SCENEDEMAND: INTENT-CONDITIONED RECURSIVE VECTOR SCENE GENERATION FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating interactive corner cases helps expose weaknesses in autonomous-driving planners and supports simulation-based training. However, existing scene-centric approaches do not explicitly align extension with the planner's intended trajectory and planning horizon. Scenes sampled without planner feedback may also lack informative interactions with the ego. Furthermore, increased difficulty alone does not guarantee training value. We propose SceneDemand, an intent-conditioned framework for on-demand recursive vector scene generation. Building on the Scenario Dreamer prior, we design a joint lane–agent diffusion generator that completes regions along the selected ego trajectory while preserving previously committed content. A temporal-coverage mechanism triggers extension when the existing map cannot support the full planning horizon, enabling scene growth to adapt to evolving ego intent. We further introduce Counterfactual Direct Planner Guidance (CDPG), which compares frozen-critic values with and without new vehicles at multiple static intent-aligned viewpoints to steer their generation toward planner-specific challenges. On Waymo, intent conditioning increases the recursive extension completion rate from 48% to 76%. Under matched training and evaluation across three policy seeds, planners trained on SceneDemand-CDPG scenes achieve 77.92% mean success on held-out Waymo Log Replay scenes, compared with 75.75% for Scenario Dreamer.

1 INTRODUCTION

Training and evaluating autonomous-driving planners require interactive scenes that remain available as the ego vehicle moves beyond the initially available map. Log replay provides realistic states but is tied to recorded driving sequences; it does not by itself provide reactive traffic for alternative ego actions or extend the map beyond the recorded scene. Generative simulators synthesize road geometry and traffic participants, and recent vectorized models make repeated scene completion increasingly practical (Tan et al., 2021; Feng et al., 2023; Pronovost et al., 2023; Rowe et al., 2025; Jiang et al., 2026).

Existing work provides important pieces of this pipeline. Scenario Dreamer learns a vectorized lane–agent latent prior, combines it with reactive NPC behavior, and supports extrapolation through diffusion inpainting (Rowe et al., 2025). VectorWorld advances structured generation toward streaming vector-graph outpainting over long interactive rollouts (Jiang et al., 2026). Controllable and adversarial generators steer traffic using semantic conditions, optimized trajectories, or preference signals (Pronovost et al., 2023; Wang et al., 2021; Hanselmann et al., 2022; Nie et al., 2026). World-model studies increasingly evaluate downstream planning performance (Zhang et al., 2026; Wang et al., 2025).

However, scene-centric generation typically samples from a learned prior, expands a geometric boundary, or follows an external control signal, without explicitly using the planner's selected future trajectory to define generation demand. This separation imposes three limitations. First, spatially specified scene extension does not explicitly measure how long the selected ego trajectory remains supported. Second, plausible traffic sampled from a scene prior alone need not challenge a particular planner. Third, increased difficulty does not guarantee improved planner training: a lower success

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.

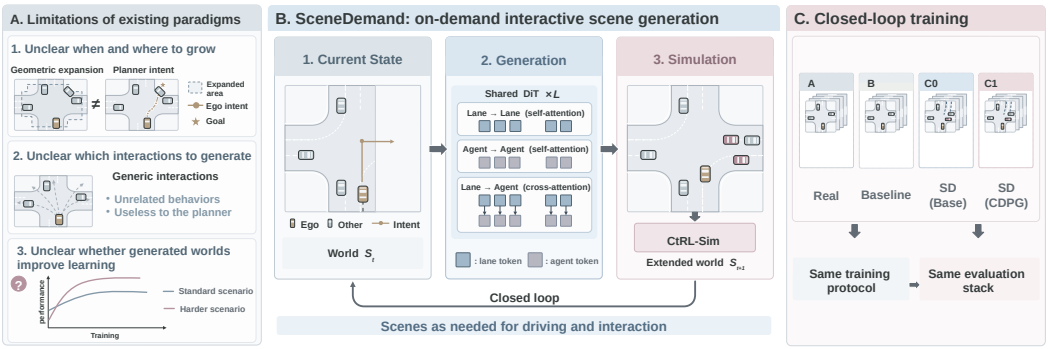

Figure 1: **Overview of SceneDemand.** (A) Limitations of scene-centric generation. (B) Planner intent drives recursive scene completion, and CtRL-Sim controls surrounding NPC behavior after each commit. (C) Four 500-scene training sources are compared using the same policy, training protocol, and closed-loop evaluation stack.

rate may reflect informative or pathological interactions. Matched training and evaluation under a common real-data protocol are therefore needed.

In this paper, we introduce SceneDemand (SD) for on-demand recursive vector scene extension (Figure 1). Our generator uses the planner-selected trajectory (*ego intent*) to identify the unsupported region and temporal coverage to trigger joint lane–agent completion; each increment is atomically committed before simulation advances. We propose Counterfactual Direct Planner Guidance (CDPG), which uses the frozen value function of a pretrained planner to guide new NPC generation toward planner-specific challenges. By comparing scene values with and without the new NPCs at multiple static intent-aligned viewpoints, CDPG estimates their marginal impact and emphasizes the largest value drops. Intent thus defines coverage demand, while CDPG targets interaction demand. On Waymo, intent conditioning enables 76% of runs to complete five recursive extensions, up from 48% without it. Across three policy seeds with matched training and evaluation, policies trained on SceneDemand-CDPG scenes attain 77.92% mean success on held-out Waymo Log Replay, exceeding those trained on Scenario Dreamer and SceneDemand-Base scenes by 2.17 and 1.95 percentage points, respectively.

Our contributions are as follows:

- **We design and train an intent-conditioned generator for recursive vector scene extension.** It jointly completes lane geometry and newly introduced NPC states in the unsupported region. Temporal coverage triggers generation, and committed scene increments condition subsequent planning and generation.
- **We propose guiding NPC generation with a pretrained planner's value function.** CDPG keeps this function frozen and compares scene values with and without new NPCs at multiple static intent-aligned viewpoints, using a smooth aggregation to emphasize the largest value drops and steer generation toward planner-specific challenges.
- **We evaluate generated scenes through controlled planner training.** Under matched initialization, source-set size, optimization budget, and evaluation settings, SceneDemand-CDPG achieves a 1.95-percentage-point higher mean Waymo Log Replay success rate than SceneDemand-Base across three policy seeds.

2 Related Work

Vectorized Scene Generation and Recursive Extension. Data-driven traffic generators model scene structure and multi-agent behavior with autoregressive, diffusion, or hybrid formulations (Tan et al., 2021; Suo et al., 2021; Feng et al., 2023; Pronovost et al., 2023). DriveSceneGen generates diverse structured scenes from scratch (Sun et al., 2024), while SLEDGE demonstrates compact latent scene synthesis (Chitta et al., 2024). Scenario Dreamer uses vectorized lane–agent latents and diffusion inpainting for environment extrapolation (Rowe et al., 2025). VectorWorld instead

performs streaming completion of heterogeneous vector graphs during long interactive rollout (Jiang et al., 2026). In SceneDemand, the planner's selected trajectory defines the unsupported region and temporal trigger, replacing fixed spatial extension criteria.

Controllable and Adversarial Scenario Generation. Controllable generators specify semantics, outcomes, or difficulty. Scenario Diffusion introduces scene-level control (Pronovost et al., 2023); adversarial simulators optimize trajectories or kinematic variables to expose planning failures (Wang et al., 2021; Hanselmann et al., 2022); and SAGE performs test-time preference steering (Nie et al., 2026). CDPG is related to diffusion guidance (Ho et al., 2020; Dhariwal & Nichol, 2021; Rombach et al., 2022), but it aggregates paired frozen-critic drops across static intent-aligned probes to target new-NPC interactions along the selected ego intent. The generated interaction persists in the recursive scene and conditions subsequent planning. Diffusion-based planning and policy generation provide complementary decision-side uses of generative models (Janner et al., 2022; Chi et al., 2023).

Reactive Multi-Agent Behavior Modeling. Multi-agent forecasting and simulation models capture interaction through transformer or language-model formulations, including AgentFormer, Scene Transformer, Motion Transformer, MotionLM, Wayformer, and Trajeglish (Yuan et al., 2021; Ngiam et al., 2022; Shi et al., 2022; Seff et al., 2023; Nayakanti et al., 2023; Philion et al., 2024). CARLA, SMARTS, nuPlan, Waymax, and ScenarioNet provide simulation interfaces for reactive evaluation (Dosovitskiy et al., 2017; Zhou et al., 2021; Caesar et al., 2021; Gulino et al., 2023; Li et al., 2023). SceneDemand generates lane geometry and new-NPC initial states; after each commit, CtRL-Sim controls NPC behavior and the ego follows its planner.

Generated Scenarios for Planner Training. Beyond reconstruction quality, latent dynamics and driving world models are evaluated for control, planning, and data generation (Hafner et al., 2023; Yang et al., 2024; Wang et al., 2023; 2024; Hu et al., 2023). Recent work studies model–policy mismatch and policy adaptation more directly, including World-In-World, AdaWM, and SPACeR (Zhang et al., 2026; Wang et al., 2025; Chang et al., 2026). Scenario Dreamer evaluates generated environments through an RL planner, while VectorWorld reports policy retraining across simulator backends (Rowe et al., 2025; Jiang et al., 2026). We assess generation and planner-targeted sampling through matched planner training.

3 SceneDemand

3.1 Formulation

Scene increments persist through subsequent planning and generation. At simulator time t, we represent the vector scene and ego intent as

$$S_t = (M_t, N_t), \qquad I_t = [q_{t,0}, \ldots, q_{t,H_I}], \\ q_{t,h} = (\Delta x_{t,h}, \Delta y_{t,h}, v_{t,h}, \psi_{t,h}). \tag{1}$$

Here, M_t is the committed lane state, N_t is the current agent state, and H_I is the final index of the discretized planner horizon. Each $q_{t,h}$ is expressed in the current ego frame: Δx and Δy are relative displacement, v is speed, and ψ is heading. We use t for simulator time, h for the discrete intent index, and τ for reverse-diffusion steps. Within a generation event, we omit t from diffusion states when unambiguous.

Intent-defined generation demand. Support for the ego planner's next decisions depends on its selected future trajectory, beyond the current geometric scene boundary. We define an intent corridor and its unsupported region as

$$\mathcal{C}(I_t) = \mathrm{Tube}(I_t; r_{\mathrm{ego}} + r_{\mathrm{margin}}), \qquad Q_t = \mathcal{C}(I_t) \setminus \Omega(M_t), \tag{2}$$

where Tube dilates the selected ego trajectory by the ego footprint radius r_{ego} plus margin r_{margin}, and $\Omega(M_t)$ is the spatial support of the committed map. Q_t is the geometric unsupported region, encoded by E_Q for conditioning. The unknown lane and active new-NPC latent rows to be generated are termed frontier slots; the slot mask introduced below distinguishes them from known content.

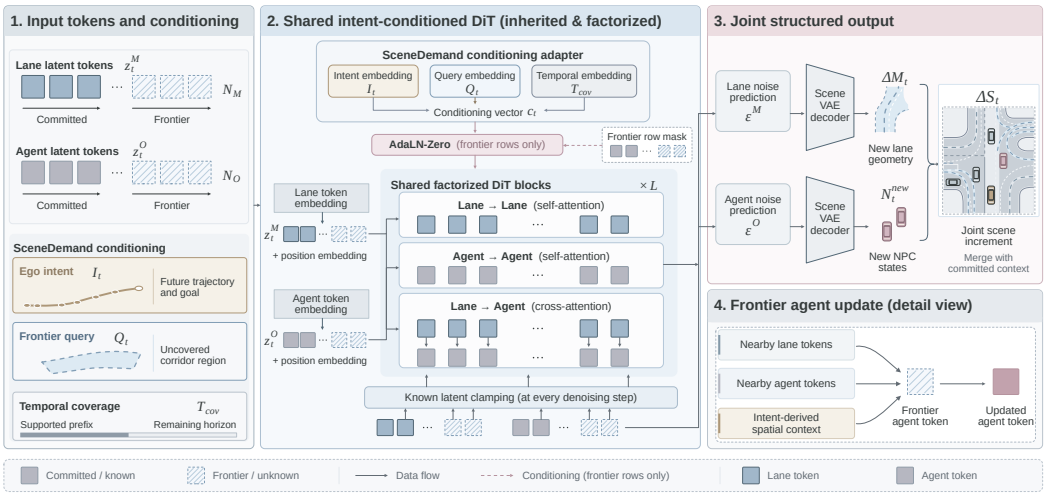

Figure 2: **Architecture of the intent-conditioned lane–agent generator.** Ego-intent, unsupported-region, and temporal-coverage features augment the inherited diffusion conditioning path only on frontier slots. Committed lane and agent latents remain clamped throughout denoising, while the shared factorized backbone predicts unknown lane and new-NPC latents in one generation event.

