UniDBO: A Unified Dual-Branch One-Step Denoising Framework for Autonomous Driving Scenario Generation

Da Zhao Yuhang Chen Jialin Fan Jian Sun Jie Sun*
College of Transportation & Key Laboratory of Road and Traffic Engineering
of Ministry of Education, Tongji University, Shanghai, China
{da_zhao,yuhangchen,jialin_fan,sunjian,jie_sun}@tongji.edu.cn

Abstract

Traffic scenario generation is essential for training, testing, and safety validation of autonomous vehicles (AVs), especially when real-world data coverage is limited. Existing methods mostly face a three-way trade-off among open-loop trajectory fitting, closed-loop simulation robustness, and inference efficiency. Autoregressive methods are typically efficient in open-loop fitting but susceptible to error accumulation and policy-induced state-distribution shift in long-horizon closed-loop rollouts. Diffusion-based methods provide strong multi-modal behavior modeling and competitive generation quality, but iterative denoising substantially increases inference latency and limits simulation throughput. Moreover, repeated refinement under a fixed scene condition is not fully aligned with the continuous feedback and frequent replanning required by closed-loop simulation. To address this challenge, we propose **UniDBO**, a **Uni**fied **D**ual-**B**ranch **O**ne-step denoising framework for autonomous driving scenario generation, which uses a shared scene encoder and two jointly optimized complementary branches with distinct roles—a continuous-scale branch (CS) for open-loop trajectory fitting and a discrete high-noise branch (DHN) for robust closed-loop simulation. Through joint training, UniDBO coordinates open-loop fitting and closed-loop robustness while preserving one-step inference efficiency. We evaluate UniDBO with different benchmarks, covering in-domain open-loop and closed-loop evaluation, zero-shot cross-domain generalization, and system-level closed-loop AV testing in unseen scenarios. Results show that UniDBO improves the open-loop/closed-loop balance while maintaining high inference efficiency, and achieves competitive zero-shot generalization on the evaluated benchmarks. Code is available at `https://canalone.github.io/UniDBO-page/`.

1 Introduction

Training, testing, and safety validation of autonomous driving systems require large-scale, realistic, and diverse traffic scenarios [1–3]. Although public benchmarks such as WOMD and INTERACTION have substantially advanced motion forecasting and traffic modeling [4, 5], real-world data coverage remains limited and cannot fully meet the needs of autonomous driving development and validation [6, 7]. This limitation makes scalable data-driven scenario generation essential to autonomous driving development and validation [6, 8–10], with accurate behavior modeling, stable long-horizon closed-loop simulation, and efficient inference for large-scale testing together constituting the fundamental requirements for such generators [6, 8, 11].

Prevailing data-driven traffic generation methods can be mainly categorized as autoregressive and diffusion-based paradigms. Autoregressive methods, such as SMART, BehaviorGPT, and CAT-K,

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

typically offer efficient generation, while their sequential generation mechanism can accumulate prediction errors over long-horizon closed-loop rollouts and induce state-distribution shift [12–15]. Diffusion-based methods, including MotionDiffuser, Scenario Diffusion, and VBD, provide strong capacity for jointly modeling continuous multi-agent behaviors, but their iterative denoising process requires repeated denoiser evaluations at each replanning step and substantially increases inference latency [16–21]. Moreover, repeated refinement under a fixed scene condition is not fully aligned with closed-loop simulation, where traffic states continuously evolve through feedback and frequent replanning. As illustrated in Figure 1(a,b), these complementary strengths and limitations indicate that, although existing methods retain competitive open-loop prediction performance, they still struggle to jointly achieve robust closed-loop simulation and efficient inference.

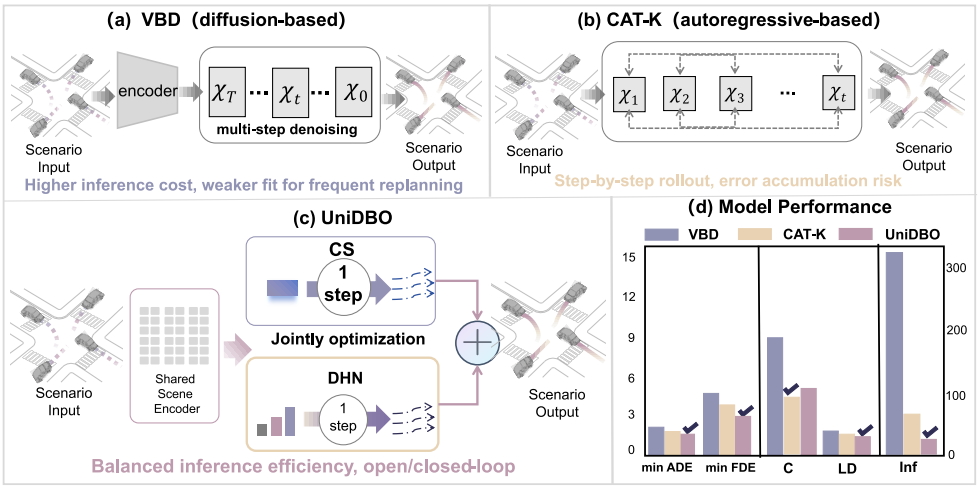

Figure 1: Trajectory fitting accuracy, simulation robustness, and inference efficiency trade-off of different scenario-generation paradigms. Panels (a–c) compare multi-step diffusion, autoregressive modeling, and UniDBO, respectively. Panel (d) reports representative metrics on WOMD/Waymax from Table 1; check marks indicate the best reported value for each metric. C, LD, and Inf denote Collision rate, Log Divergence, and inference latency (ms/step), respectively. Open-loop and closed-loop metrics are plotted on the left y-axis, and Inf is plotted on the right y-axis.

Having considered these challenges, we aim to replace iterative diffusion sampling with direct one-step generation while preserving its joint multi-agent modeling capability. Recent one-step generation studies have shown that, when the training objective is explicitly designed for direct generation, substantially reducing the number of denoising steps does not necessarily sacrifice generation quality [22–26]. For closed-loop multi-agent simulation, however, the model must jointly generate multi-agent controls in a single denoising pass, while prediction errors can further affect subsequent interactions through closed-loop feedback. This leads to a key design requirement, that the closed-loop branch should be specialized for direct control generation, while the shared scene representation should still benefit from supervision over a broader range of noise levels. Prior work on joint learning further shows that shared representations with task-specific branches can integrate complementary supervision while preserving objective-specific specialization [27–29].

Motivated by these observations, we propose **UniDBO** (Figure 1(c)), a unified dual-branch one-step denoising framework for multi-agent scenario generation. UniDBO jointly trains two complementary branches through a shared scene encoder: a discrete high-noise branch (DHN) with a progressive high-noise curriculum for direct one-step recovery in closed-loop simulation, and a continuous-scale branch (CS) over the configured continuous noise range to provide auxiliary supervision for the shared scene representation and support open-loop trajectory fitting. At inference, UniDBO uses mode-aware activation: CS for open-loop evaluation and DHN for closed-loop simulation, with each query activating only its designated branch and requiring a single denoiser evaluation. As shown in Figure 1(d), UniDBO substantially improves inference efficiency while retaining competitive open-loop recovery accuracy and strong closed-loop robustness. Our contributions are threefold:

- We introduce **UniDBO**, a mode-aware dual-branch one-step denoising framework that balances trajectory fitting, closed-loop robustness, and inference efficiency.

- We propose progressive high-noise curriculum learning with joint dual-noise supervision through a shared scene encoder. DHN specializes in direct one-step recovery for closed-loop simulation, while CS provides complementary supervision over the configured continuous noise range, strengthening the shared representation while preserving branch-specific specialization without additional inference cost.
- We conduct systematic mechanism analyses and comprehensive experiments on WOMD/Waymax, INTERACTION, and OnSite. Results show that UniDBO achieves a favorable balance between closed-loop robustness and inference efficiency while retaining competitive open-loop performance and maintaining cross-domain generalization and system-level closed-loop testing performance.

2 Related Work

2.1 Multi-Agent Traffic Behavior Generation

Data-driven traffic generation aims to generate realistic and diverse traffic scenarios [30] and model socially consistent multi-agent interactions under closed-loop feedback [1, 6, 31, 32]. Prevailing approaches primarily fall into autoregressive and diffusion-based paradigms. Autoregressive methods factor the joint future into sequential motion-token or trajectory-patch predictions [12–14, 33]. SMART adopts next-token prediction [12], CAT-K adapts tokenized models to closed-loop rollout [14], and ProSim introduces multimodal prompts for controllable generation [34]. These methods are efficient, but sequential prediction can propagate local errors and induce state-distribution shift under long-horizon feedback [15]. Conversely, diffusion-based methods jointly model continuous multi-agent behavior, supporting controllable prediction, scene generation, and traffic simulation [16–18, 35]. SceneDiffuser unifies scene initialization and closed-loop rollout [36], while subsequent methods extend diffusion generation to city-scale simulation and cross-domain transfer [2, 37]. However, multi-step sampling is computationally expensive and repeatedly refines predictions under a fixed condition, which is not fully aligned with frequent closed-loop feedback and replanning. Overall, existing methods still struggle to balance closed-loop robustness and inference efficiency while retaining competitive open-loop performance.

2.2 One-Step Denoising and Joint Learning

Recent studies suggest that reducing the number of denoising steps need not compromise generation quality. DPM-Solver and UniPC enable few-step sampling through efficient solvers [38, 39], whereas other methods learn direct generative mappings using specialized objectives based on distillation [22–24, 40], consistency learning [41–43], distribution matching [44], and flow models [45, 46]. For dynamic tasks, OneDP and MDG demonstrate efficient action generation and traffic-behavior reconstruction [25, 47], while DIDO further shows that task-specialized one-step models can outperform their multi-step counterparts [26]. However, concentrating training on high-noise direct recovery may narrow the available supervision and limit scene representation learning. Joint training can mitigate this limitation by transferring complementary supervision through shared representations; prior work has explored feature sharing [27], task trade-offs [28], loss weighting and gradient balancing [29, 48], and gradient conflict mitigation [49, 50]. Accordingly, UniDBO jointly trains two complementary branches through a shared scene encoder, combining high-noise recovery with continuous-scale supervision to balance open-loop trajectory fitting and closed-loop simulation robustness while preserving one-step inference efficiency.

3 Methodology

3.1 Problem Formulation

Scenario generation task. We formulate multi-agent scenario generation as scene-conditioned joint control-sequence generation. At replan s, the condition $\mathcal{X}^{(s)} = (\mathbf{s}_{1:h}^{(s)}, \mathcal{M}, \mathcal{T}^{(s)})$ contains agent histories, the fixed HD map, and traffic-light states. Given the map, agent set, scene layout, and initial states, UniDBO generates future controls and trajectories for the existing agents. For agent i in action block j, the control $\mathbf{x}_{0,i,j} = [a_{i,j}, \dot{\psi}_{i,j}]$ consists of longitudinal acceleration and yaw rate. The model

samples normalized joint controls and decodes them into future trajectories:

$$\hat{\mathbf{x}}_{0,s} \sim p_\theta(\cdot \mid \mathcal{X}^{(s)}), \qquad \hat{\mathbf{y}}_s = \mathcal{K}(\hat{\mathbf{x}}_{0,s}; \mathcal{X}^{(s)}), \tag{1}$$

where $\hat{\mathbf{x}}_{0,s} \in \mathbb{R}^{A\times K\times 2}$, A is the number of agents, $K = T_f/\Delta_a$ is the number of action blocks with Δ_a frames per block, and $\mathcal{K}$ de-normalizes and integrates the controls using simulator-consistent kinematics.

Open-loop evaluation and closed-loop simulation. Open-loop evaluation keeps the logged scene condition fixed and measures trajectory fitting without simulator feedback. In contrast, closed-loop simulation repeatedly queries the model using the current simulator state, with executed predictions shaping subsequent scene conditions. At each replan, only the first r predicted frames are executed before the scene condition is updated:

$$\hat{\mathbf{u}}_s = \Pi_r(\mathcal{P}_v(\hat{\mathbf{y}}_s)), \qquad \hat{\mathbf{s}}_{s+1} = F_r(\hat{\mathbf{s}}_s, \hat{\mathbf{u}}_s, \mathcal{E}_s), \qquad \mathcal{X}^{(s+1)} = \mathcal{C}(\hat{\mathbf{s}}_{s+1}), \tag{2}$$

where $\mathcal{P}_v$ denotes evaluation-time speed postprocessing (identity when unused), Π_r selects the execution prefix, F_r advances the simulator under exogenous context $\mathcal{E}_s$, and $\mathcal{C}$ constructs the next scene condition.

