2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

ForSim: Stepwise Forward Simulation for Traffic Policy Fine-Tuning

Keyu Chen[1], Wenchao Sun[1], Hao Cheng[1], Zheng Fu[1], Sifa Zheng[1]

***Abstract*— As the foundation of closed-loop training and evaluation in autonomous driving, traffic simulation still faces two fundamental challenges: covariate shift introduced by open-loop imitation learning and limited capacity to reflect the multi-modal behaviors observed in real-world traffic. Although recent frameworks such as RIFT have partially addressed these issues through group-relative optimization, their forward simulation procedures remain largely non-reactive, leading to unrealistic agent interactions within the virtual domain and ultimately limiting simulation fidelity. To address these issues, we propose *ForSim*, a stepwise closed-loop forward simulation paradigm. At each virtual timestep, the traffic agent propagates the virtual candidate trajectory that best spatiotemporally matches the reference trajectory through physically grounded motion dynamics, thereby preserving multimodal behavioral diversity while ensuring intra-modality consistency. Other agents are updated with stepwise predictions, yielding coherent and interaction-aware evolution. When incorporated into the RIFT traffic simulation framework, *ForSim* operates in conjunction with group-relative optimization to fine-tune traffic policy. Extensive experiments confirm that this integration consistently improves safety while maintaining efficiency, realism, and comfort. These results underscore the importance of modeling closed-loop multimodal interactions within forward simulation and enhance the fidelity and reliability of traffic simulation for autonomous driving. Project Page: https://currychen77.github.io/ForSim/**

I. Introduction

Traffic simulation forms the foundation of modern autonomous driving, providing the basis for closed-loop training and evaluation [1]. It requires simulating multi-agent behavior over time in a way that reflects real-world driving behaviors within a closed-loop environment. This introduces two central challenges: *covariate shift* and *multimodality*.

Covariate shift is a long-standing issue in open-loop imitation learning (IL), where models trained on offline expert demonstrations often struggle when deployed in closed-loop settings due to compounding errors and distributional drift. A classical solution is online learning [2], which unrolls the policy and queries an expert to generate new demonstrations [3]. However, in multi-agent traffic simulation, frequent expert querying becomes prohibitively expensive and unscalable. To address this challenge, recent studies integrate imitation learning (IL) with reinforcement learning (RL) under a closed-loop paradigm [4]–[6]. This hybrid approach leverages the complementary strengths of the two paradigms: IL provides sample-efficient initialization by distilling expert demonstrations, while RL enables policies to adapt through interaction-driven feedback in closed-loop execution. By jointly exploiting these advantages, such methods reduce covariate shift by grounding policy updates in causal links between observations, actions, and rewards, while leveraging expert data to accelerate the learning of realistic driving behaviors. In contrast to IL-RL hybrid approaches, CAT-K [7] adopts a purely supervised strategy through closed-loop fine-tuning, aligning each rollout with the closest among top-k expert trajectories. This alignment preserves the rollout's consistency with ground-truth (GT) demonstrations, thereby maintaining valid supervision signals throughout the closed-loop process. However, despite this improvement, such methods typically rely on optimizing the policy using only the most likely rollout stored in the buffer, which corresponds to a single behavior modality over the rollout horizon. Consequently, they struggle to capture the multimodal driving behaviors in real-world traffic. Addressing this limitation requires not only a dataset with coverage of diverse driving modalities but also a policy capable of maintaining exploration during closed-loop execution to avoid mode collapse. This places substantial demands on both data diversity and the multi-modal modeling capacity of the policy.

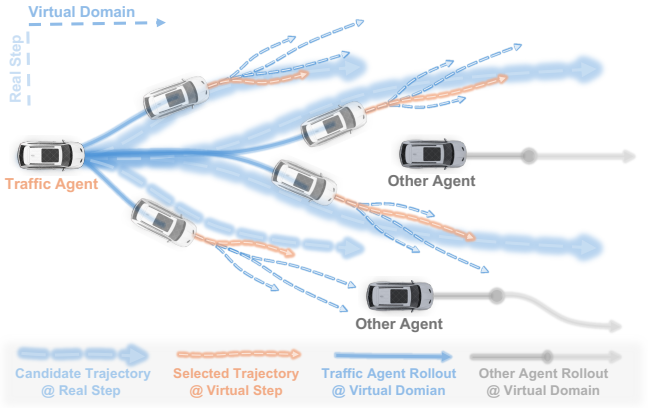

Fig. 1. ***ForSim*** introduces stepwise unrolling of multimodal candidate trajectories under closed-loop dynamics. At each virtual timestep, the traffic agent selects the spatiotemporal aligned trajectory to preserve modality consistency, while propagation through the PID controller and kinematic bicycle model ensures physical plausibility. Other agents follow stepwise predictions, ensuring interactive and coherent evolution.

To leverage the strengths of both IL and RL, RIFT [8] first performs open-loop IL pre-training on real-world datasets to capture realistic driving behaviors, and then fine-tunes the policy via group-relative optimization to mitigate covariate shift while preserving multimodality. However, RIFT conducts non-reactive forward simulations: only the first step is closed-loop, while subsequent steps are rolled out in open-loop across all candidate trajectories. This leads to unrealistic agent interactions in the virtual time domain, limiting the

*This work was supported by the National Natural Science Foundation of China under Grant 52572497 and Grant 52221005.

[1]Keyu Chen, Wenchao Sun, Hao Cheng, Zheng Fu, Sifa Zheng are with the School of Vehicle and Mobility, Tsinghua University, Beijing, China.

fidelity of the forward simulation.

Building on these observations, we introduce *ForSim*, a stepwise closed-loop forward simulation paradigm designed to capture the interaction within the virtual forward simulation. In the real-time domain, each traffic agent generates multiple candidate trajectories through the traffic policy, representing distinct plausible modalities. *ForSim* performs forward simulation independently for each candidate, treating it as a reference trajectory in an associated virtual domain. At every virtual timestep, the agent propagates the trajectory that best spatiotemporally aligns with this reference, ensuring intra-modality consistency throughout the rollout. Propagation adheres to physically grounded dynamics, ensuring physical plausibility and dynamic coherence. Other agents are propagated along predicted trajectories that are updated at each virtual timestep, ensuring responsiveness to evolving interactions. This iterative procedure guarantees that all candidate modalities evolve coherently under closed-loop conditions, simultaneously preserving multimodal behavioral diversity, intra-modality consistency, and physical plausibility. The resulting multimodal rollouts are evaluated in a stepwise fashion within the RIFT framework, enabling fine-grained optimization of traffic policy through group-relative objectives.

Our contributions are summarized as follows:

- We propose *ForSim*, a stepwise closed-loop forward simulation paradigm that ensures behavioral multimodality, intra-modality consistency, and physical plausibility, thereby enabling fine-grained optimization of traffic policy within the RIFT framework.
- We demonstrate that *ForSim* outperforms existing forward simulation paradigms by modeling closed-loop multimodal interactions, while further enhancing the realism and reliability of traffic simulation.

