REPT: RE-TARGETED POST-TRAINING FOR MULTI-AGENT TRAFFIC SIMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Traffic agents for closed-loop evaluation are trained by imitating logged driving, and reinforcement learning post-training is meant to repair the interactions imitation gets wrong: the merges and unprotected turns where a driver could yield or push ahead, but the log shows only what one driver did. We show that current post-training does not fix these scenes, and trace the reason to where its reward is placed. Existing methods reward each rollout for resembling recorded driving and apply that reward across the whole training set. Most of the gradient therefore goes to scenes the model already drives well. On the difficult scenes every rollout fails, so rewards are nearly equal, the update is weak, and the little signal left simply reinforces the least bad failure. RePT (Re-targeted Post-Training) makes two changes. It post-trains only on the scenes the supervised model scores lowest, chosen once before training. And because these scenes collapse onto one failing behavior before any alternative can succeed, it adds a diversity bonus: rollouts are clustered by their joint trajectories, since what must stay distinct is whether an agent yields or pushes ahead, each is paid for standing outside the others' clusters, and none is paid once its realism falls below the group. Alternatives thus stay in play until the realism reward can tell which one clears the interaction. On WOSAC 2025, RePT sets the highest realism reported and is the only post-training we test, ours or published, that improves the hardest scenes at all, with no change to architecture or inference cost.

1 INTRODUCTION

Whether a self-driving car gets into a dense lane of traffic is not up to the car alone; it depends on whether the driver behind the gap lets it in. Testing such a car therefore needs other drivers who respond to it; this is closed-loop simulation, and its drivers are themselves learned models that drive one short motion token at a time, trained to copy recorded human driving. Most of driving is routine, and copying handles it well. The trouble is the moments the test is for, where the recorded drivers themselves did not all do the same thing: in one merge the driver yields, in the next an equally reasonable driver holds the gap. A model trained to match both on average slows down, speeds up, and commits to neither, and the car under test ends up facing a driver that does not exist on the road. Reinforcement learning post-training was introduced to repair these scenes, and it does raise the benchmark score. But when we ask where the improvement comes from, it comes from the scenes that were already fine; the difficult scenes, the reason post-training exists, are left where imitation left them.

Why does the improvement land there? Existing methods draw scenes from the full training set, let the model drive each scene several times, and score every such rollout for realism, that is, for how closely it resembles recorded driving. The bar a rollout has to clear is set by the model's own driving: in some methods it is the score the base model earned on that scene, in others it is the average score of the other rollouts the model produced for the same scene. Consider what this bar does on the two kinds of scene. On a scene the model already drives well, its rollouts are all plausible and differ in small ways, and those small differences decide which ones clear the bar, so every such scene delivers a clear and consistent push. On a scene the model drives badly, the rollouts start out varied, since the scene admits several resolutions and the supervised model has committed to none. Neither way of scoring them picks out the one worth keeping. Measured against a fixed threshold they all fall short and are pushed down together. Measured against each other, the rollouts score almost the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

same because all of them fail, so the update is small, and it favors whichever failed least. That rollout gets reinforced; the others get nothing for having tried something different. After a few updates the policy produces only the reinforced behavior, and the alternatives are no longer sampled, so if one of them was a viable behavior, the policy will never find out.

Anyone who has followed reinforcement learning for language models will find this familiar. There, RL with verifiable rewards has been shown to sharpen the base model's distribution rather than extend it (Yue et al., 2025), and group relative policy optimization (GRPO) (Shao et al., 2024) is known to go silent on prompts where every rollout agrees (Yu et al., 2025). The field has answered in two ways. It skips the prompts that carry no signal and chooses the rest by difficulty (Foster et al., 2025), and it protects or rewards diversity, through entropy bonuses (Cui et al., 2025) or by paying a rollout for being unlike its siblings (Li et al., 2025). Neither remedy can be used here as it is. Skipping the silent groups costs a language model little, since a wrong answer is caught by the verifier and a hard problem can wait. Here the groups that go silent are the difficult scenes themselves, so skipping them means giving up on them. And traffic simulation has no verifier: realism is a continuous score, not a right or wrong, so a rollout that stands out by leaving the road still counts as different and would be paid for it. Tokens are also the wrong place to measure difference. Neighboring tokens are nearly the same movement, and two different token strings can be the same maneuver, such as yielding a moment earlier or later. What must be kept apart is whether the agents yielded or pushed through, and that shows only in their joint trajectories. And a simulator cannot afford to sharpen at all. It is scored on the spread of its rollouts, not on any single one, and a planner tested against it has to meet every driver the road contains, not the average one. Finally, because we ask where the gain comes from, the scenes that receive the reward must be fixed before training; re-choosing them as the policy improves would fold a curriculum into the answer. What is needed is a signal kept alive inside the difficult scenes rather than around them: rollouts pulled apart in trajectory space, and paid for only while each remains a plausible way to drive.

RePT (Re-targeted Post-Training) follows from this. It post-trains only on the scenes the supervised model scores lowest, chosen once and never re-selected. Within each scene it draws K rollouts, clusters them in trajectory space, and rewards each by how few of its siblings share its cluster, withholding the diversity bonus from any rollout whose realism falls a margin below the group so that diversity is paid for only among plausible ways to drive. Architecture and inference cost are unchanged. We show three things. (i) Full-set post-training of a state-of-the-art tokenized traffic model concentrates its gradient on the scenes the supervised model already handles and leaves the hardest quartile unimproved, under threshold-based, group-relative objectives, and adversarial objectives alike. (ii) The failure on difficult scenes is a diversity failure: without the diversity bonus the rollouts narrow, visible as a drop in the kinematic score, and aiming the reward at them does not by itself recover them. (iii) A realism-gated diversity bonus in trajectory space preserves the signal exactly there; with targeting, it lifts the hardest quartile above the supervised model and reaches the best realism reported on the Waymo Open Sim Agents Challenge (WOSAC) test split (Montali et al., 2023).

2 Related Work

Tokenized traffic simulation. Casting multi-agent motion as next-token prediction was introduced by MotionLM (Seff et al., 2023) and Trajeglish (Philion et al., 2024), which discretize trajectories into a vocabulary of short motion segments and generate the joint future of all agents autoregressively. SMART (Wu et al., 2024) scaled the formulation and won the 2024 WOSAC (Montali et al., 2023) on the Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021), which has since been the standard evaluation for this line and made SMART its common backbone. TrajTok (Zhang et al., 2026) revisits the tokenizer, building a vocabulary in which tokens lie on a regular grid of displacements, and holds the highest WOSAC score of any SMART-based model trained without reinforcement learning. We post-train it with its tokenizer and architecture unchanged, starting from its pre-trained checkpoint (Zhang et al., 2026). Every model in this line is trained by imitation of logged driving, which is the starting point our post-training assumes.

