# BRIDGING LOCAL OBSERVATION AND GLOBAL SIMULATION IN CLOSED-LOOP TRAFFIC MODELING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Autoregressive traffic simulators learn from local, ego-centric driving logs but must generate globally coherent multi-agent interactions. Limited observation range leaves the context incomplete for non-ego agents, particularly those far from the ego vehicle. Simulators may therefore learn actions without observing the conditions that induced them and reproduce those actions in contexts where they are inappropriate—a problem we term the ***local-to-global context mismatch***. We introduce ***CRAFT***, *Contextual pReference Alignment Framework for Traffic Simulation*, a lightweight, plug-and-play framework that uses closed-loop rollouts to expose and rectify such behaviors. Agnostic to the base simulator's architecture, motion prior, and training strategy, CRAFT operates post hoc without retraining or modifying the simulator. CRAFT labels grouped rollouts using traffic-rule and driving-quality criteria, then trains a separate Contextual Preference Evaluator (CPE) through contrastive preference learning over normal and failure cases within trajectories and across rollouts. At inference, the CPE uses lookahead to evaluate candidate scene evolutions and reweights action probabilities toward contextually appropriate behavior. Across three state-of-the-art simulators on the Waymo Open Motion Dataset, CRAFT reduces scene-level collision, offroad, and traffic-violation rates by up to 31.2%, 14.9%, and 33.2%, respectively, while improving motion realism. These results ***establish contextual preference alignment*** as a general and efficient mechanism for mitigating local-to-global mismatch ***across heterogeneous traffic simulators***.

## 1 INTRODUCTION

Realistic traffic simulation enables scalable training and evaluation of autonomous driving systems in complex multi-agent environments (Suo et al., 2021; Feng et al., 2023; Yan et al., 2023; Gulino et al., 2023). Recent autoregressive simulators learn multi-agent motion from large-scale driving logs through next-token prediction, achieving strong performance in modeling traffic behavior (Seff et al., 2023; Philion et al., 2023; Wu et al., 2024). By repeatedly generating and executing motion predictions, these models support realistic closed-loop simulation of multi-agent traffic.

Nevertheless, the driving logs on which these models rely may not capture all the information underlying the recorded behavior (Fig. 1a). Sensor range and occlusion restrict observations of surrounding agents (Ettinger et al., 2021), leaving some of the information underlying their actions outside the recorded context. This information gap can make it difficult to learn when a demonstrated behavior is appropriate (Swamy et al., 2022; Nguyen et al., 2026). In traffic simulation, a learned response may persist even when its underlying condition is absent. We term this gap between fitting locally observed demonstrations and generating contextually justified behavior the *local-to-global (L2G) context mismatch*. Consistent with this failure mode, the base simulator continues to decelerate after we remove surrounding vehicles that motivate slowing down (Fig. 1b–c). This mismatch is an underexamined yet consequential problem in traffic simulation. It can undermine the realism of multi-agent interactions, as inappropriate actions influence other agents' responses and subsequent scene evolution. This raises a central question: how can a simulator retain the motion patterns learned from logs while selecting behaviors appropriate for the evolving scene?

We propose **CRAFT**, a **C**ontextual p**R**eference **A**lignment **F**ramework for **T**raffic Simulation, to *align learned behaviors with the interaction context of the simulated scene*. We generate multiple



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.

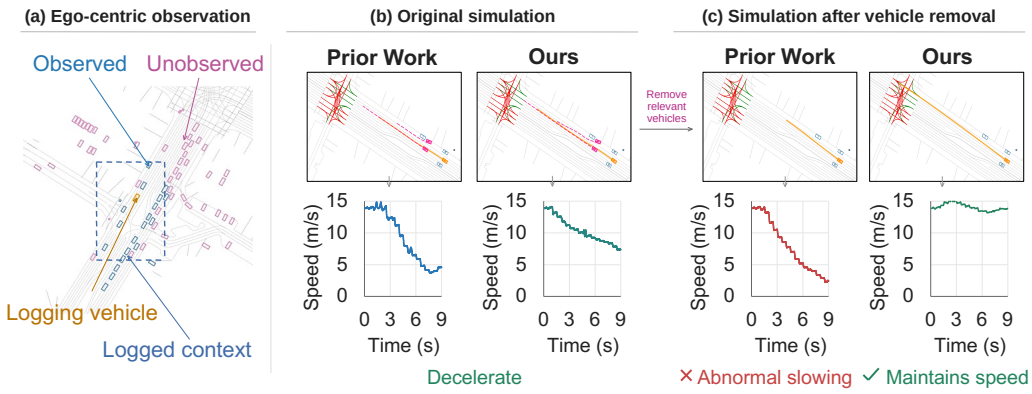



Figure 1: **Local observations and scene intervention.** (a) Ego-centric logs capture only part of the context underlying observed behavior. (b–c) The base simulator retains unnecessary deceleration after relevant surrounding vehicles are removed, whereas CRAFT maintains speed.

closed-loop rollouts from each logged initial scene, exposing alternative behaviors and interactions among the same agents. Traffic and driving criteria label these behaviors, providing supervision for a Contextual Preference Evaluator (CPE). CPE combines label prediction with contrastive preference learning: within trajectories, it compares normal and failure segments; across rollouts, it compares alternative behaviors of the same agent at matched time steps. At each simulation step, CPE evaluates the scene states induced by candidate actions through one-step lookahead. The resulting scores reweight each agent's action probabilities before sampling.

This design separates the criterion for contextual appropriateness from the simulator's learned motion distribution. Each simulator supplies its own candidate action probabilities, while CPE judges the scene those actions produce. The same evaluation criterion can therefore be combined with different motion priors at the sampling stage, without changing the simulators' training objectives or parameters. We test whether a single CPE learned from one simulator's rollouts can improve action selection across differently trained simulators. Experimental results show that CRAFT reduces abnormal behaviors while preserving motion realism. Our contributions are as follows:

**(1) An underexamined simulation mismatch.** We characterize the gap between fitting incomplete-context demonstrations and generating behavior justified by the simulated scene, explaining how it can persist even under perfect imitation.

**(2) Decoupled contextual alignment.** We learn contextual judgments from grouped closed-loop rollouts and use candidate scene evaluation to guide frozen simulators, separating behavioral correction from the simulator's training objective.

**(3) Policy-agnostic guidance.** One CPE trained on base simulator rollouts improves three state-of-the-art simulators without retraining, demonstrating that contextual guidance transfers across pretrained and post-trained motion distributions.

## 2 Related Work

**Traffic Modeling and Closed-Loop Learning.** Traffic simulators learn joint motion from driving logs using autoregressive models (Seff et al., 2023; Philion et al., 2023; Zhou et al., 2024; Wu et al., 2024), diffusion (Huang et al., 2026; Zhou et al., 2025), or hierarchical policies (Xu et al., 2023), with recent extensions to evolving agent populations (Tan et al., 2025; Yang et al., 2025). To reduce errors under self-generated histories, CAT-K uses demonstration-guided rollouts (Zhang et al., 2025), while constraint-based and reinforcement-learning refinement optimize behavioral compliance (Zhang et al., 2023; Peng et al., 2024; Wang et al., 2026; Ahmadi et al., 2026). DAgger and SimScale broaden supervision to learner-induced or perturbed states (Ross et al., 2011; Tian et al., 2026). Effective imitation also depends on the information available to explain the demonstrated action. Causal confusion can make policies exploit spurious correlations (De Haan et al., 2019); unobserved expert context can induce repetition of past actions in sequence policies (Swamy et al., 2022). LEAD addresses information asymmetry by aligning expert visibility and navigation intent

with learner inputs (Nguyen et al., 2026). Such alignment requires access to the expert's information and demonstration process, which fixed logs of surrounding drivers generally do not provide.

**Behavioral Preference Alignment.** Preference alignment introduces behavioral judgments beyond imitation. TrafficRLHF obtains these judgments from human comparisons (Cao et al., 2024), while Gen-Drive uses VLM-assisted preferences to learn scene rewards for planning (Huang et al., 2025). DPA-OMF avoids additional preference annotation by ranking generated rollouts against expert occupancy measures in safety, comfort, and progress features, then fine-tunes the simulator (Tian & Goel, 2025). Such expert-referenced rankings remain anchored to logged expert trajectories, which are not necessarily reliable behavioral references because the context underlying their actions is only partially observed. *CRAFT evaluates contextual appropriateness separately from motion likelihood, improving how well learned behaviors fit the interactions in which they are generated.*

**Controllable Generation and Online Guidance.** Guidance methods differ in their control targets and interfaces. CTG and CTG++ guide diffusion trajectories with traffic constraints and language-derived objectives (Zhong et al., 2023b;a); ProSim conditions agent policies on multimodal prompts (Tan et al., 2024), and CCDiff incorporates causal interaction structure into diffusion guidance (Lin et al., 2025). SAFE-SIM and SaFeR steer generation toward safety-critical scenarios for planner testing (Chang et al., 2024; Cui et al., 2026). For ego planning, Gen-Drive ranks candidate future scenes (Huang et al., 2025), while BridgeSim combines EPDMS-based prefix rewards with adaptive replanning (Zhao et al., 2026). These approaches emphasize scenario control or ego decisions. *CRAFT provides a plug-in correction mechanism at the action-sampling interface of multi-agent autoregressive simulators, enabling behavioral improvements across pretrained and post-trained simulators without updating their parameters.*

## 3 The problem of context mismatch in traffic simulation

**Autoregressive traffic simulation.** Let $\mathbf{C}_{t-1}$ denote the scene context at time $t-1$, including the agent states at that time and the static map, and let $\mathbf{a}_t = \{a_{i,t}\}_{i=1}^{N_a}$ denote the actions of all $N_a$ agents in the scene at time $t$. The base simulator $\pi_\theta$ learns to predict recorded actions from their logged contexts through imitation learning:

$$\mathcal{L}_{\mathrm{IL}}(\theta) = -\mathbb{E}_{(C_{t-1},a_t)\sim p_{\log}}\left[\log \pi_\theta(a_t \mid C_{t-1})\right]. \tag{1}$$

Starting from an initial scene $\mathbf{C}_0$, the simulator autoregressively generates joint actions, conditioning each prediction on the scene updated by previously generated actions:

$$P_\theta\left(\hat{\mathbf{a}}_{1:T_f} \mid \mathbf{C}_0\right) = \prod_{t=1}^{T_f} \pi_\theta\Big(\hat{\mathbf{a}}_t \mid \hat{\mathbf{C}}_{t-1}\Big), \qquad \hat{\mathbf{C}}_0 = \mathbf{C}_0. \tag{2}$$

Here, hats denote generated quantities, $T_f$ is the rollout horizon, and $P_\theta$ is the joint-action sequence distribution. At each step $t$, executing $\hat{\mathbf{a}}_t$ updates the scene to $\hat{\mathbf{C}}_t$, which then conditions action generation at step $t+1$.