II. Related Work

Traffic Simulation. Recent advances in traffic simulation have explored diverse generative architectures—including VAEs [9], [10], diffusion models [11], [12], and next-token prediction [13], [14]—to improve realism [15] and long-horizon robustness [16], [17]. At the same time, controllability has been pursued via cost-based conditioning [18], language prompts [19], [20], guided sampling [21], and retrieval-based generation [22], enabling user-aligned traffic scenario generation [23], [24]. Yet, most approaches suffer from the covariate shift problem caused by the mismatch between open-loop training and closed-loop deployment, which further undermines long-term stability. Closed-loop fine-tuning strategies provide partial remedies, yet each comes with trade-offs. Hybrid IL/RL methods [4]–[6] enhance robustness but often compromise realism due to reward design challenges. Supervised approaches such as CAT-K [7] achieve strong performance but depend on scarce expert demonstrations. RLHF [25] improves human alignment but requires costly feedback and suffers from reward model instability. RIFT [8] mitigates covariate shift while preserving multimodality via group-relative optimization, but its reliance on non-reactive forward simulation limits the realism of closed-loop agent interactions within the virtual domain.

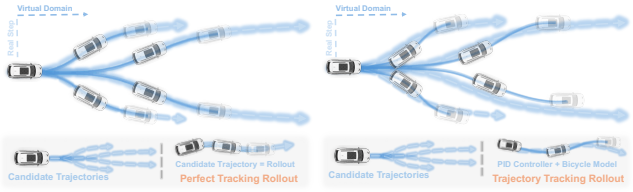

Fig. 2. **Typical rollout paradigms.** The left panel depicts *Perfect Tracking*, in which the vehicle strictly follows the planned trajectory. The right panel depicts *Trajectory Tracking*, which employs a controller and kinematic bicycle model to follow the trajectory under dynamic constraints.

Forward Simulation. In autonomous driving, forward simulation denotes virtually unrolling the future evolution of the center vehicle and other agents conditioned on a given trajectory, serving to evaluate the planned trajectories and bridging the gap between planning and execution. As illustrated in Figure 2, typical rollout paradigms for the center vehicle include *Perfect Tracking*, where the vehicle strictly follows the planned trajectory, and *Trajectory Tracking*, where a controller and kinematic bicycle model [26] are used to follow the trajectory under dynamic constraints. In practice, Trajectory Tracking is the dominant paradigm, and existing approaches primarily diverge in their modeling of responses from other agents. PDM-Closed [27] selects among candidate trajectories by unrolling each one with Trajectory Tracking, while forecasting other agents under a constant-velocity assumption [28]. PlanTF [29] and Pluto [30] integrate forward simulation into training or post-processing to better align planned trajectories and executed rollouts, but assume other agents follow log-replay. NAVSIM [31] achieves pseudo-closed-loop evaluation on the nuPlan dataset [32] by combining trajectory unrolling with log-replay [31] or rule-based updates [33]. Despite these advances, most approaches remain essentially open-loop, unrolling fixed trajectories without replanning and propagating other agents under static forecasts, neglecting dynamic inter-agent interactions. This discrepancy from the real-world plan–execute–replan cycle limits the fidelity and reliability of forward simulation.

III. Methodology

As outlined in Section I, this work aims to mitigate covariate shift in traffic simulation while preserving multimodal diversity and forward simulation reliability. A suitable foundation for this objective is RIFT [8], which couples IL pre-training, capturing realistic behaviors with route-level controllability, with RL fine-tuning in a high-fidelity simulator to reduce covariate shift and enhance style controllability.

Despite these strengths, RIFT adopts a non-reactive forward simulation paradigm, equivalent to PDM-Closed [27], where only the first step is closed-loop while subsequent steps remain open-loop. This neglects inter-agent interactions and limits long-term realism. We address this limitation by refining the forward simulation component to enable closed-loop, interaction-aware rolling of candidate trajectories, thereby ensuring more faithful evaluation of multimodal candidate

trajectories and ultimately improving the realism, reliability, and controllability of traffic simulations.

A. Preliminaries

Following RIFT [8], we construct an AV-centric closed-loop simulation in Carla [34]. The critical background vehicles (CBVs)—those most likely to interact with the AV—are controlled by the traffic policy, and we interchangeably refer to each CBV as a traffic agent in the following discussion. For each traffic agent, the policy takes as input its current feature F_{cbv}, historical features of neighboring vehicles F_{neighbor}, and vectorized map features F_{map}. These are encoded into embeddings $E_{\text{cbv}} \in \mathbb{R}^{1\times D}$, $E_{\text{neighbor}} \in \mathbb{R}^{N_{\text{neighbor}}\times D}$, and $E_{\text{map}} \in \mathbb{R}^{N_{\text{map}}\times D}$, where N_{neighbor} and N_{map} denote the number of neighboring vehicles and map elements, respectively, and D is the embedding dimension. The resulting embeddings are aggregated through Transformer encoder blocks to form the CBV-centric scene embedding E_{enc}.

To capture multimodal driving behaviors, the policy further adopts a longitudinal–lateral decoupling mechanism [30]. Specifically, high-level lateral queries $Q_{\text{lat}} \in \mathbb{R}^{N_{\text{ref}}\times D}$ are constructed from reference line information, while learnable longitudinal queries $Q_{\text{lon}} \in \mathbb{R}^{N_{\text{lon}}\times D}$ represent diverse longitudinal anchors. These are combined into multimodal navigation queries $Q_{\text{nav}} \in \mathbb{R}^{N_{\text{ref}}\times N_{\text{lon}}\times D}$, which interact with the scene embedding E_{enc} through Transformer decoder blocks. The decoder's final output Q_{dec} is passed through MLP heads to generate a set of candidate trajectories $\mathcal{T} \in \mathbb{R}^{N_{\text{ref}}\times N_{\text{lon}}\times T\times 6}$ along with their confidence scores $\mathcal{S} \in \mathbb{R}^{N_{\text{ref}}\times N_{\text{lon}}}$, where each trajectory point τ_i^t encodes $[p_x, p_y, \cos\theta, \sin\theta, v_x, v_y]$. Among these candidates, the trajectory with the highest confidence score is executed by a PID controller, ensuring that CBVs remain in closed-loop interaction with the environment.

In parallel, all candidate trajectories are unrolled through the forward simulation to generate rollouts $\widetilde{\mathcal{T}} = \{\widetilde{\tau}_i\}_{i=1}^{N_{\text{ref}}\times N_{\text{lon}}}$, which form the foundation for modality evaluation and subsequently fine-tune the traffic policy through group-relative optimization. Since our work focuses on forward simulation, the core challenge is to unroll all candidate trajectories in a way that faithfully captures real-world interactions and transitions, thereby providing a reliable foundation for rollout evaluation and policy optimization.

To address this challenge, we decompose the forward simulation into two components: (i) stepwise virtual rollout of the traffic agent and (ii) stepwise prediction propagation of other agents, which together define how candidate trajectories are unrolled under closed-loop interaction.