Reinforcement learning for traffic agents. The limits of pure imitation on difficult driving was shown by Lu et al. (2023), who add a reinforcement term to a behavior-cloning objective. Later work approaches post-training from several directions: closed-loop rule-compliance rewards (Zhang

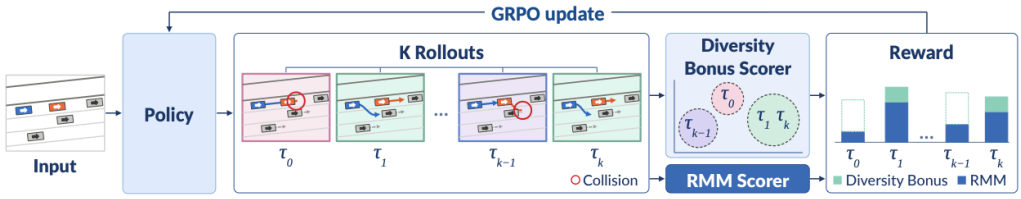

Figure 1: The RePT post-training step. The policy decodes K rollouts of one scene. The RMM scorer gives each its realism q_k; the diversity bonus scorer clusters them in trajectory space and gives $E_k = \sqrt{K/n_k}$, where n_k is the size of the rollout's cluster. The reward (right) stacks realism (blue) and bonus (green). Here the two rollouts that collided (red and purple) stand alone, but their realism falls more than m below the group mean, so their bonus is gated to zero (empty dashed bars): standing apart by colliding earns nothing. GRPO turns the rewards into group-relative advantages.

et al., 2023), human preferences (Cao et al., 2024), fine-tuning for agent behavior (Peng et al., 2024), dense rewards derived from the evaluation metric (Ahmadi et al., 2026), and adversarial imitation with a decomposed discriminator (Guo et al., 2026). SMART-R1 (Pei et al., 2026) likewise takes the benchmark metric as its reward, but optimizes it by subtracting a fixed threshold instead of comparing rollouts within a group. These methods differ in what the reward is and agree on where it is placed: scenes are drawn from the full training set, and which scenes the reward ends up acting on is never examined. We treat that region as a design variable, fixed before training, and ask what reward maximization does to the spread of the policy's rollouts once the reward is placed there.

Difficulty and the long tail. Stratifying driving data by difficulty is established practice. Makansi et al. (2021) score difficulty by the error of a constant-velocity predictor, Chen et al. (2026) by a pre-trained simulator's own likelihood, and Bronstein et al. (2023) use difficulty to select training data for a planner. A parallel line constructs difficult scenes instead of mining them (Wang et al., 2021; Rempe et al., 2022). We use a difficulty partition differently: not as a curriculum that re-ranks scenes during training, but as the fixed domain of the post-training objective, so that its gains can be attributed to where the reward was placed.

Selection and diversity in post-training. Language-model post-training has two lines of remedy for groups whose rollouts all agree. One selects prompts: DAPO (Yu et al., 2025) drops groups that yield no gradient, and LILO (Foster et al., 2025) prioritizes questions with the highest success variance. The other rewards diversity across the group, by clustering responses with a learned semantic classifier and paying the rare ones (Li et al., 2025), reweighting toward unlikely correct answers (He et al., 2025), or building preference pairs from the most diverse high-quality responses (Lanchantin et al., 2025). Outside language models, quality-diversity optimization (Pugh et al., 2016) has long kept a population of distinct high-performing solutions. Both lines share a premise: a case the model always fails is set aside for now, and spread is a means to one better answer. Neither premise holds for us. The scenes we target are exactly the ones the model always fails, and for a simulator the spread is the output itself, since the benchmark scores the distribution of its rollouts rather than any single one. We therefore keep those scenes and reward spread directly, measuring it between joint trajectories rather than on tokens, and gating the diversity bonus by realism because there is no verifier.

3 Method

Figure 1 shows the post-training step. The policy π_θ is initialized from the pre-trained TrajTok, with tokenizer and architecture left as they are; post-training changes its weights. Each step takes a batch of scenes and, for each scene s with N_s agents, samples K rollouts $\tau_0, \ldots, \tau_K$ of T steps from the current policy. Every rollout is scored with the WOSAC realism meta metric (RMM, Appendix A.1) evaluated on that rollout alone; $q_k = M(\tau_k)$ is the realism term of its reward. Everything that follows is computed over the group of K rollouts rather than over any single one: the group gives

a relative advantage without a value network, and because its members are alternative futures of the same scene, it also shows how varied the policy's driving of that scene is. Relative to running this objective on the full training set, RePT changes two things: which scenes enter training, and what a rollout is paid for.

3.1 TARGETING: A FIXED POOL OF DIFFICULT SCENES

Post-training is restricted to the scenes the supervised model handles worst. Before any policy update we score every training scene with TrajTok over 32 rollouts, as in evaluation, and take the lowest quartile by RMM as the pool. Difficulty is measured by the reward itself: a difficult scene is one where the base policy earns little realism. The pool is fixed before training and never re-selected.

3.2 CLUSTERING AND THE DIVERSITY BONUS

Spread within a group is measured in trajectory space rather than on tokens (Figure 1), since token strings do not track maneuvers. Stacking the planar positions of all agents over all steps of rollout k into a vector v_k, one (x, y) pair per agent per 10Hz step, the distance between two rollouts is normalized so that it reads as a per-agent, per-step displacement,

$$d(j,k) \;=\; \frac{\|v_j - v_k\|_2}{\sqrt{N_s T}}. \tag{1}$$

A single greedy pass in sampling order assigns each rollout to the first cluster whose representative lies within τ_{div} of it, or opens a new cluster with itself as representative; n_k is the size of the cluster containing rollout k, counting every rollout regardless of its realism. A rollout that stands alone is rewarded more than one that duplicates its siblings:

$$E_k \;=\; \begin{cases} \sqrt{K/n_k}, & q_k \;\geq\; \bar{q} - m, \\ 0, & \text{otherwise}, \end{cases} \qquad \bar{q} \;=\; \frac{1}{K}\sum_j q_j, \tag{2}$$

so E_k is largest, $\sqrt{K}$, for a rollout that is alone in its cluster ($n_k = 1$), and smallest, 1, when all K rollouts fall into a single cluster ($n_k = K$). The condition on q_k is a realism gate: a rollout whose realism falls more than a margin m below the group mean earns no bonus, which stops the diversity bonus from paying for degenerate behavior that merely happens to be far from everything else. The diversity bonus enters the reward additively,

$$R_k \;=\; q_k + \beta_{\mathrm{div}} E_k. \tag{3}$$

Since the advantage below is standardized within the group, what matters for each term is how much it varies across the K rollouts, not its absolute size. At the settings we use, the bonus varies within a group on the same order as the realism term, so both contribute to the advantage, and the gate ensures that the bonus is never paid to a rollout whose realism falls more than m below the group mean. If all rollouts of a group fall into one cluster and pass the gate, they receive the same bonus, which leaves every advantage unchanged; on an easy scene, where the rollouts drive alike, the bonus therefore does nothing. If one rollout drives differently, it is alone in its cluster and receives a larger bonus than the others. Its advantage is then positive even when its realism is no higher than theirs, so the update makes the policy more likely to produce that rollout again, and it keeps being sampled in later steps.

