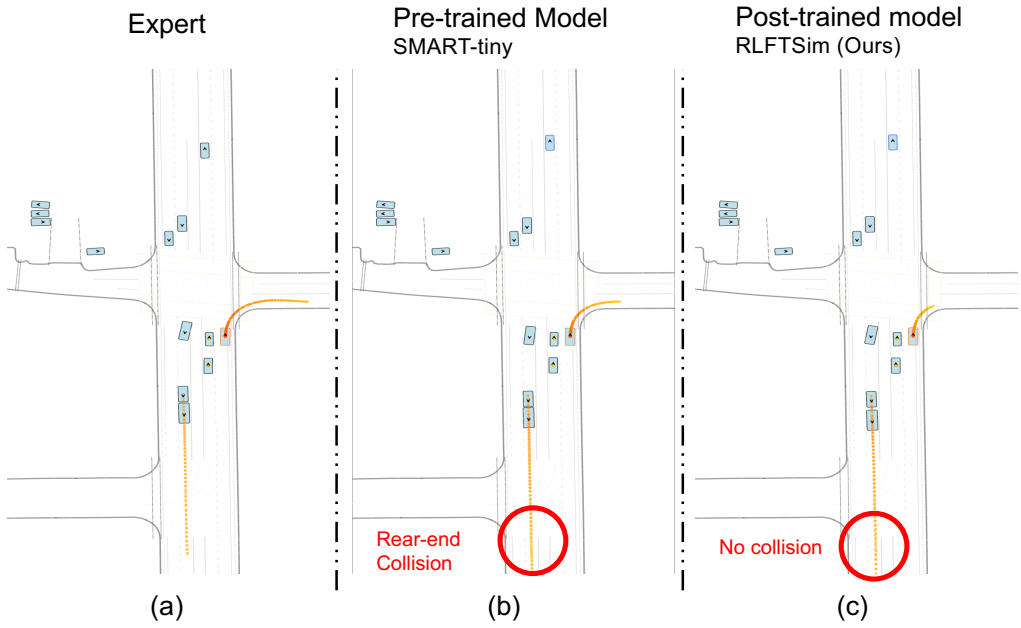


Figure S3. **Qualitative Sample - Collision 2.** (b) For the pre-trained model, there is a rear-end collision between two vehicles in the focused zone. (c) However, the post-trained model avoids this accident.

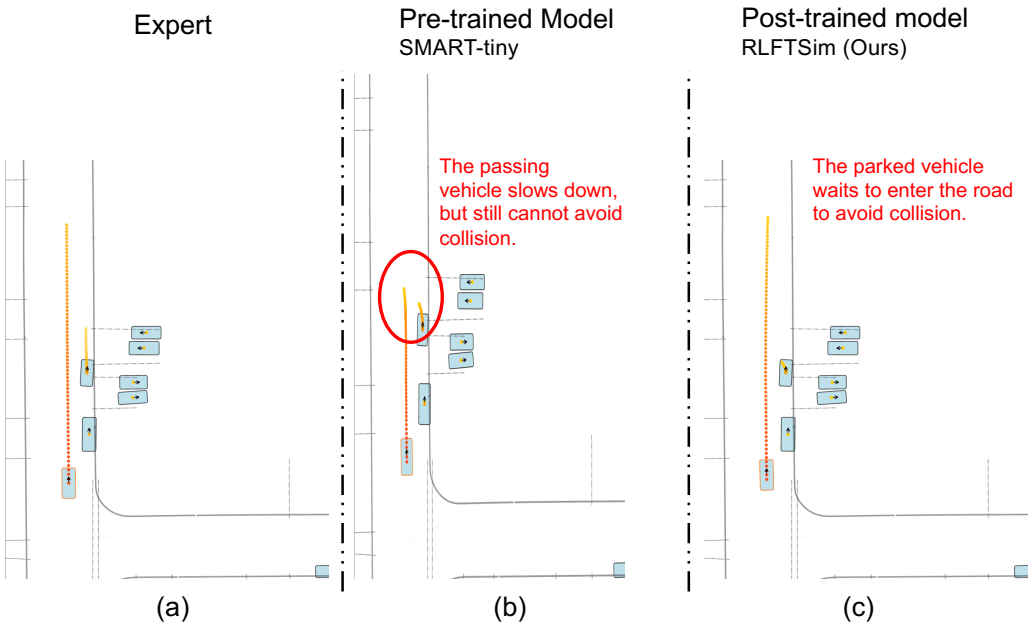


Figure S4. **Qualitative Sample - Collision 3.** (a) The parked vehicle starts to move forward, but does not enter the road to avoid a collision. (b) For the pre-trained model, the parked vehicle attempts to enter the road, which leads to a collision with the passing vehicle. The passing vehicle tries to slow down, but it cannot avoid the collision. (c) The parked vehicle waits for the road to clear and then enters the road.