B. Traffic Agent Stepwise Virtual Rollout

In contrast to Perfect Tracking, which directly reuses the planned trajectory, and Trajectory Tracking, which rigidly tracks a fixed trajectory without adaptation—both illustrated in Figure 2—our approach embraces a *stepwise planning–execution–replanning* process that dynamically adapts to real-time state changes. This framework more faithfully captures real-world driving dynamics, where agents continually revise their plans in response to evolving states and interactions. Concretely, at the initial real-world time step ($t = 0$) we generate $N_{\text{ref}} \times N_{\text{lon}}$ candidate trajectories and forward-simulate each for one step, yielding $N_{\text{ref}} \times N_{\text{lon}}$ rollout branches at virtual time $\tilde{t}$=1. Subsequently, at each virtual step $\tilde{t}$, each branch selects a candidate index $(i_{\tilde{t}}, j_{\tilde{t}})$ based on a specified paradigm and propagates its state by tracking the selected trajectory via a PID controller coupled with a kinematic bicycle model:

$$x_{\tilde{t}+1} = \mathcal{F}_{\text{PID+Bike}}\big(x_{\tilde{t}},\ \tau_{(i_{\tilde{t}},j_{\tilde{t}})}(x_{\tilde{t}})\big), \quad \tilde{t} = 1, \ldots, T, \quad (1)$$

where $x_{\tilde{t}}$ denotes the virtual state at $\tilde{t}$ and $\mathcal{F}_{\text{PID+Bike}}$ represents one-step execution via PID tracking followed by kinematic bicycle propagation. To instantiate this framework, we explore three stepwise rollout paradigms, illustrated in Figure 3: the *Max-Likelihood Rollout*, the *Mode-Consistent Rollout*, and the *Trajectory-Aligned Rollout*.

Max-Likelihood Rollout. At each virtual time step $\tilde{t}$, each branch selects the candidate with the highest confidence score predicted at the current virtual state:

$$(i_{\tilde{t}}, j_{\tilde{t}}) \ = \ \arg\max_{i,j}\ s_{(i,j)}(x_{\tilde{t}}). \quad (2)$$

This strategy ensures that each branch consistently follows the most likely mode throughout the rollout. Nevertheless, this strategy inevitably causes branches to converge toward the dominant mode. Because the policy is robust to small state perturbations, initial deviations introduced by diverse candidates are often ignored, ultimately collapsing the intended multimodal behavior.

Mode-Consistent Rollout. An alternative strategy enforces each branch to remain in the same candidate mode selected at the initial step from $t = 0$ to $\tilde{t} = 1$. Specifically, if a branch is seeded at $\tilde{t} = 1$ with candidate index $(i^{\text{ref}}, j^{\text{ref}})$, then it maintains $(i_{\tilde{t}}, j_{\tilde{t}}) = (i^{\text{ref}}, j^{\text{ref}})$ at all subsequent virtual steps and propagates accordingly via $\mathcal{F}_{\text{PID+Bike}}$ accordingly. While this approach guarantees explicit mode consistency, it introduces misalignment: trajectories corresponding to the same candidate mode at different virtual states are not perfectly coherent, and the resulting accumulation of deviations can cause unintended mode shifts—ultimately compromising rollout stability.

Trajectory-Aligned Rollout. To address the limitations of the previous strategies, we propose the Trajectory-Aligned paradigm. Each branch is still initialized at virtual time $\tilde{t}$=1 using a specific candidate index $(i^{\text{ref}}, j^{\text{ref}})$, corresponding to the trajectory selected during the initial transition from t=0 to $\tilde{t}$=1. This reference trajectory remains fixed throughout the rollout. At every subsequent virtual step $\tilde{t} \geq 2$, the branch dynamically selects the candidate whose trajectory—after temporal alignment—is closest to the fixed reference, as measured by the Average Displacement Error (ADE):

$$(i_{\tilde{t}}, j_{\tilde{t}}) \ = \ \arg\min_{(i,j)}\ \text{ADE}_{\text{align}}\Big(\tau_{(i,j)}(x_{\tilde{t}}),\ \tau_{(i^{\text{ref}},j^{\text{ref}})}\Big), \quad (3)$$

where $\text{ADE}_{\text{align}}$ denotes ADE after temporal alignment of the reference trajectory to the virtual evaluation window.

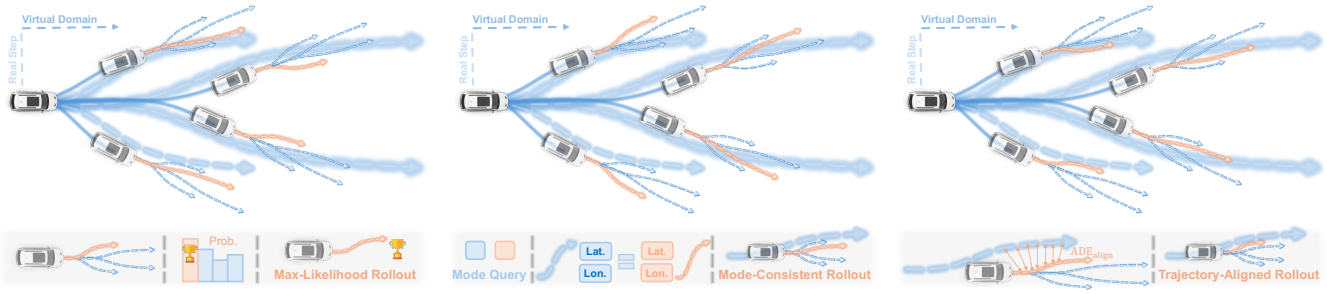

Fig. 3. **Illustration of three stepwise rollout paradigms for traffic agents:** Max-Likelihood Rollout, Mode-Consistent Rollout, and Trajectory-Aligned Rollout. Max-likelihood Rollout rapidly collapses multimodal diversity, while Mode-Consistent Rollout suffers from accumulated misalignment across virtual states. Trajectory-Aligned Rollout, in contrast, selects at each step the candidate trajectory closest to its initial mode—measured by Average Displacement Error (ADE)—thereby preserving multimodal fidelity and yielding physically coherent rollouts.

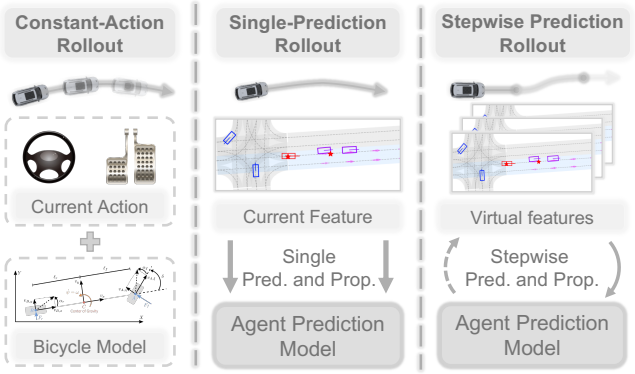

Fig. 4. **Other agents rollout paradigms:** Constant-Action Rollout propagates the current action in an open-loop manner; Single-Prediction Rollout yields more plausible trajectories but remains open-loop; and Stepwise Prediction Rollout enables closed-loop, interaction-aware rollout.