**Context mismatch.** Logged actions can depend on scene conditions missing from the recorded context. Let $U \in \{0,1\}$ indicate the presence of a condition omitted from the recorded context $C$, and let $p^*$ be the joint distribution of $a$, $C$, and $U$ underlying log collection. Under realizability and exact population optimization, the base simulator fits the logged marginal for almost every $C$:

$$\pi_\theta(a \mid C) = p_{\log}(a \mid C) = \sum_{u\in\{0,1\}} p^*(a \mid C, U=u)\, p^*(U=u \mid C). \tag{3}$$

For such a $C$, if $0 < p^*(U=1 \mid C) < 1$ and $p^*(\cdot \mid C, U=0) \neq p^*(\cdot \mid C, U=1)$, then

$$\pi_\theta(\cdot \mid C) \neq p^*(\cdot \mid C, U=u), \qquad u \in \{0,1\}. \tag{4}$$

This analysis identifies a limitation of fitting incomplete-context demonstrations: agreement with the logged marginal need not imply agreement with scene-specific behavior. *We therefore seek to mitigate the adverse effects of learning from incomplete-context demonstrations by incorporating additional prior knowledge at test time.*

**Effect of training context.** To assess how missing context in driving logs affects closed-loop behavior, we vary the number of surrounding agents visible to the model during training while keeping all other training conditions fixed (Fig. 2). Under the same rollout evaluation, adding more context lowers all three event rates relative to ego-only training. Collision rates are lowest with 32 agents, while offroad and traffic-violation rates are lowest with all available agents. These results demonstrate the importance of training context for closed-loop behavioral rationality.

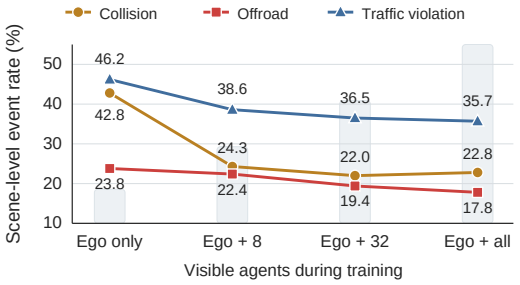



Figure 2: **Effect of training-context availability.**

# 4 METHOD

We outline CRAFT as follows. In Sec. 4.1, we construct grouped simulator rollouts and derive behavioral supervision using traffic and driving criteria. In Sec. 4.2, we train CPE to learn contextual behavioral preferences from these rollouts. In Sec. 4.3, we use the learned preferences to guide candidate action selection during autoregressive simulation.

## 4.1 GROUPED ROLLOUT SUPERVISION

Learning from learner-induced states addresses sequential distribution shift in imitation learning (Ross et al., 2011). Here, we use simulator rollouts to discover erroneous context–behavior pairings for evaluator supervision. Starting from logged scenes $\{\mathbf{C}_0^m\}_{m=1}^{\mathcal{M}}$, we generate $N$ stochastic closed-loop rollouts per scene using different random seeds (Fig. 3), exposing alternative interactions and abnormal behaviors under specific context:

$$\hat{\mathbf{C}}_{1:T_f}^{(m,j)} = \mathrm{Rollout}(\mathbf{C}_0^m; \epsilon_j), \qquad j = 1, \ldots, N. \tag{5}$$

Here, $m$ indexes the scene id and $j$ indexes the rollout with random seed. We apply an automatic labeler $\mathcal{A}$ to identify abnormal behavior using the traffic and driving criteria. For each agent $i$ at time $t$, the labeler assigns $y_{i,t} = 1$ if the behavior satisfies these criteria and $y_{i,t} = 0$ otherwise:

$$\mathbf{Y}_{1:T_f}^{(m,j)} = \mathcal{A}\left(\hat{\mathbf{C}}_{1:T_f}^{(m,j)}\right), \qquad \mathbf{Y}_{1:T_f}^{(m,j)} \in \{0,1\}^{N_a \times T_f}. \tag{6}$$

We group the labeled rollouts by their initial scene to form the preference dataset:

$$\mathcal{D}_{\mathrm{pref}} = \{\mathcal{G}_m\}_{m=1}^{\mathcal{M}}, \qquad \mathcal{G}_m = \left\{\left(\hat{\mathbf{C}}_{1:T_f}^{(m,j)}, \mathbf{Y}_{1:T_f}^{(m,j)}\right)\right\}_{j=1}^{N}. \tag{7}$$

## 4.2 CONTEXTUAL PREFERENCE LEARNING

Using $\mathcal{D}_{\mathrm{pref}}$, we train CPE to assess each agent's behavior through the scene states it produces. The input includes the evolving agent states and the map, so the same motion can receive different scores under different interactions. Given a rollout $\hat{\mathbf{C}}_{0:T_f}^{m,j}$, CPE produces a matrix of agent-time scores:

$$\mathbf{R}^{m,j} = \pi_\phi\left(\hat{\mathbf{C}}_{0:T_f}^{m,j}\right), \qquad \mathbf{R}^{m,j} \in [0,1]^{N_a \times T_f}, \tag{8}$$

Each entry $R_{i,t}^{m,j}$ scores agent $i$'s behavior as reflected in the scene evolution up to time $t$ in rollout $(m, j)$. CPE combines a frozen scene encoder with a trainable temporal convolutional preference decoder. The score at time $t$ uses only scene states up to and including $t$; states after $t$ cannot affect it.

CPE learns both absolute label agreement and relative behavioral preferences. Binary cross-entropy anchors each score to its agent-time label. Suppressing the scene and rollout indices, the objective over the $T_f$ labeled steps is

$$\mathcal{L}_{\mathrm{bce}} = -\frac{1}{N_a T_f} \sum_{i=1}^{N_a} \sum_{t=1}^{T_f} \left[y_{i,t} \log R_{i,t} + (1 - y_{i,t}) \log(1 - R_{i,t})\right]. \tag{9}$$

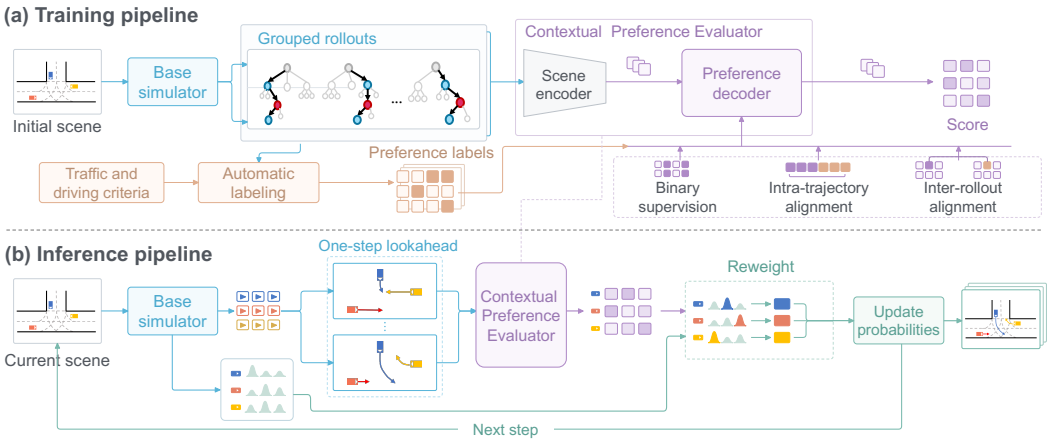



Figure 3: **Overview of CRAFT.** Top: Grouped simulator rollouts provide supervision for learning contextual behavioral preferences. Bottom: CPE scores one-step candidate scenes and reweights action probabilities during autoregressive generation.

Pairwise behavior comparisons provide supervision for learned models (Christiano et al., 2017). CPE derives these comparisons from automatic behavioral labels. Within a trajectory, labeled normal and failure segments provide temporal comparisons. Across a rollout group, labels compare alternative behaviors of the same agent at the same step. These objectives complement absolute classification by penalizing incorrect score orderings.

For intra-trajectory alignment, $\mathcal{P}_{\text{intra}}$ contains agents with both a labeled normal segment $T_i^+$ and a subsequent labeled failure segment $T_i^-$. The comparison follows these labels, rather than assuming that every earlier action is preferable to every later one. Let $\bar{R}_{i,T} = |T|^{-1} \sum_{t \in T} R_{i,t}$ be the mean segment score. We use a margin ranking loss with margin $\delta > 0$:

$$\mathcal{L}_{\text{intra}} = \frac{1}{|\mathcal{P}_{\text{intra}}|} \sum_{i \in \mathcal{P}_{\text{intra}}} \max\left(0, \bar{R}_{i,T_i^-} - \bar{R}_{i,T_i^+} + \delta\right). \tag{10}$$

For inter-rollout alignment, $\mathcal{P}_{\text{inter}}$ contains label-ordered pairs $(i, t, j^+, j^-)$ within the same initial-scene group. Rollouts $j^+$ and $j^-$ supply preferred and less preferred behavior for agent $i$ at step $t$, each scored in its own surrounding context. The shared initial condition enables comparison of alternative futures; their later scene states need not be identical. We optimize the Bradley–Terry objective (Bradley & Terry, 1952):

$$\mathcal{L}_{\text{inter}} = -\frac{1}{|\mathcal{P}_{\text{inter}}|} \sum_{(i,t,j^+,j^-) \in \mathcal{P}_{\text{inter}}} \log \frac{\exp\left(R_{i,t}^{(j^+)}\right)}{\exp\left(R_{i,t}^{(j^+)}\right) + \exp\left(R_{i,t}^{(j^-)}\right)}. \tag{11}$$

The overall training objective is

$$\mathcal{L}_{\text{CPE}} = \lambda_1 \mathcal{L}_{\text{bce}} + \lambda_2 \mathcal{L}_{\text{intra}} + \lambda_3 \mathcal{L}_{\text{inter}}, \tag{12}$$

where $\lambda_1$, $\lambda_2$, and $\lambda_3$ balance absolute token-level supervision and relative preference constraints.