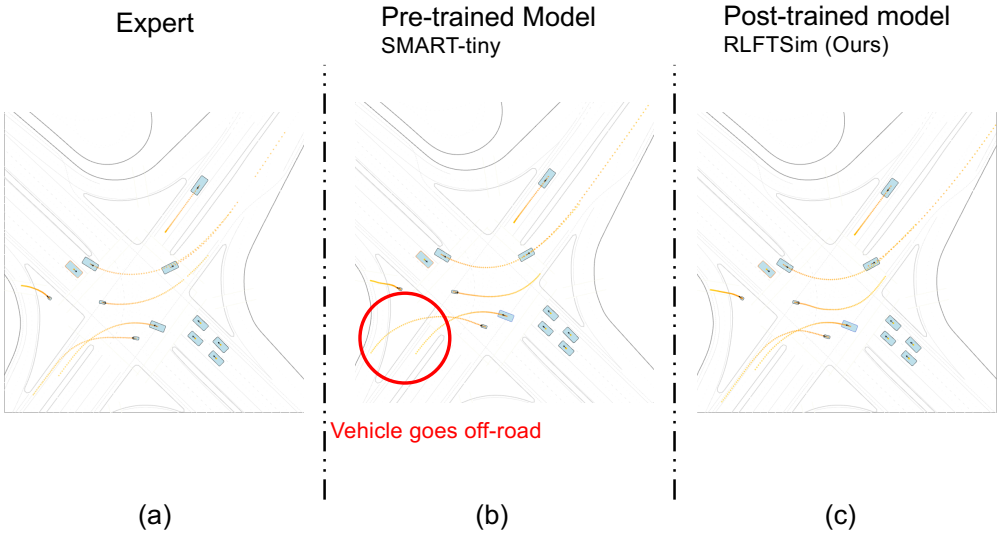


Figure S5. **Qualitative Sample - Off-road 1.** (b) The cyclist does not respect the drivable area and goes off-road. For the expert data (a) and the RLFTSim model, the cyclist adheres to the drivable area.

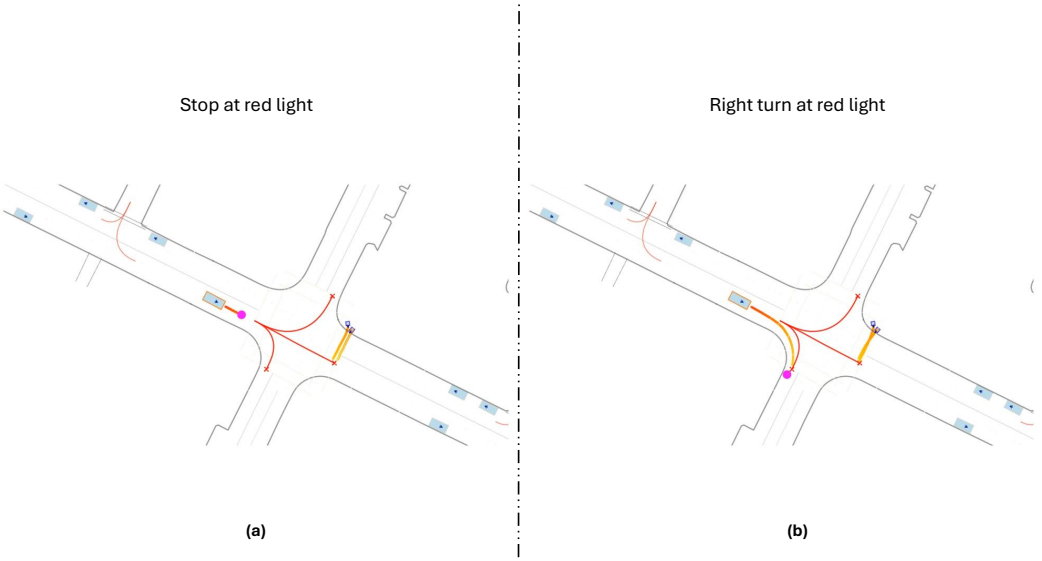


Figure S6. **Qualitative Sample - GCFT Red Light** Fine-tuning with GCFT enables greater control over scenario diversity. In this example, the base model only produces rollouts where the ego vehicle stops at the red light. In contrast, GCFT allows for the generation of rollouts where the ego performs either a right turn at the red light (b) or a full stop (a).