This formulation enforces fidelity to the original modality while avoiding mode collapse toward the Max-Likelihood Rollout and the temporal drift inherent in Mode-Consistent Rollout. Coupled with state propagation through $\mathcal{F}_{\text{PID+Bike}}$, Trajectory-Aligned Rollout preserves multimodal consistency and guarantees rollouts that are both physically plausible and dynamically coherent, making it the most effective paradigm.

C. Other Agents Stepwise Prediction Rollout

In addition to unrolling the traffic agent, accurately simulating other agents is critical for realistic interactions. We explore three paradigms, illustrated in Figure 4.

Constant-Action Rollout. The most commonly adopted baseline assumes that each other agent maintains its current action throughout the entire rollout horizon. The recorded control input at the current time step is propagated forward using a kinematic bicycle model. While this method ensures physical feasibility, it operates in a purely open-loop fashion and fails to capture interaction-induced behavior shifts, often resulting in unrealistic agent responses.

Single-Prediction Rollout. A more advanced alternative utilizes a predictor to forecast agent trajectories based on the current states. This predicted trajectory is then directly propagated through time. Compared to Constant-Action Rollout, this method yields more plausible estimates of future states. Since predictions are generated only at initialization, the simulation remains open-loop and fails to capture interactive behaviors in response to evolving traffic dynamics.

Stepwise Prediction Rollout. To address these limitations, we employ a Stepwise Prediction paradigm in which other agents update their predicted trajectories at every virtual time step, conditioning on the evolving virtual state. Each updated prediction is then propagated forward one step. This design synchronizes the planning frequency of other agents with that of the traffic agent, enabling responsive behaviors and interaction-aware updates. Consequently, it facilitates closed-loop dynamics and significantly improves the realism of inter-agent interactions over long horizons.

At the core of this paradigm lies an agent prediction model that produces horizon-T trajectories for all other agents, denoted as $\hat{\mathbf{y}} \in \mathbb{R}^{N\times T\times 6}$, where each state encodes the agent's position, orientation, and velocity. Ground-truth supervision derives from future observations over the first T_f steps, forming the target trajectories $\mathbf{y}^\star$. To balance fidelity to ground-truth data with physical realism, we integrate multiple complementary loss terms into a composite objective:

$$\mathcal{L}_{\text{pred}} = w_{\text{anchor}}\mathcal{L}_{\text{anchor}} + w_{\text{kin}}\mathcal{L}_{\text{kin}} + w_{\text{smooth}}\mathcal{L}_{\text{smooth}}, \quad (4)$$

where the individual loss terms are defined as

$$\mathcal{L}_{\text{anchor}} = \text{SmoothL1}\left(\hat{\mathbf{y}}_{0:T_f-1}, \mathbf{y}^\star\right), \quad (5)$$

$$\mathcal{L}_{\text{kin}} = \text{SmoothL1}\left(\mathbf{v}_t, \frac{\mathbf{p}_{t+1}-\mathbf{p}_{t-1}}{2\Delta t}\right), \quad (6)$$

$$\mathcal{L}_{\text{smooth}} = \|\Delta\mathbf{v}_t\|_1 + \|\Delta^2\mathbf{v}_t\|_1, \quad (7)$$

with $\mathbf{p}_t = (x_t, y_t)$, $\mathbf{v}_t = (v_t^x, v_t^y)$, $\Delta\mathbf{v}_t = \mathbf{v}_{t+1} - \mathbf{v}_t$, and $\Delta^2\mathbf{v}_t = \Delta\mathbf{v}_{t+1} - \Delta\mathbf{v}_t$. This objective balances short-term accuracy, kinematic consistency, and temporal smoothness, resulting in predictions that are both physically plausible and stable for long-horizon closed-loop simulations. We set $w_{\text{anchor}} = 1.0$, $w_{\text{kin}} = 0.5$, and $w_{\text{smooth}} = 0.5$.

D. Group-Relative Policy Fine-Tuning

Building on the results of forward simulation, we fine-tune the traffic policy in a closed-loop manner by leveraging the group-relative optimization principle introduced in RIFT [8]. For each traffic agent's real-world state s, $G = N_{\text{ref}} \times N_{\text{lon}}$ candidate trajectories are forward-simulated to generate rollout branches, which are subsequently evaluated using a stepwise reward model to obtain rollout-level returns $\mathcal{R} = \{r_i\}_{i=1}^{G}$. Formally, given a set of rollouts $\widetilde{\mathcal{T}} = \{\widetilde{\tau}_i\}_{i=1}^{G}$,

Algorithm 1 Closed-Loop Traffic Policy Optimization

Input: IL pre-trained traffic policy $\pi_{\theta_{\text{init}}}$, IL pre-trained predictor $\mathcal{M}_{\text{pred}}$, buffer $\mathcal{D}$
Policy $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration $= 1, \ldots, I$ **do** ▷ RL fine-tuning
  Update the old policy $\pi_{\theta_{old}} \leftarrow \pi_\theta$
  while $\mathcal{D}$ not full **do** ▷ Collect rollout data
    for step $= 1, \ldots, T$ **do**
      Generate $\{\tau_i\}_{i=1}^{G}$ from $\pi_{\theta_{old}}$ ▷ Policy inference
      Derive $\{\widetilde{\tau}_i\}_{i=1}^{G}$ via Alg. 2 ▷ Forward simulation
      Compute $\{r_i\}_{i=1}^{G}$, $\{\hat{A}_i\}_{i=1}^{G}$ for each $\widetilde{\tau}_i$ with Eq. (8)
      Store transition into buffer $\mathcal{D}$
    end for
  end while
  for *ForSim* iteration $= 1, \ldots, \mu$ **do** ▷ Fine-tuning
    Sample mini-batch transitions from the buffer $\mathcal{D}$
    Update predictor $\mathcal{M}_{\text{pred}}$ via Eq. (4) ▷ Train predictor
    Update traffic policy π_θ via Eq. (9) ▷ Train policy
  end for
end for
Output: RL fine-tuned traffic policy