3.2 Architecture of UniDBO

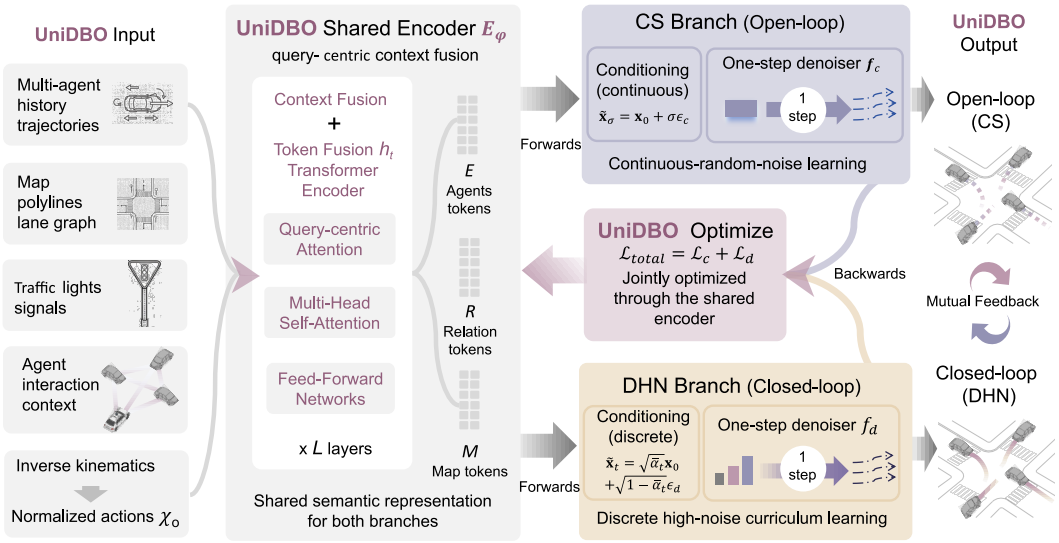

Figure 2: Overall framework of UniDBO. During training, DHN and CS share a scene encoder: DHN learns one-step recovery of joint multi-agent controls through a progressive high-noise curriculum, while CS provides auxiliary supervision across the continuous noise range and supports open-loop trajectory fitting. During inference, only the branch corresponding to the evaluation mode is used: CS for open-loop evaluation and DHN for closed-loop simulation, each requiring one denoising step.

We introduce UniDBO, a unified dual-branch one-step denoising framework for multi-agent scenario generation. UniDBO jointly trains two branches with complementary objectives through a shared scene encoder: DHN uses a progressive high-noise curriculum to strengthen direct joint control recovery for closed-loop robustness, while CS provides auxiliary supervision across the continuous noise range to enrich the shared scene representation and support open-loop trajectory fitting. At inference, UniDBO uses mode-aware activation: CS for open-loop evaluation and DHN for closed-loop simulation. Each call uses only the shared encoder and its designated recovery head, preserving one-step inference efficiency. The overall architecture is illustrated in Figure 2.

Shared scene representation. UniDBO first encodes the scene condition $\mathcal{X}$, including agent histories, interactions, map context, and traffic-light states. The resulting representation conditions two branch-specific recovery heads, each of which predicts the joint control tensor from a noisy input:

$$\mathbf{H} = E_\phi(\mathcal{X}), \qquad \hat{\mathbf{x}}_0^b = f_b(\tilde{\mathbf{x}}^b, \mathbf{H}, \nu_b), \quad b \in \{d, c\}, \tag{3}$$

where $\tilde{\mathbf{x}}^b$ and ν_b denote the noisy controls and noise condition for branch b. The parameters are $\theta = (\phi, \theta_d, \theta_c)$: DHN and CS share E_ϕ and have separate heads f_d and f_c, parameterized by θ_d and θ_c.

Discrete high-noise curriculum learning. During closed-loop rollout, control errors at each replan can propagate through simulator feedback and affect subsequent scene states. To limit this propagation, DHN uses one-step joint control recovery under high-noise conditions as a surrogate objective for closed-loop robustness. Specifically, at optimization step g, the lower bound of the discrete timestep range is progressively increased as:

$$\begin{aligned} \kappa_g &= \kappa_{\text{start}} + \min\left(\frac{g}{G_{\text{cur}}}, 1\right)(\kappa_{\text{end}} - \kappa_{\text{start}}), \\ t_{\min}(g) = \text{round}(\kappa_g t_{\max}), \qquad q_g &= \mathcal{U}_d(\{t_{\min}(g), \ldots, t_{\max}\}), \qquad t \sim q_g, \end{aligned} \tag{4}$$

where κ_{start} and κ_{end} are the initial and final curriculum ratios, G_{cur} is the curriculum duration in optimization steps, and $\mathcal{U}_d$ denotes a discrete uniform distribution. For each sampled t, we apply forward corruption to the clean controls and recover the joint controls in a single model evaluation:

$$\tilde{\mathbf{x}}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_d, \qquad \hat{\mathbf{x}}_0^d = f_d(\tilde{\mathbf{x}}_t, \mathbf{H}, t), \qquad \boldsymbol{\epsilon}_d \sim \mathcal{N}(0, \mathbf{I}), \tag{5}$$

where $(\tilde{\mathbf{x}}^d, \nu_d) = (\tilde{\mathbf{x}}_t, t)$, and $\bar{\alpha}_t = \prod_{m=0}^{t}(1-\beta_m)$ is the cumulative signal coefficient. As training progresses, the support of q_g contracts toward $t^\star = t_{\max}$, progressively concentrating optimization on direct recovery from severe corruption. Appendices C.3 and C.5 further analyze how this surrogate objective relates to closed-loop recovery risk and finite-horizon state deviation.

Randomized continuous-scale noise learning. CS learns control recovery across the continuous noise range to support open-loop trajectory fitting and provide multi-scale supervision to the shared scene encoder. Specifically, CS independently samples a noise scale and recovers clean controls from the resulting perturbed input:

$$\tilde{\mathbf{x}}_\sigma = \mathbf{x}_0 + \sigma\boldsymbol{\epsilon}_c, \qquad \hat{\mathbf{x}}_0^c = f_c(\tilde{\mathbf{x}}_\sigma, \mathbf{H}, \sigma), \qquad \boldsymbol{\epsilon}_c \sim \mathcal{N}(0, \mathbf{I}), \quad \log\sigma \sim \mathcal{U}(\log\sigma_{\min}, \log\sigma_{\max}), \tag{6}$$

where $(\tilde{\mathbf{x}}^c, \nu_c) = (\tilde{\mathbf{x}}_\sigma, \sigma)$. The CS loss updates both its separate recovery head f_c and the shared encoder, allowing continuous-scale supervision to complement DHN's high-noise learning. Appendix C.4 further analyzes how joint training affects open-loop recovery and bounds trajectory errors in terms of the recovery risk.

3.3 Joint Training and Mode-Aware One-Step Evaluation

Joint dual-noise branch training. DHN focuses on direct recovery under high-noise conditions, whereas CS covers the configured continuous noise range. To integrate these two complementary recovery signals, UniDBO jointly trains the two branches through a shared scene encoder while equipping them with separate recovery heads. Both branches predict complete control sequences, which are decoded into future trajectories as:

$$\hat{\mathbf{y}}^b = \mathcal{K}(\hat{\mathbf{x}}_0^b; \mathcal{X}), \qquad b \in \{c, d\}.$$

For each branch, $\mathcal{L}_{\text{sup}}^b$ combines Smooth-L1 position reconstruction and wrapped heading error over valid future frames, while DHN additionally uses the Smooth-L1 control-recovery loss $\mathcal{L}_{\text{act}}^d$. The joint training objective is

$$\mathcal{L}_{\text{total}} = w_c\mathcal{L}_{\text{sup}}^c + w_d\mathcal{L}_{\text{sup}}^d + w_a\mathcal{L}_{\text{act}}^d, \qquad (w_c, w_d, w_a) = (1.25, 1.25, 0.05). \tag{7}$$

The losses from both branches update the shared scene encoder, whereas each recovery head is updated only by its corresponding branch loss. The shared-encoder gradient is therefore:

$$\nabla_\phi\mathcal{L}_{\text{total}} = \underbrace{w_c\nabla_\phi\mathcal{L}_{\text{sup}}^c}_{\mathbf{g}_c} + \underbrace{w_d\nabla_\phi\mathcal{L}_{\text{sup}}^d + w_a\nabla_\phi\mathcal{L}_{\text{act}}^d}_{\mathbf{g}_d}. \tag{8}$$

Through this joint optimization, CS transfers continuous-scale recovery signals to the shared scene representation, thereby complementing DHN's high-noise learning. Proposition 2 gives a sufficient local condition for positive auxiliary transfer. Detailed losses and masks are provided in Appendix A.1; loss-weight sensitivity is reported in Appendix D.4.

Mode-aware one-step evaluation. To balance open-loop trajectory fitting, closed-loop simulation robustness, and inference efficiency, UniDBO activates only the shared scene encoder and the recovery head corresponding to the evaluation mode. For open-loop evaluation, CS performs one-step trajectory fitting under a fixed logged scene condition. At each closed-loop replan, DHN leverages its ability to recover directly from high-noise inputs to generate joint controls from an independent Gaussian query under the updated scene condition. Both routes require one denoising step:

$$\hat{\mathbf{x}}_0 = \begin{cases} f_c(\tilde{\mathbf{x}}_\sigma, E_\phi(\mathcal{X}), \sigma), & \text{open-loop evaluation,} \\ f_d(\mathbf{z}_s, E_\phi(\mathcal{X}^{(s)}), t^\star), & \text{closed-loop simulation,} \end{cases} \tag{9}$$

During closed-loop simulation, the simulator executes the decoded trajectory prefix according to Equation 2 and uses the updated scene state for the next replan. Algorithm 1 provides the complete training procedure.

4 Experiments

We comprehensively evaluate UniDBO to answer three core research questions: **RQ1: Performance and Efficiency.** Can UniDBO balance open-loop trajectory fitting with closed-loop simulation robustness while maintaining high inference efficiency? **RQ2: Mechanism Analysis.** How do one-step denoising and progressive high-noise curriculum learning affect closed-loop simulation, and how does CS-assisted joint training balance open-loop recovery with closed-loop robustness? **RQ3: Cross-Domain and System-Level Applicability.** Do these properties transfer to zero-shot INTERACTION simulation and system-level AV testing on unseen OnSite cases? Detailed experimental settings, metric definitions, and supplementary results are provided in Appendix B and D.

4.1 Main Results with WOMD/Waymax

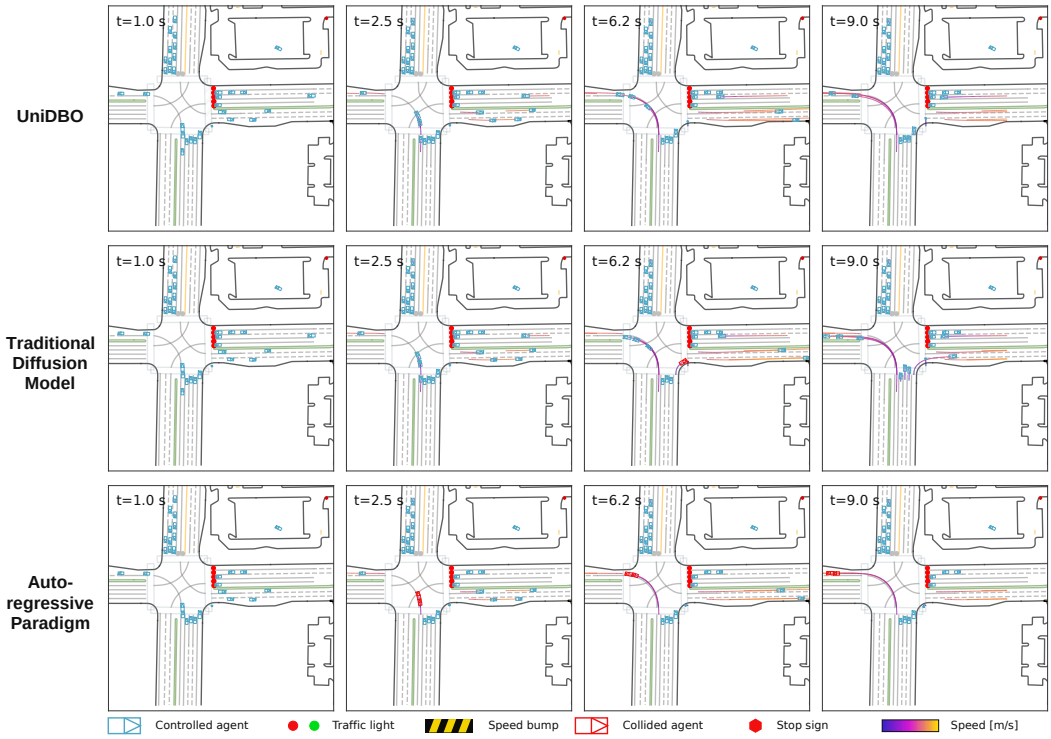

Figure 3: Qualitative closed-loop rollout comparison in a signalized multi-lane WOMD/Waymax scene. Columns show four time-aligned simulator states; rows correspond to UniDBO, the conventional iterative diffusion model VBD, and the autoregressive model SMART. In the shown case, UniDBO maintains lane-consistent turning and passage through the junction, whereas VBD and SMART exhibit collisions near the crossing. Blue boxes mark controlled agents, colored traces show their executed trajectories, and red outlines mark collided agents.