Temporal coverage is the largest supported discrete prefix of the current intent,

$$T_{\mathrm{cov}}(M_t, I_t) = \max\Big\{h \in \{0, \dots, H_I\} \;\Big|\; \mathrm{Tube}\big(I_t[0{:}h]; r_{\mathrm{ego}} + r_{\mathrm{margin}}\big) \subseteq \Omega(M_t)\Big\}, \qquad g_t = \mathbb{I}[T_{\mathrm{cov}}(M_t, I_t) < H_I]\,. \tag{3}$$

We set $T_{\mathrm{cov}} = 0$ when no nonzero prefix is supported, and $\mathbb{I}[\cdot]$ denotes the indicator function. Thus Q_t specifies the unsupported region, while g_t triggers generation when the supported prefix falls below the planner horizon. Both quantities change when the planner selects a new trajectory.

Given these quantities, the proposed scene increment contains lane geometry and newly introduced NPCs,

$$\Delta S_t = (\Delta M_t, N_t^{\mathrm{new}}) \sim \mathcal{K}_{\theta,c}^{(g_t)}(\cdot \mid S_t, I_t, Q_t, T_{\mathrm{cov}}(M_t, I_t))\,, \tag{4}$$

with the trigger-gated kernel

$$\mathcal{K}_{\theta,c}^{(g)} = (1-g)\,\delta_{\varnothing} + g\,\mathcal{K}_{\theta,c}, \qquad c \in \{\mathrm{Base}, \mathrm{CDPG}\}. \tag{5}$$

The mode c is an inference-time switch between base and planner-guided sampling, and $\delta_{\varnothing}$ is a point mass at the null increment. When $g_t = 0$, the transition returns this null increment. Because I_t is recomputed after each simulator transition, the next generation request depends on the updated simulator state.

3.2 Model Architecture and Training

We train the joint generator using Scenario Dreamer's released vector latent representation and checkpoints (Rowe et al., 2025). At each triggered event, ego intent, unsupported region, and temporal coverage augment the diffusion context,

$$h_t = h_t^{\mathrm{diff}} + E_I(I_t) + E_Q(Q_t) + E_T\big(T_{\mathrm{cov}}(M_t, I_t)\big), \tag{6}$$

where h_t^{diff} is the inherited scene-conditioning context and E_I, E_Q, and E_T map these three conditioning inputs to the shared dimension. These added conditions are applied to frontier slots. Lane and object slots are then completed by the same denoising backbone,

$$(\Delta z_t^M, z_t^{N,\mathrm{new}}) \sim p_\theta(\cdot \mid z_t^{\mathrm{known}}, m_t, h_t), \qquad (\Delta M_t, N_t^{\mathrm{new}}) = D_\phi(\Delta z_t^M, z_t^{N,\mathrm{new}}). \tag{7}$$

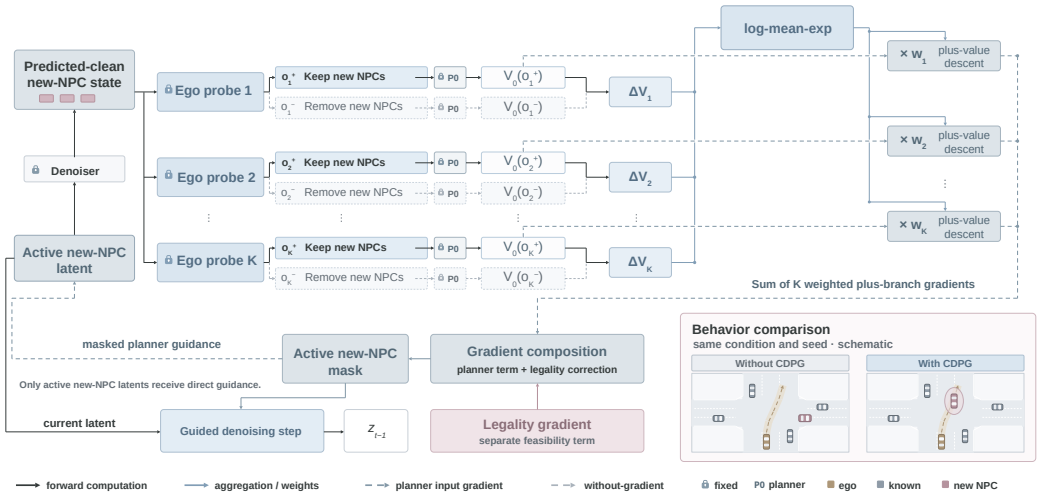

Figure 3: **Intent-aligned viewpoints for CDPG.** Static ego probes inspect the same frontier NPCs from the current pose and future locations along the selected intent. Paired values measure the marginal new-NPC impact at each probe; log-mean-exp emphasizes the largest drops before composition with the legality term.

Here, z_t^{known} contains committed latents, m_t separates known content from unknown lane and active new-NPC slots, and D_ϕ denotes the frozen scene decoders. Both types of unknown slots are predicted jointly.

Generator training. For diffusion training, let $z_t = (\Delta z_t^M, z_t^{N,\text{new}})$ denote the clean frontier latents and let $x_{t,\tau}$ be their noised state at event t and reverse step τ. We train the shared generator with masked epsilon prediction,

$$\begin{aligned}\mathcal{L}_G = \mathbb{E}\Big[&\|m_t^M \odot [\epsilon^M - \epsilon_\theta^M(x_{t,\tau}, \tau, m_t, h_t)]\|_2^2 \\ &+\lambda_{\text{obj}}\|m_t^N \odot [\epsilon^N - \epsilon_\theta^N(x_{t,\tau}, \tau, m_t, h_t)]\|_2^2\Big].\end{aligned} \tag{8}$$

m_t^M and m_t^N select the trainable lane and object frontier slots, ϵ^M and ϵ^N are the injected Gaussian noises for the two branches, and λ_{obj} weights the object prediction term. Every recursive event uses this objective with horizon-independent weights; repeated request–generate–commit transitions determine rollout duration. Appendices B.1 and B.7 give training-pair construction and optimizer settings.

3.3 Inference-Time Counterfactual Direct Planner Guidance

Intent-aligned critic probes. A current-pose critic probe may miss frontier NPC interactions farther along the selected intent. Rather than differentiate through costly planner–simulator rollouts that entangle guidance with NPC behavior, we evaluate the same predicted-clean scene from K static intent-aligned ego probes, $\mathcal{A}_t = \{a_{t,k}\}_{k=1}^K$.

Predicted-clean new-NPC state. At a selected reverse step τ, CDPG reconstructs the current-event new-NPC latent,

$$\hat{x}_{0,\tau}^N = \frac{x_\tau^N - \sqrt{1-\bar{\alpha}_\tau}\,\epsilon_\theta^N(x_\tau, \tau, h_t)}{\sqrt{\bar{\alpha}_\tau}}. \tag{9}$$

Here x_τ is the noisy scene latent and $\bar{\alpha}_\tau$ the cumulative DDPM signal schedule. Existence decisions and observation indices are fixed with stop-gradient; decoded continuous position, velocity, and heading remain differentiable.

Counterfactual marginal value drop. For each probe, $o_{\tau,k}^+$ retains the current-event new NPCs, whereas $o_{\tau,k}^-$ masks only those NPCs while preserving the probe state, committed scene, route,

map, and existing agents. The paired difference $\Delta V_{\tau,k}$ measures the marginal value degradation attributable to the new NPCs at probe k. Using each probe's no-new-NPC value as a local baseline accounts for intrinsic critic-value differences among probe locations. With differentiable observation adapter $\mathcal{O}_0$ and frozen NPC decoder D_ϕ^N, we construct these pairs and aggregate their value drops as

$$o^+_{\tau,k} = \mathcal{O}_0\big(S_t, I_t, a_{t,k}, D^N_\phi(\hat{x}^N_{0,\tau})\big), \quad o^-_{\tau,k} = \mathrm{Mask}_{\mathrm{new}}(o^+_{\tau,k}),$$
$$\Delta V_{\tau,k} = V_0(o^-_{\tau,k}) - V_0(o^+_{\tau,k}), \qquad E_{\mathrm{hard},\tau} = \tau_a \left[\log \sum_{k=1}^{K} e^{\Delta V_{\tau,k}/\tau_a} - \log K\right]. \tag{10}$$

Smooth aggregation and direct guidance. Positive $\Delta V_{\tau,k}$ denotes lower value with the new NPCs present. The log-mean-exp objective smoothly emphasizes the probe at which those NPCs have the strongest planner impact, with temperature $\tau_a > 0$ controlling this emphasis. The masked branches are detached from the new-NPC latent, so their values calibrate the cross-probe weights rather than supplying an additional direct gradient path. The applied hardness gradient is the resulting weighted combination of the present-NPC critic gradients:

$$w_{\tau,k} = \frac{e^{\Delta V_{\tau,k}/\tau_a}}{\sum_{j=1}^{K} e^{\Delta V_{\tau,j}/\tau_a}}, \qquad \nabla_{x^N_\tau} E_{\mathrm{hard},\tau} = -\sum_{k=1}^{K} w_{\tau,k} \nabla_{x^N_\tau} V_0(o^+_{\tau,k}). \tag{11}$$

Masked gradient composition. With $M_{\mathrm{new}} = M^{\mathrm{unk}}_{\mathrm{obj}} \odot M_{\mathrm{act}}$ selecting active unknown object slots, the implemented update is

$$g^{\mathrm{hard}}_\tau = \mathrm{Normalize}\big(M_{\mathrm{new}} \odot \nabla_{x^N_\tau} E_{\mathrm{hard},\tau}\big), \quad g^{\mathrm{legal}}_\tau = \mathrm{Normalize}\big(M_{\mathrm{new}} \odot \nabla_{x^N_\tau} B_{\mathrm{legal}}\big),$$
$$d_\tau = \mathrm{ClipNorm}\big(g^{\mathrm{hard}}_\tau - g^{\mathrm{legal}}_\tau\big), \quad \epsilon^{N,\mathrm{CDPG}}_\tau = \epsilon^N_\theta(x_\tau, \tau, h_t) - s_\tau \sqrt{1-\bar{\alpha}_\tau}\, d_\tau, \tag{12}$$
$$\nabla_{x^M_\tau} E_{\mathrm{hard},\tau} = 0, \qquad \nabla_{x^{\mathrm{known}}_\tau} E_{\mathrm{hard},\tau} = 0.$$

B_{legal} combines four unit-weight penalties for ego clearance, inter-agent clearance, lane support, and kinematics (Appendix B.2). Hardness and legality gradients are normalized separately before subtraction and norm clipping. This local guidance update is restricted to active current-event new-NPC latents; lane latents, known content, intent, route, and planner parameters receive no direct update. CtRL-Sim and simulator transitions remain outside the differentiation graph. Base uses $s_\tau = 0$; CDPG uses $s_\tau > 0$ only on the guided-step set $\mathcal{G}$. Algorithm S2 gives the reverse steps, and Appendix A.3 provides the structural analysis. After DDPM-100, the scene increment is decoded and processed by frozen cleanup before the online commit (Appendix B.3).

3.4 Implementation Details

Intent extraction. The planner-selected trajectory is expressed in the current ego frame and resampled to 16 waypoints over a 3.0 s horizon. Appendix B.7 gives the normalization and embedding settings.
Probe aggregation. The probes use log-mean-exp temperature $\tau_a = 0.25$.

Generation trigger. The 0.1 s coverage test uses a 5.0 m corridor margin and triggers below the planner horizon (Eq. 3).

Simulator integration. Scene increments are committed without overwriting known content. New NPCs are registered by the simulator-history adapter. CtRL-Sim then predicts NPC actions, the ego executes its planner-selected intent, and the simulator advances the vehicle states. The updated S_{t+1} determines the next intent, coverage test, and unsupported region. Algorithm S1 summarizes these transitions.

4 Experiments

4.1 Experimental Setup and Metrics

We evaluate SceneDemand on the Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021) using DDPM-100 and a shared 9 s quantitative window. CtRL-Sim controls NPCs; the evaluated planner controls the ego. CDPG queries frozen P_0 and its critic, and Base/CDPG share behavior tilt $+10$. Random Real, Scenario Dreamer, Base, and CDPG each supply 500 training environments. All

policies share pretrained initialization, PPO settings (Schulman et al., 2017), 500k interactions, and three policy seeds. Evaluation uses the same 500 held-out Waymo Log Replay scenes, three evaluator seeds, and closed-loop stack. We measure **scene quality** using Fréchet distance (FD), endpoint distance (EPD), static collisions, and Agent Jensen–Shannon divergence (JSD); **recursive continuation** using coverage, map starvation (MSR), five-commit completion, route-lane utilization, and geometric continuity; **frozen-planner difficulty** using collision/off-route/success (C/O/S), progress, and E_{hard}; and **training utility** using success, failure rates, and progress. E_{hard} is an internal diagnostic, not a closed-loop outcome. Arrows indicate preferred directions. Appendices B and C.1 detail settings and metrics; Appendix D.1 discusses limitations.