Algorithm 2 *ForSim*: Stepwise Forward Simulation

Input: Reference trajectories $\{\tau_i\}_{i=1}^{G}$, traffic policy $\pi_{\theta_{\text{init}}}$, predictor $\mathcal{M}_{\text{pred}}$
for $\tau_i \in \{\tau_i\}_{i=1}^{G}$ **do**
  Propagate τ_i via $\mathcal{F}_{\text{PID+Bike}}$ to obtain x_1^{center}
  Predict other agents via $\mathcal{M}_{\text{pred}}$ to obtain x_1^{agents}
  Record initial rollout state $x_1 = (x_1^{\text{center}}, x_1^{\text{agents}})$
  for Virtual step $\tilde{t} = 1, \ldots, T$ **do** ▷ Stepwise rollout
    Generate trajectories from $x_{\tilde{t}}$, select via Eq. (3), then propagate via Eq. (1) to obtain $x_{\tilde{t}+1}^{\text{center}}$ ▷ traffic agent
    Predict $x_{\tilde{t}+1}^{\text{agents}}$ via $\mathcal{M}_{\text{pred}}(x_{\tilde{t}})$ ▷ other agents
    Record next rollout state $x_{\tilde{t}+1} = (x_{\tilde{t}+1}^{\text{center}}, x_{\tilde{t}+1}^{\text{agents}})$
  end for
end for
Output: Rollouts $\{\widetilde{\tau}_i\}_{i=1}^{G}$

we define:

$$r_i = \sum_{t=0}^{T} \gamma^t \left[\text{RM}(\widetilde{\tau}_i^t)\right], \hat{A}_i = \frac{r_i - \text{mean}(\mathcal{R})}{\text{std}(\mathcal{R})}, \forall i = 1, \ldots, G \quad (8)$$

where $\hat{A}_i$ represents its normalized group-relative advantage. This normalization promotes high-return behaviors while preserving the multimodality inherent in the candidate set.

To update the policy, we adopt the group-relative optimization framework of RIFT, which employs a dual-clip mechanism to bound policy updates, stabilize training, and preserve alignment with user-preferred styles. The resulting optimization objective is:

$$\mathcal{J}(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G} \psi(\rho_i(\theta),\, \hat{A}_i)\right], \quad \rho_i(\theta) = \frac{\pi_\theta(\tau_i|s)}{\pi_{\theta_{\text{old}}}(\tau_i|s)},$$
$$\psi(\rho, \hat{A}) = \begin{cases} \min\big(\rho\,\hat{A},\ \text{clip}(\rho, 1-\epsilon, 1+\epsilon)\,\hat{A}\big), & \hat{A} \geq 0, \\ \max\Big(\min\big(\rho\,\hat{A},\ \text{clip}(\rho, 1-\epsilon, 1+\epsilon)\,\hat{A}\big),\ c\,\hat{A}\Big), & \hat{A} < 0 \end{cases}, \quad (9)$$

where $\rho_i(\theta)$ denotes the trajectory-level likelihood ratio between the updated and old traffic policies.

In this manner, the traffic policy is fine-tuned directly on forward-simulated rollouts, enabling improved controllability, mitigating covariate shift, and enhancing training stability in a closed-loop setting. The overall training pipeline (Algorithm 1) repeatedly invokes the forward simulation module (Algorithm 2) to generate virtual rollouts for policy updates.

IV. Experiment

In this section, we empirically evaluate the effectiveness of *ForSim* in enhancing the realism and controllability of traffic simulations. Our experiments are designed to answer the following key research questions: **Q1**: Does the closed-loop forward simulation in *ForSim* improve traffic simulation quality compared to baseline approaches? **Q2**: What are the individual contributions of the rollout paradigms within *ForSim* to the overall performance? **Q3**: Why do the proposed rollout paradigms work effectively in practice?

We begin by outlining the experimental setup, including the simulation platform, evaluation metrics, and baselines. We then conduct a comprehensive empirical comparison, demonstrating the superiority of *ForSim* under various quantitative and qualitative criteria. Finally, we analyze the design of the rollout paradigms in *ForSim*, providing insights into why they outperform existing approaches and how they contribute to improved closed-loop simulation fidelity.

A. Experimental Setup

a) Simulation Environment: All experiments are conducted in the CARLA simulator [34], following the RIFT framework [8], which facilitates AV-centric traffic simulation by assigning learned traffic policy to critical background vehicles (CBVs) that may interact with the autonomous vehicle (AV) along its designated route. RIFT adopts a two-stage learning pipeline: trajectory-level realism and route-level controllability are established through open-loop IL pretraining, while style-level diversity and covariate shift mitigation are subsequently addressed via RL fine-tuning.

To isolate the impact of rollout paradigms, the overall training pipeline, including the reward formulation, learning protocol, and evaluation procedure, is kept consistent with RIFT. The only modification lies in the forward simulation component, which is replaced by our proposed stepwise closed-loop paradigm. This controlled design enables an isolated and systematic examination of how rollout paradigms affect the realism and controllability of traffic flow.

b) Baselines: To ensure fair comparison, we adopt the same baselines as RIFT, including RIFT itself as a strong reference. All methods are fine-tuned on the scoring head to isolate the effect of forward simulation. For prediction-based rollouts, the prediction head is also fine-tuned to ensure consistency between rollout and policy optimization.

Pure RL/IL: Models trained exclusively via RL or IL, including *Pluto* [30] and *FREA* [24].

RLFT/SFT: Fine-tuning methods based on a pre-trained *Pluto* model, including *RIFT-Pluto*, *PPO-Pluto*, *REINFORCE-Pluto*, *GRPO-Pluto*, and *SFT-Pluto*.

Hybrid: Approaches that combine reinforcement and supervised fine-tuning, including *RTR-Pluto* and *RS-Pluto*.

All RL-based methods are trained under the normal style reward, following the standard configuration defined in RIFT.

TABLE I

COMPARISON IN CONTROLLABILITY AND REALISM. METRICS ARE EVALUATED UNDER THE PDM-LITE [35] AV SETTING ACROSS THREE RANDOM SEEDS, WITH THE **BEST** AND THE **SECOND-BEST** RESULTS HIGHLIGHTED ACCORDINGLY.

Method	Type	Kinematic			Interaction				Map	Comfort
		S-SW ↑	S-WD ↓	A-SW ↑	CPK ↓	RP ↑	2D-TTC ↑	ACT ↑	ORR ↓	UC ↓
Pluto [30]	IL	0.88 ± 0.01	5.81 ± 0.06	0.90 ± 0.01	5.06 ± 2.69	564.14 ± 114.41	2.50 ± 1.48	2.44 ± 1.39	0.24 ± 0.15	**56.45** ± 4.14
FREA [24]	RL	0.93 ± 0.01	5.10 ± 0.14	0.93 ± 0.01	30.42 ± 5.28	292.81 ± 68.54	2.71 ± 1.40	2.67 ± 1.41	9.01 ± 2.09	72.40 ± 1.72
SFT-Pluto	SFT	0.88 ± 0.02	6.01 ± 0.19	0.87 ± 0.02	6.33 ± 2.23	780.48 ± 41.05	2.20 ± 1.64	2.12 ± 1.51	**0.06** ± 0.07	68.14 ± 4.91
RS-Pluto [6]	SFT+RLFT	0.93 ± 0.00	5.40 ± 0.15	0.92 ± 0.01	**4.11** ± 3.90	819.40 ± 74.07	2.27 ± 1.45	2.23 ± 1.43	1.05 ± 0.31	70.31 ± 4.07
RTR-Pluto [5]	SFT+RLFT	0.85 ± 0.00	6.24 ± 0.16	0.81 ± 0.03	6.98 ± 2.59	481.60 ± 70.19	2.55 ± 1.60	2.47 ± 1.51	0.08 ± 0.09	**55.58** ± 4.76
PPO-Pluto	RLFT	0.95 ± 0.01	4.96 ± 0.31	0.90 ± 0.02	6.89 ± 3.19	683.57 ± 38.12	2.66 ± 1.50	2.60 ± 1.43	**0.07** ± 0.13	58.29 ± 2.70
REINFORCE-Pluto	RLFT	0.92 ± 0.01	5.63 ± 0.19	0.90 ± 0.02	6.98 ± 0.86	813.70 ± 24.76	2.39 ± 1.64	2.30 ± 1.55	1.37 ± 1.13	68.10 ± 1.22
GRPO-Pluto [36]	RLFT	0.94 ± 0.04	4.96 ± 0.89	**0.96** ± 0.00	7.24 ± 4.04	892.65 ± 65.27	2.65 ± 1.44	2.61 ± 1.48	0.10 ± 0.08	78.58 ± 0.59
RIFT-Pluto [8]	RLFT	**0.97** ± 0.01	**4.46** ± 0.43	0.93 ± 0.01	6.83 ± 2.62	**995.33** ± 84.62	**2.74** ± 1.30	**2.71** ± 1.32	0.36 ± 0.20	76.90 ± 2.82
ForSim-Pluto (ours)	RLFT	**0.96** ± 0.01	**4.31** ± 0.72	**0.94** ± 0.01	**3.95** ± 1.57	**1005.15** ± 32.28	**2.95** ± 1.40	**2.92** ± 1.46	0.34 ± 0.31	57.12 ± 3.03