Table 1: Performance comparison of methods on WOMD/Waymax for in-domain open-loop recovery, closed-loop simulation, and inference efficiency. Best values are **bolded**, and second-best values are underlined.

Method	Open-loop Metrics		Closed-loop Metrics					Efficiency Metrics (ms/step)	
	minADE↓	minFDE↓	Collision (%)↓	Off-road (%)↓	Wrong-way (%)↓	Kinematic Infeasibility (%)↓	Log Divergence↓	Inf.↓	Total↓
CDPT [2]	0.825	2.228	7.362	4.964	2.103	0.0970	1.687	290.876	431.267
CAT-K [14]	0.785	2.941	**4.034**	5.666	1.295	0.7251	1.433	42.905	234.156
SMART [12]	1.106	3.358	4.906	5.390	1.258	0.5879	1.642	149.938	338.085
VBD [18]	1.416	3.688	10.571	5.657	2.429	0.0833	1.971	311.682	479.823
UniDBO	**0.778**	**1.907**	4.773	**3.721**	**1.244**	**0.0228**	**1.322**	**6.948**	**175.981**

Table 1 summarizes UniDBO's in-domain performance on WOMD/Waymax (**RQ1**). UniDBO achieves the best results on four of the five closed-loop metrics while maintaining strong auxiliary open-loop recovery. Although CAT-K has the lowest Collision rate, UniDBO ranks second and reduces Collision by 54.8% and 35.2% relative to VBD and CDPT, respectively. Its low Off-road, Kinematic Infeasibility, and Log Divergence indicate road-compliant, feasible, and stable closed-loop behavior across evaluated traffic scenes. Qualitative results in Figure 3 illustrate this trend: UniDBO maintains lane-consistent, coordinated motion, whereas VBD and SMART collide near the crossing. Regarding efficiency, UniDBO requires only 6.948 ms/step for model inference, yielding $44.8\times$ and $6.2\times$ speedups over VBD and CAT-K, respectively, and achieves the lowest total runtime. Overall, UniDBO effectively balances closed-loop robustness and inference efficiency while maintaining strong auxiliary open-loop recovery performance.

4.2 Cross-Domain Generalization with INTERACTION

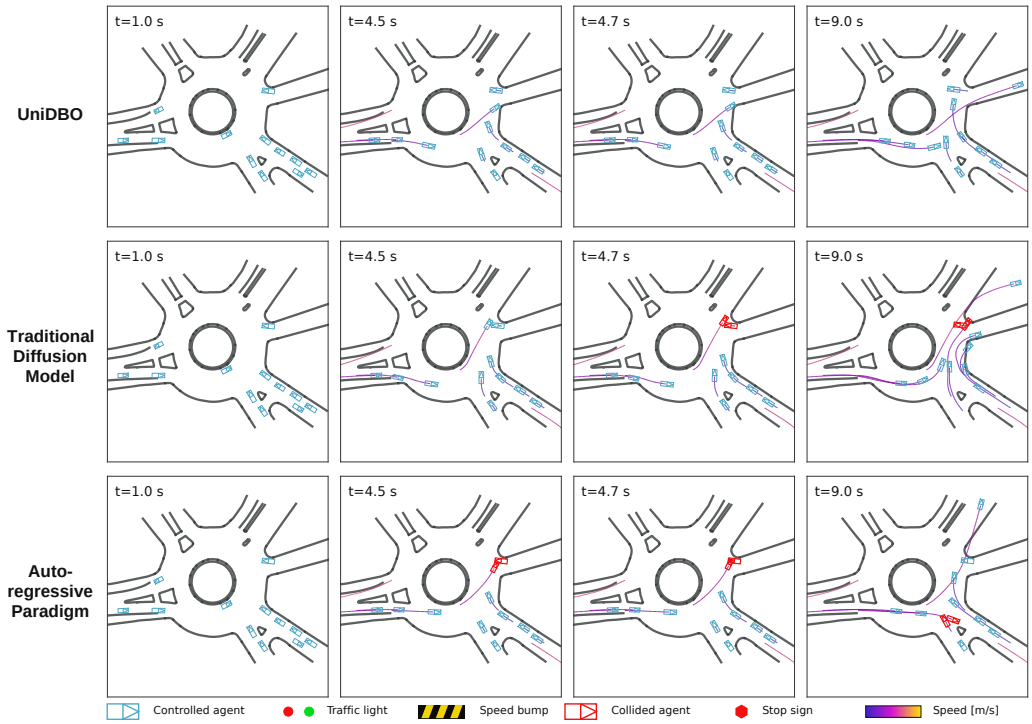

Figure 4: Qualitative zero-shot closed-loop rollout comparison in an INTERACTION roundabout. UniDBO is evaluated without INTERACTION training or fine-tuning. The layout, method order, and annotations follow Figure 3. In the shown case, UniDBO produces more coordinated yielding and merging, whereas the baselines exhibit local interaction conflicts near the merge.

A key advantage of UniDBO is its ability to transfer to unseen traffic domains without retraining (**RQ3**). Table 2 demonstrates this capability on INTERACTION: UniDBO achieves the best results on three of the five closed-loop metrics and ranks second only to CAT-K in Collision and Wrong-way, while also obtaining the lowest open-loop recovery errors. In the roundabout shown in Figure 4, UniDBO generates coordinated yielding and merging, whereas VBD and SMART exhibit local

interaction conflicts near the merge. Together, these results indicate that UniDBO achieves competitive zero-shot cross-domain generalization.

Table 2: Performance comparison of methods on INTERACTION for zero-shot cross-domain generalization. Best values are **bolded**, and second-best values are underlined.

Method	Open-loop Metrics		Closed-loop Metrics				
	minADE↓	minFDE↓	Collision (%)↓	Off-road (%)↓	Wrong-way (%)↓	Kinematic Infeasibility (%)↓	Log Divergence↓
CDPT [2]	0.627	1.228	6.426	10.287	5.125	0.0940	2.301
CAT-K [14]	0.541	1.348	**1.652**	8.773	**0.454**	0.5866	2.380
VBD [18]	0.939	2.495	8.146	21.160	5.454	0.1539	3.078
SMART [12]	2.015	2.927	3.518	13.003	1.531	2.9565	2.433
UniDBO	**0.406**	**0.793**	2.847	**7.040**	1.521	**0.0041**	**2.036**

4.3 Mechanism Analysis

We analyze the sources of UniDBO's one-step performance in open-loop recovery and closed-loop simulation through three comparisons (**RQ2**). We first compare one-step and iterative denoising, then evaluate progressive high-noise curriculum learning under one-step deployment, and finally assess joint dual-noise training by adding CS auxiliary supervision to DHN.

4.3.1 One-Step Denoising

Table 3: Performance comparison of one-step and iterative denoising on WOMD/Waymax using the same Full-range noise model. Inf. and Total denote amortized model and end-to-end latency (ms/step), respectively. Best values are **bolded**.

Configuration	NFE↓	Collision (%)↓	Off-road (%)↓	Wrong-way (%)↓	Kinematic Infeasibility (%)↓	Log Divergence↓	Inf.↓	Total↓
Full-range noise	50	10.571	5.657	2.429	0.0833	**1.971**	311.682	479.823
Full-range noise	**1**	**9.539**	**5.128**	**1.915**	**0.0039**	1.988	**7.968**	**172.715**

We first compare one-step and 50-step denoising using the Full-range noise model. As shown in Table 3, one-step denoising improves four of the five closed-loop metrics, with slightly higher Log Divergence, while reducing denoiser calls from 50 to one. Both configurations use the same external replanning interval, indicating that more internal denoiser evaluations do not necessarily improve closed-loop performance. This result is consistent with the conditional analysis in Theorem 1: prediction errors may propagate and accumulate during iterative reverse sampling [21, 51], and one-step recovery yields lower control error when their net contribution exceeds the refinement gain. Appendix D.3 provides the complete comparison.

4.3.2 Progressive High-Noise Curriculum Learning

Table 4: Performance comparison of one-step training designs on WOMD/Waymax under NFE$=1$ deployment. Best values are **bolded**, and second-best values are underlined.

Configuration	Open-loop Metrics		Closed-loop Metrics				
	minADE↓	minFDE↓	Collision (%)↓	Off-road (%)↓	Wrong-way (%)↓	Kinematic Infeasibility (%)↓	Log Divergence↓
Full-range noise	1.416	3.688	9.539	5.128	1.915	0.0039	1.988
DHN	2.780	7.955	6.507	3.832	1.396	0.0484	1.334
CS	0.847	2.133	11.335	5.229	2.608	**0.0002**	2.560
UniDBO	**0.778**	**1.907**	**4.773**	**3.721**	**1.244**	0.0228	**1.322**

We then compare Full-range noise with progressive high-noise curriculum learning under NFE$=1$ deployment. DHN outperforms Full-range noise on most closed-loop metrics (Table 4) and achieves lower 1–8 s means across all six drift measures (Figure 5; Appendix Table 14). The NFE$=0$ control in Appendix Table 13 shows that removing both input noise and denoising degrades closed-loop performance, further supporting the effectiveness of noise-conditioned one-step recovery. These results support high-noise direct recovery as a robustness proxy for closed-loop drift. Appendices C.3 and C.5 connect terminal-query recovery risk to closed-loop state drift through finite-horizon error propagation.

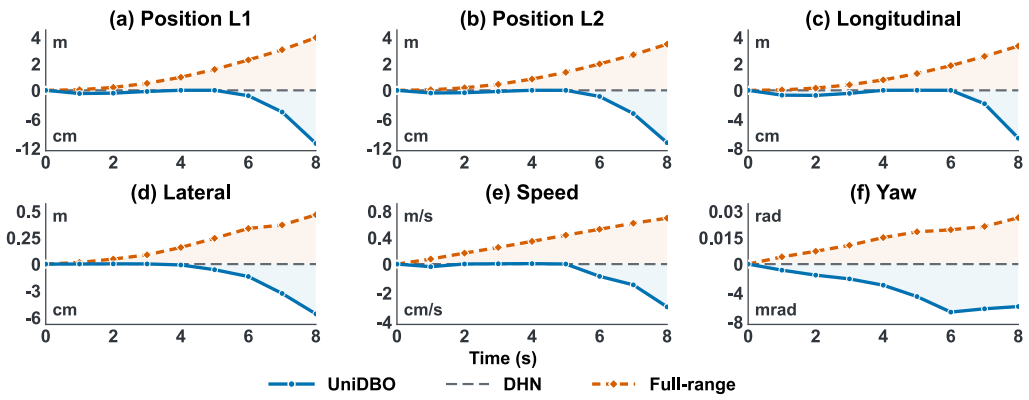

Figure 5: One-step closed-loop drift differences on WOMD/Waymax at aligned 0–8 s snapshots. Panels (a)–(f) show scene-macro position-L1, position-L2, longitudinal, lateral, speed, and yaw drift relative to DHN. The gray dashed line marks the DHN zero baseline; orange and blue curves show Full-range-minus-DHN and UniDBO-minus-DHN differences, respectively. Positive values indicate greater drift than DHN, and negative values indicate lower drift. The positive and negative halves use separately labeled scales and units.

4.3.3 Joint Dual-Noise Training

Joint supervision and branch specialization. UniDBO jointly trains separate CS and DHN recovery heads through a shared scene encoder and uses mode-aware activation at inference, preserving one-step denoising efficiency. Compared with DHN alone, joint training improves all five closed-loop metrics while yielding more accurate auxiliary open-loop recovery (Table 4). Table 15 in Appendix D.3.3 further shows that, under matched sampling strategies, CS provides more accurate open-loop recovery, whereas DHN performs better on most closed-loop metrics, highlighting their complementarity. Mixing both noise regimes within a single recovery head partly mitigates this trade-off, but DHN's improved open-loop recovery comes with higher Collision than under the high-noise curriculum. UniDBO instead preserves branch-specific specialization through separate recovery heads while transferring complementary supervision through the shared encoder, achieving a better overall balance between open-loop recovery and closed-loop simulation. The local shared-encoder analysis in Appendix C.4 provides a sufficient gradient-alignment condition under which joint encoder updates lower the DHN training loss more than DHN-only updates.