4.2 Scene Generation Quality and Recursive Continuation

Scene generation quality. Both SceneDemand variants reduce FD and EPD relative to reproduced Scenario Dreamer and exhibit no static collisions (Table 1). However, higher Agent JSD shows that better geometry does not imply closer agreement with reference traffic statistics. CDPG changes Agent JSD only slightly relative to Base while retaining zero initial overlaps, suggesting that its added interaction pressure is not explained by colliding initializations. Thus, geometric plausibility, agent-distribution fidelity, and closed-loop difficulty should be assessed separately; no single quality metric captures all three.

Table 1: **Generated lane–agent scene quality on Waymo.** FD measures distributional fidelity; EPD, lane-successor endpoint consistency; static collision rate, initialization feasibility; and Agent JSD, alignment of agent statistics. Lower is better throughout.

Method	Source	Perceptual	Map structure	Safety	Agent distribution
		FD↓	EPD (m)↓	Static Collision (%)↓	Agent JSD↓
DriveSceneGen† (Sun et al., 2024)	Reported	40.59	0.010	0.20	–
VectorWorld† (Jiang et al., 2026)	Reported	0.9400	0.0940	4.6900	0.098000
Scenario Dreamer (Rowe et al., 2025)	Reproduced	29.5167	0.1346	5.6089	**0.033320**
SceneDemand-Base	Ours	27.5996	**0.1191**	**0.0000**	0.104896
SceneDemand-CDPG	Ours	**27.2110**	0.1269	**0.0000**	0.103869

† Reported-only values, not matched comparisons: DriveSceneGen follows VectorWorld's Waymo comparison (sourced there from Scenario Dreamer); VectorWorld follows its reported Waymo evaluation.

Recursive-generation settings. We separate demand localization from trigger timing (Table 2). **SD w/o Intent-Conditioned Generation** retains live-intent temporal triggering but masks the learned intent condition and replaces the intent-derived frontier query with an intent-independent seeded route fallback. It therefore changes where and what to generate, not when. **SD w/o Temporal** retains the intent-conditioned frontier and generator inputs but uses the frozen spatial trigger.

Table 2: **Intent-conditioned recursive continuation.** Generation settings are compared in terms of temporal support, five-commit completion, route use, and structural consistency. Intent-conditioned generation specifies where and what to generate; temporal coverage determines when to extend.

Method	Component settings		Recursive support			Structure		
	Intent-conditioned generation	Trigger	Temporal Cov. (%)↑	MSR (%)↓	5-Commit (%)↑	Route-Lane Util. (%)↑	Endpoint Gap↓	Successor Cont. (%)↑
SD w/o Intent-Conditioned Generation	✗	Temporal	98.32	0.09	48.00	23.63	0.072	98.20
SD w/o Temporal	✓	Spatial	95.77	0.15	68.00	20.90	**0.071**	96.67
SceneDemand-Base	✓	Temporal	97.52	**0.05**	**76.00**	**24.07**	0.088	96.80

Recursive continuation. Intent conditioning raises five-commit completion from 48% to 76%, despite slightly lower temporal coverage, showing that local coverage alone does not ensure repeated continuation. Its modest route-use gain and worse endpoint gap and successor continuity indicate that the benefit is demand-aligned completion, not uniformly better geometry. Temporal triggering

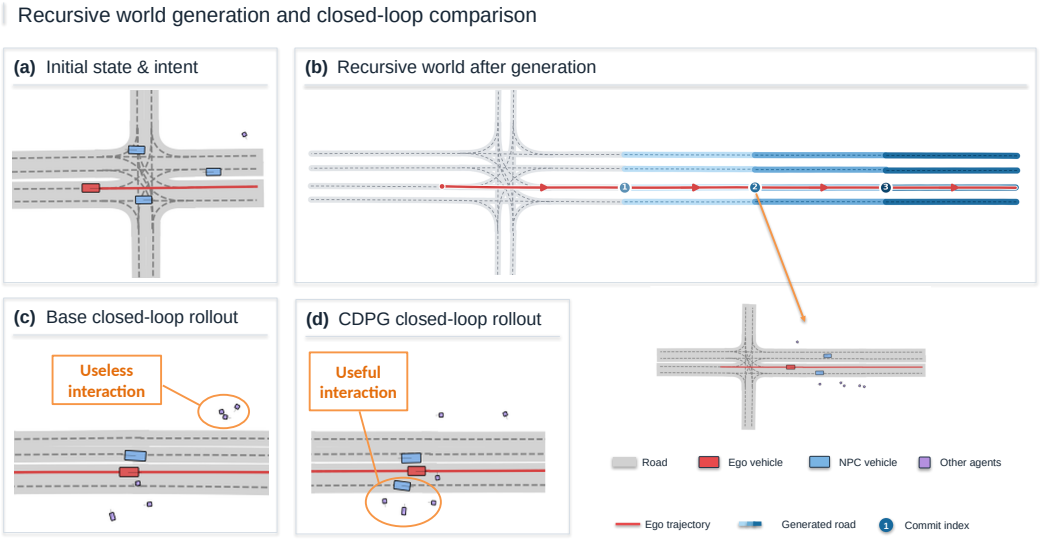

Figure 4: **Recursive scene extension and matched Base–CDPG rollouts.** A single case shows (a) the initial state and ego intent, (b) recursive road extension with commit markers 1–3, and (c,d) Base and CDPG rollouts. Blue segments show generated road; the red line shows the ego trajectory; orange callouts highlight the compared interactions.

independently raises completion from 68% to 76% and reduces starvation from 0.15% to 0.05%. Together, these results motivate coupling *where to generate* with *when support is needed*, rather than using either condition alone.

Qualitative interpretation. In Figure 4(a,b), successive extensions remain in one persistent scene: committed increments support later planning rather than being independently resampled. Panels (c,d) place the highlighted CDPG interaction closer to the ego path than the Base case, illustrating why generation must address both route support and interaction relevance. Table 3 tests the latter effect under matched controls.

Long-horizon recurrence. Beyond the quantitative evaluation window, SceneDemand supports continuous closed-loop scene generation for 50 s and longer. Appendix C.2.3 provides layouts for tracing accumulated scene growth, evolving intent, commit events, and active-NPC counts in a representative 50 s rollout.

4.3 Effect of CDPG on Frozen-Planner Difficulty

We compare Base and CDPG with the same CtRL-Sim model and tilt $+10$ (Table 3). Within Scenario Dreamer ("Dreamer"), adversarial tilt lowers success by 2.78 points, illustrating behavior-side difficulty control. Our fixed-tilt comparison instead isolates generation-side guidance.

Table 3: **CDPG and frozen-planner difficulty.** Base/CDPG share CtRL-Sim tilt $+10$; Dreamer Adv. uses -10. C/O/S denotes collision/off-route/success. E_{hard} is the internal value-drop diagnostic.

Environment	Generation and rollout settings			Frozen pretrained planner evaluation		
	Source	NPC behavior model / tilt	Horizon	C / O / S (%)	Progress (m)	E_{hard} ↑
Dreamer Normal	Reproduced	CtRL-Sim / $+10$	55 m / 9 s	26.39 / 4.17 / 68.06	44.16	–
Dreamer Adv.	Reproduced	CtRL-Sim / -10	55 m / 9 s	27.78 / 4.17 / 65.28	44.03	–
SceneDemand-Base	Ours	CtRL-Sim / $+10$	9 s	8.00 / 32.00 / 60.00	35.92	0.0141
SceneDemand-CDPG	Ours	CtRL-Sim / $+10$	9 s	**12.00 / 36.00 / 52.00**	**32.79**	**0.0487**

Planner-targeted difficulty. CDPG lowers success from 60% to 52% and progress from 35.92 m to 32.79 m; collision and off-route rates each rise by four points. The pressure therefore extends beyond

direct collisions to disrupted route following. The larger E_{hard} (0.0141 to 0.0487) is consistent with the intended guidance, while the closed-loop outcomes establish its behavioral effect. With NPC behavior settings fixed, this comparison supports shaping generated configurations rather than merely increasing behavior aggressiveness. Training utility is tested separately below.

4.4 Planner Training and Cross-Environment Evaluation

Planner training performance. Under matched training, CDPG improves mean Waymo Log Replay success over Base by 1.95 points and progress by $1.56\,\text{m}$ (Table 4). The success gain is accompanied mainly by fewer off-route and timeout outcomes, with little collision-rate change. This suggests improved task completion rather than a collision-only benefit.

Table 4: **Matched planner training with Waymo Log Replay evaluation.** Mean±sample standard deviation over three policy seeds under Section 4.1; CDPG versus Base is the primary comparison.

Policy	Training source	Waymo Log Replay evaluation				
		Success↑	Collision↓	Off-route↓	Timeout↓	Progress↑
Pretrained planner	None	73.00	17.84	5.83	3.33	60.93
Random Real	WOMD Real	73.33±1.92	18.00±2.06	**5.50±2.42**	**3.17±0.62**	61.50±0.15
Scenario Dreamer	Scenario Dreamer	75.75±2.42	12.67±1.65	6.08±2.07	5.50±0.71	61.04±0.16
SceneDemand-Base	SceneDemand-Base	75.97±2.25	12.00±1.18	7.83±1.71	4.20±2.71	61.26±1.67
SceneDemand-CDPG	SceneDemand-CDPG	**77.92±1.98**	**11.92±1.30**	6.67±1.18	3.50±0.24	**62.82±0.20**

Relative to Random Real, generated-source training mainly lowers collisions. Base and Scenario Dreamer yield similar success, whereas CDPG adds a further gain, linking targeted interactions to training utility. Real-data training still has fewer off-route and timeout outcomes, so the improvement is not uniform across failures.

Cross-environment evaluation. CDPG has the highest mean success in all three sets (Table 5). Gains over Base are 1.95 points on Waymo Log Replay, 0.53 on Dreamer-Neg, and 0.78 on SD-Challenge. The consistent direction suggests utility beyond one evaluation source, while the smaller challenge-set gains show that the benefit is environment-dependent, not a uniform robustness improvement.

Table 5: **Cross-environment success.** Trained-policy results are mean±std over the same seeds. Dreamer denotes Scenario Dreamer; SD denotes SceneDemand. All policies share evaluation settings.

Policy	Success (%)↑		
	Waymo Log Replay	Dreamer-Neg	SD-Challenge
Pretrained planner	73.00	62.28	52.00
Random Real	73.33±1.92	63.13±0.80	52.44±2.04
Scenario Dreamer	75.75±2.42	63.39±3.81	51.78±2.07
SceneDemand-Base	75.97±2.25	63.43±3.26	52.11±2.39
SceneDemand-CDPG	**77.92±1.98**	**63.96±3.60**	**52.89±2.78**

5 Conclusion

In this paper, we presented SceneDemand for on-demand recursive vector scene generation. To align scene extension with planner intent, we introduced a joint lane–agent diffusion generator and a temporal-coverage mechanism that triggers growth only when the current map cannot support the planning horizon. We further proposed CDPG to steer newly generated vehicles toward planner-specific interaction challenges. Together, these components address both the need for continued scene coverage and the need for informative interactions. Experiments show improved recursive continuation and higher mean success on held-out Waymo Log Replay scenes under matched planner training. These results suggest that coupling scene generation with ego intent and planner feedback is a practical path toward more effective simulation-based training. Future work will explore explicit control over traffic density, agent behavior style, interaction type, and difficulty level.

Reproducibility Statement

Implementation details, configurations, manifests, and evaluation definitions will accompany the anonymous submission.

AI Use Statement

Generative AI tools assisted with writing and language polishing, literature search and discovery, and research conceptualization and execution. The authors reviewed the AI-assisted content against the underlying experiments and source materials and take responsibility for the final content. Generative AI was not used to create the reported experimental measurements or to substitute synthetic results for missing experiments.

References

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

Wei-Jer Chang, Akshay Rangesh, Kevin Joseph, Matthew Strong, Masayoshi Tomizuka, Yihan Hu, and Wei Zhan. Spacer: Self-play anchoring with centralized reference models. In *International Conference on Learning Representations (ICLR)*, 2026.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin C. M. Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Kashyap Chitta, Daniel Dauner, and Andreas Geiger. Sledge: Synthesizing driving environments with generative models and rule-based traffic. In *European Conference on Computer Vision (ECCV)*, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16, 2017.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9710–9719, 2021.

Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3567–3575, 2023.

Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, John D. Co-Reyes, Rishabh Agarwal, Rebecca Roelofs, Yao Lu, Nico Montali, Paul Mougin, Zoey Yang, Brandyn White, Aleksandra Faust, Rowan McAllister, Dragomir Anguelov, and Benjamin Sapp. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Niklas Hanselmann, Katrin Renz, Kashyap Chitta, Apratim Bhattacharyya, and Andreas Geiger. King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients. In *European Conference on Computer Vision (ECCV)*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915, 2022.

Chaokang Jiang, Desen Zhou, Jiuming Liu, and Kevin Li Sun. Vectorworld: Efficient streaming world model via diffusion flow on vector graphs. *arXiv preprint arXiv:2603.17652*, 2026.

Quanyi Li, Zhenghao Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2980–2987, 2023.

Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David J. Weiss, Benjamin Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations (ICLR)*, 2022.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021.

Tong Nie, Yuewen Mei, Yihong Tang, Junlin He, Jie Sun, Haotian Shi, Wei Ma, and Jian Sun. Steerable adversarial scenario generation through test-time preference alignment. In *International Conference on Learning Representations (ICLR)*, 2026.

Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *International Conference on Learning Representations (ICLR)*, 2024.

Ethan Pronovost, Meghana Reddy Ganesina, Noureldin Hendy, Zeyu Wang, Andres Morales, Kai Wang, and Nicholas Roy. Scenario diffusion: Controllable driving scenario generation with diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 68873–68894, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Luke Rowe, Roger Girgis, Anthony Gosselin, Liam Paull, Christopher Pal, and Felix Heide. Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17207–17218, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8579–8590, 2023.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 6531–6543, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021b.

S. Sun, Z. Gu, T. Sun, J. Sun, C. Yuan, Y. Han, D. Li, and M. H. Ang. Drivescenegen: Generating diverse and realistic driving scenarios from scratch. *IEEE Robotics and Automation Letters*, 9(8): 7007–7014, 2024.

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.

Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Scenegen: Learning to generate realistic traffic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 892–901, 2021.

Hang Wang, Xin Ye, Feng Tao, Chenbin Pan, Abhirup Mallik, Burhan Yaman, Liu Ren, and Junshan Zhang. Adawm: Adaptive world model based planning for autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2025.

Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9909–9918, 2021.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14749–14759, 2024.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, 2024.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M. Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9813–9823, 2021.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M. Patel, Paul Pu Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-in-world: World models in a closed-loop world. In *International Conference on Learning Representations (ICLR)*, 2026.

Ming Zhou, Jun Luo, Julian Villella, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, Montgomery Alban, Iman Fadakar, Zheng Chen, Chongxi Huang, Ying Wen, Kimia Hassanzadeh, Daniel Graves, Zhengbang Zhu, Yihan Ni, Nhat Nguyen, Mohamed Elsayed, Haitham Ammar, Alexander Cowen-Rivers, Sanjeevan Ahilan, Zheng Tian, Daniel Palenicek, Kasra Rezaee, Peyman Yadmellat, Kun Shao, Dong Chen, Baokuan Zhang, Hongbo Zhang, Jianye Hao, Wulong Liu, and Jun Wang. Smarts: An open-source scalable multi-agent rl training school for autonomous driving. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 155 of *Proceedings of Machine Learning Research*, pp. 264–285, 2021.

APPENDIX

This appendix consolidates the supplementary material into four parts: method formulation and analysis, implementation and experimental protocol, supplementary results and visualizations, and limitations and broader impact.

Reading guide.

A. Method Formulation, Algorithms, and Structural Analysis

Pages 14–17. Section A.1 defines the persistent scene, intent, frontier support, notation, and recursive kernel. Section A.2 gives the recursive-execution and CDPG algorithms. Section A.3 covers implementation invariants, coverage monotonicity, and the local mean-shift interpretation.

B. Implementation and Experimental Protocol

Pages 18–21. Sections B.1–B.3 describe conditioning, guidance, and simulator integration. Sections B.4–B.6 specify training sources, matched planner evaluation, and metrics. Section B.7 records the operating configuration and hyperparameters.

C. Supplementary Results and Visualizations

Pages 22–31. Section C.1 presents the parameter-sensitivity results and their interpretation. Section C.2 groups visualizations of dataset-aligned recursion, a collision failure case, and the long-horizon sequence.

D. Limitations and Broader Impact

Page 32. Section D.1 discusses representation scope and planner specificity. Section D.2 addresses the intended safety-research use and responsible interpretation of generated scenarios.

Table S1: **Evidence map.** Main-paper evidence and corresponding appendix support.

Claim	Main-paper evidence	Appendix support
Intent and temporal coverage define when/where	Table 2	Algorithm S1, Secs. A.1, B.3
CDPG increases scenario difficulty for a fixed planner	Table 3	Algorithm S2, Secs. A.3, B.2
CDPG source yields higher mean WLR success	Table 4	Sec. B.5
Recursive execution mechanism	Section 3.2	Algorithm S1, Sec. B.3

Navigation and artifact status. Section references above are clickable. Supplementary equations, algorithms, figures, and tables retain their S-prefixed numbering. Section C.2 uses the supplied rollout visualizations. Figures S3 and S4 show one collision failure case; Figures S5–S9 show a separate 50 s run.

A METHOD FORMULATION, ALGORITHMS, AND STRUCTURAL ANALYSIS

A.1 EXTENDED NOTATION AND RECURSIVE FORMULATION

A.1.1 STATE, INTENT, AND FRONTIER SUPPORT

The persistent scene state is $S_t = (M_t, N_t)$, where M_t denotes committed lane geometry and N_t denotes the current agent state. The selected ego intent $I_t = [q_{t,0}, \ldots, q_{t,H_I}]$ is expressed in the current ego frame and contains displacement, speed, and heading along the selected future trajectory. The intent corridor

$$\mathcal{C}(I_t) = \mathrm{Tube}(I_t; r_{\mathrm{ego}} + r_{\mathrm{margin}})$$

defines the spatial support relevant to the current planner. The unsupported region $Q_t = \mathcal{C}(I_t) \backslash \Omega(M_t)$ therefore changes whenever either the planner intent or committed map changes.

Table S2: **Core notation used across the recursive generator.**

Symbol	Meaning
$S_t = (M_t, N_t)$	Persistent scene state at simulator time t.
I_t	Selected ego future trajectory produced by the current planner.
$\mathcal{C}(I_t)$	Intent corridor obtained by dilating the selected future trajectory.
Q_t	Unsupported portion of the intent corridor.
T_{cov}	Largest supported index of the discretized planner-intent prefix.
g_t	Generation trigger, equal to one when future support is insufficient.
ΔS_t	Generated scene increment $(\Delta M_t, N_t^{\mathrm{new}})$.
S_t^+	Post-commit persistent state before the simulator transition.
P_0, V_0	Frozen pretrained target planner and its critic/value function.
K	Number of static intent-aligned ego probes on the same decoded scene.
E_{hard}	Log-mean-exp aggregated multi-anchor counterfactual value drop.

The distinction between geometric distance and temporal coverage matters because a locally short map can still fully support a slow or turning intent, whereas a large map can fail immediately when the selected future leaves the represented corridor. The trigger therefore measures support in the planner's own temporal coordinates rather than in meters alone.

A.1.2 RECURSIVE KERNEL AND PERSISTENT STATE

For reference, the recursive generation kernel from the main paper can be written as

$$\Delta S_t \sim \mathcal{K}_{\theta,c}^{(g_t)}(\cdot \mid S_t, I_t, Q_t, T_{\mathrm{cov}}(M_t, I_t)), \tag{S1}$$

with $c \in \{\mathrm{Base}, \mathrm{CDPG}\}$. When $g_t = 0$, the kernel returns the null increment; when $g_t = 1$, a shared latent completion event proposes new lane and NPC content. Generated content is committed before the simulator transition, so later generation operates on the accumulated scene rather than reconstructing a fresh local scene.

This formulation separates three time scales that are easy to conflate in implementation: (i) simulator time t, (ii) future-intent index h, and (iii) diffusion reverse time τ. We use different symbols throughout the paper because the planner intent changes after simulator transitions, whereas diffusion time exists only inside one generation event.

Table S3: **Time variables and their roles.**

Variable	Domain	Role
t	simulator steps	Simulator state transition.
h	$\{0, \ldots, H_I\}$	Discrete index within the current planner intent.
τ	DDPM reverse steps	Denoising within one generation event.

A.2 ALGORITHMS

Algorithm S1 Intent-Conditioned Recursive Vector Scene Generation

Require: Initial committed world S_0, ego planner P, generator p_θ, CtRL-Sim NPC behavior policy π_{CtRL}, mode c

for $t = 0, 1, 2, \ldots$ until episode termination **do**
  $I_t \leftarrow P(S_t)$ ▷ selected ego future trajectory
  compute $T_{\mathrm{cov}}(M_t, I_t)$ and $Q_t = \mathcal{C}(I_t) \setminus \Omega(M_t)$
  $\Delta S_t \leftarrow \varnothing$
  if $T_{\mathrm{cov}}(M_t, I_t) < H_I$ **then**
    $\Delta S_t \sim \mathcal{K}_{\theta,c}(\cdot \mid S_t, I_t, Q_t, T_{\mathrm{cov}})$
  end if
  $S_t^+ \leftarrow \mathrm{Commit}(S_t, \Delta S_t)$ ▷ preserve committed content
  $H_t^{\mathrm{new}} \leftarrow \Gamma(N_t^{\mathrm{new}})$ ▷ register new-NPC history
  $A_t^{\mathrm{npc}} \sim \pi_{\mathrm{CtRL}}(\cdot \mid S_t^+, H_t \cup H_t^{\mathrm{new}})$
  $S_{t+1} \leftarrow \mathcal{T}(S_t^+, \mathrm{Exec}(I_t), A_t^{\mathrm{npc}})$
end for

Here N_t^{new} is the NPC component of ΔS_t, empty for a null increment. We use $\mathrm{Commit}(S_t, \varnothing) = S_t$ and $\Gamma(\varnothing) = \varnothing$. Scene updates and history registration precede CtRL-Sim's NPC action prediction; the planner is queried again after the simulator transition.

Algorithm S2 Multi-Anchor Counterfactual Direct Planner Guidance

Require: Noisy lane/object latents x_T, frozen critic V_0, observation adapter $\mathcal{O}_0$, mode c
Require: Static intent-aligned probes $\mathcal{A}_t = \{a_{t,k}\}_{k=1}^K$, temperature τ_a, guided steps $\mathcal{G}$

for $\tau = T, T-1, \ldots, 1$ **do**
  predict $\epsilon_\theta(x_\tau, \tau, h_t)$ and reconstruct $\hat{x}_{0,\tau}^N$ using Eq. 9
  if $\tau \in \mathcal{G}$ and $c = \mathrm{CDPG}$ **then**
    decode active current-event NPCs; fix existence decisions and observation indices
    for $k = 1, \ldots, K$ **do**
      $o_{\tau,k}^+ \leftarrow \mathcal{O}_0(S_t, I_t, a_{t,k}, D_\phi^N(\hat{x}_{0,\tau}^N))$
      $o_{\tau,k}^- \leftarrow \mathrm{Mask}_{\mathrm{new}}(o_{\tau,k}^+)$
      $\Delta V_{\tau,k} \leftarrow V_0(o_{\tau,k}^-) - V_0(o_{\tau,k}^+)$
    end for
    $E_{\mathrm{hard},\tau} \leftarrow \tau_a[\mathrm{logsumexp}_k(\Delta V_{\tau,k}/\tau_a) - \log K]$
    $g_\tau^{\mathrm{hard}} \leftarrow \mathrm{Normalize}(M_{\mathrm{new}} \odot \nabla_{x_\tau^N} E_{\mathrm{hard},\tau})$
    $g_\tau^{\mathrm{legal}} \leftarrow \mathrm{Normalize}(M_{\mathrm{new}} \odot \nabla_{x_\tau^N} B_{\mathrm{legal}})$
    $d_\tau \leftarrow \mathrm{ClipNorm}(g_\tau^{\mathrm{hard}} - g_\tau^{\mathrm{legal}})$
    $\epsilon_\tau^N \leftarrow \epsilon_\theta^N(x_\tau, \tau, h_t) - s_\tau \sqrt{1 - \bar{\alpha}_\tau}\, d_\tau$
  end if
  suppress direct guidance on lane and known-content latents
  perform the standard DDPM reverse update and re-clamp known content
end for
decode the scene increment and run frozen cleanup

A.3 Structural Analysis

A.3.1 Implementation Invariants and Enforcement

The following invariants record the boundaries enforced by the recursive state update and masked planner guidance.