TABLE II

ABLATION OVER CENTER ROLLOUT PARADIGMS. ENTRIES WITH * USE CONSTANT-ACTION FOR SURROUNDING AGENTS.

Paradigms	Efficiency Metrics		Infraction Metrics	
	BR ↓	RP ↑	ORR ↓	CPK ↓
Perfect Tracking*	**0.00** ± 0.00	993.63 ± 48.01	**0.26** ± 0.22	**4.00** ± 0.83
Trajectory Tracking*	3.33 ± 5.77	**1087.56** ± 36.41	0.67 ± 0.57	4.00 ± 1.47
Max-Likelihood	**0.00** ± 0.00	910.97 ± 9.51	**0.11** ± 0.05	4.76 ± 3.53
Mode-Consistent	3.33 ± 5.77	940.94 ± 73.70	0.28 ± 0.29	4.55 ± 1.34
Trajectory-Aligned	**0.00** ± 0.00	**1005.15** ± 32.28	0.34 ± 0.31	**3.95** ± 1.57

TABLE III

ABLATION OVER OTHERS ROLLOUT PARADIGMS.

Paradigms	Efficiency Metrics		Infraction Metrics	
	BR ↓	RP ↑	ORR ↓	CPK ↓
Constant-Action	**0.00** ± 0.00	**1128.77** ± 96.24	**0.91** ± 0.30	4.97 ± 1.57
Single-Prediction	**0.00** ± 0.00	**1170.28** ± 29.17	2.31 ± 0.29	**3.41** ± 1.44
Stepwise Prediction	**0.00** ± 0.00	1005.15 ± 32.28	**0.34** ± 0.31	**3.95** ± 1.57

B. Traffic Simulation Quality (Q1)

a) Metrics: Following the WOSAC evaluation framework [37], we assess simulation quality using four categories of metrics: *kinematic*, *interaction*, *map*, and *comfort*.

Kinematic metrics evaluate the distributional realism of agent motion, including the Shapiro–Wilk test on speed (S-SW) and acceleration (A-SW), and the Wasserstein distance on speed (S-WD).

Interaction metrics include Collision per Kilometer (CPK), Route Progress (RP), and safety-critical indicators: 2D-TTC [38] and ACT [39].

Map metric measures spatial compliance via the Off-Road Rate (ORR), indicating time spent outside drivable areas.

Comfort metric captures driving smoothness via the Uncomfortable Rate (UC), defined as the proportion of time acceleration exceeds comfort thresholds, following RIFT [8].

b) Main Results and Analysis: *ForSim* demonstrates marked improvements in safety-related metrics, notably reducing CPK, 2D-TTC, and ACT, as presented in Table I. These results highlight the effectiveness of modeling closed-loop interactions, which enable agents to anticipate and respond to dynamic traffic conditions more effectively. Importantly, this enhanced safety is achieved without compromising efficiency or realism: kinematic characteristics and RP are preserved or even marginally improved relative to RIFT. At the same time, a reduced ORR suggests more spatially compliant behavior. Additionally, *ForSim* improves comfort substantially, benefiting from the use of the PID controller and the kinematic bicycle model that ensures physical plausibility.

C. Rollout Paradigm Analysis (Q2 & Q3)

a) Ablation over Rollout Paradigms: We conduct ablation studies over both traffic agent and other agent rollout paradigms, as shown in Tables II and III. For the traffic agent (Table II), we compare three closed-loop variants: Max-Likelihood Rollout, Mode-Consistent Rollout, and Trajectory-Aligned Rollout, while fixing other agents to the Stepwise Prediction Rollout. In contrast, Perfect Tracking and Trajectory Tracking bypass the traffic policy during forward simulation and are unaffected by other agent transitions. We therefore retain the standard Constant-Action Rollout for other agents, consistent with prior work [27].

Among traffic agent paradigms, the Trajectory-Aligned Rollout offers the most favorable trade-off between safety and efficiency, achieving the lowest BR and CPK with competitive RP and ORR. By selecting the spatiotemporally aligned trajectory that best matches the initial reference, this strategy enforces physical modality consistency. When integrated with stepwise rollout, it captures interaction-aware transitions that more faithfully emulate real-world multi-agent interactions.

For other agents (Table III), we fix the traffic agent to the Trajectory-Aligned Rollout and evaluate three rollout paradigms: Constant-Action Rollout, Single-Prediction Rollout, and Stepwise Prediction Rollout. Constant-Action Rollout, as an open-loop strategy, neglects evolving traffic dynamics, resulting in poor responsiveness and higher CPK. Single-Prediction Rollout improves short-horizon accuracy but lacks long-horizon supervision, leading to error accumulation and frequent off-road deviations, reflected in higher ORR. In contrast, Stepwise Prediction Rollout leverages short-horizon accuracy in a closed-loop manner, continuously adapting the agent's behavior to dynamic contexts, thereby achieving competitive ORR and CPK while maintaining high RP and eliminating blocking. These findings underscore the effectiveness of closed-loop modeling in capturing responsive and interactive multi-agent behavior.