3.3 GROUP-RELATIVE OBJECTIVE AND UPDATE

Advantages are computed within the group, with a floor on the denominator:

$$A_k \;=\; \frac{R_k - \frac{1}{K}\sum_j R_j}{\max(\mathrm{std}_j(R_j),\; \sigma_{\min})}. \tag{4}$$

The floor matters on the scenes we target. Once a group has narrowed, its rewards differ only by noise, and without the floor that difference would be scaled up to the advantage a genuinely varied group produces. The floor leaves the numerator alone, so such a group updates the policy in proportion to how little its rollouts actually differ. The advantage of a rollout is shared by all of its

Table 1: WOSAC 2025 test-split leaderboard, one submission per method. Best values in bold.

Method	RMM ↑	Kinematic ↑	Interactive ↑	Map-based ↑	minADE ↓
SMART (Wu et al., 2024)	0.7814	0.4854	0.8089	0.9153	1.393
UniMM (Lin et al., 2026)	0.7829	0.4914	0.8089	0.9161	1.295
CAT-K (Zhang et al., 2025)	0.7846	0.4931	0.8106	0.9177	1.307
TrajTok (Zhang et al., 2026)	0.7852	0.4887	0.8116	**0.9207**	1.318
RLFTSim (Ahmadi et al., 2026)	0.7857	0.4927	0.8129	0.9183	1.325
SMART-R1 (Pei et al., 2026)	0.7858	0.4944	0.8110	0.9201	**1.289**
DecompGAIL (Guo et al., 2026)	0.7864	0.4919	**0.8152**	0.9176	1.421
RePT (ours)	**0.7879**	**0.4970**	0.8140	0.9206	1.359

Table 2: Ablation on the validation split, mean ± 95% CI over 20 seeds. Best values in bold.

Model	RMM ↑	Kinematic ↑	Interactive ↑	Map-based ↑	minADE ↓
BASE	$0.7871^{\pm.0001}$	$0.4890^{\pm.0001}$	$0.8144^{\pm.0002}$	$\mathbf{0.9225}^{\pm.0004}$	$\mathbf{1.264}^{\pm.0005}$
FULL	$0.7869^{\pm.0001}$	$0.4872^{\pm.0001}$	$0.8153^{\pm.0002}$	$0.9216^{\pm.0003}$	$1.276^{\pm.0004}$
FULL+DIV	$0.7875^{\pm.0001}$	$0.4891^{\pm.0001}$	$0.8157^{\pm.0002}$	$0.9217^{\pm.0003}$	$1.294^{\pm.0005}$
EASY+DIV	$0.7858^{\pm.0001}$	$0.4873^{\pm.0001}$	$0.8149^{\pm.0002}$	$0.9189^{\pm.0003}$	$1.312^{\pm.0004}$
RANDOM+DIV	$0.7874^{\pm.0001}$	$0.4885^{\pm.0001}$	$0.8165^{\pm.0002}$	$0.9207^{\pm.0003}$	$1.318^{\pm.0003}$
HARD	$0.7873^{\pm.0002}$	$0.4883^{\pm.0001}$	$0.8161^{\pm.0001}$	$0.9210^{\pm.0004}$	$1.299^{\pm.0004}$
REPT	$\mathbf{0.7882}^{\pm.0002}$	$\mathbf{0.4897}^{\pm.0001}$	$\mathbf{0.8171}^{\pm.0002}$	$0.9217^{\pm.0004}$	$1.274^{\pm.0004}$

tokens, across every agent. Not every agent-step is simulated: WOMD marks each agent as valid or not at each step, and the invalid ones take no token. Let $m^i_{k,t}$ be 1 where agent i is valid at step t and 0 otherwise, V_s the number of agent-steps that remain, and $a^i_{k,t}$ the token agent i takes at step t. Then

$$\mathcal{L} \;=\; -\sum_{k=1}^{K} \frac{1}{V_s} \sum_{i=1}^{N_s} \sum_{t=1}^{T} m^i_{k,t}\, A_k \log \pi_\theta\big(a^i_{k,t}\big) \;+\; \beta_{\mathrm{KL}}\, \mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})\,. \tag{5}$$

In each step the policy drives every scene of the batch K times, the rollouts are scored, and one gradient step on the loss raises the probability of every token of a rollout with positive advantage and lowers it for a rollout with negative advantage, in proportion to the advantage, while the divergence term keeps the policy near the reference. The divergence term is estimated on one rollout per group, so the reference forward pass costs the same for any K.

4 EXPERIMENTS

4.1 SETUP

We evaluate on WOSAC 2025 over WOMD. Each scene gives one second of history, and the policy drives every valid agent jointly for the following eight seconds at 10 Hz, 16 motion tokens per agent. This is repeated for 32 rollouts, and the metrics compare the distribution they form against the logged trajectory (Appendix A.1).

Leaderboard standings are computed by the evaluation server on the held-out test split, whose logged trajectories we cannot see. Every ablation and analysis uses the validation split; scoring all 44,097 scenes is costly, so following prior practice (Zhang et al., 2025; Pei et al., 2026; Guo et al., 2026) we use the same 880-scene subset for every configuration.

Configurations. Crossing the training pool with the diversity bonus gives four post-trained models: FULL and FULL+DIV train on all scenes, HARD on the fixed lowest quartile without the diversity bonus, and REPT on that quartile with the bonus. EASY+DIV trains on the highest quartile with the bonus, and RANDOM+DIV on a random quarter of each quartile with the bonus, as controls. All start from the same supervised checkpoint, BASE.

Difficulty bins. Where results are split by difficulty, the 880 scenes are ranked by the RMM of BASE and cut into four bins of 220, Q1 being the hardest. This is the score that defines the training pool, so difficulty means the same thing in training and evaluation.

Uncertainty. Each configuration is trained with 20 seeds. Tables report the mean over seeds, and intervals, where shown, are 95% confidence intervals across seeds.

minADE. Minimum Average Displacement Error is the error of whichever of the 32 rollouts lands closest to the logged trajectory. At a fixed sample budget it depends on how tightly a policy concentrates its samples, so it measures a different axis than realism, and we use it to read the sample distribution of a policy.

4.2 IMPLEMENTATION DETAILS

Objective settings. Each scene is fanned into $K = 4$ rollouts. The diversity bonus is weighted by $\beta_{\text{div}} = 0.2$, the clustering threshold is $\tau_{\text{div}} = 3.0$, and the realism gate uses a margin of $m = 0.03$. When advantages are standardized within a group, the denominator is floored at $\sigma_{\min} = 0.02$. The divergence term against the reference policy is weighted by $\beta_{\text{KL}} = 0.1$; the reference is a frozen copy of the policy at the start of post-training and is never updated. The remaining details are given in Appendix A.2.

4.3 QUANTITATIVE RESULTS

Table 1 reports results on the WOSAC 2025 test split, where REPT reaches the best RMM among existing methods. Against TrajTok, the supervised model we post-train, the interactive score rises and the map-based score is unchanged, so the improvement in interaction is not traded against map compliance. The kinematic score rises too, while in the controlled comparison of Table 2 it is nearly flat.

Compared with TrajTok, the model we start from, REPT gains 0.0027 in RMM. Weighting each component as the metric does, 0.0017 of that comes from kinematics, 0.0011 from interaction, and none from map compliance. The interaction gain is the one our method is designed to produce, and it is about the same on the validation subset, 0.0012 (Table 2). How it is obtained also differs from the other post-trained entries. DecompGAIL and RLFTSim raise their interactive scores but lose 0.002 to 0.003 in map compliance relative to TrajTok; REPT raises interaction and keeps map compliance unchanged.