Committed-content persistence. After a scene commit, its map tokens are marked as known. Every later reverse step re-clamps those latent values, preserving previously committed map geometry across subsequent generation events. CtRL-Sim predicts NPC actions, and the simulator updates vehicle states through the closed-loop transition.

Direct-gradient locality. The support mask M_{new} in Eq. 12 enforces the locality of the injected planner gradient. Non-guided slots follow the shared denoiser and known-content clamping path.

Multi-anchor paired counterfactual input. At a fixed reverse step, $o^{+}_{\tau,k}$ and $o^{-}_{\tau,k}$ share the same committed scene, intent, probe state, map, route, existing agents, and decoded scene. Only current-event NPC activity differs. The paired baseline therefore compares the added NPCs at a common probe state; the resulting gradient identity is derived below.

Base-mode denoising equivalence. Setting $s_\tau = 0$ removes the additive perturbation while preserving the scene prior, conditioning, masks, decoder, and simulator. This equivalence concerns the reverse-diffusion updates.

A.3.2 Monotonicity of Temporal Coverage

The temporal trigger is defined relative to both the map and the selected intent. One useful property is monotonicity with respect to committed map support.

Proposition 1 (Coverage monotonicity under map expansion)**.** *Let M_a and M_b be two committed map states satisfying $\Omega(M_a) \subseteq \Omega(M_b)$. For any fixed selected intent I, the temporal coverage obeys*

$$T_{\text{cov}}(M_a, I) \leq T_{\text{cov}}(M_b, I). \tag{S2}$$

Proof. For a fixed intent I, every prefix whose intent tube is contained in $\Omega(M_a)$ is also contained in $\Omega(M_b)$ because $\Omega(M_a) \subseteq \Omega(M_b)$. The feasible index set for M_a is therefore a subset of the feasible index set for M_b, and its maximum is no larger.

For a fixed intent, committing valid frontier content weakly increases temporal support as the represented map expands. In closed loop, the planner may select a different intent after the simulator transition, so $T_{\text{cov}}(M_{t+1}, I_{t+1})$ can still fall after map growth. This policy dependence distinguishes temporal coverage from a fixed distance-to-boundary trigger.

A.3.3 Local Mean-Shift Interpretation of Multi-Anchor CDPG

At a selected reverse step, condition on the current state x_τ and context h_t. The base object transition is Gaussian,

$$q_\theta(x_{\tau-1}^N \mid x_\tau, h_t) = \mathcal{N}(\mu_\theta(x_\tau, h_t), \Sigma_\tau)\,. \tag{S3}$$

Equation 12 defines a local direction d_τ by separately normalizing the masked hardness and legality gradients, subtracting them, and norm-clipping the result. Holding this direction fixed within the reverse step gives

$$\mu_{\theta,\mathrm{CDPG}} = \mu_\theta + \eta_\tau d_\tau, \tag{S4}$$

where η_τ absorbs the DDPM coefficients and guidance scale. This is the selected-step mean shift of the implemented sampler, including its composed direction. Separate normalization and clipping need not define the gradient of a single global scalar potential; no exact globally normalized energy-tilted target is asserted.

Counterfactual reweighting. For each probe, masking gives $\partial o_{\tau,k}^- / \partial x_\tau^N = 0$. Differentiating the log-mean-exp objective yields

$$\begin{gathered}\frac{\partial E_{\mathrm{hard},\tau}}{\partial \Delta V_{\tau,k}} = w_{\tau,k}, \qquad \nabla_{x_\tau^N} \Delta V_{\tau,k} = -\nabla_{x_\tau^N} V_0(o_{\tau,k}^+), \\ \nabla_{x_\tau^N} E_{\mathrm{hard},\tau} = -\sum_{k=1}^{K} w_{\tau,k} \nabla_{x_\tau^N} V_0(o_{\tau,k}^+).\end{gathered} \tag{S5}$$

For $K = 1$, $w_{\tau,1} = 1$ and the counterfactual baseline is constant with respect to the new-NPC latent, so the hardness update reduces to masked direct critic guidance. For $K > 1$, probe-specific baselines affect the cross-probe weights and can therefore change the combined direction; a common baseline cancels from the softmax. These are structural properties of the update. The matched experiments evaluate the complete CDPG package rather than isolating the empirical contribution of counterfactual weighting.

B Implementation and Experimental Protocol

B.1 Shared Latent Completion and Conditioning

SceneDemand uses the representation and checkpoints specified in Section 3.2. Ego-intent, unsupported-region, and temporal-coverage embeddings condition one shared denoising event whose mask distinguishes committed tokens from unknown frontier content, preserving joint lane–agent completion within a single generator.

Table S4: **Conditioning channels in the recursive generator.**

Channel	Source	Role
h_t^{diff}	shared diffusion context	Scene-prior conditioning.
$E_I(I_t)$	selected ego intent	Future-corridor alignment.
$E_Q(Q_t)$	unsupported region	Local generation support.
$E_T(T_{\text{cov}})$	temporal coverage	Urgency in planner time.
Mask m_t	committed/frontier partition	Clamp history and expose the frontier.

The generator objective treats recursive events uniformly with horizon-independent weighting. A rollout is formed by repeated application of the same transition/continuation mechanism, leaving its duration to the closed-loop recurrence.

Pseudo-time construction. Training examples are organized as committed-state / future-frontier pairs under one frozen preprocessing pipeline. The intent horizon, resampling, normalization, corridor margin, and temporal-support resolution used by the completed experiments are reported in Section B.7; the committed/frontier mask semantics are summarized in Table S4.

B.2 CDPG Implementation Details

At a guided reverse step, CDPG decodes $\hat{x}_{0,\tau}^N$ and constructs the K static intent-aligned observation pairs defined in Section 3.3. Their paired value drops enter Eq. 10 with $\tau_a = 0.25$.

Gradient paths and composition. Existence decisions and observation indices use stop-gradient; continuous new-NPC position, velocity, and heading remain differentiable. Hardness and legality gradients are separately normalized, subtracted, and norm-clipped to form the epsilon update in Eq. 12. Appendix A.3 derives the gradient paths and local mean shift.

Feasibility barrier. The frozen implementation groups its differentiable feasibility penalties as

$$B_{\text{legal}} = b_{\text{ego}} + b_{\text{agent}} + b_{\text{lane}} + b_{\text{kin}},$$

with unit weights. The terms penalize ego clearance, inter-agent clearance, lane support, and kinematic violations; the kinematic term applies the recorded heading and speed limits. The frozen thresholds are 2 m ego clearance, 1 m inter-agent clearance, 3 m on-lane distance, 0.75 rad heading error, and a 0–45 m/s speed range, with temperature 0.5 in the differentiable barrier. These values are held fixed during the parameter sweeps in Section C.1.

B.3 Online Commit and Simulator Integration

Each scene increment registers its lane geometry and new-NPC states together before simulation advances. The atomic, non-overwrite commit preserves known map content and keeps agent bookkeeping synchronized. For a null increment, the pre-event scene is unchanged and no new spawn history is introduced; subsequent simulation still uses the persistent scene.

Table S5: **Online recurrence interfaces.**

Stage	Input	Output / invariant
Planner query	Current S_t	I_t
Coverage check	M_t, I_t	T_{cov}, Q_t, g_t
Shared completion	Committed latents and conditions	$\Delta M_t, N_t^{\text{new}}$
Atomic commit	Scene increment	Persistent S_t^+
Spawn-history adapter	N_t^{new}	CtRL-Sim-compatible H_t^{new}
NPC behavior prediction	$S_t^+, H_t \cup H_t^{\text{new}}$	CtRL-Sim NPC actions
World transition	Ego and NPC actions	S_{t+1}

New NPCs are registered before the next behavior-model step. CtRL-Sim predicts their subsequent actions, which the simulator uses to advance surrounding-vehicle states. The scene-update and history-registration settings are common to Base and CDPG.

B.4 Dataset and Training-Source Construction

Table S6 summarizes the four source sets constructed under the dataset-aligned protocol in Section 4.1. WOMD supplies the real motion corpus (Ettinger et al., 2021); the other sources use generated scenes.

Table S6: **Planner-training sources used in the main comparison.**

Training source	Environments	Duration	Purpose
WOMD / Waymo Log Replay	500	9 s	Real-data reference.
Scenario Dreamer	500	9 s	Generative baseline.
SceneDemand-Base	500	9 s	Recursive generator without CDPG.
SceneDemand-CDPG	500	9 s	Planner-targeted recursive generator.

Each source set contains a single source type. The separate 50-s rollout is analyzed qualitatively outside the logged-data comparison.

Temporal alignment. WOMD clips span 9.1 s (91 steps at 10 Hz) (Ettinger et al., 2021), and Scenario Dreamer's Waymo-matched 55 m setting uses a 9 s episode timeout (Rowe et al., 2025). The shared 9 s evaluation window aligns temporal exposure across training sources; it is an evaluation choice, not a limit of the recursive formulation.

B.5 Planner Training and Evaluation Protocol

Frozen controls. All trained policies start from pretrained planner P_0 and use the PPO configuration in Section B.7. Each policy receives 500k interactions on 500 environments, with policy seeds $\{0, 1, 2\}$. Evaluation uses 500 held-out Waymo Log Replay scenes, evaluator seeds $\{0, 1, 2\}$ per scene, and a 9-s horizon (90 simulation steps). These settings apply to every training source.

Checkpoint selection. The same checkpoint-selection rule is used for all four sources, and every reported planner result uses the policy checkpoint at 500k interactions. The cross-environment evaluation sets are Waymo Log Replay, Dreamer-Neg (the Scenario Dreamer set), and SD-Challenge (the SceneDemand set). A direct VectorWorld policy comparison is not included because its horizon, evaluator, and simulation backend differ from this matched protocol.

B.6 Evaluation Metrics

This section gives the formal definitions of the scene-quality, recursive-continuation, and planner-evaluation metrics introduced in Section 4.1.

Scene-quality metrics. FD summarizes distributional discrepancy between generated and reference scene features under the shared evaluation implementation; we report the same scalar definition for every row evaluated by our code. EPD measures geometric consistency across directed lane-successor edges. For the evaluated successor set $\mathcal{E}_{\text{succ}}$,

$$\text{EPD} = \frac{1}{|\mathcal{E}_{\text{succ}}|} \sum_{(i,j)\in\mathcal{E}_{\text{succ}}} \left\| p_i^{\text{end}} - p_j^{\text{start}} \right\|_2 . \tag{S6}$$

Static collision rate is the percentage of generated initial states that violate the shared overlap test. Agent JSD compares generated and reference histograms of the evaluated agent attributes. For normalized histograms P and Q and $M = (P + Q)/2$,

$$\text{JSD}(P, Q) = \tfrac{1}{2}\,\text{KL}(P\|M) + \tfrac{1}{2}\,\text{KL}(Q\|M). \tag{S7}$$

The main table reports the aggregate Agent JSD produced by the frozen evaluator.

Recursive-continuation metrics. Temporal Coverage is computed from the planner-time support defined in the main paper and averaged after clipping to the current intent horizon. For evaluated states $\{(M_i, I_i)\}_{i=1}^{N}$,

$$\text{TemporalCoverage} = \frac{100}{N} \sum_{i=1}^{N} \min\left(\frac{T_{\text{cov}}(M_i, I_i)}{H_I}, 1 \right). \tag{S8}$$

MSR is the percentage of evaluated states flagged by the frozen map-starvation predicate; keeping that predicate fixed across variants prevents the trigger ablation from changing the metric definition itself. Five-commit completion is the fraction of source traces that reach five recursive commits. Route-Lane Utilization measures the fraction of generated lane support used by the committed ego route, while Endpoint Gap and Successor Continuity diagnose geometric and topological consistency across recursive lane joins. These metrics are reported together because high temporal coverage alone can be obtained by generating excessive or structurally poor map support.

Planner-facing and policy-learning metrics. For frozen-planner evaluation on generated scenarios, we report collision, off-route, and success rates, route progress, and the multi-probe value-drop score E_{hard} defined in Eq. 10. Lower success and progress indicate greater difficulty for the target planner. Policy-learning evaluation additionally reports timeout. Success, collision, off-route, and timeout use the common Waymo Log Replay termination logic; progress is the traveled route distance at termination. Trained-policy results are summarized as mean$\pm$sample standard deviation over three policy seeds, with success interpreted jointly with collision and progress.

B.7 Configuration and Hyperparameter Reporting

The operating configuration used in all headline experiments is summarized below. Section C.1 reports the corresponding parameter-sensitivity experiments.