### 4.3 Test-Time Preference Alignment

At inference time, CRAFT reweights the base simulator's motion distribution to reduce the probability of actions that are inappropriate for the current context (Fig. 3). Exhaustively evaluating the joint action space requires $K^{N_a}$ scene evaluations for $K$ candidates per agent, exceeding the computational budget for online guidance. We therefore use an independent marginal approximation with one-step lookahead. At each step $t$, we sample $K$ joint actions and simulate their immediate scene outcomes. CPE produces agent-wise scores, evaluating each action in the interaction context induced by the

other agents' randomly sampled actions. For agent $i$ and candidate $k$, we update the action probability as

$$P^{\text{post}}_{i,t,k} \propto P^{\text{prior}}_{i,t,k} \exp\Big(\beta R^{(k)}_{i,t}\Big). \tag{13}$$

We normalize the updated probabilities over each agent's candidate pool and sample actions independently to advance the simulation. Algorithm 1 summarizes the procedure.

**Algorithm 1:** Inference Sampling Guidance

**Input :** Initial scene context $\mathbf{C}_0$; Base simulator $\pi_\theta$; Contextual Preference Evaluator $\pi_\phi$; Scenario horizon $T_f$; Candidates $K$; Chunk size $K_c$; Guidance scale $\beta$.

**for** $t = 1, \dots, T_f$ **do**
- Extract Top-$K$ candidate action pool $\mathbf{A}_t \in \mathbb{R}^{N_a \times K}$ and priors $\mathbf{P}^{\text{prior}}_t$ from $\pi_\theta(\cdot|\mathbf{C}_{t-1})$;
- Partition candidate indices into chunks $\{\mathcal{K}_b\}_{b=1}^{\lceil K/K_c \rceil}$, where $|\mathcal{K}_b| \le K_c$;
- **for** *candidate chunk* $\mathcal{K}_b$ **do**
  - Generate lookahead scenarios: $\{\tilde{\mathbf{C}}^{(k)}_t\}_{k\in\mathcal{K}_b} \leftarrow \{\texttt{Step}(\mathbf{C}_{t-1}, \tilde{\mathbf{a}}^{(k)}_t)\}_{k\in\mathcal{K}_b}$ ;
  - Evaluate scores: $\mathbf{R}^{\mathcal{K}_b}_t \leftarrow \pi_\phi(\{\tilde{\mathbf{C}}^{(k)}_{0:t}\}_{k\in\mathcal{K}_b})$, where $\mathbf{R}^{\mathcal{K}_b}_t = [\mathbf{R}^{(k)}_t]_{k\in\mathcal{K}_b} \in \mathbb{R}^{N_a\times|\mathcal{K}_b|}$ ;
  - Update probabilities: $\mathbf{P}^{\text{unnorm}}_{:,t,\mathcal{K}_b} \leftarrow \mathbf{P}^{\text{prior}}_{:,t,\mathcal{K}_b} \odot \exp(\beta \mathbf{R}^{\mathcal{K}_b}_t)$ ;
- Normalize probabilities across candidates: $\mathbf{P}^{\text{post}}_{i,t,k} \leftarrow \frac{P^{\text{unnorm}}_{i,t,k}}{\sum_{j=1}^{K} P^{\text{unnorm}}_{i,t,j}}$ ;
- Sample $a_{i,t} \sim \texttt{Categorical}(\mathbf{P}^{\text{post}}_{i,t,:})$ for all agents $i$;
- Execute the actual step update: $\mathbf{C}_t \leftarrow \texttt{Step}(\mathbf{C}_{t-1}, \{a_{i,t}\}_{i=1}^{N_a})$ ;

**return** $\mathbf{C}_{1:T_f}$.

# 5 Experiments

## 5.1 Experimental Setup

**Data and base simulators.** We use the Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021) for training and validation. We evaluate CRAFT on different base simulators to test its compatibility. For every method, we initialize the simulator from the same logged scenes and generate autoregressive closed-loop rollouts for the same horizon. Unless otherwise stated, CRAFT uses $K = 32$ candidates, guidance scale $\beta = 2$, and the full preference-training set. Our training and testing experiments were conducted on 4 Nvidia RTX 5090 GPUs.

**Metrics.** WOSAC (Montali et al., 2023) is a widely used benchmark for traffic simulation, but log matching can penalize contextual correction and only partially capture behavioral rationality (Appendix D). We therefore evaluate behavioral rationality through event rates and motion realism through aggregate feature distributions, considering all moving agents. P-Sc/P-Ag denote fractions of scenes/agents with at least one event. We summarize collision, offroad, and traffic-violation P-Sc rates and define the realism score as the mean Jensen–Shannon distance (JSD) for speed, angular speed, and nearest-agent distance following prior studies (Lu et al., 2024; Rowe et al., 2025):

$$S_{\text{CL}} = 100\left[1 - \tfrac{1}{3}\left(r_{\text{col}} + r_{\text{off}} + r_{\text{traffic}}\right)\right], \tag{14}$$

$$S_{\text{real}} = \tfrac{1}{3}\left(\text{JSD}_{\text{spd}} + \text{JSD}_{\text{ang}} + \text{JSD}_{\text{dist}}\right). \tag{15}$$

Here $r \in [0, 1]$; higher $S_{\text{CL}}$ and lower $S_{\text{real}}$ are better. JSD compares pooled logged and simulated distributions, reducing dependence on individual logged futures (Appendix D.3).

**Baselines.** We compare pretrained and post-trained simulators, together with direct rule-based guidance. (1) *Pretrained imitation learning models*: GUMP (Hu et al., 2024) is a generative motion model for multi-task motion generation, and SMART (Wu et al., 2024) is an autoregressive multi-agent simulator trained via next-token behavior cloning. (2) *Post-training refinement models*: CAT-K (Zhang et al., 2025) refines tokenized traffic models through closed-loop supervised fine-tuning, while R1Sim (Wang et al., 2026) and RLFTSim (Ahmadi et al., 2026) apply reinforcement-learning fine-tuning with handcrafted rewards to optimize generated trajectories. (3) *Auto-labeler*: We additionally compare with a direct rule-based guidance baseline, where the same human-aligned criteria used for auto-labeling are applied online to score candidate actions.

Table 1: **Closed-loop simulation performance across base simulators.** All CRAFT rows use the same CPE checkpoint without retraining. Spd., Ang., and Dist. denote speed, angular speed, and nearest-agent distance; P-Ag. and P-Sc. denote per-agent and per-scene rates, respectively. Realism score $S_{\text{real}}$ is the mean of the three reported JSDs, in units of $10^{-2}$. Parentheses show score changes from the corresponding Top-$K$ baseline: decreases in realism score and increases in closed-loop score indicate improvements.

| Base Model | Strategy | JSD ($\times 10^{-2}$) ↓ | | | Realism score ↓ | Collision (%) ↓ | | Offroad (%) ↓ | | Traffic (%) ↓ | Closed-loop score ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Spd. | Ang. | Dist. | | P-Ag. | P-Sc. | P-Ag. | P-Sc. | P-Sc. | |
| Log | Log replay | 0.00 | 0.00 | 0.00 | 0.00 | 0.56 | 5.20 | 1.59 | 14.40 | 20.70 | 86.57 |
| GUMP | Top-K | 4.78 | 5.41 | 11.36 | 7.18 | 4.06 | 36.30 | 3.80 | 26.90 | 32.70 | 68.03 |
| SMART | Top-K | 1.07 | 3.23 | 0.56 | 1.62 | 3.13 | 15.10 | 2.52 | 19.10 | 38.80 | 75.67 |
| CAT-K | Top-K | 1.02 | 2.96 | **0.53** | 1.50 | 3.41 | 15.70 | 2.54 | 19.50 | 36.50 | 76.10 |
| R1Sim | Top-K | 1.05 | 3.16 | 0.56 | 1.59 | 3.36 | 15.50 | 2.23 | 16.30 | 36.60 | 77.20 |
| RLFTSim | Top-K | 1.06 | 3.02 | 0.68 | 1.59 | 2.28 | 11.10 | 1.82 | 12.80 | 46.30 | 76.60 |
| CAT-K | Auto-labeler | 1.19 | 3.02 | 0.67 | 1.63 | 3.21 | 15.50 | 2.57 | 20.30 | 36.70 | 75.83 |
| SMART | CRAFT | 0.91 | 2.20 | 0.72 | 1.28 (−0.34) | 2.60 | 11.90 | 2.06 | 17.10 | **25.90** | **81.70** (+6.03) |
| CAT-K | CRAFT | 0.92 | 2.20 | 0.69 | 1.27 (−0.23) | 2.46 | 10.80 | 2.26 | 16.60 | 26.60 | **82.00** (+5.90) |
| RLFTSim | CRAFT | **0.87** | **2.19** | 0.71 | **1.26** (−0.33) | **2.14** | **10.10** | **1.73** | **11.60** | 40.30 | **79.33** (+2.73) |

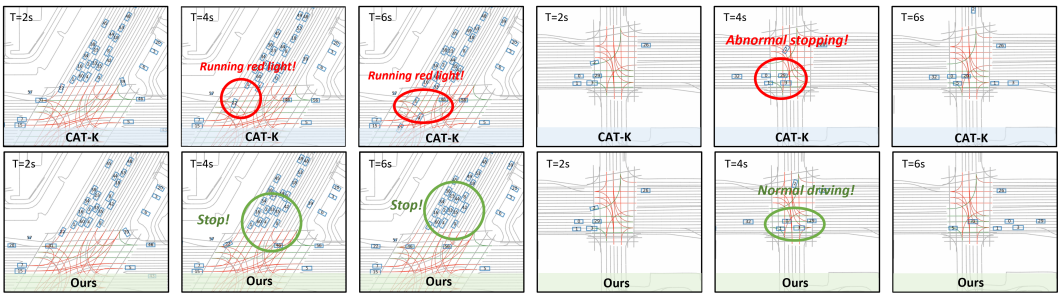



Figure 4: **Context-dependent behavioral correction.** CAT-K (top) and CRAFT (bottom) start from identical initial scenes. Left: CRAFT stops for a red light. Right: CRAFT maintains progress where CAT-K stops unnecessarily.