The kinematic gain is not the same on the two splits: 0.0083 on test, 0.0007 on the validation subset. We think the two splits contain different kinds of scenes. Going from the subset to the test split, BASE scores lower on the interactive and map-based groups but the same on kinematics, so the test split appears to hold more scenes with contested interactions. Those are the scenes where the diversity bonus spreads the rollouts out, and the kinematic score rewards that spread: it checks whether the speeds and accelerations of the 32 rollouts include the logged one, which happens more often when the rollouts are spread. This is our interpretation, not a measured result; the interaction gain does not depend on it.

minADE is higher than TrajTok's, that is, worse. With 32 rollouts it falls as a policy concentrates its samples, so a term that spreads them is expected to raise it; Table 2 shows that the diversity bonus itself lowers minADE on the hardest quartile, from HARD to REPT.

4.4 TARGETING AND THE DIVERSITY BONUS

Table 2 crosses the training pool with the presence of the diversity bonus, and two rows tell most of the story. FULL leaves RMM where BASE had it. REPT raises it most, and by more than the gains of HARD and FULL+DIV over BASE combined; neither change is sufficient on its own. The two controls isolate where the bonus is placed. RANDOM+DIV matches REPT in pool size and bonus and differs only in which scenes it holds; it stays close to FULL+DIV and below REPT, so a smaller pool does not by itself produce the gain. EASY+DIV has the lowest RMM of all and is also worse than BASE on map compliance and minADE. Among the three pools of the same size,

RMM follows the difficulty of the pool, and the same objective gives the best row of the table on the hardest quartile and the worst on the easiest.

The three groups show what each change does. Without the diversity bonus, FULL and HARD fall below BASE on the kinematic score: reward maximization narrows the rollouts of a group, and the kinematic likelihoods, which rise when the rollouts cover the logged value, fall as the rollouts narrow. With the bonus, FULL+DIV and REPT recover it. EASY+DIV does not, since on scenes whose rollouts already agree the bonus seldom has anything to separate, and RANDOM+DIV recovers only part of it. The interactive score rises in every configuration, more with targeting, and more still once the diversity bonus is added. The map-based score falls a little in every configuration; on the full pool and the hardest quartile the bonus does not add to the fall, and EASY+DIV falls the most.

minADE is where the effect of the diversity bonus depends on the pool: REPT is below HARD, while FULL+DIV is above FULL. The two pools hold different scenes. On a difficult scene the supervised model settles on the average of the resolutions the scene admits; when the diversity bonus pulls the rollouts apart, one branch takes the resolution the logged trajectory took, and minADE falls. On a scene the supervised model already handles, its rollouts are all plausible and differ only in small ways, so a rollout can stand apart only by moving away from the logged trajectory, and as long as it remains plausible it passes the realism gate and is paid. The hardest quartile excludes scenes of this second kind, and there the diversity bonus lowers minADE. The full pool includes them, and there minADE rises even as RMM gains from the spread the kinematic likelihoods reward; RANDOM+DIV, whose pool holds them in the same share, is at least as high as FULL+DIV. EASY+DIV trains on them alone and has the worst map-based score in the table. What the diversity bonus does depends on where it is placed, and targeting decides where.

4.5 WHERE THE GAIN FALLS WITH DIFFICULTY

The learning signal is already uneven across the difficulty bins before any update is taken. At the supervised checkpoint we backpropagate each sampled scene alone, record the norm of its gradient, and compute each quartile's share of the summed norm; with equally many scenes per quartile, a signal independent of difficulty would give each a quarter. Figure 2 shows the easiest quartile taking more than its share and the hardest less. The update leans toward the scenes the policy already handles before the objective has been optimized at all.

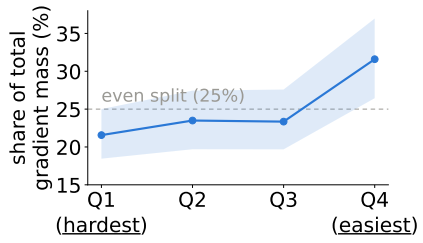

Figure 2: Share of the total gradient mass by difficulty quartile, measured at the supervised checkpoint. The quartiles hold equally many scenes, so 25% is the even split; the band is a bootstrap 95% interval.

The signal that does reach the difficult scenes narrows them, and that is the problem. Reward maximization sharpens the policy on every scene: the rollout that scores highest is reinforced and the others stop being drawn (Yue et al., 2025). On an easy scene the rollouts already sit in one cluster near the logged trajectory, so sharpening keeps what is already right. On a difficult scene the rollouts spread over several clusters, one for each way the model has half-learned to resolve it, and none of them scores well yet. Whichever scores a little higher is kept, the rest are dropped, and the right resolution may be among the dropped. Table 2 shows the trace: the kinematic score, which falls when the rollouts stop covering the logged value, is below BASE in FULL and HARD. The bonus pays a rollout for being in a cluster of its own, so the alternatives stay in play long enough for the reward to find the one that clears the interaction; the clusters survive and the kinematic score returns in FULL+DIV and REPT. On the full pool the bonus leaves Q4 untouched, since a group in one cluster pays every rollout the same bonus; FULL and FULL+DIV tie there in Table 3.

What this leaves for the difficult scenes is in Table 3. FULL and FULL+DIV leave Q1 below BASE, and so do SMART-R1 and DecompGAIL on the same base: four objectives — group-relative, with a diversity term, threshold-based, adversarial — and none of them lifts it. Aiming the realism reward at those scenes does not lift it either; HARD falls below as well. Only REPT rises above, and Q1 is where REPT and HARD differ most. Both exceed BASE on the middle quartiles, most on the second, and on Q4 the cost of restricting the pool shows and the bonus reduces it. The three pools of

Table 3: RMM by difficulty quartile on the validation split. The 880 scenes are split into four bins of 220 by the RMM of BASE. Rows above the line are the configurations of Table 2; rows below it are published post-training methods applied to BASE. Mean ± 95% CI over 20 seeds; best in bold.

Model	Q1 (hardest)	Q2	Q3	Q4 (easiest)
BASE	$0.7162^{\pm.0004}$	$0.7786^{\pm.0002}$	$0.8054^{\pm.0001}$	$0.8483^{\pm.0001}$
FULL	$0.7143^{\pm.0004}$	$0.7789^{\pm.0001}$	$0.8057^{\pm.0002}$	$\mathbf{0.8488}^{\pm.0001}$
FULL+DIV	$0.7136^{\pm.0004}$	$0.7801^{\pm.0002}$	$\mathbf{0.8074}^{\pm.0001}$	$\mathbf{0.8488}^{\pm.0001}$
EASY+DIV	$0.7077^{\pm.0004}$	$0.7796^{\pm.0002}$	$\mathbf{0.8074}^{\pm.0001}$	$0.8485^{\pm.0001}$
RANDOM+DIV	$0.7142^{\pm.0003}$	$0.7801^{\pm.0003}$	$0.8068^{\pm.0001}$	$0.8485^{\pm.0001}$
HARD	$0.7149^{\pm.0005}$	$0.7809^{\pm.0002}$	$0.8059^{\pm.0002}$	$0.8474^{\pm.0001}$
REPT	$\mathbf{0.7179}^{\pm.0005}$	$\mathbf{0.7810}^{\pm.0002}$	$0.8059^{\pm.0001}$	$0.8481^{\pm.0001}$
SMART-R1 (Pei et al., 2026)	$0.7148^{\pm.0005}$	$0.7784^{\pm.0001}$	$0.8069^{\pm.0001}$	$\mathbf{0.8488}^{\pm.0001}$
DecompGAIL (Guo et al., 2026)	$0.7150^{\pm.0005}$	$0.7794^{\pm.0002}$	$0.8067^{\pm.0001}$	$0.8487^{\pm.0001}$