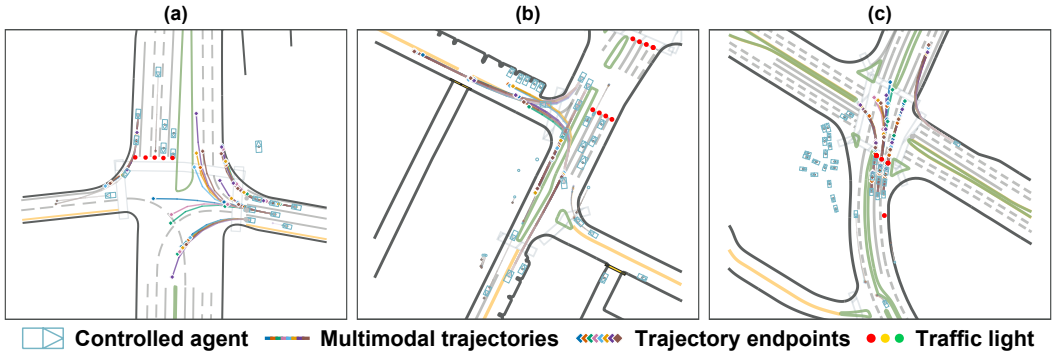

Figure 6: Stochastic multi-agent futures on WOMD/Waymax. Three initial scenes each show eight UniDBO closed-loop rollouts from independent Gaussian-query sequences. Cyan outlines mark controlled vehicles; colors identify joint rollouts across agents; diamonds and circles mark endpoints and traffic lights, respectively. The initial conditions, model, and simulator settings are fixed within each scene.

Closed-loop drift. DHN uses direct high-noise recovery as a trainable proxy for closed-loop robustness. With CS-assisted joint training, UniDBO retains DHN's low-drift behavior and further reduces the mean of all six drift measures (Figure 5; Appendix Table 14). This trend is consistent with the analyses of recovery risk and closed-loop error propagation in Appendices C.3 and C.5. Low drift is not confined to a single generated trajectory: independent Gaussian-query sequences yield distinct joint multi-agent futures, while closed-loop quality and drift remain stable across eight realizations (Figure 6; Appendix D.5).

4.4 System-Level AV Testing within OnSite Challenge

Following the official OnSite Challenge evaluation protocol [52], we further evaluate the effectiveness of UniDBO in system-level closed-loop AV testing on unseen scenarios (**RQ3**). As shown in Table 5, UniDBO achieves the best overall performance, while maintaining competitive comfort and test effectiveness. Although MJTG and CAT-K achieve higher kinematic consistency scores, UniDBO delivers a more balanced performance across the evaluation dimensions. These results demonstrate its ability to generate realistic and diverse traffic scenarios that support robust external-planner evaluation in safety-critical environments. Dataset details and score definitions are provided in Appendices B.1 and B.2.

Table 5: System-level AV testing results on OnSite Challenge. Higher is better; best values are **bolded**, and second-best values are underlined.

Method	Safety Score↑	Kinematic Consistency↑	Comfort Score↑	Test Effectiveness↑	Total Score↑
IDM [53]	16.73	5.84	5.77	6.06	34.40
SAC [54]	16.11	4.59	7.04	9.26	37.00
MJTG [55]	19.61	**13.08**	5.94	7.16	45.80
EGPO [56]	26.03	5.31	**8.77**	13.71	53.82
VBD [18]	29.96	7.47	7.62	16.18	61.23
SMART [12]	22.88	8.60	6.88	13.24	51.60
CAT-K [14]	28.02	10.96	8.12	12.78	59.88
CDPT [2]	29.96	7.76	7.65	**17.92**	63.29
UniDBO	**29.98**	8.91	8.25	17.07	**64.21**

5 Conclusion

This work introduced UniDBO, a unified dual-branch one-step denoising framework for autonomous driving scenario generation. By jointly training CS and DHN through a shared scene encoder while activating only the designated branch at inference, UniDBO balances open-loop trajectory fitting and closed-loop robustness without sacrificing inference efficiency. Experiments on WOMD/Waymax, INTERACTION, and OnSite demonstrate strong in-domain performance, zero-shot cross-domain generalization, and utility for system-level AV testing.

Limitations. UniDBO currently models trajectory-level vehicle dynamics and was evaluated under fixed benchmark protocols. Collision-related behavior also remains challenging in complex multi-agent interactions. These limitations suggest several directions for future work. First, the framework could be extended to photorealistic traffic simulation. Second, risk-sensitive and safety-constrained generation objectives could help reduce residual collision risk. Finally, broader validation across diverse out-of-domain scenarios and AV testing platforms would provide a more comprehensive assessment of generalization. Broader impacts are discussed in Appendix E.

Acknowledgments and Disclosure of Funding

This research is partially supported by the National Natural Science Foundation of China (Nos. 52125208, 52402374), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM123), and the Fundamental Research Funds for the Central Universities.

References

[1] Xintao Yan, Zhengxia Zou, Shuo Feng, Haojie Zhu, Haowei Sun, and Henry X Liu. Learning naturalistic driving environment with statistical realism. *Nature communications*, 14(1):2037, 2023.

[2] Yuhang Chen, Jie Sun, Jialin Fan, and Jian Sun. Transferring causal driving patterns for generalizable traffic simulation with diffusion-based distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(1):110–118, 2026.

[3] Chengkai Xu, Yiming Cui, Jiaqi Liu, Yicheng Guo, Cheng Qin, Geyuan Zhang, Xinwei Dong, Shiyu Fang, Peng Hang, and Jian Sun. A survey on end-to-end autonomous driving training from the perspectives of data, strategy, and platform. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–20, 2026.

[4] 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*, pages 9710–9719, 2021.

[5] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kümmerle, Hendrik Königshof, Christoph Stiller, Arnaud de La Fortelle, and Masayoshi Tomizuka. INTERACTION dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019.

[6] 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*, pages 10400–10409, 2021.

[7] Xiyan Jiang, Xiaocong Zhao, Yiru Liu, Zirui Li, Peng Hang, Lu Xiong, and Jian Sun. A naturalistic trajectory dataset with dense interaction for autonomous driving. *Scientific Data*, 12:1084, 2025.

[8] 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. *Advances in Neural Information Processing Systems*, 36:7730–7742, 2023.

[9] 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*, volume 36, pages 3894–3920, 2023.

[10] 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*, 2026.

[11] Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, Brandyn White, and Dragomir Anguelov. The Waymo open sim agents challenge. *Advances in Neural Information Processing Systems*, 36:59151–59171, 2023.

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

[13] Zikang Zhou, Haibo Hu, Xinhong Chen, Jianping Wang, Nan Guan, Kui Wu, Yung-Hui Li, Yu-Kai Huang, and Chun Jason Xue. BehaviorGPT: Smart agent simulation for autonomous driving with next-patch prediction. In *Advances in Neural Information Processing Systems*, volume 37, pages 79597–79617, 2024.

[14] Zhejun Zhang, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5422–5432, 2025.

[15] Stéphane Ross, Geoffrey Gordon, and Drew 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*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635. PMLR, 2011.

[16] Chiyu Max Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, and Dragomir Anguelov. MotionDiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9644–9653, 2023.

[17] Ethan Pronovost, Meghana Reddy Ganesina, Noureldin Hendy, Zeyu Wang, Andres Morales, Kai Wang, and Nick Roy. Scenario diffusion: Controllable driving scenario generation with diffusion. *Advances in Neural Information Processing Systems*, 36:68873–68894, 2023.

[18] Zhiyu Huang, Zixu Zhang, Ameya Vaidya, Yuxiao Chen, Jaime Fernández Fisac, and Chen Lv. Versatile behavior diffusion for generalized traffic agent simulation. *IEEE Transactions on Intelligent Transportation Systems*, 27(7):8681–8697, 2026.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[21] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 26245–26265. PMLR, 2023.

[22] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 62307–62331. PMLR, 2024.

[23] Mingyuan Zhou, Huangjie Zheng, Yi Gu, Zhendong Wang, and Hai Huang. Adversarial score identity distillation: Rapidly surpassing the teacher in one step. In *International Conference on Learning Representations*, 2025.

[24] Yifei Wang, Weimin Bai, Colin Zhang, Debing Zhang, Weijian Luo, and He Sun. Uni-Instruct: One-step diffusion model through unified diffusion divergence instruction. In *Advances in Neural Information Processing Systems*, volume 38, pages 57097–57132, 2025.

[25] Zhendong Wang, Max Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, Ming-Yu Liu, and Yu Zeng. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 63399–63416. PMLR, 2025.

[26] Jing Lyu, Shuanghao Bai, Runze Xiao, Zhenyu Liao, Wenxing Tan, Zihan Tang, Ruochuan Shi, Cheng Peng, Yuheng Ji, Yihao Wang, Badong Chen, Pengwei Wang, Zhongyuan Wang, and Xiaoguang Zhao. DIDO: Distilling interaction-centric dynamics into one-step denoising for world action models. *arXiv preprint arXiv:2609.15570*, 2026.

[27] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016.

[28] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[29] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018.

[30] 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)*, pages 3567–3575. IEEE, 2023.

[31] Jie Sun and Jiwon Kim. Modelling two-dimensional driving behaviours at unsignalised intersection using multi-agent imitation learning. *Transportation Research Part C: Emerging Technologies*, 165:104702, 2024.

[32] Shihan Wang, Ying Ni, Chengsheng Miao, Jian Sun, and Jie Sun. A multiagent social interaction model for autonomous vehicle testing. *Communications in Transportation Research*, 5(2): 100183, 2025.

[33] 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*, pages 8579–8590, 2023.

[34] Shuhan Tan, Boris Ivanovic, Yuxiao Chen, Boyi Li, Xinshuo Weng, Yulong Cao, Philipp Kraehenbuehl, and Marco Pavone. Promptable closed-loop traffic simulation. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 5087–5105. PMLR, 2025.

[35] Haohong Lin, Xin Huang, Tung Phan, David Hayden, Huan Zhang, Ding Zhao, Siddhartha Srinivasa, Eric Wolff, and Hongge Chen. Causal composition diffusion model for closed-loop traffic generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27542–27552, 2025.

[36] Chiyu Max Jiang, Yijing Bai, Andre Cornman, Christopher Davis, Xiukun Huang, Hong Jeon, Sakshum Kulshrestha, John Lambert, Shuangyu Li, Xuanyu Zhou, Carlos Fuertes, Chang Yuan, Mingxing Tan, Yin Zhou, and Dragomir Anguelov. SceneDiffuser: Efficient and controllable driving simulation initialization and rollout. In *Advances in Neural Information Processing Systems*, volume 37, pages 55729–55760, 2024.

[37] Shuhan Tan, John Lambert, Hong Jeon, Sakshum Kulshrestha, Yijing Bai, Jing Luo, Dragomir Anguelov, Mingxing Tan, and Chiyu Max Jiang. SceneDiffuser++: City-scale traffic simulation via a generative world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1570–1580, 2025.

[38] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, volume 35, pages 5775–5787, 2022.

[39] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pages 49842–49869, 2023.

[40] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[41] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32211–32252. PMLR, 2023.

[42] Zehao Dou, Minshuo Chen, Mengdi Wang, and Zhuoran Yang. Theory of consistency diffusion models: Distribution estimation meets fast sampling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11592–11612. PMLR, 2024.

[43] Junlong Lyu, Zhitang Chen, and Shoubo Feng. Sampling is as easy as keeping the consistency: Convergence guarantee for consistency models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 33664–33685. PMLR, 2024.

[44] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024.

[45] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. InstaFlow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024.

[46] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, pages 75460–75482, 2025.

[47] Zhiyu Huang, Zewei Zhou, Tianhui Cai, Yun Zhang, and Jiaqi Ma. MDG: Masked denoising generation for multi-agent behavior modeling in traffic environments. *arXiv preprint arXiv:2511.17496*, 2025.

[48] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[49] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836, 2020.

[50] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 18878–18890, 2021.

[51] Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *International Conference on Learning Representations*, 2024.

[52] OnSite. OnSite challenge platform. OnSite Autonomous Driving Algorithm Challenge, 2026.

[53] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805–1824, 2000.

[54] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 2018.

[55] Ye Tian, Wentao Zheng, Yinghao Shao, He Zhang, and Jian Sun. MJTG: A multi-vehicle joint trajectory generator for complex and rare scenarios. *IEEE Transactions on Vehicular Technology*, 74(10):15026–15039, 2025.

[56] Jialin Fan, Ying Ni, Yujia Zhao, Jie Sun, and Jian Sun. Optimizing multi-agent behavior for interactive autonomous driving with equilibrium value estimation. *SSRN preprint 5367491*, 2025.

[57] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pages 26565–26577, 2022.

[58] Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, and Holger Caesar. Towards learning-based planning: The nuPlan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 629–636. IEEE, 2024.

[59] Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[60] Zhiyao Ren, Yibing Zhan, Liang Ding, Gaoang Wang, Chaoyue Wang, Zhongyi Fan, and Dacheng Tao. Multi-step denoising scheduled sampling: Towards alleviating exposure bias for diffusion models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(5): 4667–4675, 2024.