### 5.2 Closed-Loop Simulation Performance

**Contextual guidance improves both realism and rationality.** Table 1 shows consistent improvements in behavioral rationality and motion realism across the base simulators. CRAFT reduces collision, offroad, and traffic-violation rates while better matching aggregate motion distributions. The slight increase in spacing JSD may reflect larger inter-agent buffers encouraged by downweighting collision-prone candidates. On CAT-K, CRAFT also outperforms direct rule-based guidance on both summary scores, highlighting the value of learned contextual evaluation. A single CPE trained on CAT-K rollouts also improves SMART and RLFTSim without retraining, demonstrating transfer across different training paradigms.

The paired rollouts in Fig. 4 illustrate how these improvements depend on the interaction context. From identical initial scenes, CAT-K proceeds when a red light requires stopping, but stops unnecessarily when the intersection permits continued motion. CRAFT corrects both responses: it stops for the red light and maintains progress through the unobstructed intersection. Together with the distributional results, these examples support contextual correction that retains useful motion capabilities rather than simply favoring slower or more conservative behavior.

### 5.3 Contextual Evaluation and Guidance

**Complementary supervision improves evaluation and guidance.** Token-level scoring improves open-loop $F_1$, enabling finer-grained identification of behavioral anomalies at the level of individual action candidates (Fig. 5). Rollout supervision improves closed-loop performance by exposing states where the base simulator generates actions inconsistent with their surrounding context; these failures provide targeted examples for learning to recognize context–action mismatches. The intra-trajectory

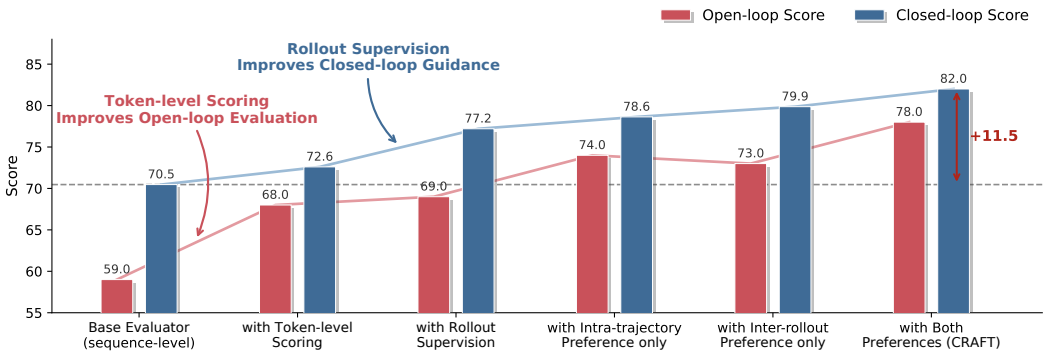
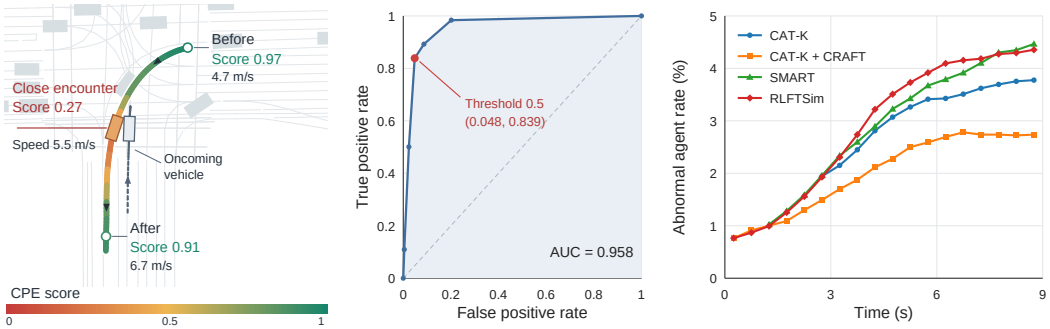



Figure 5: **Ablation of CPE training.** The first three settings cumulatively introduce token-level scoring and simulator-rollout supervision. The last three independently add intra-trajectory, inter-rollout, or both preference objectives to the same rollout-trained evaluator. Open-loop and closed-loop scores denote $100F_1$ and $S_{\mathrm{CL}}$.




(a) Offline trajectory scoring. (b) CPE ROC analysis. (c) Abnormal-agent rates over time.

Figure 6: **Contextual evaluation and closed-loop guidance.** (a) One vehicle before, during, and after a close encounter. Colors encode CPE scores; the dashed path shows the oncoming vehicle. Vehicle boxes show the closest-approach frame. (b) CPE achieves an AUC of 0.958 on test dataset. (c) CRAFT suppresses abnormal-agent rates over the rollout.

and inter-rollout objectives add explicit relative comparisons to binary supervision, encouraging CPE to learn finer distinctions in behavioral appropriateness across scene contexts. Both improve open-loop $F_1$ and closed-loop performance, with their combination achieving the best results.

**CPE scores reflect contextual appropriateness.** Figure 6a illustrates CPE's ability to assess whether a vehicle's behavior is appropriate for its interaction context. We use CPE to evaluate the CAT-K generated trajectories offline. As the vehicle approaches oncoming traffic, the original trajectory continues to accelerate rather than slowing to yield. CPE assigns lower scores to this context-inconsistent response, with the score dropping from 0.97 before the encounter to 0.27 at closest approach. After the vehicles pass, the score recovers to 0.91 despite a further increase in speed. Thus, the score does not simply penalize faster or slower motion: it reflects whether the behavior is appropriate for the surrounding interaction. On the test dataset, CPE achieves an AUC of 0.958 (Fig. 6b), complementing this qualitative example with aggregate discrimination performance.

**Active guidance improves on passive rollout selection.** We compare stepwise guidance with selecting completed rollouts using CPE or the auto-labeler. Offline selection with CPE or the auto-labeler lowers event rates relative to unfiltered sampling (Table 2), showing that the base simulator already produces better-behaved futures but does not reliably favor them during generation. Online CRAFT further reduces all three event rates, demonstrating the benefit of actively guiding generation over passively filtering completed rollouts. Its advantage over the base simulators also widens over time, with consistently lower abnormal-agent rates (Fig. 6c). These results highlight the role of test-time guidance in directing the simulator's existing motion capabilities toward context-appropriate behavior throughout a rollout.

Table 2: **Offline selection and online guidance on CAT-K.** Ranking retains 32 of 64 rollouts; AL denotes auto-labeler. Col./Off./Tra. are scene-level rates (%); CL is closed-loop score.

| Strategy | Col. ↓ | Off. ↓ | Tra. ↓ | CL ↑ |
|---|---|---|---|---|
| Top-$K$ | 15.70 | 19.50 | 36.50 | 76.10 |
| AL rank | 12.80 | 17.90 | 34.30 | 78.33 |
| CPE rank | 13.90 | 17.90 | 34.00 | 78.07 |
| CRAFT | **10.80** | **16.60** | **26.60** | **82.00** |

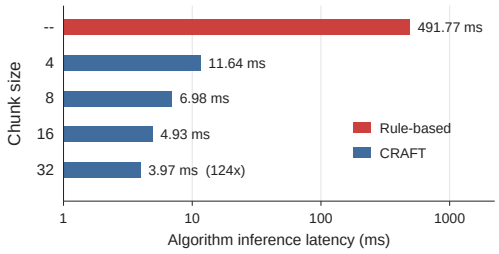



Figure 7: **Inference latency.**

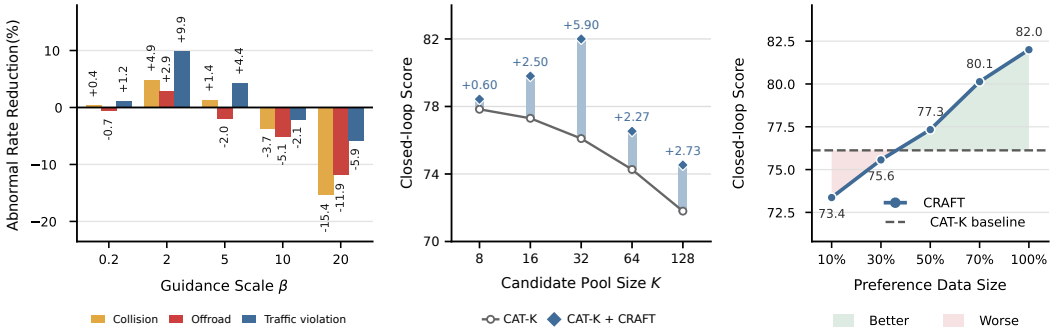



Figure 8: **Sensitivity of contextual guidance on CAT-K.** Left: abnormal rate reduction over unguided CAT-K across guidance scales $\beta$. Middle: closed-loop scores with and without CRAFT at matched candidate-pool sizes $K$. Right: closed-loop score versus the scaling of preference-training data.

### 5.4 Sensitivity and Computational Efficiency

**Effective guidance balances contextual preferences with the motion prior.** Performance peaks at $\beta = 2$, while stronger guidance degrades all three event rates (Fig. 8). CPE evaluates contextual appropriateness rather than modeling the full motion distribution; excessive reweighting may therefore override useful motion preferences encoded by the base simulator. Expanding the top-$K$ pool admits lower-probability actions, potentially making reliable selection harder. Although CRAFT outperforms the matched CAT-K baseline at every tested $K$, its score peaks at $K = 32$ and then declines alongside the baseline. This suggests that contextual reweighting improves candidate selection but cannot fully offset degradation in the proposal distribution. By contrast, increasing preference data consistently improves performance under fixed inference settings, supporting better-learned contextual judgments.

**Millisecond-scale evaluation supports real-time guidance.** CRAFT reduces candidate-evaluation latency from 491.77 ms for the rule-based auto-labeler to 3.97 ms at a chunk size of 32, achieving a 124× speedup (Fig. 7). For a fixed candidate pool, increasing the chunk size from 4 to 32 provides a 2.9× speedup. This low evaluation overhead supports contextual correction at each autoregressive step, making CRAFT better suited to latency-constrained real-time traffic simulation.

## 6 Conclusion

We study an underexamined mismatch between fitting incomplete driving logs and generating behavior justified by the simulated scene. Motivated by the limitations of incomplete-context imitation, CRAFT learns contextual preferences from closed-loop rollouts and uses them to improve action selection at test time. Experiments show that a single CPE improves SMART, CAT-K, and RLFTSim without adaptation, raising closed-loop scores by 2.73–6.03 points, supporting contextual evaluation as a reusable complement to motion-prior learning.