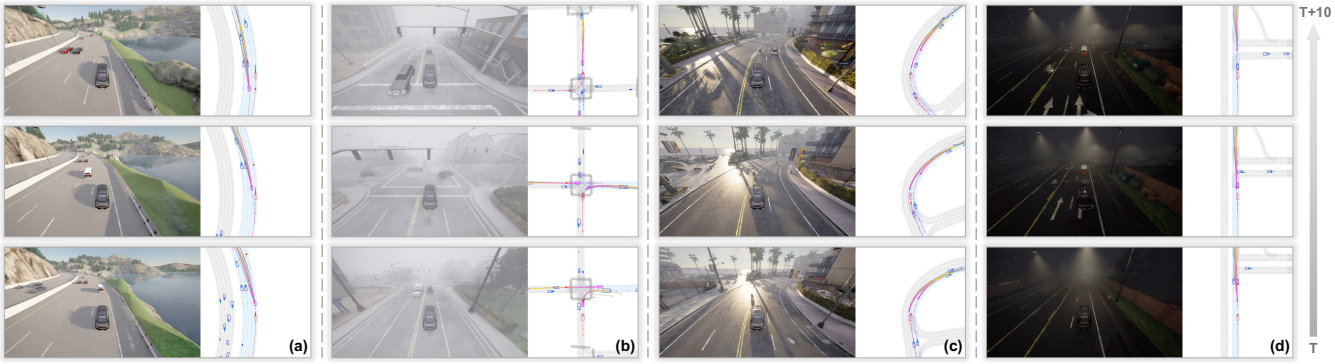

Fig. 5. **Representative scenarios of *ForSim*.** The traffic agent (CBV) is marked in purple, AV (PDM-Lite [35]) is in red, and other agents are in blue.

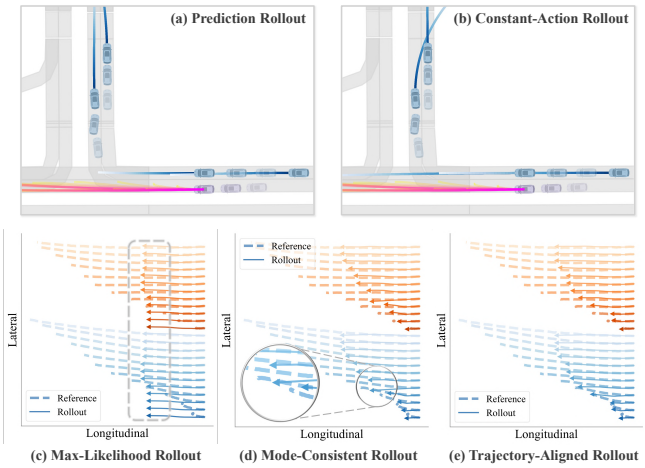

Fig. 6. **Comparison of rollout paradigms.** For other agents, (a)–(b): Prediction Rollout improves long-horizon stability over Constant-Action Rollout. For the traffic agent, (c)–(e): Trajectory-Aligned Rollout preserves multimodal coherence, while Max-Likelihood Rollout and Mode-Consistent Rollout suffer from mode collapse and temporal inconsistency.

Together, these results show that combining the Trajectory-Aligned Rollout with Stepwise Prediction Rollout is the most effective forward simulation configuration, substantially improving safety while preserving efficiency.

b) Representative Scenarios Analysis: Figure 5 presents representative scenarios that highlight the effectiveness of *ForSim* under the proposed forward simulation paradigm. By modeling closed-loop virtual interactions, the traffic policy learns to select context-aware, risk-sensitive behaviors.

In Figure 5(a), the agent performs a high-efficiency lane-change strategy by shifting from the outer to the inner lane during a low-curvature turn, highlighting its ability to anticipate and optimize future motion. Figure 5(b) illustrates that, despite multiple off-road candidates near an intersection, the agent selects a reference-aligned trajectory that adheres to spatial constraints. Figure 5(c) presents two cyclists recovering from off-center positions during a turn, demonstrating robustness to covariate shift and the capacity for corrective behavior. Figure 5(d) shows an overtaking maneuver initiated in response to a lead vehicle's deceleration, where the agent avoids potential collision while preserving driving efficiency.

These cases collectively demonstrate that modeling closed-loop interactions during forward simulation enables the traffic policy to internalize risk-aware, efficient, and realistic behaviors across diverse scenarios.

c) Qualitative Analysis of Rollout Paradigms: Figure 6 provides qualitative comparisons of rollout paradigms. Figure 6(a)–(b) contrast two rollout paradigms for other agents: Prediction Rollout and Constant-Action Rollout. The former yields more stable and accurate long-horizon rollouts, as its trained predictor generates physically plausible futures and is less sensitive to transient control noise. In contrast, the Constant-Action Rollout propagates only the initial action throughout the horizon, resulting in severe error accumulation and reduced realism.

Figure 6(c)–(e) visualize traffic agent rollouts under three paradigms. The Max-Likelihood Rollout, as discussed in Section III-B, collapses diverse intention modes by repeatedly selecting the highest-probability trajectory, limiting diversity and reducing intra-group advantage separation. The Mode-Consistent Rollout enforces explicit modal consistency by fixing the selected mode, but suffers from temporal inconsistency—trajectories from the same mode across different virtual states may not be physically aligned, leading to modality drift. In contrast, the Trajectory-Aligned Rollout enforces physical consistency by selecting the candidate that best matches the reference trajectory in spatiotemporal distance. This approach preserves multimodal diversity while enhancing physical plausibility and intra-modality consistency.

V. Conclusions

In this work, we proposed *ForSim*, a stepwise closed-loop forward-simulation paradigm that addresses key limitations in existing traffic simulation frameworks, particularly their non-reactivity and lack of interactive fidelity in prior rollout procedures. *ForSim* employs the Trajectory-Aligned Rollout for traffic agents at each virtual timestep and propagates them with physically grounded dynamics, thereby preserving multimodal diversity, intra-modality consistency, and physical plausibility. Meanwhile, other agents are updated via Stepwise Prediction Rollout, enabling responsive, interaction-aware evolution under closed-loop conditions. Integrated into the RIFT framework, *ForSim* facilitates interaction-aware simulation and fine-grained policy optimization via group-relative objectives. Experiments demonstrate that *ForSim* improves safety-related metrics without compromising efficiency, realism, or

comfort, validating the importance of modeling closed-loop multimodal interactions in virtual simulation. These results establish *ForSim* as a general and effective paradigm for enhancing the fidelity and reliability of traffic simulation.