the same size separate the effect of placement from that of pool size. With the same bonus, REPT lifts Q1, RANDOM+DIV leaves it below BASE as FULL+DIV does, and EASY+DIV leaves it lowest of all. Training on fewer scenes does not help Q1 unless those scenes are chosen by difficulty, and spread learned on scenes whose rollouts already agree does not carry the structure the difficult scenes need. No definition of the reward lifts the hardest quartile unless it is placed there.

4.6 QUALITATIVE RESULTS

Figure 3 shows two scenes from the hardest quartile, a dense merge and an unprotected left turn. Each admits two resolutions, yield or go first, wait or proceed, and the supervised model commits to neither: trained by maximum likelihood, it settles on the average of the two, which in a merge is worse than either. In (a) it neither yields to the mainline vehicle nor pulls ahead of it and collides; in (b) it enters the intersection without waiting and collides with the oncoming vehicle. Collision ties for the largest weight in RMM, so a scene where the model collides in most of its rollouts lands in the lowest quartile; that is what makes these scenes hard. The BASE row shows one such rollout: it slows down, speeds up, and collides.

The FULL, FULL+DIV and HARD rows all still collide. FULL and FULL+DIV resolve neither scene; the reward placed over the full set went mostly to scenes the model already handled, and what the bonus taught there does not carry over. HARD aimed the reward at scenes of exactly this kind and still collides: its group has narrowed onto one failing behaviour, as the previous section described, and rollouts on one path score alike, so there is nothing left to rank. Table 3 shows the same at scale, where each of these three leaves Q1 below BASE.

The bonus changes what a rollout is paid for. One that brakes or accelerates more decisively than its siblings leaves their cluster and is paid for standing apart, before it has cleared the interaction, as long as its realism stays within the margin of the group. It keeps being drawn, and once it does clear the interaction the collision term rewards it and the policy commits to that resolution.

The REPT row shows the result. In (a) the rollout accelerates and merges ahead of the mainline vehicle; in (b) it waits at the stop line for the oncoming vehicle to pass and then turns. Neither slows down and speeds up again on the way. The alternatives that survive are drivable because of the gate: a rollout that stood apart by leaving the lane would have lost its realism and with it the bonus. Appendix A.5 compares the same scenes under post-training methods.

5 CONCLUSION

This paper treats *where the reward acts* as a design variable in reinforcement learning post-training for traffic simulation. A reward placed over the full training set is easiest to earn on the scenes the policy already handles, so post-training refines what imitation learned and leaves the difficult interactions as they were. RePT fixes the region where the supervised model is weakest before training, post-trains only there, and pairs this targeting with a realism-gated diversity bonus.

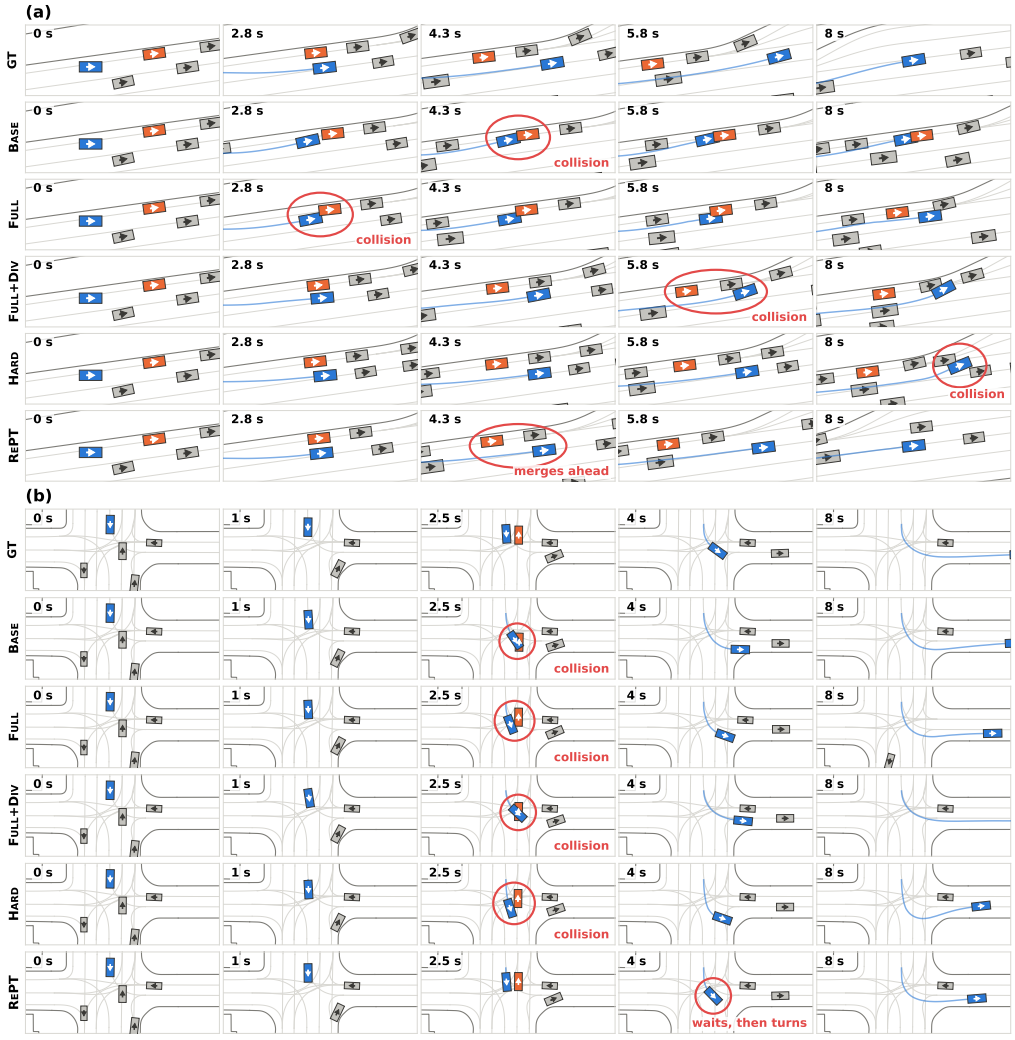

Figure 3: Two scenes from the hardest quartile: (a) a dense merge, (b) an unprotected left turn. Rows: logged trajectory, then BASE, FULL, FULL+DIV, HARD and REPT; columns advance in time. Each row shows one rollout of the configuration. Blue is the agent of interest, orange the vehicle it interacts with. Red circles mark the collision in each row that collides and the resolution in the REPT row.