CRAFT remains limited by candidate coverage: it cannot select suitable actions absent from the proposal pool (Appendix F). Improving action search or adapting proposal distributions dominated by inappropriate behaviors are promising directions. Appendices A–F provide implementation details, additional analyses, and guidance limitations.

### AI USE STATEMENT

In this work, we used generative AI tools to identify related work and assist with code for statistical analysis, plotting, and visualization. We have not used generative AI tools to develop the research motivation, write the initial manuscript draft, or design the core method; these tasks were completed by the authors. Additionally, we used generative AI tools to polish author-written text. We have reviewed all AI-assisted work. We verified the additional references identified by LLMs to ensure that they were genuine and not fabricated. We also manually reviewed LLM-generated visualization code to ensure consistency between the data presented in the visualizations and those reported in the manuscript text. The authors finalized the interpretation of experimental results and the scientific conclusions. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

Sections 4.1–4.3 describe CRAFT's rollout supervision, preference learning, and test-time guidance. Appendix A documents data preprocessing, model architecture, and training and inference settings. Appendix C details CRAFT15K construction, data quantities, and behavioral annotation. Evaluation metrics and the human evaluation protocol are specified in Appendices D.3 and D.5, respectively. Appendix F.2 reports the profiling hardware, latency, and memory costs. We plan to publicly release the code and trained model weights upon acceptance.

## REFERENCES

Ehsan Ahmadi, Hunter Schofield, Behzad Khamidehi, et al. 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)*, 2026.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Yulong Cao, Boris Ivanovic, Chaowei Xiao, and Marco Pavone. Reinforcement learning with human feedback for realistic traffic simulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Wei-Jer Chang, Francesco Pittaluga, Masayoshi Tomizuka, et al. SAFE-SIM: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Paul F. Christiano, Jan Leike, Tom Brown, et al. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Daphne Cornelisse, Julian Hunt, Zixu Zhang, et al. Human-like autonomy emerges from self-play and a pinch of human data. *arXiv preprint arXiv:2606.19370*, 2026.

Jinlong Cui, Fenghua Liang, Guo Yang, et al. Safer: Safety-critical scenario generation for autonomous driving test via feasibility-constrained token resampling. *arXiv preprint arXiv:2603.04071*, 2026.

Pim De Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, 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 (ICCV)*, pp. 9710–9719, 2021.

Shuo Feng, Haowei Sun, Xintao Yan, et al. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953):620–627, 2023.

Cole Gulino, Justin Fu, Wenjie Luo, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Yihan Hu, Siqi Chai, Zhening Yang, et al. Solving motion planning tasks with a scalable generative model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 386–404, 2024.

Zhiyu Huang, Xinshuo Weng, Maximilian Igl, et al. Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3445–3451, 2025.

Zhiyu Huang, Zixu Zhang, Ameya Vaidya, et al. Versatile behavior diffusion for generalized traffic agent simulation. *IEEE Transactions on Intelligent Transportation Systems*, 27(7):8681–8697, 2026.

Haohong Lin, Xin Huang, Tung Phan, et al. Causal composition diffusion model for closed-loop traffic generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27542–27552, 2025.

Jack Lu, Kelvin Wong, Chris Zhang, et al. Scenecontrol: Diffusion for controllable traffic scene generation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16908–16914, 2024.

Nico Montali, John Lambert, Paul Mougin, et al. The waymo open sim agents challenge. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 59151–59171, 2023.

Long Nguyen, Micha Fauth, Bernhard Jaeger, et al. Lead: Minimizing learner-expert asymmetry in end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39775–39785, 2026.

Zhenghao Peng, Wenjie Luo, Yiren Lu, et al. Improving agent behaviors with RL fine-tuning for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. *arXiv preprint arXiv:2312.04535*, 2023.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635, 2011.

Luke Rowe, Roger Girgis, Anthony Gosselin, et al. Ctrl-sim: Reactive and controllable driving agents with offline reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, pp. 3600–3621, 2025.

Ari Seff, Brian Cera, Dian Chen, et al. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8579–8590, 2023.

Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10400–10409, 2021.

Gokul Swamy, Sanjiban Choudhury, J. A. Bagnell, and Steven Z. Wu. Sequence model imitation learning with unobserved contexts. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Shuhan Tan, Boris Ivanovic, Yuxiao Chen, et al. Promptable closed-loop traffic simulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2024.

Shuhan Tan, John Lambert, Hong Jeon, et al. SceneDiffuser++: City-scale traffic simulation via a generative world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1570–1580, 2025.

Haochen Tian, Tianyu Li, Haochen Liu, et al. SimScale: Learning to drive via real-world simulation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Ran Tian and Kratarth Goel. Direct post-training preference alignment for multi-agent motion generation models using implicit feedback from pre-training demonstrations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Ziyan Wang, Peng Chen, Ding Li, et al. Learning rollout from sampling: An r1-style tokenized traffic simulation model. *IEEE Robotics and Automation Letters*, 11(5):6336–6343, 2026.

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

Lingyu Xiao, Zexin Feng, and Xintao Yan. Long-term traffic simulation via structured autoregressive modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. BITS: Bi-level imitation for traffic simulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

Xintao Yan, Zhengxia Zou, Shuo Feng, et al. Learning naturalistic driving environment with statistical realism. *Nature communications*, 14(1):2037, 2023.

Xiuyu Yang, Shuhan Tan, and Philipp Krähenbühl. Long-term traffic simulation with interleaved autoregressive motion and scenario generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Chris Zhang, James Tu, Lunjun Zhang, et al. Learning realistic traffic agents in closed-loop. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 229, pp. 800–821, 2023.

Zhejun Zhang, Peter Karkus, Maximilian Igl, et al. Closed-loop supervised fine-tuning of tokenized traffic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5422–5432, 2025.

Seth Z Zhao, Luobin Wang, Hongwei Ruan, et al. Bridgesim: Unveiling the ol-cl gap in end-to-end autonomous driving. *arXiv preprint arXiv:2604.10856*, 2026.

Ziyuan Zhong, Davis Rempe, Yuxiao Chen, et al. Language-guided traffic simulation via scene-level diffusion. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 229, pp. 144–177, 2023a.

Ziyuan Zhong, Davis Rempe, Danfei Xu, et al. Guided conditional diffusion for controllable traffic simulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3560–3566. IEEE, 2023b.

Yunsong Zhou, Naisheng Ye, William Ljungbergh, et al. Decoupled diffusion sparks adaptive scene generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27760–27770, 2025.

Zikang Zhou, Haibo Hu, Xinhong Chen, et al. Behaviorgpt: Smart agent simulation for autonomous driving with next-patch prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 79597–79617, 2024.

# APPENDIX

Appendix A describes the WOMD setup, model architecture, training, and inference configurations. Appendix B examines the role of context in behavioral errors. Appendix C describes CRAFT15K construction and annotation, and analyzes the failure patterns exposed by the generated rollouts. Appendix D explains the evaluation metrics and examines the benefits and costs of correction, including human judgments of scene realism. Appendix E illustrates closed-loop recovery and stability, followed by the limits of test-time guidance in Appendix F.

## A EXPERIMENTAL DETAILS

### A.1 WOMD AND PREPROCESSING

We use the Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021). Each scenario provides 1 s of observations and an 8 s rollout horizon at 10 Hz. The base simulators are trained on the training split. Framework comparisons use 2% of the validation split, with the same initial scenes and rollout horizon across methods. During training, we select up to 32 target agents within 100 m of the ego vehicle that are valid at the final observation step. Inference retains all initially valid agents.

### A.2 MODEL ARCHITECTURE

Following SMART (Wu et al., 2024) and CAT-K (Zhang et al., 2025), we build map and motion vocabularies using $k$-disk clustering. The map vocabulary has 1,024 tokens, each representing a 5 m polyline with 10 consecutive segments. The motion vocabulary has 2,048 tokens, each representing a 0.5 s trajectory fragment. Map polylines and motion fragments are assigned to their nearest vocabulary entries.

We initialize the CPE scene encoder from pretrained CAT-K and freeze its parameters. The encoder combines temporal agent features with agent–map and agent–agent attention. A preference decoder with five residual causal temporal convolutional layers and a scalar scoring head produces an agent-time score in $[0, 1]$. Each score uses only scene history up to that step. Table 3 gives the model configuration.

Table 3: Scene-encoder and preference-decoder configuration.

| **Shared representation** | | **Agent encoder** | |
|---|---|---|---|
| Hidden dimension | 128 | Attention layers | 6 |
| Attention heads | 8 | Historical samples | 11 |
| Head dimension | 16 | Future samples | 80 |
| Fourier frequency bands | 64 | Temporal span | 30 |
| Encoder dropout | 0.1 | Temporal stride | 5 |
| | | Agent–agent radius (m) | 60 |
| **Map and interaction encoding** | | **Preference decoder** | |
| Map attention layers | 3 | Temporal convolutional layers | 5 |
| Map–map radius (m) | 10 | Kernel size | 6 |
| Map–agent radius (m) | 30 | GroupNorm groups | 16 |
| Maximum map–map neighbors | 100 | MLP layers | 4 |
| Maximum map–agent neighbors | 300 | Decoder dropout | 0.15 |
| Maximum agent–agent neighbors | 300 | Output dimension | 1 |

### A.3 MODEL TRAINING

We optimize the preference decoder on CRAFT15K (Appendix C) for 32 epochs, keeping the pretrained scene encoder frozen. Training uses four NVIDIA RTX 5090 GPUs. Table 4 lists the learning rate, batch size, and objective hyperparameters.

Table 4: CPE training configuration.

| Training setting | Value | Objective setting | Value |
|---|---|---|---|
| Learning rate | $10^{-5}$ | BCE weight $\lambda_1$ | 0.5 |
| Batch size | 8 | Intra-trajectory weight $\lambda_2$ | 1.5 |
| Epochs | 32 | Inter-rollout weight $\lambda_3$ | 1.0 |
| | | Ranking margin $\delta$ | 0.5 |

### A.4 Model Inference

At each autoregressive step, CRAFT evaluates one-step candidate scenes with CPE and reweights the base simulator's action probabilities before sampling. Table 5 gives the default inference settings. Candidate scenes are evaluated in chunks; the profiled chunk sizes and their computational costs are reported in Appendix F.2.