[61] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pages 24081–24125, 2024.

[62] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, volume 38, pages 167283–167308, 2025.

[63] Ke Guo, Zhenwei Miao, Wei Jing, Weiwei Liu, Weizi Li, Dayang Hao, and Jia Pan. LASIL: Learner-aware supervised imitation learning for long-term microscopic traffic simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15386–15395, 2024.

Appendix

The appendix is organized into five parts. Appendix A provides training objectives, implementation settings, and algorithms. Appendix B describes datasets, evaluation metrics and protocols, and comparison methods. Appendix C analyzes the design choices and their implications for open-loop recovery and closed-loop drift. Appendix D presents supplementary results, ablations, and stochastic-diversity analyses. Appendix E discusses broader impacts, potential risks, and mitigation recommendations.

A. Method and Training Details 16
A.1 Training Targets and Objectives 16
A.2 Implementation and Hyperparameter Selection 17
B. Experimental Setup and Evaluation Protocols 18
B.1 Datasets and Distribution Diagnostics 18
B.2 Evaluation Metrics and Protocols 19
B.3 Baselines and Comparison Implementations 21
C. Theoretical Analysis 21
C.1 Overview of Theoretical Claims 21
C.2 Direct One-Step versus Iterative DDPM Reverse Sampling 22
C.3 High-Noise Direct Recovery as a Closed-Loop Robustness Proxy 23
C.4 Auxiliary Transfer from Continuous-Scale Supervision 24
C.5 From Direct Control Recovery to Closed-Loop State Drift 25
D. Supplementary Experimental Results 26
D.1 Supplementary WOMD/Waymax Results 26
D.2 Supplementary Zero-Shot INTERACTION Results 28
D.3 Mechanism and Ablation Studies 29
D.4 Weight Sensitivity 32
D.5 Stochastic Diversity and Multimodality 33
E. Broader Impact 34

A Method and Training Details

This section specifies the training objectives, implementation settings, hyperparameter selection, and mode-aware training and inference procedures for UniDBO.

A.1 Training Targets and Objectives

Validity masks and supervision targets. We define the targets and validity masks used by the branch objectives in Section 3.3. Let Ω and Ω_a denote valid supervised trajectory frames and action blocks:

$$\Omega = \{(n,i,\tau) \mid m^{\text{valid}}_{n,i,\tau} = 1,\ m^{\text{int}}_{n,i} = 1,\ \tau \in \{1,\dots,T_f\}\}, \tag{10}$$

$$\Omega_a = \{(n,i,j) \mid m^{\text{act}}_{n,i,j} = 1,\ m^{\text{int}}_{n,i} = 1,\ j \in \{1,\dots,K\}\}, \tag{11}$$

where m^{valid} marks valid future frames, $m^{\text{act}}_{n,i,j} = 1$ when the control target for action block j is valid, and m^{int} selects supervised agents. All averages use nonempty valid sets.

Trajectory reconstruction loss. For a vector $\mathbf{v}$, define the summed element-wise Smooth-L1 penalty

$$\text{SL1}(\mathbf{v}) = \sum_k \varphi(v_k), \qquad \varphi(u) = \begin{cases} \frac{1}{2}u^2, & |u| < 1, \\ |u| - \frac{1}{2}, & |u| \geq 1. \end{cases} \tag{12}$$

For each branch $b \in \{c, d\}$, the position and heading losses are

$$\mathcal{L}^b_{\text{state}} = \frac{1}{|\Omega|} \sum_{(n,i,\tau)\in\Omega} \text{SL1}\left(\hat{\mathbf{y}}^{b,xy}_{n,i,\tau} - \mathbf{y}^{xy}_{n,i,\tau}\right), \tag{13}$$

$$\mathcal{L}^b_{\text{yaw}} = \frac{1}{|\Omega|} \sum_{(n,i,\tau)\in\Omega} \left|\Delta_\pi\left(\hat{\psi}^b_{n,i,\tau} - \psi_{n,i,\tau}\right)\right|, \tag{14}$$

where

$$\Delta_\pi(\delta) = \operatorname{atan2}(\sin\delta, \cos\delta). \tag{15}$$

The corresponding supervised trajectory loss is

$$\mathcal{L}^b_{\text{sup}} = \mathcal{L}^b_{\text{state}} + \mathcal{L}^b_{\text{yaw}}. \tag{16}$$

DHN control recovery and joint objective. In addition to trajectory supervision, DHN uses the control-recovery loss

$$\mathcal{L}^d_{\text{act}} = \frac{1}{|\Omega_a|} \sum_{(n,i,j)\in\Omega_a} \text{SL1}\big(\hat{\mathbf{x}}^d_{0,n,i,j} - \mathbf{x}_{0,n,i,j}\big)\,. \tag{17}$$

The branch objectives are

$$\mathcal{L}_d = w_d\mathcal{L}^d_{\text{sup}} + w_a\mathcal{L}^d_{\text{act}}, \qquad \mathcal{L}_c = w_c\mathcal{L}^c_{\text{sup}}. \tag{18}$$

The complete joint objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_d + \mathcal{L}_c. \tag{19}$$

Both objectives update the shared encoder. Each objective updates only its corresponding recovery head: $\mathcal{L}_d$ updates f_d, whereas $\mathcal{L}_c$ updates f_c.

Joint training. Algorithm 1 summarizes joint training, where $\mathbf{m} = (m^{\text{valid}}, m^{\text{act}}, m^{\text{int}})$ collects the masks defined in Appendix A.1. The branches share scene and target representations but receive independent corruptions.

Algorithm 1 UniDBO training with a shared scene encoder and branch-specific recovery heads.

Require: Data $\mathcal{D}$; encoder E_ϕ; heads $f_c(\cdot;\theta_c)$ and $f_d(\cdot;\theta_d)$; decoder $\mathcal{K}$
Require: Weights (w_c, w_d, w_a); optimizer Opt; updates G_{train}; noise settings from Table 6
for $g = 1$ to G_{train} **do**
  Sample $(\mathcal{X}, \mathbf{x}_0, \mathbf{y}, \mathbf{m}) \sim \mathcal{D}$ and compute $\mathbf{H} \leftarrow E_\phi(\mathcal{X})$
  Sample $t \sim q_g$ and $\boldsymbol{\epsilon}_d \sim \mathcal{N}(0, \mathbf{I})$; form $\tilde{\mathbf{x}}_t$ by Equations 4–5
  Predict $\hat{\mathbf{x}}^d_0 \leftarrow f_d(\tilde{\mathbf{x}}_t, \mathbf{H}, t; \theta_d)$; decode $\hat{\mathbf{y}}^d \leftarrow \mathcal{K}(\hat{\mathbf{x}}^d_0; \mathcal{X})$; compute $\mathcal{L}_d$
  Sample $\log\sigma \sim \mathcal{U}(\log\sigma_{\min}, \log\sigma_{\max})$ and $\boldsymbol{\epsilon}_c \sim \mathcal{N}(0, \mathbf{I})$; set $\tilde{\mathbf{x}}_\sigma \leftarrow \mathbf{x}_0 + \sigma\boldsymbol{\epsilon}_c$
  Predict $\hat{\mathbf{x}}^c_0 \leftarrow f_c(\tilde{\mathbf{x}}_\sigma, \mathbf{H}, \sigma; \theta_c)$; decode $\hat{\mathbf{y}}^c \leftarrow \mathcal{K}(\hat{\mathbf{x}}^c_0; \mathcal{X})$; compute $\mathcal{L}_c$
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_c + \mathcal{L}_d$; update $(\phi, \theta_c, \theta_d)$ with Opt
end for

A.2 Implementation and Hyperparameter Selection

Table 6 summarizes the architecture, noise schedules, optimization settings, and inference configuration used in our experiments, and separately lists the default epoch budget of the released training configuration.

Table 6: UniDBO architecture, training, and inference settings.

Group	Setting	Value
Architecture	Max. agents / frames / control dim.	$32/80/2$
Architecture	Encoder layers	6
Controls	Mean / std. / normalized clamp	$[0,0]/[1,0.15]/[-5,5]$
DHN	Steps / cosine parameters	50 / cosine / `s/e/tau`$= 0/1/1$
DHN	Curriculum ratio / duration	$0.7 \rightarrow 1.0$ / 180,000 updates
DHN	Inference timestep	$t^\star = 49$
CS	Noise range / sampling	$[0.01, 1.0]$ / log-uniform
Objective	(w_c, w_d, w_a)	$(1.25, 1.25, 0.05)$
Optimization	Per-device batch size	8
Released configuration	Default epoch budget	15
Inference	Open-loop / closed-loop head	CS / DHN
Closed loop	Modules / query / NFE per replan	encoder + DHN / independent Gaussian / 1
Closed loop	Frames per replan	$r = 10$

Noise schedules and curriculum. DHN uses a 50-step cosine schedule with $t_{\max} = t^\star = 49$, curriculum ratios $(\kappa_{\text{start}}, \kappa_{\text{end}}) = (0.7, 1.0)$, and $G_{\text{cur}} = 180{,}000$ updates. Over the training budget, the support of q_g contracts from $\{34, \ldots, 49\}$ to $\{44, \ldots, 49\}$, while retaining $t^\star$ for closed-loop inference. DHN's forward corruption follows discrete DDPM Gaussian noising [19], whereas CS uses continuous-scale additive Gaussian perturbation [57] with $\log \sigma \sim \mathcal{U}(\log 0.01, \log 1)$ to provide auxiliary supervision and support open-loop trajectory fitting.

Training resources and computational cost. Full-data training takes 45.54 h on eight NVIDIA L40 GPUs, 56 CPU cores, and 1,000 GB of host memory. The current joint model has 11.77M learnable parameters.

B Experimental Setup and Evaluation Protocols

This section specifies the datasets, cross-domain diagnostics, evaluation metrics and protocols, statistical reporting, and baseline implementations used to evaluate UniDBO.

B.1 Datasets and Distribution Diagnostics

WOMD/Waymax data. UniDBO is trained on 486,995 processed WOMD scenarios [4]. Open-loop evaluation uses 44,097 WOMD validation scenarios, each comprising 10 history frames and 80 logged future frames. In-domain closed-loop evaluation uses a fixed ordered set of 1,000 Waymax scenarios [8] with eight-second rollouts.

INTERACTION data. INTERACTION [5] is used for zero-shot cross-domain evaluation without target-domain training or fine-tuning. Open-loop evaluation uses 11,794 scenarios from the official validation split, each comprising 10 history frames and 30 logged future frames. Closed-loop evaluation uses 500 high-density scenarios, initialized from 10 history frames and rolled out for 80 frames.

OnSite data. OnSite [52] comprises 800 organizer-provided scenarios for system-level AV evaluation. Each scenario includes a fixed HD map, the states of existing agents, and a 31-frame scene prefix.

Data access, terms, and provenance. WOMD and Waymax are available through the official Waymo channels (`https://waymo.com/open/`); WOMD is used for non-commercial research under the Waymo Dataset License Agreement for Non-Commercial Use (`https://waymo.com/open/terms/`). INTERACTION is available through its official portal (`https://interaction-dataset.com/`), and its processing tools use the BSD 3-Clause License (`https://github.com/interaction-dataset/interaction-dataset`). OnSite scenarios are provided through the organizer platform (`https://onsite.com.cn/`) and used only for evaluation under its rules. We referenced parts of the Apache-2.0-licensed VBD implementation (`https://github.com/SafeRoboticsLab/VBD`) [18].

Cross-domain diagnostics. We apply the same feature extractor to WOMD Train and the UniDBO-Waymax, INTERACTION, and OnSite rollout sets to characterize cross-domain differences. For each scene, we compute speed P90 and acceleration-magnitude P95 from valid vehicle states, and pair-distance P05 and interaction-pair density from valid vehicle pairs. Figure 7 shows the per-scene distributions of all four; Table 7 aggregates the first three scene-level statistics using their corresponding cross-scene percentiles.

Table 7: Cross-domain motion and interaction statistics (cross-scene percentiles).