Table S7: **Generator and simulator settings used in the completed experiments.**

Component	Field	Recorded setting
Scene representation	Representation / checkpoint family	Released vector latents and checkpoints (Section 3.2).
Diffusion	Reverse steps	DDPM-100.
Known content	Treatment	Clamped at every denoising step.
Base mode	Planner guidance	$s_\tau = 0$.
CDPG mode	Planner guidance	$s_\tau > 0$ on the selected guided steps.
CDPG support	Direct gradient	Active new-NPC slots only.
Lane / known tokens	Direct CDPG gradient	Zero.
NPC behavior, SD	CtRL-Sim behavior tilt	+10.
Dreamer Normal	CtRL-Sim behavior tilt	+10.
Dreamer Adv.	CtRL-Sim behavior tilt	-10.
Quantitative SD horizon	Rollout	9 s / 5 commits.

The exact operating values used by the completed experiments are given here in text; file paths, checkpoint hashes, and release-engineering identifiers are reserved for the accompanying release materials.

Intent, coverage, and query settings. The intent adapter uses a $3.0\,\mathrm{s}$ horizon (30 steps at 10 Hz), 16 resampled waypoints, and embedding width 80. Relative position is normalized by 32 m, speed by 30 m/s, and heading is represented with $\cos / \sin$. The selected corridor margin is 5.0 m, chosen from the 3.0 / 4.0 / 5.0 m sweep, and temporal support is evaluated at $0.1\,\mathrm{s}$ resolution.

Shared latent diffusion model. The shared LDM is optimized with AdamW at learning rate 2.5×10^{-6}, weight decay 10^{-5}, 500 warmup steps, a constant schedule, and 12k training steps. Training uses batch size 16 per GPU on 2×A100 GPUs with bf16 mixed precision. The selected object-loss coefficient is $\lambda_{\mathrm{obj}} = 0.05$, chosen from $\{0.05, 0.10, 0.20\}$.

CDPG settings. The guidance scale is $s_\tau = 0.5$; the other tested values are 1.0 and 2.0. Guidance is applied at reverse steps $\{70, 55, 40\}$; the sparse and dense alternatives are $\{55\}$ and $\{85, 70, 55, 45, 40\}$, respectively. Hardness and legality gradients are separately normalized before subtraction and norm clipping at 1.0.

Feasibility settings. The feasibility barrier uses ego clearance 2 m, inter-agent clearance 1 m, on-lane distance 3 m, heading error 0.75 rad, speed range 0–45 m/s, temperature 0.5, and four unit-weight terms.

PPO settings. Planner training uses Adam with learning rate 3×10^{-4}, $\gamma = 0.99$, GAE $\lambda = 0.95$, PPO clip 0.2, entropy coefficient 10^{-4}, value coefficient 0.3, and maximum gradient norm 0.5. Each update uses rollout size 16,384, minibatch size 16,384, and 4 epochs, with a total budget of 500k interactions.

C SUPPLEMENTARY RESULTS AND VISUALIZATIONS

C.1 PARAMETER SENSITIVITY ABLATIONS

The parameter ablations below examine the sensitivity of the completed experiments to the operating configuration. Each sweep varies one parameter at a time while keeping the remaining protocol fixed. Bold parameter values identify the settings used across all headline comparisons.

Table S8: **CDPG guidance-parameter ablation.** Guidance strength and guided reverse steps are varied independently. Bold parameter values identify the settings used in the headline experiments.

Parameter	Setting	Actual value	Success (%)↓	Progress (m)↓	E_{hard} ↑
s_τ	Reference	**0.5**	18.18	6.74	0.0595
	Mid	1.0	27.27	5.14	0.0626
	High	2.0	55.56	8.66	0.0577
$\mathcal{G}$	Sparse	$\{55\}$	19.44	4.99	0.0592
	Reference	$\{\mathbf{70}, \mathbf{55}, \mathbf{40}\}$	27.27	5.14	0.0626
	Dense	$\{85, 70, 55, 45, 40\}$	15.15	5.09	0.0585

Table S9: **Recursive-generation parameter ablation.** Corridor margin and object-loss coefficient are varied independently. Bold parameter values denote the selected settings used in the final experiments.

Parameter	Setting	Actual value	Temporal Cov. (%)↑	MSR (%)↓	5-Commit (%)↑	EPD (m)↓
r_{margin}	Small	3.0 m	97.82	0.155	56.00	0.295
	Mid	4.0 m	96.23	0.344	44.00	0.285
	Selected	**5.0 m**	98.02	0.109	60.00	0.230
λ_{obj}	Selected	**0.05**	97.83	0.132	72.00	0.267
	Mid	0.10	96.23	0.344	44.00	0.285
	High	0.20	97.33	0.246	48.00	0.367

Reading the CDPG sweep. Table S8 shows that larger guidance strength does not monotonically increase closed-loop difficulty: planner success rises from 18.18% at $s_\tau = 0.5$ to 55.56% at 2.0, while 1.0 yields the lowest progress and largest E_{hard} within that sweep. Thus, maximizing the critic diagnostic and minimizing successful completion are not interchangeable objectives. In the guided-step sweep, dense guidance gives the lowest success, sparse guidance the lowest progress, and the reference schedule the largest value drop. No schedule is uniformly strongest on all three metrics. These results motivate treating the headline configuration as a fixed operating choice, rather than assuming that more guidance steps or greater strength necessarily produce more useful challenges.

Reading the recursive-generation sweep. In Table S9, the 5.0 m corridor gives the best coverage, lowest starvation, highest completion, and lowest EPD among the tested margins. This suggests that sufficient frontier width can support both continuation and boundary consistency, although the non-monotonic 3.0–4.0 m results caution against a simple larger-is-always-better rule. The lowest object-loss weight, $\lambda_{\text{obj}} = 0.05$, likewise performs best on all reported metrics in its sweep, including 72% five-commit completion. Higher object weighting does not improve recursive utility and worsens endpoint consistency, highlighting the need to balance the lane and agent branches for continuation rather than emphasizing object prediction alone.

Operating configuration. The final configuration is $r_{\text{margin}} = 5.0$ m, $\lambda_{\text{obj}} = 0.05$, $s_\tau = 0.5$, and $\mathcal{G} = \{70, 55, 40\}$. This configuration is held fixed across all headline experiments and matched comparisons.

Every sweep uses the same horizon-independent recursive-event objective, and rollout length arises from repeated application of the same transition mechanism.

C.2 Supplementary Visualizations

The following figures examine local recursive continuation, the approach to a collision failure, and a continuous 50 s rollout. The local snapshots, accumulated map, and temporal diagnostics provide complementary views of scene growth and vehicle interactions.

C.2.1 Dataset-Aligned 9-s Recursive Continuation

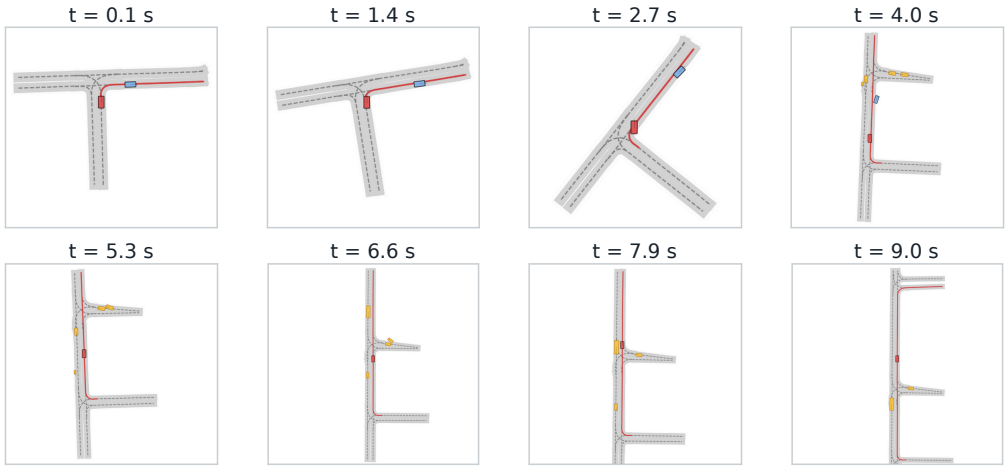

Figure S1: **Recursive continuation across the dataset-aligned 9-s evaluation window.** Time-ordered snapshots from 0.1 to 9.0 s show the evolving road context, ego route/intent, and surrounding vehicles during one continuous rollout.

Route-aligned scene evolution. Figure S1 follows the ego from an initial junction into a road corridor with successive side branches. The early panels show the route turning through the junction; the later panels place the ego farther along the continuing road, with additional connections visible around its selected future. Across these changes, the red route remains aligned with the represented road. The sequence makes the role of intent conditioning concrete: the relevant scene context advances with the ego and the direction of its planned motion.

Road support and surrounding traffic. Vehicles are visible both along the continuing corridor and near connecting branches. Their positions relative to the route change as the ego progresses, so the rollout contains evolving traffic context as well as road geometry. The sequence reaches the 9 s evaluation horizon while continuing to display road support ahead of the ego. Figure S2 examines individual generation events within this local continuation setting, making the added road segments and vehicles easier to distinguish from the previously available context.

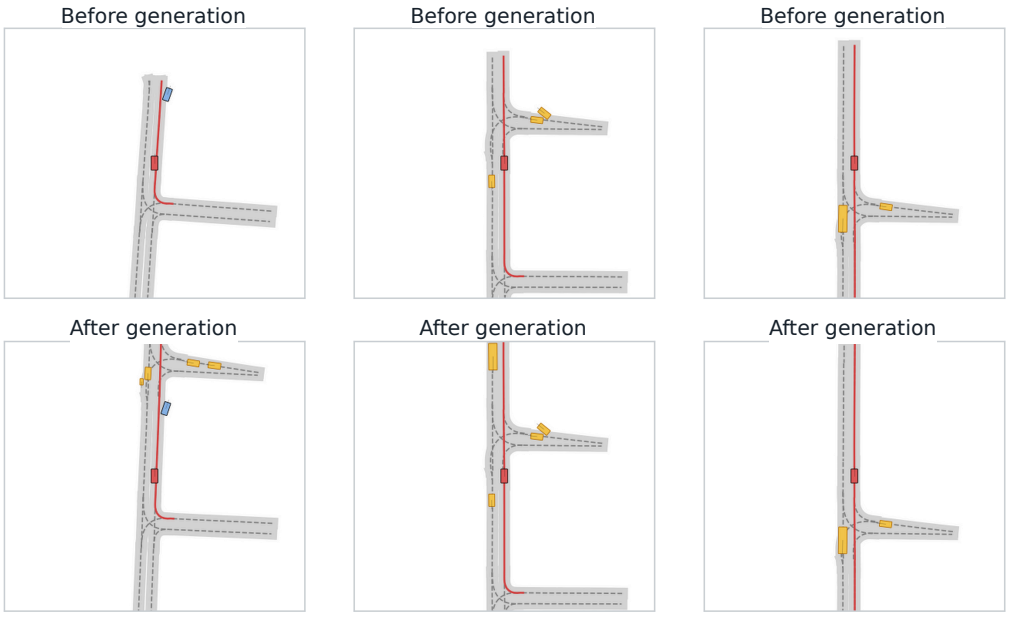

Figure S2: **Commit-level views of recursive frontier growth.** Each column shows the scene before generation (top) and after generation (bottom), revealing local road continuation and newly introduced vehicles around the ego route.

Local completion at the frontier. In Figure S2, each column isolates one generation event. The top and bottom views share recognizable road segments, providing a reference for locating the new content. The first column exposes a continuation beyond the earlier road endpoint, including an additional branch near the upper part of the scene. The other columns show local completion around an existing junction and the continuing route. Together, the three events illustrate how an update extends the usable neighborhood while retaining its connection to the road already surrounding the ego.

Joint addition of lanes and vehicles. The lower panels contain newly visible vehicles alongside the completed road context. Their placement gives the added geometry an immediate traffic interpretation: the continuation includes agents situated on or near road segments relevant to the ego's future motion. The spatial relationship among the route, the connecting lanes, and the vehicles is therefore central to reading each update. These views expose the local scene increments that underpin the longer progression in Figure S1, linking a single generation-and-commit event to the evolving rollout.