Table 5: Default CRAFT inference configuration.

| Setting | Value | Setting | Value |
|---|---|---|---|
| Candidate-pool size $K$ | 32 | Sampling temperature | 1 |
| Guidance scale $\beta$ | 2 | Lookahead depth | 1 step |

## B Role of Context in Behavioral Errors

We test how access to surrounding context affects closed-loop behavioral errors. Table 6 varies perception range and visible training agents using matched 1M-parameter models. The visible-agent comparison fixes the scenes, target actions, architecture, tokenizer, and training budget, varying only the surrounding agents visible during training. Collision, offroad, and traffic-violation rates are P-Sc percentages; Stop denotes the abnormal-stopping rate.

Table 6: Effects of context availability on behavioral errors.

| Setting | Collision ↓ | Offroad ↓ | Traffic ↓ | Stop ↓ |
|---|---|---|---|---|
| *Perception range* | | | | |
| 30 m | 26.90 | 22.00 | 37.50 | 9.98 |
| 45 m | 25.50 | 20.00 | 37.50 | 9.69 |
| 60 m | 22.00 | 19.40 | 36.50 | 9.46 |
| *Visible training context* | | | | |
| Ego only | 42.80 | 23.80 | 46.20 | 10.35 |
| Ego + 8 agents | 24.30 | 22.40 | 38.60 | 9.70 |
| Ego + 32 agents | 22.00 | 19.40 | 36.50 | 9.46 |
| Ego + all agents | 22.80 | 17.80 | 35.70 | 9.24 |

**Richer context improves behavioral rationality.** Adding surrounding agents reduces collision incidence from 42.8% to 22.0%, while also improving offroad, traffic-violation, and abnormal-stopping rates. Expanding perception range produces a similar trend. The benefits extend across behaviors that depend on different scene elements, supporting the role of context in learning when a motion is appropriate. These findings motivate CRAFT's use of the current simulated context to evaluate candidate actions.

## C CRAFT15K and Behavioral Supervision

We describe CRAFT15K's construction and annotation, then examine what the generated rollouts reveal about the base simulator's behavioral failures.

### C.1 Constructing Grouped Rollouts

We construct CRAFT15K from 15,000 logged initial scenes. CAT-K generates $N = 32$ stochastic closed-loop rollouts per scene using different random seeds, yielding 480,000 rollout candidates before preprocessing and validity filtering.

Rollouts from the same initial scene form one group. Agent identities and time indices align alternative futures, while each rollout retains its own evolving traffic context. Grouping preserves temporal sequences and cross-rollout comparisons.

### C.2 Grounding Behavioral Supervision

The automatic labeler evaluates each agent's motion together with the surrounding agents, road geometry, and traffic signals in the generated scene. It assigns $y_{i,t} = 1$ when an agent-time token satisfies all six behavioral criteria and $y_{i,t} = 0$ otherwise. Table 7 specifies the abnormal behaviors covered by each criterion and summarizes their shares among negative annotations.

Spatial checks relate motion to nearby agents, the drivable area, and lane directions. Signal and stopping checks use the relevant traffic signal, leading blockers, and upcoming turns to judge whether stopping or continued motion is appropriate. Jitter checks detect back-and-forth motion. These conditions ground the labels in the interactions represented by each rollout.

Figure 9 illustrates the six criteria, with adjacent temporal bars marking positive labels in green and negative labels in red for the circled agents. A rollout can contain both reasonable and abnormal segments. The step-wise annotations localize these failures, preserving reasonable segments as positive supervision instead of rejecting an entire trajectory because of a local error.

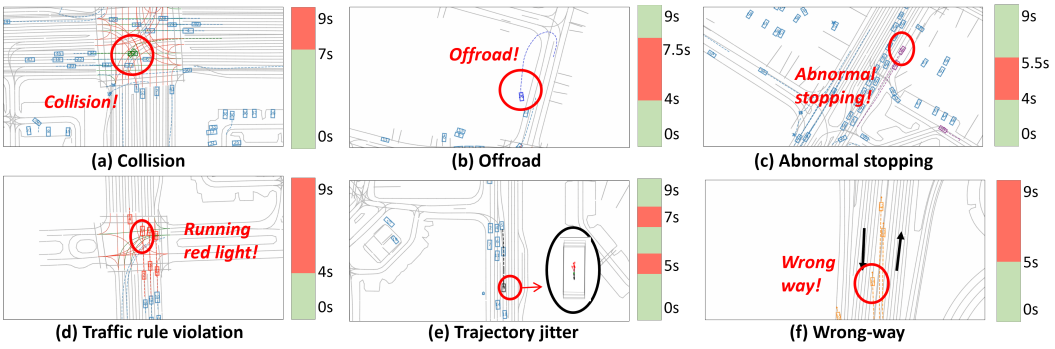



Figure 9: Behavioral failures and step-wise annotations in CRAFT15K.

Table 7: Behavioral criteria and label composition in CRAFT15K.

| Criterion | Abnormal behavior | Train (%) | Test (%) |
|---|---|---|---|
| Collision | Collision or hazardous spatial conflict with another agent. | 38.2 | 40.1 |
| Abnormal stopping | Stopping without a leading blocker, red signal, or upcoming large-angle turn. | 27.1 | 25.3 |
| Traffic violation | Red-light running or unreasonable inaction under green for the relevant movement. | 15.1 | 13.9 |
| Offroad | Departure from the drivable area. | 12.1 | 12.2 |
| Wrong-way driving | Large heading disagreement with the traversed lane. | 2.5 | 2.5 |
| Trajectory jitter | Abnormal back-and-forth lateral or longitudinal motion. | 5.0 | 5.8 |

After preprocessing, the preference-training and test sets contain approximately 17.1 million and 1.1 million labeled tokens; 13.42% and 13.19% are negative, respectively. Labels retain their agent, time, and rollout associations to supervise the corresponding CPE outputs.

### C.3 Failure Patterns in Self-Generated Rollouts

Collision, abnormal stopping, traffic violation, and offroad events account for 92.5% of negative training annotations and 91.5% of negative test annotations (Table 7). These dominant failure modes directly undermine behavioral rationality by compromising collision avoidance, appropriate progress, signal compliance, and road adherence. They can also destabilize subsequent simulation: the resulting agent states become part of the shared scene that conditions later actions, allowing local errors to disrupt surrounding interactions and propagate over time. The generated data therefore highlight frequent, disruptive behaviors whose correction is important for both behavioral rationality and closed-loop stability.

## D Evaluating Contextual Correction

The Waymo Open Sim Agents Challenge (WOSAC) (Montali et al., 2023) is a widely used benchmark for multi-agent traffic simulation. However, logged behavior may depend on context that is absent from the simulated scene. In such cases, evaluating each agent's logged features under its simulated feature distributions can penalize reasonable responses to the available context. We therefore evaluate motion realism using Jensen–Shannon distance (JSD) between aggregate feature distributions, reducing dependence on agent-specific correspondence with the log.

Our scenario-level statistics further show only a modest correlation between WOSAC scores and reasonable rollout rates: high-scoring scenarios can contain frequent behavioral failures, while lower-scoring scenarios can still produce reasonable rollouts. We therefore measure behavioral rationality separately using direct event rates. The following qualitative and quantitative analyses motivate these two metric choices, respectively.

### D.1 Motion Realism under Changing Context

The L2G context mismatch can also affect evaluation. WOSAC evaluates the likelihood of each agent's logged motion and interaction features under that agent's simulated feature distributions. A logged action may depend on conditions absent from the represented scene, so this agent-specific correspondence can penalize a reasonable response to the simulated context. Related discussions distinguish agreement with recorded behavior from the quality of alternative driving behaviors (Cornelisse et al., 2026; Xiao et al., 2026).

Figure 10 illustrates the distinction. CAT-K slows toward a stop despite the absence of a leading blocker, whereas CRAFT maintains progress. CAT-K more closely reproduces the logged deceleration and receives the higher WOSAC score. Log matching and contextual correction can thus favor different behaviors.

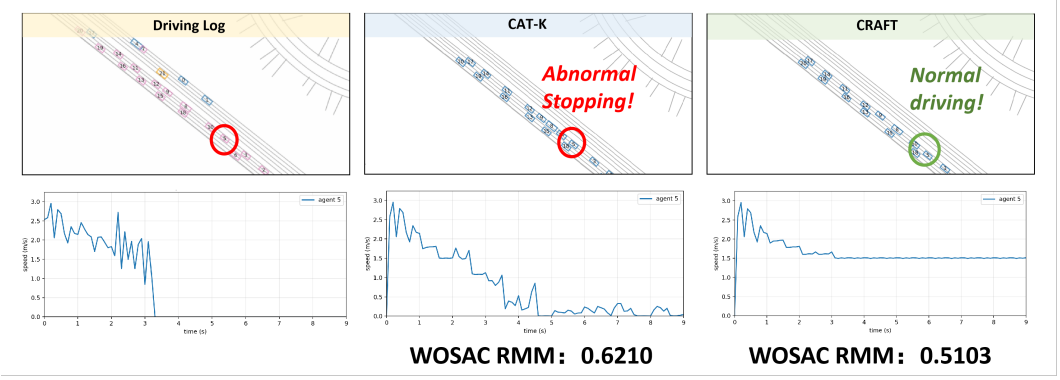



Figure 10: Log matching and contextual correction in a simulated scene.

This comparison motivates measuring motion realism through aggregate feature distributions. Pooling features across agents reduces dependence on reproducing an individual logged future, while retaining the log as a reference for typical motion statistics. Appendix D.3 describes the resulting JSD and event-rate metrics.

### D.2 Behavioral Rationality beyond Realism Scores

**High WOSAC scores can accompany behavioral failures.** Across 1,000 scenarios, Figure 11 plots each scenario's mean WOSAC score against its reasonable rollout rate, the fraction of rollouts that pass the behavioral checks. Higher WOSAC scores are associated with more reasonable rollouts on average, but the rank correlation is modest (Spearman's $\rho = 0.286$). The LOWESS curve summarizes this trend, while vertical bars show the local 10th–90th percentile range of reasonable rollout rates. This spread remains wide at higher WOSAC scores, indicating substantial differences in behavior among scenarios with similar scores.

A high WOSAC score can therefore accompany repeated behavioral failures, and a lower score does not necessarily imply less reasonable behavior. Agreement with logged feature distributions cannot establish behavioral rationality (Cornelisse et al., 2026; Xiao et al., 2026). We therefore measure behavioral failures directly using event rates, alongside JSD for motion realism.