Data source / rollout set	Speed P90 (m/s)	Accel. P95 (m/s^2)	Pair-distance P05 (m)
WOMD Train	18.38	10.35	6.37
UniDBO-Waymax	18.42	3.60	5.17
INTERACTION	11.35	1.46	5.78
OnSite	21.25	19.07	3.33

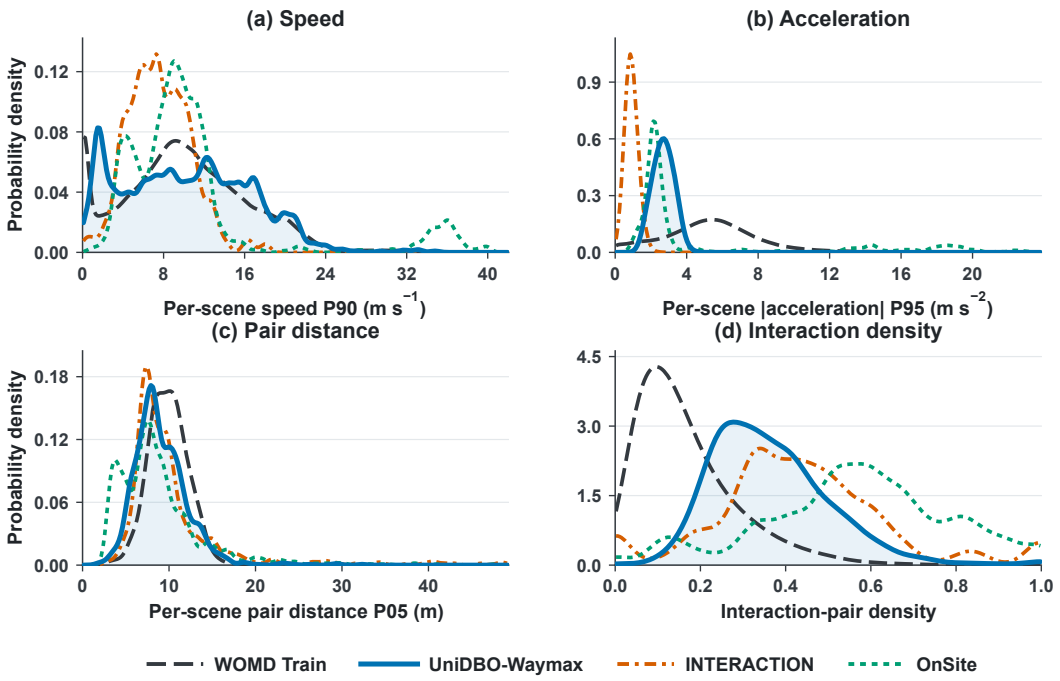

Figure 7: Cross-domain per-scene distributions of (a) speed P90, (b) acceleration-magnitude P95, (c) pair-distance P05, and (d) interaction-pair density.

UniDBO-Waymax retains a speed scale comparable to WOMD Train but has shorter pair distances. INTERACTION has lower speed P90 and acceleration-magnitude P95, whereas OnSite has higher values for both, shorter pair distances, and higher interaction-pair density. These diagnostics characterize differences among the evaluated domains, but do not establish exhaustive scenario coverage or safety-sufficient validation.

B.2 Evaluation Metrics and Protocols

Closed-loop evaluation. For WOMD/Waymax and INTERACTION, DHN generates joint controls from the current simulated history once per second. The simulator executes the first 10 predicted frames at 10 Hz before replanning. We compute the five closed-loop quality metrics in Table 8 using the same Waymax implementations for both datasets [8, 11].

Open-loop evaluation. We follow each method's open-loop validation protocol used during training and compute minADE and minFDE on the validation set to measure trajectory fitting accuracy. UniDBO's denoising recovery evaluation follows VBD and CDPT [2, 18].

Closed-loop drift. We measure drift between simulated and logged controlled-agent states at aligned 1–8 s snapshots, including the terminal state [4, 8, 11, 58]. For scenario n, snapshot k, and agent i, let $(\hat{\mathbf{p}}_{nki}, \hat{\mathbf{v}}_{nki}, \hat{\psi}_{nki})$ and $(\mathbf{p}_{nki}, \mathbf{v}_{nki}, \psi_{nki})$ denote simulated and logged position, velocity, and yaw. Define $\Delta\mathbf{p}_{nki} = \hat{\mathbf{p}}_{nki} - \mathbf{p}_{nki} = (\Delta x_{nki}, \Delta y_{nki})$, with logged-heading tangent $\mathbf{t}_{nki} = (\cos\psi_{nki}, \sin\psi_{nki})$ and normal $\mathbf{n}_{nki} = (-\sin\psi_{nki}, \cos\psi_{nki})$. The six drift measures are

$$\begin{aligned} e^{\mathrm{L1}}_{nki} &= |\Delta x_{nki}| + |\Delta y_{nki}|, & e^{\mathrm{L2}}_{nki} &= \sqrt{\Delta x^2_{nki} + \Delta y^2_{nki}}, \\ e^{\mathrm{long}}_{nki} &= |\mathbf{t}^{\mathsf{T}}_{nki}\Delta\mathbf{p}_{nki}|, & e^{\mathrm{lat}}_{nki} &= |\mathbf{n}^{\mathsf{T}}_{nki}\Delta\mathbf{p}_{nki}|, \\ e^{\mathrm{speed}}_{nki} &= \left|\|\hat{\mathbf{v}}_{nki}\|_2 - \|\mathbf{v}_{nki}\|_2\right|, & e^{\mathrm{yaw}}_{nki} &= \left|\mathrm{atan2}(\sin\Delta\psi_{nki}, \cos\Delta\psi_{nki})\right|, \end{aligned} \tag{20}$$

with $\Delta\psi_{nki} = \hat{\psi}_{nki} - \psi_{nki}$.

Table 8: Closed-loop quality and drift metric definitions, units, and directions; lower is better.

Metric	Definition	Unit
Collision	Simulated footprint overlaps another valid object	% ↓
Off-road	Simulated footprint leaves the drivable surface	% ↓
Wrong-way	Travel outside legal forward paths or against permitted direction	% ↓
Kinematic Infeasibility	State transition violates vehicle-kinematics constraints	% ↓
Log Divergence	L_2 position deviation from time-aligned logged behavior	m ↓
Position L1	Absolute coordinate-wise position error	m ↓
Position L2	Euclidean position error	m ↓
Longitudinal	Absolute position error along the logged heading	m ↓
Lateral	Absolute position error normal to the logged heading	m ↓
Speed	Absolute difference in speed magnitude	m/s ↓
Yaw	Absolute wrapped heading difference	rad ↓

Let $\mathcal{A}_{nk}$ denote the valid controlled-agent set at snapshot k, and let $\mathcal{K}_n$ contain the noninitial 1–8 s snapshots, including the terminal 8-s state. For any drift quantity q, the reported scene-macro value is

$$\mu_{nk}^{(q)} = \frac{1}{|\mathcal{A}_{nk}|} \sum_{i \in \mathcal{A}_{nk}} e_{nki}^{(q)}, \qquad \bar{\mu}_n^{(q)} = \frac{1}{|\mathcal{K}_n|} \sum_{k \in \mathcal{K}_n} \mu_{nk}^{(q)}, \qquad M^{(q)} = \frac{1}{N_q} \sum_{n=1}^{N_q} \bar{\mu}_n^{(q)}, \tag{21}$$

where N_q is the number of valid scenarios, so each scenario receives equal weight.

OnSite scoring. All methods are evaluated on the same OnSite scenarios with the same external ego planners and scoring protocol [2, 52]. The OnSite score has four components assessing safety, motion-distribution similarity, comfort, and testing utility. Kinematic Consistency compares speed, longitudinal acceleration, and angular velocity distributions using Jensen–Shannon divergence [59]; Comfort measures exceedances of acceleration, jerk, and yaw-rate thresholds. Table 9 gives the score mappings, where $T_{\mathrm{TTC}<\tau_{\mathrm{TTC}}}$ is the duration with time to collision below the prescribed threshold, T_{total} is the scenario duration, and r_{real} and c_{raw} are the raw realism and comfort measures.

Table 9: OnSite component-score mappings (higher is better).

Component	Range	Score mapping
Safety Score	0–30	$30\big(1 - T_{\mathrm{TTC}<\tau_{\mathrm{TTC}}}/T_{\mathrm{total}}\big)$ if collision-free; otherwise 0
Kinematic Consistency	0–20	$20(1 - \sqrt{r_{\mathrm{real}}})$
Comfort Score	0–10	$10 - 2c_{\mathrm{raw}}$
Test Effectiveness	0–40	$40\times$ benchmark scene-score average
Total Score	0–100	Sum of the four components

Test Effectiveness compares external ego-planner safety, efficiency, and comfort in generated scenes with the corresponding recorded scenes; lower relative planner performance in a generated scene yields a higher score. A dynamic-constraint violation sets all components and the Total Score to zero while retaining the scene in the valid-submission denominator; invalid submissions are handled separately.

Runtime accounting. For N_{step} simulator steps in the same Waymax environment [8], let $T_{\mathrm{inf}}, T_{\mathrm{trans}}, T_{\mathrm{met}}$ denote cumulative inference, transition, and metric-computation times in seconds. We report

$$\begin{aligned} \mathrm{Inf.} &= 1000\, T_{\mathrm{inf}}/N_{\mathrm{step}}, \\ \mathrm{Total} &= 1000\,(T_{\mathrm{inf}} + T_{\mathrm{trans}} + T_{\mathrm{met}})/N_{\mathrm{step}} \quad (\mathrm{ms/step}). \end{aligned} \tag{22}$$

Both quantities are amortized over simulator steps rather than measured per replanning call.

B.3 Baselines and Comparison Implementations

Comparison methods and interface. The main WOMD/Waymax and zero-shot INTERACTION comparisons include CAT-K [14], SMART [12], VBD [18], and CDPT [2]. The OnSite comparison additionally includes IDM [53], SAC [2, 54], MJTG [55], and EGPO [56]. The OnSite results for overlapping methods in Table 5 are consistent with those reported in CDPT [2]. We adapt Continuous DHN, Diffusion Forcing, and Self-Forcing to the same scene inputs, normalized control representation, scene encoder, kinematic decoder, and evaluation interface as UniDBO. These adaptations use the same position, heading, and control-loss definitions, with method-specific weights.

Continuous DHN control. Continuous DHN interpolates the discrete cumulative noise schedule. With $q = \omega t_{\max}$, $k = \lfloor q \rfloor$, and $\delta = q - k$, we define

$$\bar{\alpha}(\omega) = (1-\delta)\bar{\alpha}_k + \delta\bar{\alpha}_{\lceil q \rceil}, \qquad \tilde{\mathbf{x}}_\omega = \sqrt{\bar{\alpha}(\omega)}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}(\omega)}\,\boldsymbol{\epsilon}, \tag{23}$$

where ω is sampled from the curriculum-defined high-noise interval within $[0, 1]$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. Continuous DHN conditions recovery on interpolated noise levels rather than discrete timestep indices.

Diffusion Forcing adaptation. The Diffusion Forcing adaptation partitions the 80-frame target into $K = 40$ synchronized two-frame control blocks, samples one timestep t_j for each scene and block, and applies temporally causal attention:

$$\tilde{\mathbf{x}}_{t_j}^{(j)} = \sqrt{\bar{\alpha}_{t_j}}\,\mathbf{x}_0^{(j)} + \sqrt{1-\bar{\alpha}_{t_j}}\,\boldsymbol{\epsilon}^{(j)}, \qquad \mathcal{A}\big((i,j),(i',j')\big) = \mathbf{1}[j' \le j], \tag{24}$$

where $\boldsymbol{\epsilon}^{(j)} \sim \mathcal{N}(0, \mathbf{I})$ is sampled independently across blocks. Each block attends to all agents in the current and preceding blocks, but not future blocks. Recovered controls are decoded with the shared kinematic decoder.

Self-Forcing adaptation. The Self-Forcing adaptation predicts controls from terminal noise, retains and decodes the first control block, and appends the generated state to the subsequent context:

$$\hat{\mathbf{x}}^{(j)} = f_{\mathrm{SF}}(\boldsymbol{\epsilon}^{(j)}, E_\phi(\mathcal{X}^{(j)}), t^\star), \qquad \mathcal{X}^{(j+1)} = \mathrm{Update}\Big(\mathcal{X}^{(j)}, \mathrm{sg}\Big[\mathcal{K}(\hat{\mathbf{x}}_1^{(j)}; \mathcal{X}^{(j)})\Big]\Big), \tag{25}$$

where sg is the stop-gradient operator, $\hat{\mathbf{x}}_1^{(j)}$ is the first predicted control block, and $\boldsymbol{\epsilon}^{(j)} \sim \mathcal{N}(0, \mathbf{I})$. Subsequent predictions use model-generated histories, but gradients do not propagate through these context updates.

C Theoretical Analysis

This section provides theoretical explanations for UniDBO's three design choices: direct one-step recovery, progressive high-noise training, and CS-assisted joint learning. We also connect per-replan recovery error to finite-horizon closed-loop state drift and briefly consider auxiliary open-loop recovery. We first summarize the claims and evidence and then present the derivations.

C.1 Overview of Theoretical Claims

Table 10 summarizes what each result establishes and how it relates to the controlled experiments in Section 4.3 and Appendix D.3.

Table 10: Theoretical claims and corresponding empirical tests.