RePT reaches the best RMM on WOSAC 2025, with the interactive score up and map compliance unchanged. On the validation split, every full-pool configuration, two published methods included, leaves the hardest quartile below the supervised model; so do targeting without the bonus and the same objective on a random pool of the same size, and so, worst of all, does the same objective placed on the easiest quartile. Only targeting the hardest quartile with the bonus lifts it. No definition of the reward lifts that quartile unless it is placed there.

The qualitative results show why. On a difficult scene the realism reward alone narrows the group onto one behaviour that need not clear the interaction, while the bonus keeps the alternatives sampled until the reward can tell. A simulator's output is a distribution, and a planner under test should face the range of behaviours it would meet on the road, not one averaged behaviour.

Limitations and future work. Restricting post-training to the lowest quartile is a design choice, and where the boundary should fall is open. The bonus measures diversity by one distance in trajectory space, which may not capture every kind of multimodality, and the reward is a proxy for realism, so implausibility it does not measure is not addressed.

AI USE STATEMENT

In this work, we used generative AI tools (Claude, Anthropic) for the following tasks with required disclosure: (i) implementing methods—the algorithms and model architecture were designed by the authors, and the AI assistant was used to help write portions of the implementation code based on these designs; (ii) providing feedback on research methodology and experimental design, such as suggesting ablations and baselines; (iii) assisting in the interpretation of experimental results; and (iv) translating parts of the manuscript into English. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, write proofs or provide ingredients for proofs, propose or refine research hypotheses, clean or reformat datasets, or perform qualitative or thematic data analysis. Additionally, we used generative AI tools for tasks with recommended disclosure, namely drafting and editing parts of the manuscript to improve readability, assisting in creating figures, and searching for and summarizing related literature and brainstorming research ideas.

We have reviewed all AI-assisted work. AI-assisted code was validated with unit tests and sanity checks on its outputs, and the full experimental pipeline was verified by reproducing the reported results of baseline methods before running our own experiments. All AI-edited and AI-translated text was reviewed by all authors to ensure that it preserves the intended meaning and technical accuracy. References and claims suggested by the AI assistant were checked against the original sources, and no reference was included without verification by the authors. Feedback on methodology and result interpretation was treated as input for discussion only; all experimental design choices, analyses, and conclusions were made by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only the publicly available Waymo Open Motion Dataset, accessed under its non-commercial research license; no new data were collected, and the study involves no human subjects. The agents we train are intended solely for the closed-loop evaluation of autonomous-driving systems in simulation. Because RePT uses the WOSAC realism metric as its reward, gains on that metric should not be read as a guarantee of realism or safety beyond what the metric measures. We are not aware of other ethical concerns raised by this work.

REPRODUCIBILITY STATEMENT

The post-training objective is fully specified in section 3: the trajectory-space distance (Equation 1), the clustering rule and the realism-gated diversity bonus (Equation 2), the reward (Equation 3), the group-relative advantage with its floor (Equation 4), and the loss with the divergence term (Equation 5). The construction of the fixed pool of difficult scenes—scoring every training scene with the supervised model over 32 rollouts and keeping the lowest quartile by RMM—is described in subsection 3.1 and Appendix A.2, together with the two control pools. All hyperparameters of the objective (K, β_{div}, τ_{div}, m, $\sigma_{\min}$, β_{KL}) are given in subsection 4.2, and the optimizer, learning rate, gradient clipping, batch composition, number of steps, and hardware are given in Appendix A.2; Appendix A.4 reports the sensitivity of the results to the diversity hyperparameters. The supervised starting point is a pre-trained TrajTok checkpoint that we reproduced from the authors' public code release, with tokenizer, architecture, and training recipe unchanged. Appendix A.1 details the dataset splits, the WOSAC evaluation protocol, and the composition and weights of the realism meta metric; validation-split metrics are computed with the official Waymo Open Dataset evaluation code, and test-split results are produced by the WOSAC evaluation server. The validation subset and the difficulty quartiles used for all ablations are defined in subsection 4.1, and per-component results are reported in Appendix A.3.

REFERENCES

Ehsan Ahmadi, Hunter Schofield, Behzad Khamidehi, Fazel Arasteh, Jinjun Shan, Lili Mou, Dongfeng Bai, and Kasra Rezaee. Rlftsim: Realistic and controllable multi-agent traffic simulation via reinforcement learning fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39734–39743, June 2026.

Eli Bronstein, Sirish Srinivasan, Supratik Paul, Aman Sinha, Matthew O'Kelly, Payam Nikdel, and Shimon Whiteson. Embedding synthetic off-policy experience for autonomous driving via zero-shot curricula. In *Conference on robot learning*, pp. 188–198. PMLR, 2023.

Yulong Cao, Boris Ivanovic, Chaowei Xiao, and Marco Pavone. Reinforcement learning with human feedback for realistic traffic simulation. In *2024 IEEE international conference on robotics and automation (ICRA)*, pp. 14428–14434. IEEE, 2024.

Yu-Hsiang Chen, Wei-Jer Chang, Yi-Ting Chen, and Masayoshi Tomizuka. Ecosim: Data efficient fine-tuning for controllable traffic simulation. In *European Conference on Computer Vision*, pp. 20–36. Springer, 2026.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9710–9719, 2021.

Thomas Foster, Anya Sims, Johannes Forkel, and Jakob Foerster. Lilo: Learning to reason at the frontier of learnability. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 48941–48974. Curran Associates, Inc., 2025. doi: 10.52202/085713-1634. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/45d5823950cab9a88e10c6e8dbff351c-Paper-Conference.pdf.

Ke Guo, Haochen Liu, Xiaojun Wu, and Chen Lv. Decompgail: Learning realistic traffic behaviors with decomposed multi-agent generative adversarial imitation learning. In *International Conference on Learning Representations*, volume 2026, pp. 83791–83808, 2026.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25559–25571, 2025.

Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*, 2025.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025.

Longzhong Lin, Xuewu Lin, Kechun Xu, Haojian Lu, Lichao Huang, Rong Xiong, and Yue Wang. Unimm: A unified mixture model framework for multi-agent simulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Rebecca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7553–7560. IEEE, 2023.

Osama Makansi, Özgün Çiçek, Yassine Marrakchi, and Thomas Brox. On exposing the challenging long tail in future prediction of traffic actors. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13127–13137. IEEE, 2021.

Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, et al. The waymo open sim agents challenge. *Advances in Neural Information Processing Systems*, 36:59151–59171, 2023.

Muleilan Pei, Shaoshuai Shi, and Shaojie Shen. Advancing multi-agent traffic simulation via r1-style reinforcement fine-tuning. In *International Conference on Learning Representations*, volume 2026, pp. 130791–130807, 2026.

Zhenghao Peng, Wenjie Luo, Yiren Lu, Tianyi Shen, Cole Gulino, Ari Seff, and Justin Fu. Improving agent behaviors with rl fine-tuning for autonomous driving. In *European Conference on Computer Vision*, pp. 165–181. Springer, 2024.

Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Z59Rb5bPPP.

Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

Davis Rempe, Jonah Philion, Leonidas J Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17284–17294. IEEE, 2022.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8545–8556. IEEE, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9904–9913. IEEE, 2021.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. Smart: Scalable multi-agent real-time motion generation via next-token prediction. *Advances in Neural Information Processing Systems*, 37: 114048–114071, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=4OsgYD7em5.