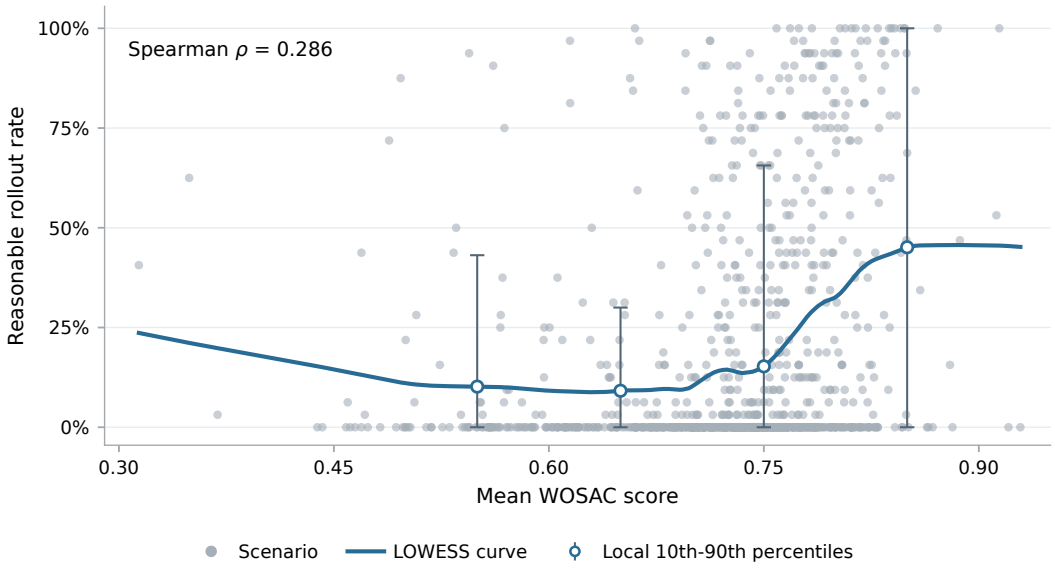



Figure 11: WOSAC realism and behavioral rationality.

### D.3 Evaluation Metric Design

WOSAC directly evaluates the ego vehicle and WOMD's designated prediction targets (up to eight agents), whereas a scene can contain dozens of agents (Ettinger et al., 2021).[1] WOSAC evaluates each agent's logged features under that agent's simulated feature distributions, so a reasonable response can score lower when it departs from that particular logged future. To assess contextual correction without requiring each agent to reproduce its own logged behavior, we measure motion realism through pooled feature distributions across all valid moving agents and behavioral rationality through direct event rates.

**Aggregate motion distributions.** Within each scene, we pool speed, angular speed, and nearest-agent distance from all valid moving agents over the rollout horizon. Speed and angular speed characterize translational and rotational motion; nearest-agent distance characterizes interaction spacing. For each feature, the resulting distribution can be written as

$$p = \sum_i w_i p_i, \tag{16}$$

where $p_i$ is the normalized feature histogram of agent $i$, and $w_i$ is its fraction of all valid feature samples. We construct the logged distribution $p$ and simulated distribution $q$ separately using shared

[1] See the official WOSAC agent-selection implementation. Other simulated agents still affect interaction metrics, such as collisions involving evaluated agents.

bins and compare them using Jensen–Shannon distance following prior studies (Lu et al., 2024; Rowe et al., 2025):

$$\mathrm{JSD}(p,q) = \sqrt{\tfrac{1}{2}D_{\mathrm{KL}}(p\|m) + \tfrac{1}{2}D_{\mathrm{KL}}(q\|m)}, \qquad m = \tfrac{1}{2}(p+q). \tag{17}$$

The realism score $S_{\mathrm{real}}$ in equation 15 equally weights the three feature distances, analogous to the equal weighting of event rates in $S_{\mathrm{CL}}$. Lower values indicate closer distributions. Table 1 reports both the individual distances and $S_{\mathrm{real}}$ in units of $10^{-2}$. With the sample weights fixed, a change in one agent's feature distribution contributes to the pooled distribution in proportion to its weight. This limits the influence of isolated context mismatches while retaining the log as a reference for scene-level motion statistics, without requiring each agent to reproduce its own logged behavior.

**Behavioral event definitions.** We evaluate all valid moving agents and exclude stationary objects to reduce pre-existing overlap artifacts. Collision detection checks overlap between oriented vehicle bounding boxes using the separating axis theorem. offroad detection checks vehicle-box corners against nearby road edges for departures from the drivable area. Traffic violations include red-light running and unreasonable inaction under a green signal for the relevant movement. These checks assess the simulated scene directly and do not require agreement with an agent's logged trajectory.

For event type $e$, let $N_{\mathrm{sc},e}$ count evaluated scenes with at least one such event, and $N_{\mathrm{ag},e}$ count evaluated agents involved in at least one event. With $N_{\mathrm{sc}}$ evaluated scenes and $N_{\mathrm{ag}}$ evaluated moving agents, we report

$$r_e^{\text{P-Sc}} = \frac{N_{\mathrm{sc},e}}{N_{\mathrm{sc}}}, \qquad r_e^{\text{P-Ag}} = \frac{N_{\mathrm{ag},e}}{N_{\mathrm{ag}}}. \tag{18}$$

Each scene or agent is counted at most once per event type, regardless of how often the event recurs. P-Sc measures how broadly failures affect scenes; P-Ag measures the fraction of agents affected. We report both for collision and offroad events, and report traffic violations at the scene level because signal applicability varies across agents. Tables express rates as percentages. The closed-loop score $S_{\mathrm{CL}}$ in equation 14 equally weights the three P-Sc rates and converts their average to a score where higher is better. Our event rates differ from official WOSAC rates in agent selection, event definitions, and aggregation.

The two metric families provide complementary evidence: realistic aggregate motion can contain unsafe interactions, while avoiding events by remaining stationary need not reproduce realistic motion. We therefore examine JSD and event rates together when assessing contextual correction.

### D.4 WOSAC Results

As discussed in Appendix D.1, WOSAC rewards similarity to logged behavior. When the simulated context differs from that of the log, reproducing a logged action can receive a higher score even when that action is unreasonable in the simulated scene. This preference conflicts with CRAFT's objective of correcting behavior according to the available context, making the log-matching score unsuitable as the primary criterion for evaluating contextual correction. For completeness, we nevertheless report results from the 2025 WOSAC evaluator on the WOMD Sim Agents validation split with 32 stochastic rollouts per scene (Table 8). Base denotes the original simulator and $\beta$ denotes the CRAFT guidance scale.

**Agent-specific log matching and motion realism.** Although CRAFT receives lower WOSAC scores, these scores alone do not imply that its simulated scenes are less realistic. As discussed in Appendix D.3, agreement with an individual logged future is distinct from realism in the simulated context. CRAFT's contextual corrections may depart from recorded actions and thus reduce the log-matching score. Under the same evaluator, CRAFT reduces collision, offroad, and traffic-violation rates across all three base simulators at the default guidance scale.

**Effect of guidance strength.** On CAT-K, moderate guidance ($\beta = 0.5$) reduces the reported event rates with a smaller change in the WOSAC composite score. The default setting applies stronger correction and further reduces these failures, while moving the score farther from the base model. This trend is consistent with the tension between correcting behavior for the simulated context and reproducing logged features.

Table 8: Complete WOSAC results.

| | SMART | | CAT-K | | | | RLFTSim | |
|---|---|---|---|---|---|---|---|---|
| Metric | Base | $\beta = 2$ | Base | $\beta = .2$ | $\beta = .5$ | $\beta = 2$ | Base | $\beta = 2$ |
| Composite realism ↑ | .7630 | .7190 | .7647 | .7647 | .7575 | .7189 | .7634 | .7176 |
| **Kinematic score** ↑ | .4595 | .3493 | .4588 | .4585 | .4453 | .3420 | .4493 | .3485 |
| Linear speed | .327 | .202 | .329 | .328 | .298 | .194 | .294 | .185 |
| Linear acceleration | .388 | .276 | .387 | .387 | .378 | .268 | .387 | .279 |
| Angular speed | .482 | .378 | .479 | .479 | .468 | .368 | .474 | .377 |
| Angular acceleration | .641 | .541 | .640 | .640 | .637 | .538 | .642 | .553 |
| **Interactive score** ↑ | .7948 | .7630 | .7993 | .7989 | .7930 | .7583 | .8007 | .7719 |
| Collision event | .951 | .933 | .958 | .958 | .955 | .927 | .966 | .951 |
| Time to collision | .825 | .802 | .828 | .827 | .820 | .801 | .823 | .801 |
| Nearest-object distance | .374 | .299 | .374 | .373 | .361 | .294 | .365 | .295 |
| **Map-based score** ↑ | .8663 | .8225 | .8658 | .8660 | .8590 | .8105 | .8615 | .8185 |
| Traffic-light event | .948 | .912 | .948 | .949 | .943 | .905 | .941 | .895 |
| Road-edge distance | .651 | .568 | .647 | .647 | .633 | .561 | .633 | .555 |
| Offroad event | .933 | .905 | .934 | .934 | .930 | .888 | .936 | .912 |
| Collision rate (%) ↓ | 5.50 | 4.00 | 4.70 | 4.70 | 4.50 | 3.90 | 3.90 | 3.40 |
| Offroad rate (%) ↓ | 12.50 | 11.88 | 12.50 | 12.50 | 12.20 | 11.58 | 11.40 | 11.30 |
| Traffic-violation rate (%) ↓ | 3.90 | 2.10 | 3.50 | 3.50 | 3.20 | 2.30 | 3.80 | 3.20 |
| ADE (m) ↓ | 3.999 | 3.440 | 3.881 | 3.867 | 4.032 | 3.549 | 4.301 | 3.958 |
| minADE (m) ↓ | 1.963 | 3.043 | 1.985 | 1.979 | 2.254 | 3.343 | 2.371 | 2.749 |

**CRAFT preserves multimodal diversity.** Table 9 reports diversity across all scenarios and compliant scenarios, using 32 simulated futures per scene and CAT-K as the base simulator for CRAFT. We report mean ± standard deviation in meters for pairwise endpoint distance (PED) and longitudinal/lateral endpoint standard deviation (Long./Lat.). Across the two groups, CRAFT generally has lower endpoint dispersion than the baselines, but its values remain within a comparable range. These results suggest that CRAFT preserves substantial multimodal diversity while improving behavioral rationality.