Design question	Main conclusion	Empirical evidence
Why one-step?	Reverse-path error can exceed refinement gain	Full-range 1 vs. 50 NFE
Why high-noise?	Severe-noise recovery is a robustness proxy	Full-range vs. DHN
Why joint training?	Aligned CS gradients can aid DHN; open-loop bound is conditional	Branch and single-head ablations
Why lower drift?	Per-replan errors bound finite-horizon state drift	Aligned 1–8 s drift

The following subsections derive these claims under explicit assumptions, complementing the controlled comparisons summarized in Table 10.

C.2 Direct One-Step versus Iterative DDPM Reverse Sampling

Prior studies show that prediction errors may propagate and accumulate during iterative reverse sampling, leading to a mismatch between training and sampling inputs [21, 51, 60]. We analyze one-step and iterative DDPM control recovery from the same terminal state using an x_0-prediction denoiser.

The 50-step protocol applies stochastic DDPM posterior sampling over the zero-based timestep grid $t = 49, \dots, 0$. Define $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{j=0}^{t} \alpha_j$, and $\bar{\alpha}_{-1} = 1$. Let $\mathbf{z}_t$ denote the reverse-chain state, $\mathbf{x}_0$ the clean control target, and D_θ an x_0-prediction denoiser. One DDPM reverse transition is

$$\mathbf{z}_{t-1} = \lambda_t \mathbf{z}_t + \gamma_t D_\theta(\mathbf{z}_t, \mathcal{X}, t) + \sigma_t \boldsymbol{\xi}_t, \tag{26}$$

where

$$\gamma_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}, \qquad \lambda_t = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}, \tag{27}$$

where σ_t is the DDPM posterior standard deviation, distinct from the CS noise scale σ.

To analyze recovery error, we synchronously couple the sampled chain with an oracle chain. The chains share the terminal state and Gaussian innovations $\boldsymbol{\xi}_t$, while the oracle replaces each denoiser output with the clean control $\mathbf{x}_0$. Define

$$\mathbf{e}_t = \mathbf{z}_t - \mathbf{z}_t^\star, \qquad \boldsymbol{\delta}_t = D_\theta(\mathbf{z}_t^\star, \mathcal{X}, t) - \mathbf{x}_0. \tag{28}$$

A local Taylor expansion of the denoiser gives

$$D_\theta(\mathbf{z}_t, \mathcal{X}, t) - \mathbf{x}_0 = \boldsymbol{\delta}_t + \mathbf{J}_t \mathbf{e}_t + \mathbf{r}_t, \qquad \|\mathbf{r}_t\| = O(\|\mathbf{e}_t\|^2). \tag{29}$$

Because the posterior innovations are shared, they cancel in the difference between the two transitions, yielding

$$\mathbf{e}_{t-1} = \underbrace{(\lambda_t \mathbf{I} + \gamma_t \mathbf{J}_t)}_{\mathbf{A}_t} \mathbf{e}_t + \gamma_t \boldsymbol{\delta}_t + \gamma_t \mathbf{r}_t. \tag{30}$$

The term $\mathbf{A}_t \mathbf{e}_t$ propagates the existing path error, while $\gamma_t \boldsymbol{\delta}_t$ and $\gamma_t \mathbf{r}_t$ capture the current prediction residual and local expansion remainder, respectively. For $T = t_{\max} = 49$, the reverse process starts from $\mathbf{z}_T$ and returns the final control $\mathbf{z}_{-1}$. Unrolling Equation 30 over an L-evaluation path gives

$$\mathbf{e}_{\text{out}}^{(L)} = \mathbf{z}_{-1} - \mathbf{x}_0 = \mathbf{P}_T \boldsymbol{\delta}_T + \mathbf{h}_L, \tag{31}$$

where $\mathbf{P}_T$ collects the relevant local propagation matrices and DDPM coefficients, and $\mathbf{h}_L$ aggregates the intermediate path residuals and Taylor remainders. Direct one-step prediction instead yields $\mathbf{e}_{\text{out}}^{(1)} = \boldsymbol{\delta}_T$.

Theorem 1 (Direct-versus-iterative DDPM recovery decomposition)**.** *Define*

$$\mathcal{E}_1 = \mathbb{E}\|\boldsymbol{\delta}_T\|_2^2, \qquad \mathcal{E}_L = \mathbb{E}\|\mathbf{e}_{\text{out}}^{(L)}\|_2^2, \tag{32}$$

the refinement gain

$$\mathcal{G}_L = \mathbb{E}\|\boldsymbol{\delta}_T\|_2^2 - \mathbb{E}\|\mathbf{P}_T \boldsymbol{\delta}_T\|_2^2, \tag{33}$$

and the additional reverse-path contribution

$$\mathcal{P}_L = \mathbb{E}\|\mathbf{h}_L\|_2^2 + 2\mathbb{E}\langle \mathbf{P}_T \boldsymbol{\delta}_T, \mathbf{h}_L \rangle. \tag{34}$$

Then

$$\boxed{\mathcal{E}_L - \mathcal{E}_1 = \mathcal{P}_L - \mathcal{G}_L}, \qquad \boxed{\mathcal{E}_1 < \mathcal{E}_L \iff \mathcal{P}_L > \mathcal{G}_L}. \tag{35}$$

Proof. Substituting Equation 31 into $\mathcal{E}_L$ gives $\mathcal{E}_L = \mathbb{E}\|\mathbf{P}_T \boldsymbol{\delta}_T\|^2 + \mathbb{E}\|\mathbf{h}_L\|^2 + 2\mathbb{E}\langle \mathbf{P}_T \boldsymbol{\delta}_T, \mathbf{h}_L \rangle$. Subtracting $\mathcal{E}_1$ and collecting the corresponding terms proves both statements. □

The decomposition identifies the trade-off between terminal-residual refinement and reverse-path error. The term $\mathcal{G}_L$ measures the reduction in mean squared error of the terminal residual after propagation, whereas $\mathcal{P}_L$ includes the intermediate path residuals and their interaction with the propagated terminal residual; neither term is assumed nonnegative. When $\mathcal{P}_L > \mathcal{G}_L$, one-step recovery has lower control error, providing a conditional explanation for the comparison in Table 3.

Proposition 1 (NFE cost ordering)**.** *If one denoiser evaluation has positive cost C_D, an $L > 1$ reverse sampler has at least $(L-1)C_D$ more denoiser cost than direct one-step prediction, excluding its additional reverse-transition overhead.*

C.3 High-Noise Direct Recovery as a Closed-Loop Robustness Proxy

We analyze how progressive high-noise training supports the terminal Gaussian query used at each replan. The argument first bounds the gap between curriculum and terminal-query recovery risks, then accounts for the scene-target distribution shift at evaluation. Appendix C.5 links the resulting control-recovery risk to the measured finite-horizon drift from logged states.

For $T = t^\star = 49$, let $d = 2AK$ denote the dimension of the vectorized joint control tensor under a fixed padded agent count. The DDPM forward corruption is

$$\tilde{\mathbf{x}}_t = a_t \mathbf{x}_0 + b_t \boldsymbol{\epsilon}, \qquad a_t = \sqrt{\bar{\alpha}_t}, \quad b_t = \sqrt{1 - \bar{\alpha}_t}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}_d). \tag{36}$$

Under a shared $(\mathbf{x}_0, \boldsymbol{\epsilon})$ coupling, the schedule-induced distance to the terminal corruption is

$$d_{\mathrm{DDPM}}(t, T) = \sqrt{(a_t - a_T)^2 \mathbb{E}\|\mathbf{x}_0\|_2^2 + (b_t - b_T)^2 d}. \tag{37}$$

The implemented DHN remains $f_d(\tilde{\mathbf{x}}_t, \mathbf{H}, t)$. For analysis only, define the normalized timestep descriptor $\chi_t = t/T$, with $\chi_T = 1$; this descriptor neither changes the model input nor replaces the discrete timestep implementation. Let $\zeta = (\mathcal{X}, \mathbf{x}_0, \mathbf{y})$ collect a scene, its clean control and trajectory targets, and the associated validity masks. We use $\Psi_\theta(\zeta, \mathbf{z}, \chi)$ to denote the per-record weighted DHN loss $\mathcal{L}_d$ induced by noisy input $\mathbf{z}$ and the corresponding timestep condition. For each fixed ζ, assume that the following Lipschitz condition holds over the relevant region and timestep grid:

$$|\Psi_\theta(\zeta, \mathbf{z}, \chi) - \Psi_\theta(\zeta, \mathbf{z}', \chi')| \le L_z \|\mathbf{z} - \mathbf{z}'\|_2 + L_\chi |\chi - \chi'|, \qquad L_z, L_\chi \ge 0.$$

This assumption controls variation in both the noisy control and the explicit timestep condition while holding the supervision target fixed. Define

$$\mathcal{R}_t^d(\theta) = \mathbb{E}_{\zeta, \boldsymbol{\epsilon}}[\Psi_\theta(\zeta, \tilde{\mathbf{x}}_t, \chi_t)], \qquad \mathcal{R}_{\mathrm{qry}}^d(\theta) = \mathbb{E}_{\zeta, \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_d)}[\Psi_\theta(\zeta, \mathbf{z}, 1)],$$

and let $\mathcal{R}_g^d(\theta) = \mathbb{E}_{t \sim q_g} \mathcal{R}_t^d(\theta)$ denote the curriculum risk. We further define

$$\mathcal{A}_g = \mathbb{E}_{t \sim q_g}\left[L_z d_{\mathrm{DDPM}}(t, T) + L_\chi \frac{|t - T|}{T}\right], \qquad \Gamma_T = L_z \mathbb{E}_\zeta W_1(\mathcal{L}(\tilde{\mathbf{x}}_T \mid \zeta), \mathcal{N}(0, \mathbf{I}_d)), \tag{38}$$

where $\mathcal{A}_g$ jointly quantifies the curriculum-to-terminal mismatch in the noisy control and explicit timestep condition, whereas Γ_T captures the remaining conditional mismatch between terminal DDPM corruption and an independent Gaussian query for the same supervised record.

Lemma 1 (Curriculum-to-query risk alignment)**.** *For every fixed θ,*

$$\left|\mathcal{R}_g^d(\theta) - \mathcal{R}_{\mathrm{qry}}^d(\theta)\right| \le \mathcal{A}_g + \Gamma_T. \tag{39}$$

If $\hat{\theta}_g$ satisfies $\mathcal{R}_g^d(\hat{\theta}_g) \le \inf_\theta \mathcal{R}_g^d(\theta) + \varepsilon_g$, then, for a terminal-query-risk minimizer $\theta_{\mathrm{qry}}^\star$,

$$\mathcal{R}_{\mathrm{qry}}^d(\hat{\theta}_g) - \mathcal{R}_{\mathrm{qry}}^d(\theta_{\mathrm{qry}}^\star) \le \varepsilon_g + 2(\mathcal{A}_g + \Gamma_T). \tag{40}$$

Proof. At fixed ζ, the conditional Kantorovich–Rubinstein bound and shared forward-corruption coupling yield $|\mathcal{R}_t^d - \mathcal{R}_T^d| \le L_z d_{\mathrm{DDPM}}(t, T) + L_\chi |t - T|/T$ and $|\mathcal{R}_T^d - \mathcal{R}_{\mathrm{qry}}^d| \le \Gamma_T$. Taking the expectation over the curriculum and applying the triangle inequality proves Equation 39. Applying this bound to $\hat{\theta}_g$, invoking its ε_g-optimality for $\mathcal{R}_g^d$, and then applying the bound to $\theta_{\mathrm{qry}}^\star$ proves Equation 40. □

Corollary 1 (Nested-support alignment)**.** *For the monotone DDPM schedule, a_t decreases and b_t increases toward T, so both $d_{\mathrm{DDPM}}(t, T)$ and $|t - T|/T$ decrease with t. Consequently, removing the lowest timestep from a uniform tail support cannot increase the average joint mismatch: for $\{34, \ldots, 49\} \supset \{35, \ldots, 49\} \supset \cdots \supset \{49\}$,*

$$\boxed{\mathcal{A}_{g+1} \le \mathcal{A}_g.}$$

The corollary formalizes increasing alignment with the terminal condition: both corruption-space and timestep-conditioning mismatches decrease as the support moves toward T. If training reaches the scheduled curriculum endpoint, $q_{G_{\mathrm{cur}}}$ is concentrated at $T = 49$, so $\mathcal{A}_{G_{\mathrm{cur}}} = 0$. For checkpoints

obtained before this endpoint, the general risk bound retains the curriculum-to-terminal mismatch term $\mathcal{A}_g$. The separate mismatch Γ_T between terminal forward corruption and the Gaussian query remains.