REFERENCES

[1] D. Chen, M. Zhu, H. Yang, X. Wang, and Y. Wang, "Data-driven traffic simulation: A comprehensive review," IEEE Transactions on Intelligent Vehicles, 2024. 1

[2] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635. 1

[3] Z. Peng, W. Mo, C. Duan, Q. Li, and B. Zhou, "Learning from active human involvement through proxy value propagation," Advances in Neural Information Processing Systems, 2023. 1

[4] Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, R. Roelofs, B. Sapp, B. White, A. Faust, S. 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). IEEE, 2023, pp. 7553–7560. 1, 2

[5] C. Zhang, J. Tu, L. Zhang, K. Wong, S. Suo, and R. Urtasun, "Learning realistic traffic agents in closed-loop," in Conference on Robot Learning. PMLR, 2023, pp. 800–821. 1, 2, 6

[6] Z. Peng, W. Luo, Y. Lu, T. Shen, C. Gulino, A. Seff, and J. Fu, "Improving agent behaviors with rl fine-tuning for autonomous driving," in European Conference on Computer Vision. Springer, 2024, pp. 165–181. 1, 2, 6

[7] Z. Zhang, P. Karkus, M. Igl, W. Ding, Y. Chen, B. Ivanovic, and M. Pavone, "Closed-loop supervised fine-tuning of tokenized traffic models," in Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 5422–5432. 1, 2

[8] K. Chen, W. Sun, H. Cheng, and S. Zheng, "Rift: Closed-loop rl fine-tuning for realistic and controllable traffic simulation," arXiv preprint arXiv:2505.03344, 2025. 1, 2, 3, 4, 5, 6

[9] S. Suo, S. Regalado, S. Casas, and R. Urtasun, "Trafficsim: Learning to simulate realistic multi-agent behaviors," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10 400–10 409. 2

[10] D. Xu, Y. Chen, B. Ivanovic, and M. Pavone, "Bits: Bi-level imitation for traffic simulation," in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 2929–2936. 2

[11] K. Chitta, D. Dauner, and A. Geiger, "Sledge: Synthesizing driving environments with generative models and rule-based traffic," in European Conference on Computer Vision. Springer, 2024, pp. 57–74. 2

[12] Y. Zhou, N. Ye, W. Ljungbergh, T. Li, J. Yang, Z. Yang, H. Zhu, C. Petersson, and H. Li, "Decoupled diffusion sparks adaptive scene generation," arXiv preprint arXiv:2504.10485, 2025. 2

[13] W. Wu, X. Feng, Z. Gao, and Y. Kan, "Smart: Scalable multi-agent real-time motion generation via next-token prediction," Advances in Neural Information Processing Systems, vol. 37, pp. 114 048–114 071, 2024. 2

[14] J. Philion, X. B. Peng, and S. Fidler, "Trajeglish: Traffic modeling as next-token prediction," in The Twelfth International Conference on Learning Representations, 2023. 2

[15] S. Tan, K. Wong, S. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "Scenegen: Learning to generate realistic traffic scenes," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 892–901. 2

[16] X. Yang, S. Tan, and P. Krähenbühl, "Long-term traffic simulation with interleaved autoregressive motion and scenario generation," arXiv preprint arXiv:2506.17213, 2025. 2

[17] Z. Peng, Y. Liu, and B. Zhou, "Infgen: Scenario generation as next token group prediction," arXiv preprint arXiv:2506.23316, 2025. 2

[18] Z. Zhong, D. Rempe, D. Xu, Y. Chen, S. Veer, T. Che, B. Ray, and M. Pavone, "Guided conditional diffusion for controllable traffic simulation," in 2023 IEEE international conference on robotics and automation (ICRA). IEEE, 2023, pp. 3560–3566. 2

[19] Z. Zhong, D. Rempe, Y. Chen, B. Ivanovic, Y. Cao, D. Xu, M. Pavone, and B. Ray, "Language-guided traffic simulation via scene-level diffusion," arXiv preprint arXiv:2306.06344, 2023. 2

[20] S. Tan, B. Ivanovic, Y. Chen, B. Li, X. Weng, Y. Cao, P. Krähenbühl, and M. Pavone, "Promptable closed-loop traffic simulation," in 8th Annual Conference on Robot Learning, 2024. 2

[21] J. Lu, K. Wong, C. Zhang, S. Suo, and R. Urtasun, "Scenecontrol: Diffusion for controllable traffic scene generation," in 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 16 908–16 914. 2

[22] W. Ding, Y. Cao, D. Zhao, C. Xiao, and M. Pavone, "Realgen: Retrieval augmented generation for controllable traffic scenarios," in European Conference on Computer Vision. Springer, 2024, pp. 93–110. 2

[23] H. Lin, X. Huang, T. Phan-Minh, D. S. Hayden, H. Zhang, D. Zhao, S. Srinivasa, E. M. Wolff, and H. Chen, "Causal composition diffusion model for closed-loop traffic generation," arXiv preprint arXiv:2412.17920, 2024. 2

[24] K. Chen, Y. Lei, H. Cheng, H. Wu, W. Sun, and S. Zheng, "Frea: Feasibility-guided generation of safety-critical scenarios with reasonable adversariality," in Conference on Robot Learning. PMLR, 2025, pp. 566–586. 2, 5, 6

[25] Y. Cao, B. Ivanovic, C. Xiao, and M. Pavone, "Reinforcement learning with human feedback for realistic traffic simulation," in 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 14 428–14 434. 2

[26] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in 2017 IEEE intelligent vehicles symposium (IV). IEEE, 2017, pp. 812–818. 2

[27] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in Conference on Robot Learning. PMLR, 2023, pp. 1268–1281. 2, 6

[28] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1696–1703, 2020. 2

[29] J. Cheng, Y. Chen, X. Mei, B. Yang, B. Li, and M. Liu, "Rethinking imitation-based planners for autonomous driving," in 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 14 123–14 130. 2

[30] J. Cheng, Y. Chen, and Q. Chen, "Pluto: Pushing the limit of imitation learning-based planning for autonomous driving," arXiv preprint arXiv:2404.14327, 2024. 2, 3, 5, 6

[31] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone et al., "Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking," Advances in Neural Information Processing Systems, vol. 37, pp. 28 706–28 719, 2024. 2

[32] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," arXiv preprint arXiv:2106.11810, 2021. 2

[33] W. Cao, M. Hallgarten, T. Li, D. Dauner, X. Gu, C. Wang, Y. Miron, M. Aiello, H. Li, I. Gilitschenski et al., "Pseudo-simulation for autonomous driving," arXiv preprint arXiv:2506.04218, 2025. 2

[34] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in Conference on robot learning. PMLR, 2017, pp. 1–16. 3, 5

[35] J. Beißwenger, "PDM-Lite: A rule-based planner for carla leaderboard 2.0," https://github.com/OpenDriveLab/DriveLM/blob/DriveLM-CARLA/docs/report.pdf, 2024, accessed: 2025-04-09. 6, 7

[36] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu et al., "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," arXiv preprint arXiv:2402.03300, 2024. 6

[37] N. Montali, J. Lambert, P. Mougin, A. Kuefler, N. Rhinehart, M. Li, C. Gulino, T. Emrich, Z. Yang, S. Whiteson et al., "The waymo open sim agents challenge," Advances in Neural Information Processing Systems, vol. 36, pp. 59 151–59 171, 2023. 6

[38] H. Guo, K. Xie, and M. Keyvan-Ekbatani, "Modeling driver's evasive behavior during safety–critical lane changes: Two-dimensional time-to-collision and deep reinforcement learning," Accident Analysis & Prevention, vol. 186, p. 107063, 2023. 6

[39] S. P. Venthuruthiyil and M. Chunchu, "Anticipated collision time (act): A two-dimensional surrogate safety indicator for trajectory-based proactive safety assessment," Transportation research part C: emerging technologies, vol. 139, p. 103655, 2022. 6