Table 9: Diversity across all and compliant scenarios.

| | All scenarios | | | Compliant scenarios | | |
|---|---|---|---|---|---|---|
| Method | PED | Long. | Lat. | PED | Long. | Lat. |
| SMART | $8.98 \pm 5.51$ | $7.14 \pm 4.56$ | $2.21 \pm 3.46$ | $8.42 \pm 6.73$ | $4.84 \pm 4.73$ | $1.46 \pm 2.88$ |
| CAT-K | $9.54 \pm 5.77$ | $7.47 \pm 4.62$ | $2.54 \pm 3.78$ | $8.45 \pm 7.08$ | $4.67 \pm 4.77$ | $1.57 \pm 3.22$ |
| RLFTSim | $8.54 \pm 5.09$ | $6.66 \pm 4.16$ | $2.11 \pm 3.27$ | $7.80 \pm 6.17$ | $3.83 \pm 4.22$ | $1.10 \pm 2.45$ |
| CRAFT | $8.24 \pm 5.76$ | $6.39 \pm 4.41$ | $2.32 \pm 3.80$ | $7.30 \pm 7.38$ | $3.61 \pm 3.82$ | $0.90 \pm 2.53$ |

### D.5 Human Evaluation

To assess whether the behavioral improvements also translate into perceived scene realism, we conduct a human evaluation following the paired-comparison protocol of TrafficRLHF (Cao et al., 2024). We randomly sample 300 initial scene states from WOMD and generate one closed-loop rollout with CAT-K and one with CRAFT from each identical initial state. Three evaluators independently assess all 300 pairs, selecting CAT-K as more realistic, CRAFT as more realistic, or indistinguishable when they cannot reliably distinguish the realism of the two scenes.

Table 10: Human preferences for simulated scene realism.

| Evaluator | CAT-K preferred | CRAFT preferred | Indistinguishable |
|---|---|---|---|
| Evaluator 1 | 44 | 146 | 110 |
| Evaluator 2 | 74 | 112 | 114 |
| Evaluator 3 | 53 | 145 | 102 |
| Overall (%) | 19.0 | 44.8 | 36.2 |

Table 10 reports vote counts for each evaluator and percentages pooled over all 900 judgments. Every evaluator favors CRAFT more often than CAT-K, with an overall preference rate of 44.8% versus 19.0%. This consistent preference supports improved perceived scene realism, providing evidence of the benefits of contextual correction.

# E Closed-Loop Recovery and Stability

Figure 12 examines recovery from an initial perturbation and stability during a sharp turn. In the first case, CAT-K drifts offroad and enters an opposing lane, whereas CRAFT returns the agent toward the lane center and sustains normal driving. In the second, CRAFT keeps agents aligned with the road where CAT-K produces wrong-way and offroad motion. These examples illustrate how correcting current actions can also improve subsequent simulation states, interrupting error accumulation and supporting continued reasonable motion.

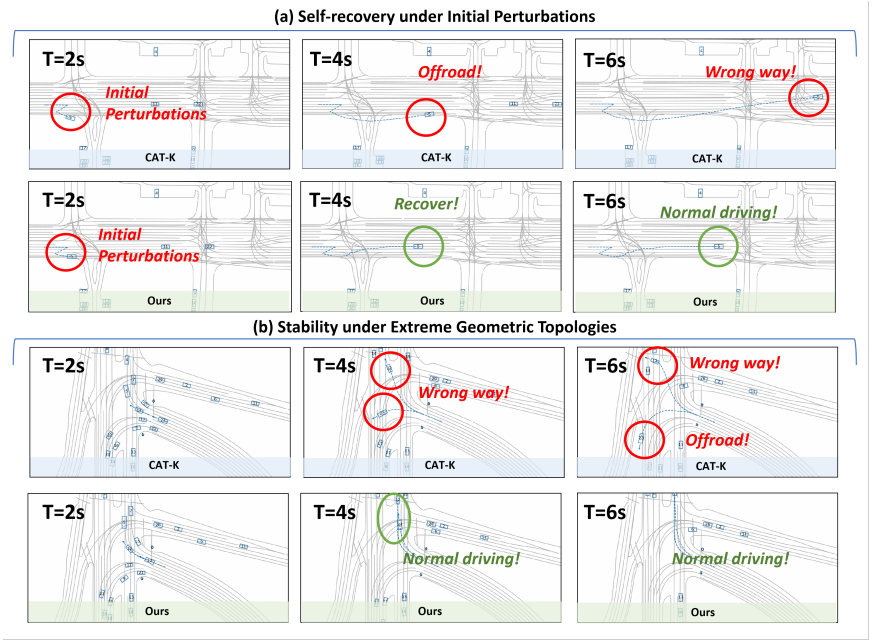



Figure 12: Closed-loop recovery and stability with CRAFT.

# F Limits of Test-Time Guidance

## F.1 A missing candidate limits correction

Figure 13 illustrates a failure near a junction: CAT-K stops unnecessarily, while CRAFT continues only slowly. A token supporting faster progress lies outside the candidate pool. CPE favors a better available action but cannot select the excluded token supporting faster progress.

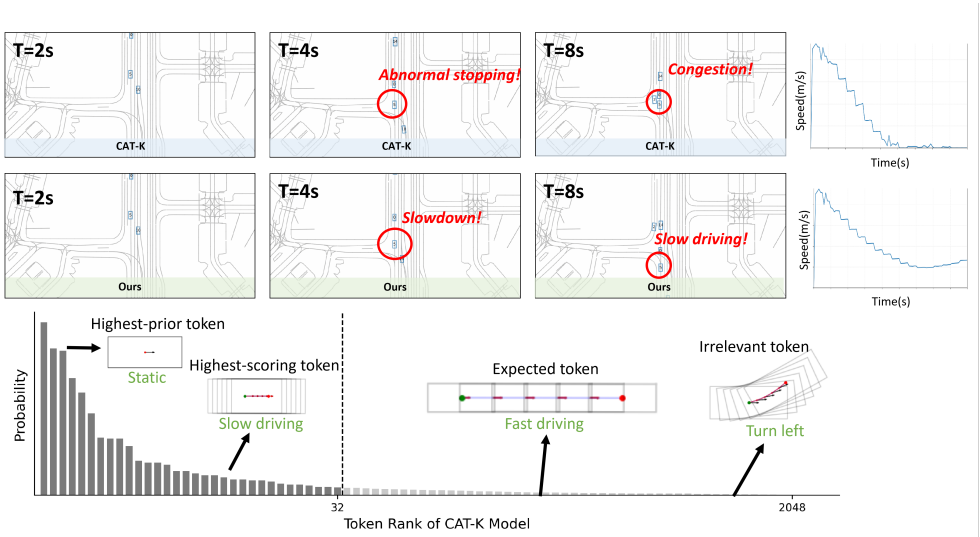



Figure 13: Failure caused by limited candidate support.

**Open-loop coverage saturates with pool size.** Figure 14 measures how often the logged motion token appears in the base simulator's Top-$K$ pool. Recall increases with pool size and is already 95.14% at the default setting, indicating broad coverage of demonstrated actions. Further expansion yields progressively smaller gains. This recall measures whether the logged action is available for selection at evaluated log positions.

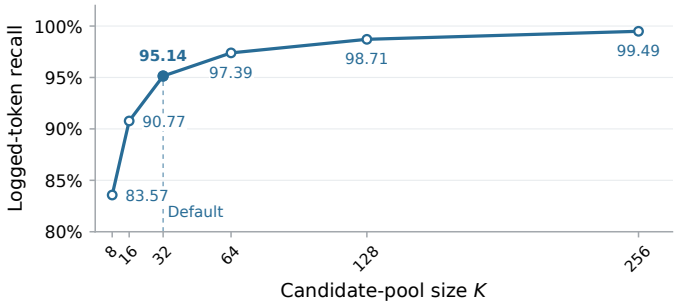



Figure 14: Open-loop logged-token recall on CAT-K.

**CRAFT benefits from a moderate candidate pool.** Table 11 reports P-Sc event rates (%) for Base and CRAFT at matched $K$. The base simulator becomes more collision-prone as the pool expands. In contrast, CRAFT's collision, offroad, and traffic-violation rates all decrease as the pool grows toward $K = 32$, showing that additional candidates can benefit contextual correction. Beyond this setting, CRAFT's event rates rise despite increasing logged-token recall. Across the tested pool sizes, CRAFT has lower event rates than the matched base in all comparisons except collision at $K = 8$. CRAFT thus mitigates much of the degradation associated with larger pools, while a moderate pool provides the best tested balance between candidate coverage and effective behavioral correction.

Table 11: Closed-loop effects of candidate-pool size on CAT-K.

| | Collision ↓ | | Offroad ↓ | | Traffic ↓ | |
|---|---|---|---|---|---|---|
| $K$ | Base | CRAFT | Base | CRAFT | Base | CRAFT |
| 8 | 12.70 | 13.30 | 19.10 | 18.70 | 34.70 | 32.70 |
| 16 | 13.90 | 12.10 | 18.10 | 17.30 | 36.10 | 31.20 |
| 32 | 15.70 | **10.80** | 19.50 | **16.60** | 36.50 | **26.60** |
| 64 | 19.40 | 16.20 | 21.00 | 19.10 | 36.80 | 35.10 |
| 128 | 24.70 | 19.80 | 23.50 | 23.20 | 36.40 | 33.40 |

### F.2 COMPUTATIONAL SCOPE

We profile on four NVIDIA RTX 5090 GPUs with $K = 32$ and batch size 12. Candidate evaluation accounts for only part of the simulation cost. At the largest profiled chunk, lookahead takes approximately 4.0 ms, whereas the complete simulation step takes 43.1 ms (Table 12). Larger chunks reduce latency by evaluating more candidates concurrently, but increase memory consumption. The full-step latency at this chunk size is below the 100 ms budget of a 10 Hz simulator, demonstrating that CRAFT supports real-time closed-loop simulation under the profiled configuration.

Table 12: Latency and memory versus candidate chunk size.

| | Full step | | Lookahead | |
|---|---|---|---|---|
| Chunk size $K_c$ | Latency (ms) | Memory (MiB) | Latency (ms) | Memory (MiB) |
| 4 | 131.166 | 294.8 | 11.642 | 48.8 |
| 8 | 72.719 | 598.3 | 6.981 | 107.5 |
| 16 | 49.265 | 1306.8 | 4.929 | 175.7 |
| 32 | 43.108 | 4074.4 | 3.974 | 345.3 |