C.2.2 Collision Failure Case

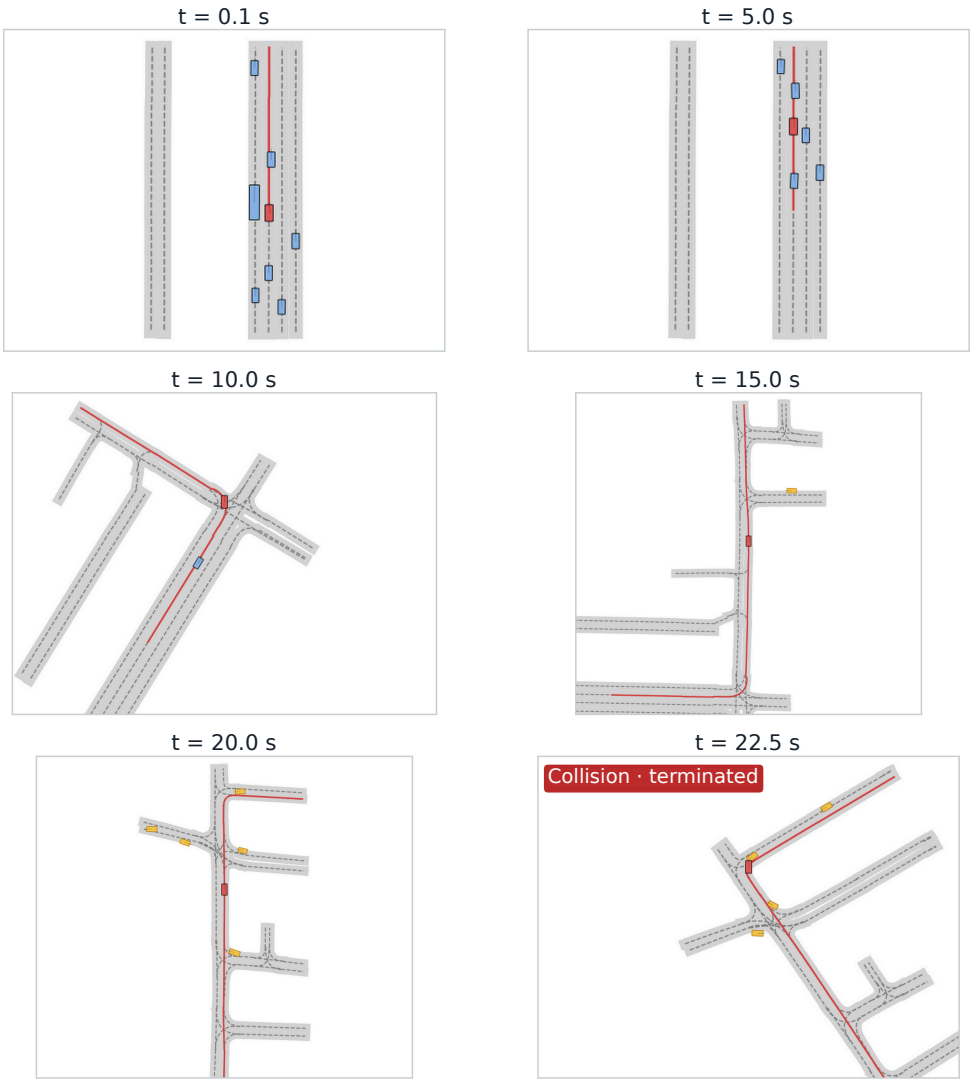

Figure S3: **A failed closed-loop rollout ending in a collision.** Snapshots from 0.1 to 22.5 s show the progression of a single run. The terminal panel marks collision termination; Figure S4 examines the same event.

Progression toward the failure. Figure S3 begins on a straight road with several surrounding vehicles. The subsequent views show the ego following its route through changing junction geometry and into a corridor with multiple side connections. At 20.0 s, the ego is approaching the interaction region examined in the close-ups. The final snapshot records collision termination at 22.5 s.

Scene context at termination. The terminal view retains the continuing road and the ego route through the collision area. The failure occurs during an interaction with traffic in this represented road context. Figure S4 follows the involved generated vehicle over the final 2.5 s, connecting the rollout-level outcome to the local approach between the two vehicles.

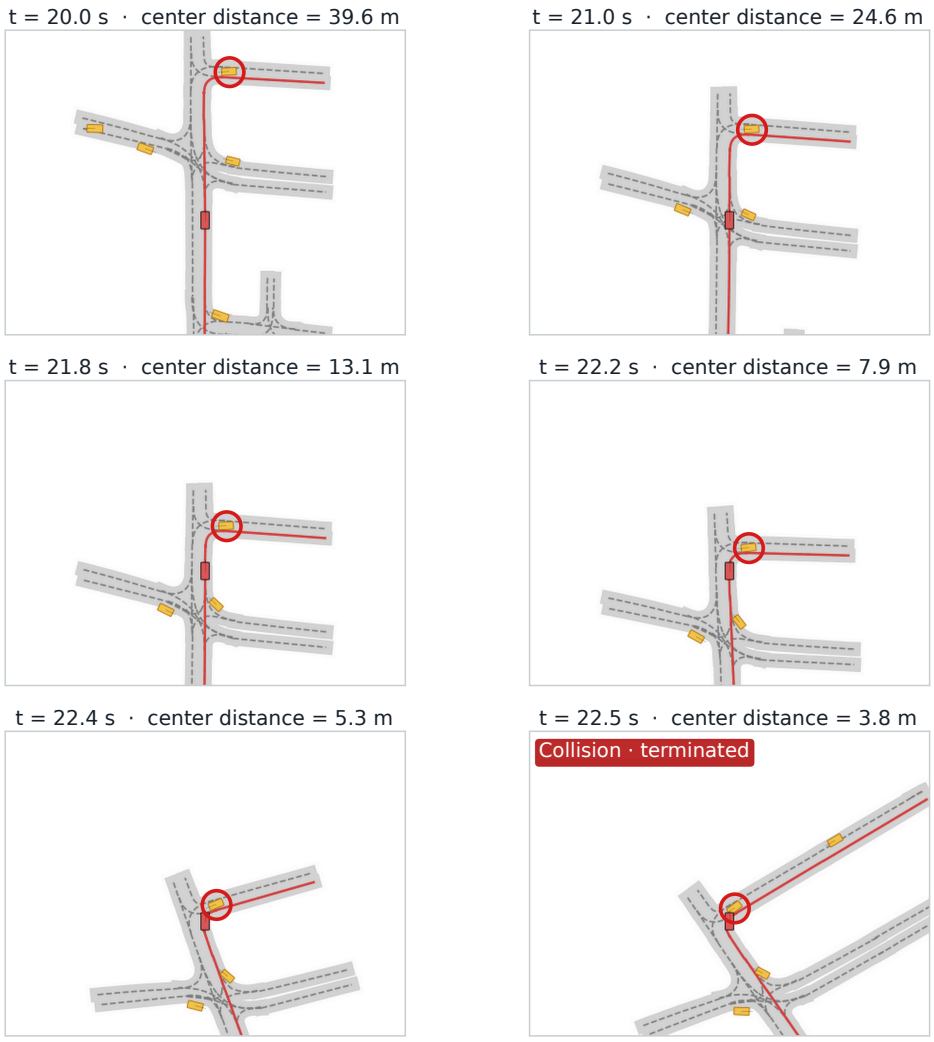

Figure S4: **Close-ups of the collision failure in Figure S3.** The red ring identifies generated NPC 69, committed at 19.5 s. From 20.0 to 22.5 s, the annotated center distance decreases from 39.6 to 3.8 m, ending at the marked collision.

Closing separation before contact. Figure S4 tracks the same generated NPC across all six panels. The annotated center distance decreases from 39.6 to 24.6, 13.1, 7.9, 5.3, and finally 3.8 m. The red ring links the vehicle's position across the changing views and makes its approach relative to the ego visible throughout the final interaction.

Timing and collision geometry. NPC 69 is committed at 19.5 s, and the collision is logged 3.0 s later. The close-ups connect its insertion into the scene with the subsequent approach and terminal contact. The final 3.8 m annotation measures center-to-center separation; the vehicles' finite footprints determine the contact geometry shown in the terminal panel.

C.2.3 Long-Horizon 50-s Recursive Generation

Figures S5–S9 examine a continuous 50 s rollout through interval snapshots, accumulated geometry, and temporal diagnostics.

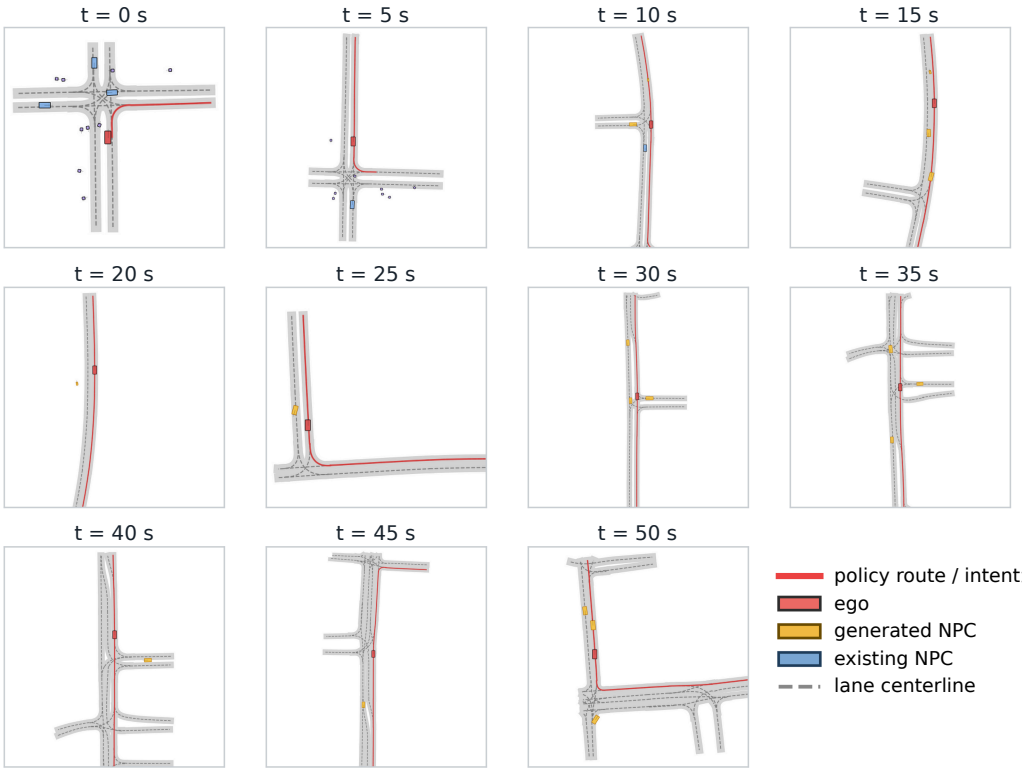

Figure S5: **Continuous 50-s recursive rollout.** Snapshots at 5 s intervals follow one continuous run. The legend distinguishes the policy route/intent, ego vehicle, generated NPCs, existing NPCs, and lane centerlines.

Road and route evolution over the full run. Figure S5 follows the sequence from the initial junction to the road configuration reached at 50 s. Between these endpoints, the local views contain straight sections, bends, and successive side connections. The route turns through the junction at the beginning, follows the continuing corridor, and encounters further turns later in the run. This temporal progression shows recursive scene support adapting to several changes in the local driving context.

Traffic within the evolving scene. The initial views include existing traffic, while generated NPCs remain visible in later parts of the sequence. Their locations change relative to the ego and nearby road branches as the run advances. The common legend makes road continuation and traffic evolution readable together. Figures S6 and S7 enlarge the first and second halves; Figures S8 and S9 connect these local checkpoints to the accumulated map and the timing of scene extensions.

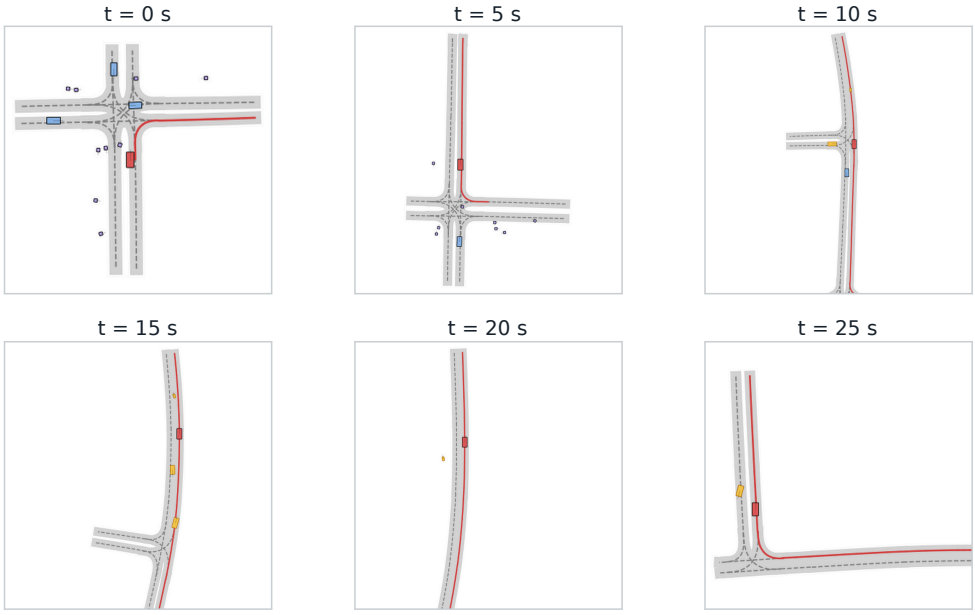

Figure S6: **Long-horizon recursive rollout, 0–25 s.** Enlarged snapshots of the first half of the continuous sequence in Figure S5, showing the evolving local road context and traffic around the ego route.