Chris Zhang, James Tu, Lunjun Zhang, Kelvin Wong, Simon Suo, and Raquel Urtasun. Learning realistic traffic agents in closed-loop. *arXiv preprint arXiv:2311.01394*, 2023.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5422–5432. IEEE, 2025.

Zhiyuan Zhang, Xiaosong Jia, Guanyu Chen, Qifeng Li, Zuxuan Wu, Yu-Gang Jiang, and Junchi Yan. Trajtok: What makes for a good trajectory tokenizer in behavior generation? In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Zvy2agYouY.

A APPENDIX

A.1 WOMD AND WOSAC 2025

Dataset. The Waymo Open Motion Dataset (Ettinger et al., 2021) contains 486,995 training, 44,097 validation and 44,920 test scenes. Every scene is recorded at 10 Hz and provides one second of observed history, eight seconds of future motion, and a high-definition map with lanes, road boundaries and traffic signals. Vehicles, pedestrians and cyclists appear together in a scene; the number of agents designated for simulation varies from scene to scene, with a median of 30 on the validation split.

Evaluation protocol. WOSAC (Montali et al., 2023) requires the policy to drive every designated agent of a scene jointly. One second of observed history is given, and the following eight seconds are generated at 10 Hz, yielding 80 steps per scene. This is repeated for 32 independent rollouts, and the metrics compare the distribution formed by those 32 rollouts against the logged trajectory. What is evaluated is therefore the distribution a policy generates, not the accuracy of any single trajectory.

The realism meta metric. RMM of WOSAC 2025 is a weighted sum of ten component likelihoods. Each component measures how well a quantity computed from the simulated trajectories agrees with the distribution observed in logged driving, and higher is more realistic. The components fall into three groups.

Kinematic (weights summing to 0.20) comprises the likelihoods of linear speed, linear acceleration, angular speed and angular acceleration, each weighted 0.05. This group asks whether the motion profile of an individual agent is consistent with human driving.

Interactive (weights summing to 0.45) comprises collision indication at 0.25, distance to the nearest object at 0.10, and time to collision at 0.10. Collision indication shares the largest weight in the metric with offroad indication, and coordination between agents is reflected in this group.

Map-based (weights summing to 0.35) comprises offroad indication at 0.25, distance to the road edge at 0.05, and traffic light violation at 0.05. This group asks whether the driving respects the road structure and the traffic controls.

The three groups sum to 1.0, and RMM is the sum of the ten component likelihoods under these weights. The per-group scores reported in our tables are the weighted means within a group, that is, normalized by the sum of the weights belonging to that group.

Minimum Average Displacement Error (minADE) is the average displacement error of whichever of the 32 rollouts drawn for a scene lands closest to the logged trajectory. It carries a weight of exactly zero in RMM and is not part of the ranking criterion. At a fixed sample budget its value depends on how tightly a policy concentrates its samples, so it measures a different axis than realism.

A.2 FURTHER IMPLEMENTATION DETAILS

Training pool. Before any policy update, the base model scores all 486,995 scenes of the training split, and the identifiers of the lowest realism quartile — 121,734 scenes, 25.0% of the split — are written to a file. The file is not regenerated during training, and every configuration that uses the lowest quartile reads the same file. The full-pool configurations use all 486,995 scenes. EASY+DIV uses the highest realism quartile, and RANDOM+DIV a random quarter of each of the four quartiles; both pools are drawn once and written to a file in the same way.

Optimization and scale. Optimization uses Adam with a constant learning rate of 1×10^{-5}. Gradients are clipped to an L_2 norm of 1.0 over all trainable parameters. Training runs on eight NVIDIA B200 GPUs and each rank processes two scenes, so an optimizer step consumes 16 scenes. Post-training runs for 30,000 steps.

A.3 PER-COMPONENT REALISM RESULTS

Table 2 reports RMM together with the three group summaries. A group summary combines components that move in different directions inside it, so it does not reveal which part of the metric

Table 4: Per-component WOSAC 2025 results on the validation split for the configurations of Table 2, mean over 20 seeds. The best value in each column is shown in bold.

Model	RMM ↑	Linear speed likeli. ↑	Linear acc. likeli. ↑	Angular speed likeli. ↑	Angular acc. likeli. ↑	Dist. to nearest obj. ↑	Collision likeli. ↑	Time to collision likeli. ↑	Dist. to road edge likeli. ↑	Offroad likeli. ↑	Traffic light likeli. ↑
BASE	0.7871	**0.3993**	0.4047	**0.5106**	0.6414	**0.3973**	0.9734	**0.8342**	**0.6875**	**0.9575**	0.9825
FULL	0.7869	0.3944	0.4061	0.5080	0.6404	0.3939	0.9774	0.8316	0.6854	0.9565	0.9835
FULL+DIV	0.7875	0.3976	0.4071	0.5097	0.6421	0.3956	0.9768	0.8329	0.6853	0.9565	**0.9838**
EASY+DIV	0.7858	0.3941	0.4066	0.5071	0.6415	0.3947	0.9758	0.8327	0.6838	0.9531	0.9835
RANDOM+DIV	0.7874	0.3964	**0.4075**	0.5088	0.6411	0.3949	0.9784	0.8334	0.6855	0.9553	0.9831
HARD	0.7873	0.3943	0.4064	0.5102	0.6424	0.3953	0.9776	0.8332	0.6857	0.9556	0.9833
REPT	**0.7882**	0.3978	0.4074	**0.5106**	**0.6431**	0.3968	**0.9789**	0.8329	0.6858	0.9565	0.9835

Table 5: Sensitivity to the diversity hyperparameters on the validation split. Each block varies one of K, τ_{div} and β_{div} and holds the other two at the adopted setting, whose value is shown in bold; the adopted row is REPT in every block. The best value in each column within a block is shown in bold.

Parameter	Value	RMM ↑	Kinematic ↑	Interactive ↑	Map-based ↑	minADE ↓
Rollouts per scene K	3	0.7877	0.4879	0.8161	**0.9225**	1.281
	4	**0.7882**	**0.4897**	**0.8171**	0.9217	**1.274**
	5	0.7875	0.4885	0.8161	0.9216	1.305
Clustering threshold τ_{div}	2.0	0.7881	**0.4901**	0.8161	**0.9224**	1.287
	3.0	**0.7882**	0.4897	**0.8171**	0.9217	**1.274**
	4.0	0.7875	0.4884	0.8165	0.9211	1.301
Bonus weight β_{div}	0.1	0.7881	**0.4900**	0.8161	0.9225	**1.263**
	0.2	**0.7882**	0.4897	**0.8171**	0.9217	1.274
	0.3	0.7881	0.4889	0.8165	**0.9226**	1.298

post-training acts on. Here the configurations of Table 2 are decomposed into all ten component likelihoods in Table 4.

The gain sits in the collision likelihood. All six post-trained configurations exceed BASE, and REPT is the highest of them. Collision likelihood shares the largest weight among the ten components with offroad likelihood, and it is the component that most directly measures how agents resolve one another's paths. The observation that the improvement appears in interaction takes this concrete form at the component level.