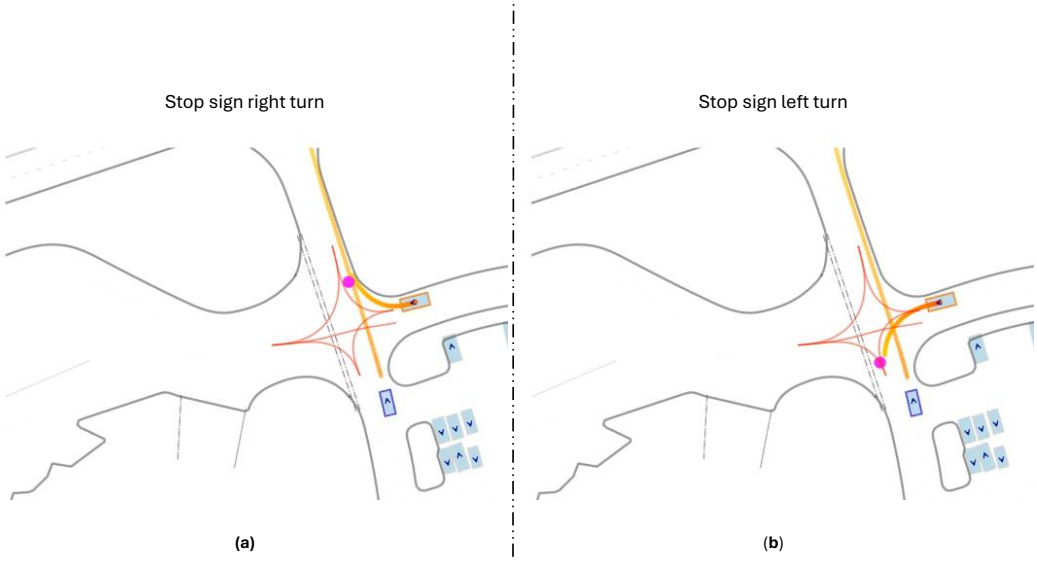


Figure S7. **Qualitative Sample - GCFT Stop Sign** Fine-tuning with GCFT enables greater control over scenario diversity. In this example, the base model only produces rollouts where the ego vehicle turns right after stopping. In contrast, GCFT allows for the generation of rollouts where the ego vehicle can turn either right (a) or left (b) at the stop sign.

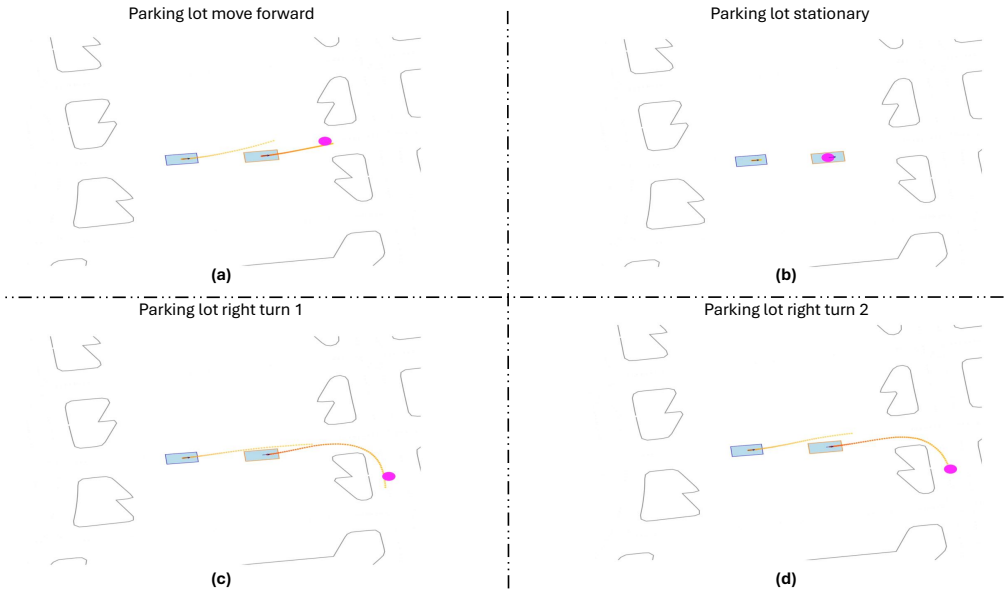


Figure S8. **Qualitative Sample - GCFT Parking Lot** Fine-tuning with GCFT allows for behavior creation from otherwise static objects. In the base SMART model, the ego vehicle always remains stationary in this scenario. After GCFT we can specify for the agent to move forward (a), remain stationary (b), or perform a right turn (c & d).