From the initial junction to the next turn. Figure S6 begins with the ego negotiating a junction surrounded by existing traffic. The 5–20 s panels follow its progression along the continuing corridor, including a gradual bend and several side connections. At 25 s, the displayed route follows a pronounced turn into the adjoining road. The enlarged views make the lane support around each change of direction visible, including the connection between the approach and the outgoing segment.

Evolution of the local traffic context. The concentration of vehicles around the initial junction gives way to a sparser configuration along the later corridor. Generated NPCs appear near the ego route and connecting roads, placing moving traffic within the extended scene. These changes show how the local environment evolves through both geometry and agent placement. The 25 s panel provides the shared endpoint with Figure S7, so the turn reached here can be followed directly into the second half of the rollout.

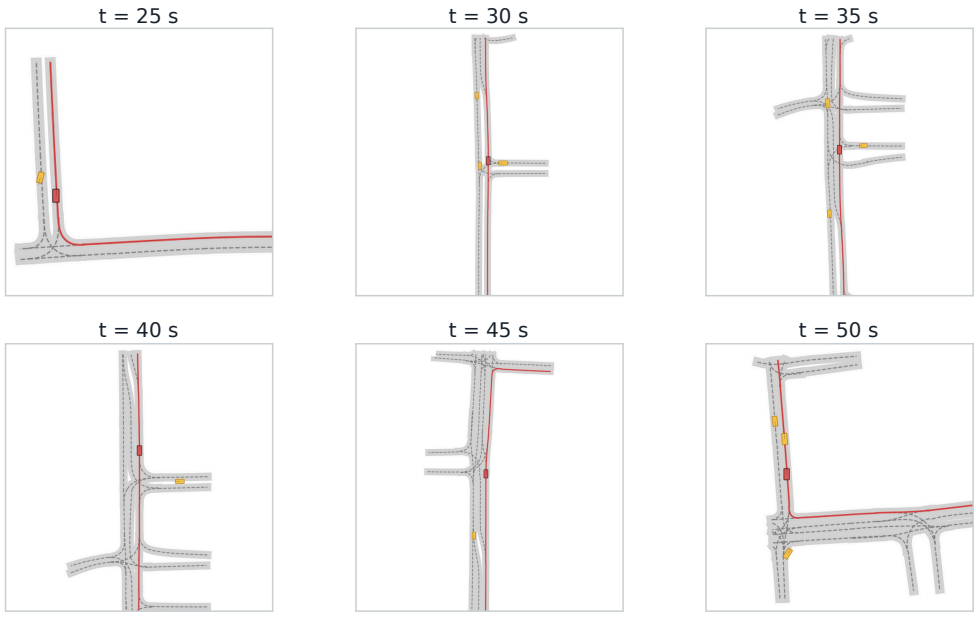

Figure S7: **Long-horizon recursive rollout, 25–50 s.** Enlarged snapshots continuing the same run shown in Figure S6. The shared 25 s snapshot connects the two halves of the sequence.

Repeated continuation in the second half. Figure S7 starts at the turn reached at 25 s and follows the ego through the next section of the scene. Subsequent panels reveal further side branches and changing junction layouts along the continuing road. Near 50 s, the route turns through another connected road configuration. The sequence thus extends the progression from the first half through additional road context and another change in the ego's selected direction.

Local support near later junctions. The route remains embedded in the displayed lane structure as the surrounding branches change, and generated NPCs are visible along the later corridor. The enlarged panels expose how the ego's immediate driving neighborhood continues to contain both road support and traffic near the end of the run. Read together with Figure S6, they show the full temporal progression; the accumulated map in Figure S8 places these successive local road configurations into a common spatial frame.

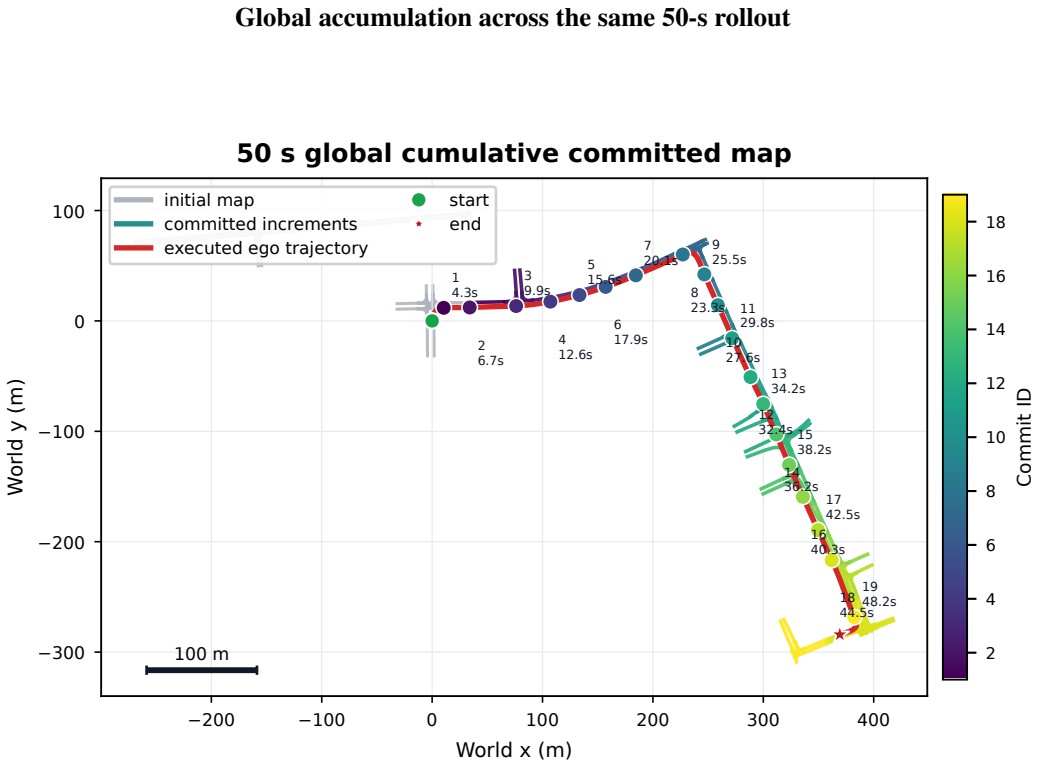

Figure S8: **Global accumulation for the 50-s rollout.** The initial map, color-coded committed increments, and executed ego trajectory are overlaid in one global coordinate system. Commit labels and times locate successive extensions; start and end markers delimit the run.

Accumulated geometry along the executed route. Figure S8 shows the initial road context near the start marker and the successive increments extending across the world frame. The ego trajectory first advances toward increasing world x, then bends toward decreasing world y, and reaches the final road section near the end marker. The colored increments follow this progression while also adding side connections. Earlier and later road sections remain visible together, making the spatial continuity of the accumulated scene explicit.

Where and when extensions are committed. The map identifies 19 commits, beginning at $4.3\,\mathrm{s}$ and ending at $48.2\,\mathrm{s}$. Their numbered markers and timestamps connect the spatial growth to the rollout's temporal order. The locations span the initial progression, the main bend, and the later corridor, showing repeated additions throughout the driven scene. Figure S9 presents the same event sequence as a cumulative curve and relates it to temporal coverage and the active traffic population.

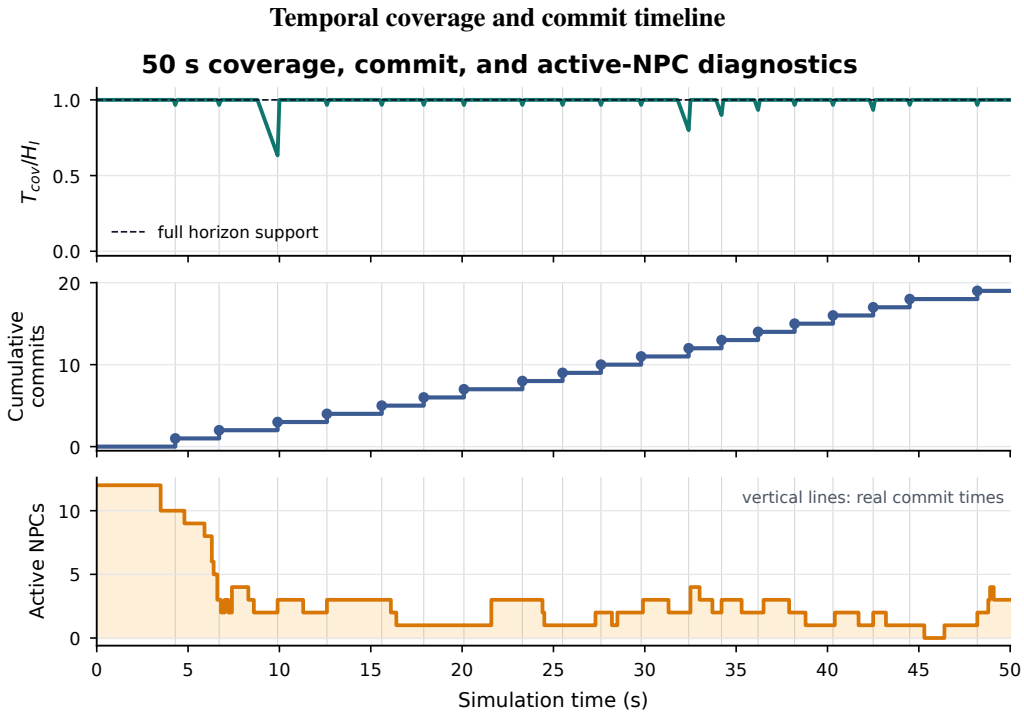

Figure S9: **Coverage and commit diagnostics for the same 50-s rollout.** Normalized temporal coverage T_{cov}/H_I (top), cumulative commits (middle), and active-NPC counts (bottom) are plotted against simulation time. Vertical lines mark commit times; the dashed line denotes full-horizon support.

Temporal support across successive extensions. The upper trace in Figure S9 stays close to full-horizon support for most of the run. Brief reductions, including the more pronounced early dip near 10 s, are followed by a return toward full coverage. These changes make the evolution of available support visible between the regularly sampled snapshots.

Commit timing over the rollout. The middle trace increases in discrete steps and reaches 19 commits, consistent with the labels in Figure S8. The intervals between steps vary as the run progresses. The vertical event markers align this extension history with the coverage trace, providing a temporal view of the repeated updates that build the accumulated road scene.

Changing active traffic population. The bottom trace begins with the largest active-NPC population, declines during the early part of the run, and then fluctuates at a lower level with several local increases. This pattern corresponds to the transition from the busy initial junction to the sparser traffic visible in the later snapshots. Together, the three traces characterize the displayed run through its available planning support, accumulated extensions, and evolving active traffic.

D LIMITATIONS AND BROADER IMPACT

D.1 LIMITATIONS

Representation scope. The current recursive world state contains lane centerlines and vehicle, pedestrian, and cyclist states. Richer traffic-light and map semantics are not explicitly modeled, so the generated scenes do not cover the full semantic complexity of real driving environments. Temporal coverage measures map support for the selected ego trajectory within this representation; it does not by itself establish compliance with traffic signals or other unmodeled road rules. The reported results should therefore be interpreted within the scope of the current lane–agent abstraction.

Planner specificity. CDPG is tied to the frozen planner P_0 and its value representation. Interactions that challenge this target need not be equally difficult or informative for other planners, and the present results do not establish transfer across planner architectures.

D.2 BROADER IMPACT

Use in safety research. Planner-oriented generation focuses simulation on interactions that expose policy weaknesses and may support targeted planner training and testing. Generated scenarios are sandboxed stress tests, not evidence that a deployed system is safe. Feasibility constraints, held-out real-data evaluation, and real-world validation provide complementary evidence; simulation results should not replace the latter.

Responsible use. The framework should be used to identify and mitigate failure modes in controlled simulation, not to reproduce hazardous interactions on public roads. Dataset provenance and scenario-generation settings should be documented so that observed failures and training gains can be interpreted in context. Evaluation across different road configurations and traffic participants can help identify gaps in scenario coverage before deployment.