The kinematic components barely move. Each of the four varies little across the configurations, and REPT holds at or above BASE on linear acceleration, angular speed and angular acceleration. What movement there is follows the spread of the rollouts rather than any single agent's motion: adding the diversity bonus raises all four components on both pools, from FULL to FULL+DIV and from HARD to REPT.

The small shift in the distance-based components is the other side of avoiding collisions. Distance to the nearest object and time to collision are likelihoods: they measure how far the distribution a simulation produces overlaps the distribution of logged driving, not whether a larger value is safer. A policy that avoids collisions more reliably keeps wider gaps, and those gaps depart from the logged distribution. Both move only slightly, and once the weights are applied they return only part of what the collision term gains.

Map compliance is preserved. Offroad likelihood and traffic light likelihood stay within a narrow band for every configuration except EASY+DIV, which loses the most offroad likelihood, as its map-based score in Table 2 already shows. That the improvement in interaction is not traded against map compliance holds at the component level as well.

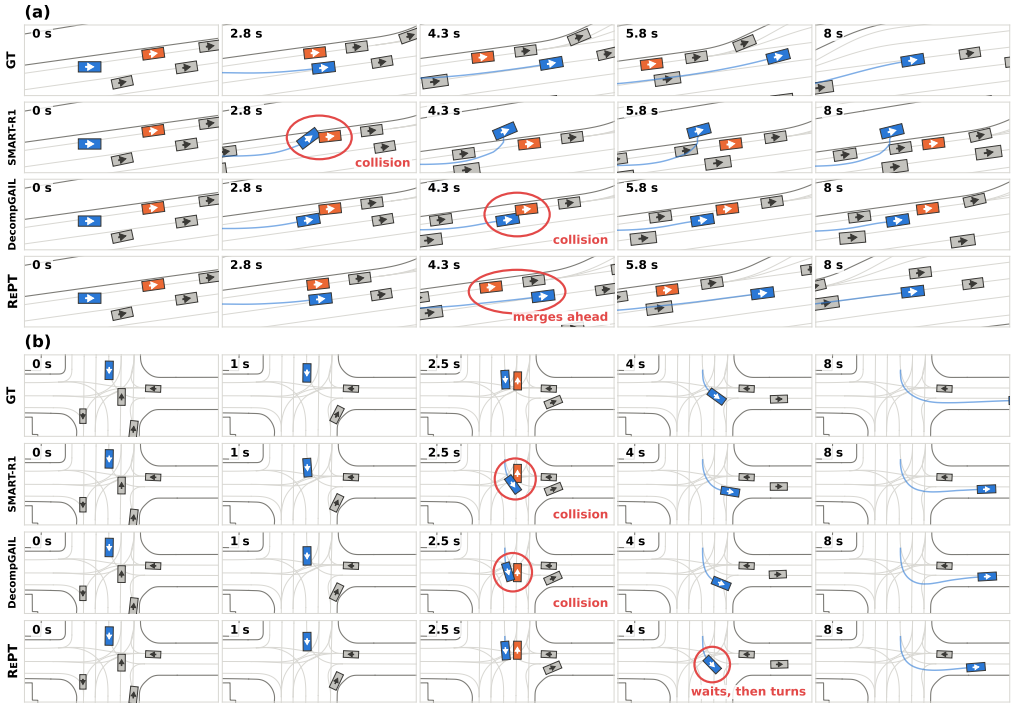

Figure 4: The two scenes of Figure 3 under post-training methods: (a) a dense merge, (b) an unprotected left turn. Rows: logged trajectory (GT), then SMART-R1, DecompGAIL and REPT, one rollout each; columns advance in time. SMART-R1 and DecompGAIL are post-trained from BASE on the same TrajTok backbone. Blue is the agent of interest, orange the vehicle it interacts with. Red circles mark the collision in each row that collides and the resolution in the REPT row.

A.4 SENSITIVITY TO THE DIVERSITY HYPERPARAMETERS

The diversity bonus depends on three values: the number of rollouts drawn per scene K, the threshold τ_{div} at which two rollouts are placed in the same cluster, and the weight β_{div} with which the diversity bonus enters the reward. The three do different work. K sets how many alternatives the comparison has to draw on, τ_{div} decides when two rollouts count as different, and β_{div} sets how heavily the resulting diversity weighs against realism. Each is varied on its own with the other two held at the adopted setting and training run to the same step, so that the three axes can be read independently. Results are reported in Table 5, and the adopted setting gives the highest RMM on all three axes.

Each axis has an interior optimum. RMM falls off when the number of rollouts is raised or lowered, and the same holds for the clustering threshold and the diversity bonus weight. The diversity bonus is therefore not a component that improves monotonically with its strength. The two ends of the threshold make this concrete: one that is too small classifies almost every rollout as distinct, so the diversity bonus becomes close to a constant, and one that is too large collects rollouts carrying genuinely different futures into a single group, so the diversity bonus disappears. In either case the structure of the group stops reaching the reward, and the purpose of the diversity bonus is defeated.

The number of rollouts is the axis tied to compute. With fewer rollouts there are fewer alternatives inside a group, so it is less clear what a relative advantage is pointing at; with more, the compute spent on one scene rises and fewer scenes are seen under the same budget. The other two axes carry no such trade, and the adopted value is the balance between these two directions.

The diversity bonus weight is the least sensitive. Halving it or raising it by half moves the metric less than either of the other two axes. What matters is that the diversity bonus is present, rather than

that its strength be tuned finely within a narrow range. Removing it altogether, which gives HARD, costs more than any point on this axis.

What the three axes move is the spread of the samples. The three group summaries barely change across the nine settings, while minADE varies visibly. The hyperparameters of the diversity bonus are therefore not pushing any single realism component directly; they control how far apart the rollouts drawn for one scene end up. The change in RMM is what follows when that spread is right.

A.5 COMPARISON WITH POST-TRAINING METHODS

Figure 4 shows the two scenes of Figure 3 under the two published methods of Table 3. SMART-R1 (Pei et al., 2026) and DecompGAIL (Guo et al., 2026) are post-trained from BASE, the TrajTok checkpoint REPT starts from, with the same backbone and their own objectives over the full training set. The rows therefore differ in the objective and in where it is placed, not in the model being post-trained.

In the merge (a), neither method resolves the interaction, and the two fail in opposite ways. SMART-R1 cuts into the mainline at 2.8 s, before it has pulled ahead of the mainline vehicle, and collides with it, at the same moment as FULL in Figure 3. DecompGAIL stays alongside the mainline vehicle, neither yielding nor pulling ahead, and collides at 4.3 s, as BASE does. At the left turn (b), both begin the turn at 2.5 s while the oncoming vehicle is still in the intersection and collide with it, the failure of BASE and of every configuration except REPT in Figure 3. REPT merges ahead in (a) and waits for the oncoming vehicle before turning in (b), as the log does.

Two scenes with one rollout per method illustrate the pattern rather than establish it; the pattern itself is in Table 3, where both methods leave Q1 below BASE and only REPT rises above it. The two objectives differ, one threshold-based and one adversarial, yet both keep the collision that places these scenes in the hardest quartile, as subsection 4.5 describes for a reward placed over the full training set.