Let μ_ζ denote the training distribution of scene-target records and $\mu^{\log}_{\zeta,s}$ the distribution of logged evaluation records at replan s. Each logged record supplies its clean controls and validity masks. Under a fixed metric on these records, define the evaluation shift as

$$\Xi^{\log}_s = W_1(\mu_\zeta, \mu^{\log}_{\zeta,s}). \tag{41}$$

Assume that $\zeta \mapsto \mathbb{E}_{\mathbf{z}} \Psi_\theta(\zeta, \mathbf{z}, 1)$ is L_{ctx}-Lipschitz on the compared record distributions. The logged-state terminal-query risk is

$$\mathcal{R}^d_{s,\log}(\theta) = \mathbb{E}_{\zeta \sim \mu^{\log}_{\zeta,s},\, \mathbf{z} \sim \mathcal{N}(0,\mathbf{I}_d)} [\Psi_\theta(\zeta, \mathbf{z}, 1)] \,.$$

Theorem 2 (High-noise recovery risk at evaluation)**.** *The DHN terminal-query recovery risk on logged evaluation records satisfies*

$$\boxed{\mathcal{R}^d_{s,\log}(\theta) \le \mathcal{R}^d_g(\theta) + \mathcal{A}_g + \Gamma_T + L_{\mathrm{ctx}} \Xi^{\log}_s.} \tag{42}$$

Proof. The distributional Lipschitz condition gives $\mathcal{R}^d_{s,\log} \le \mathcal{R}^d_{\mathrm{qry}} + L_{\mathrm{ctx}} \Xi^{\log}_s$. Equation 39 gives $\mathcal{R}^d_{\mathrm{qry}} \le \mathcal{R}^d_g + \mathcal{A}_g + \Gamma_T$. Combining these inequalities proves the result. □

Equation 42 separates curriculum recovery, terminal-query alignment, and evaluation shift. Appendix C.5 propagates this logged-state recovery risk to the measured finite-horizon drift.

C.4 Auxiliary Transfer from Continuous-Scale Supervision

For a local analysis of the shared encoder, hold the branch-specific heads fixed and define

$$\mathcal{L}_d = w_d \mathcal{L}^d_{\mathrm{sup}} + w_a \mathcal{L}^d_{\mathrm{act}}, \quad \mathcal{L}_c = w_c \mathcal{L}^c_{\mathrm{sup}}, \quad \mathbf{g}_d = \nabla_\phi \mathcal{L}_d, \quad \mathbf{g}_c = \nabla_\phi \mathcal{L}_c. \tag{43}$$

We compare the DHN-only and joint encoder updates

$$\phi^+_d = \phi - \eta \mathbf{g}_d, \qquad \phi^+_J = \phi - \eta(\mathbf{g}_d + \mathbf{g}_c). \tag{44}$$

Proposition 2 (Sufficient condition for positive auxiliary transfer)**.** *Suppose $\mathcal{L}_d$ has an L_ϕ-Lipschitz gradient along the two update segments. Then*

$$\mathcal{L}_d(\phi^+_J) - \mathcal{L}_d(\phi^+_d) \le -\eta \langle \mathbf{g}_d, \mathbf{g}_c \rangle + L_\phi \eta^2 \left(\|\mathbf{g}_d\| \, \|\mathbf{g}_c\| + \frac{1}{2} \|\mathbf{g}_c\|^2 \right). \tag{45}$$

Consequently,

$$\boxed{\langle \mathbf{g}_d, \mathbf{g}_c \rangle > L_\phi \eta \left(\|\mathbf{g}_d\| \, \|\mathbf{g}_c\| + \frac{1}{2} \|\mathbf{g}_c\|^2 \right)} \tag{46}$$

is sufficient for $\mathcal{L}_d(\phi^+_J) < \mathcal{L}_d(\phi^+_d)$.

Proof. Applying smoothness to the step from ϕ^+_d to $\phi^+_J = \phi^+_d - \eta \mathbf{g}_c$ gives

$$\mathcal{L}_d(\phi^+_J) \le \mathcal{L}_d(\phi^+_d) - \eta \langle \nabla \mathcal{L}_d(\phi^+_d), \mathbf{g}_c \rangle + \frac{L_\phi \eta^2}{2} \|\mathbf{g}_c\|^2. \tag{47}$$

Gradient Lipschitzness further gives $\|\nabla \mathcal{L}_d(\phi^+_d) - \mathbf{g}_d\| \le L_\phi \eta \|\mathbf{g}_d\|$. The Cauchy–Schwarz inequality then yields Equation 45; condition 46 makes its right-hand side strictly negative. □

This proposition gives a local sufficient condition for positive auxiliary transfer. Appendix D.3.3 reports the closed-loop DHN-versus-UniDBO ablation.

For the open-loop comparison, also define $\phi^+_c = \phi - \eta \mathbf{g}_c$.

For open-loop recovery under the evaluation corruption distribution, define

$$\mathcal{R}_o(\phi) = \mathbb{E}_{\mathcal{X}, \mathbf{x}_0, \mathbf{y}, \sigma, \boldsymbol{\epsilon}_c} \left[\ell_o(\mathcal{K}(f_c(\tilde{\mathbf{x}}_\sigma, E_\phi(\mathcal{X}), \sigma); \mathcal{X}), \mathbf{y}) \right], \tag{48}$$

where σ follows the evaluation noise-scale distribution, $\boldsymbol{\epsilon}_c \sim \mathcal{N}(0, \mathbf{I})$ is independent of the record, and ℓ_o is the average positional Smooth-L1 loss.

Proposition 3 (Conditional open-loop transfer)**.** *If $\mathcal{R}_o$ has an L_o-Lipschitz gradient along the compared segment, then*

$$\mathcal{R}_o(\phi_J^+) - \mathcal{R}_o(\phi_c^+) \le -\eta\langle \nabla\mathcal{R}_o(\phi_c^+), \mathbf{g}_d\rangle + \tfrac{1}{2}L_o\eta^2\|\mathbf{g}_d\|^2. \tag{49}$$

Proof. Apply the descent lemma to $\mathcal{R}_o$ at ϕ_c^+ along $\phi_J^+ - \phi_c^+ = -\eta\mathbf{g}_d$. □

On a bounded positional-error domain, squared error is bounded by a finite multiple c_M of Smooth-L1. Let ADE and FDE denote the displacement errors of the evaluated recovered trajectory. Then

$$\mathbb{E}[\mathrm{ADE}] \le \sqrt{c_M\mathcal{R}_o}, \qquad \mathbb{E}[\mathrm{FDE}] \le \sqrt{c_M T_f \mathcal{R}_o}. \tag{50}$$

These bounds follow from the squared-error comparison, Cauchy–Schwarz, and Jensen's inequality, assuming consistent valid-frame averaging and a valid terminal target. The bounds do not predict the reported metric values: training-gradient alignment need not imply alignment with $\nabla\mathcal{R}_o$, since the minibatch objective differs from population evaluation risk. Table 4 provides the empirical comparison.

C.5 From Direct Control Recovery to Closed-Loop State Drift

We compare the generated rollout with the time-aligned logged states used by the drift evaluator in Appendix B.2. Here the simulator state includes the recent history needed to construct the scene condition. At replan s, let $\mathbf{s}_s^{\log}$ and $\mathbf{x}_{0,s}^{\log}$ be the logged state and its clean control target, and set $\mathcal{X}_s^{\log} = \mathcal{C}(\mathbf{s}_s^{\log})$. The actual rollout uses $\hat{\mathbf{s}}_s$ and an independent Gaussian query $\mathbf{z}_s$. For analysis, use the same query at the logged state to define

$$\begin{aligned} \hat{\mathbf{x}}_{0,s}^{\log} &= f_d\big(\mathbf{z}_s, E_\phi(\mathcal{X}_s^{\log}), t^\star\big)\,, \\ \mathbf{e}_s &= \hat{\mathbf{x}}_{0,s}^{\log} - \mathbf{x}_{0,s}^{\log}, \qquad \varepsilon_s = \|M_s \odot \mathbf{e}_s\|_2, \end{aligned} \tag{51}$$

where M_s selects the valid controlled-agent action blocks and covers every control coordinate used by the executed prefix. Write $J_s = |\Omega_{a,s}|$, with $1 \le J_s \le AK$, and let $\mathcal{L}_{\mathrm{act},s}^d$ be the per-record action loss over these blocks. The risk $\mathcal{R}_{s,\log}^d$ is defined in Appendix C.3 for the same logged records and Gaussian query.

Lemma 2 (Smooth-L1 control recovery)**.**

$$\boxed{\mathbb{E}\varepsilon_s \le h\big(\mathcal{R}_{s,\log}^d\big)\,, \qquad h(R) = \sqrt{\frac{2AK}{w_a}R} + \frac{2AK}{w_a}R.} \tag{52}$$

Proof. For the Smooth-L1 penalty φ, split the valid scalar errors into $|u| < 1$ and $|u| \ge 1$. The former satisfy $u^2 = 2\varphi(u)$, while the latter satisfy $|u| \le 2\varphi(u)$. With $H_s = \sum_{(i,j)\in\Omega_{a,s}}\sum_{k=1}^{2}\varphi(e_{s,i,j,k}) = J_s\mathcal{L}_{\mathrm{act},s}^d$, the triangle inequality gives $\varepsilon_s \le \sqrt{2H_s} + 2H_s$. Since $J_s \le AK$ and the nonnegative per-record DHN loss obeys $\Psi_\theta \ge w_a\mathcal{L}_{\mathrm{act},s}^d$, taking expectations and applying Jensen's inequality to the square-root term proves Equation 52. □

Let $\mathcal{G}_r(\mathbf{x};\mathcal{X})$ be the executed-prefix map in Equation 2, and define the query-conditioned prefix policy $\hat{\pi}_{\mathbf{z}}(\mathbf{s}) = \mathcal{G}_r(f_d(\mathbf{z}, E_\phi(\mathcal{C}(\mathbf{s})), t^\star); \mathcal{C}(\mathbf{s}))$. The logged control target produces the prefix $\mathbf{u}_s^{\log} = \mathcal{G}_r(\mathbf{x}_{0,s}^{\log}; \mathcal{X}_s^{\log})$. Its one-step replay residual is $b_s = \|F_r(\mathbf{s}_s^{\log}, \mathbf{u}_s^{\log}, \mathcal{E}_s) - \mathbf{s}_{s+1}^{\log}\|$. On the compared finite-horizon states, assume finite constants L_x, L_u, L_π, L_G such that

$$\begin{aligned} \|F_r(\mathbf{s},\mathbf{u},\mathcal{E}) - F_r(\mathbf{s}',\mathbf{u}',\mathcal{E})\| &\le L_x\|\mathbf{s}-\mathbf{s}'\| + L_u\|\mathbf{u}-\mathbf{u}'\|, \\ \|\hat{\pi}_{\mathbf{z}}(\mathbf{s}) - \hat{\pi}_{\mathbf{z}}(\mathbf{s}')\| &\le L_\pi\|\mathbf{s}-\mathbf{s}'\|, \\ \|\mathcal{G}_r(\mathbf{x};\mathcal{X}_s^{\log}) - \mathcal{G}_r(\mathbf{x}';\mathcal{X}_s^{\log})\| &\le L_G\|M_s \odot (\mathbf{x}-\mathbf{x}')\|_2. \end{aligned} \tag{53}$$

The policy inequality uses the same Gaussian query on both states. Let $\Delta_s = \|\hat{\mathbf{s}}_s - \mathbf{s}_s^{\log}\|$ and $\rho = L_x + L_u L_\pi$.

Theorem 3 (Finite-horizon drift from logged states)**.** *For the rollout and logged states under the same exogenous context,*

$$\boxed{\begin{aligned}&\Delta_{s+1} \le \rho\Delta_s + L_u L_G \varepsilon_s + b_s,\\ &\Delta_{S+1} \le \rho^S \Delta_1 + \sum_{v=1}^{S} \rho^{S-v}(L_u L_G \varepsilon_v + b_v).\end{aligned}} \tag{54}$$

Consequently, with $D_s = \mathbb{E}\Delta_s$*,*

$$\boxed{\begin{aligned}D_{S+1} \le\ & \rho^S D_1\\ &+ \sum_{v=1}^{S} \rho^{S-v}\left[L_u L_G h\big(\mathcal{R}^d_{v,\log}\big) + \mathbb{E}b_v\right].\end{aligned}} \tag{55}$$

Proof. Insert $F_r(\mathbf{s}_s^{\log}, \hat{\pi}_{\mathbf{z}_s}(\mathbf{s}_s^{\log}), \mathcal{E}_s)$ and $F_r(\mathbf{s}_s^{\log}, \mathbf{u}_s^{\log}, \mathcal{E}_s)$ between the generated next state and the logged next state. The first two differences are at most $(L_x + L_u L_\pi)\Delta_s$ and $L_u L_G \varepsilon_s$, respectively, by Equation 53; the last is b_s. This proves the one-step inequality. Iteration proves the finite-horizon inequality, and taking expectations with Lemma 2 proves Equation 55. □

Theorem 2 gives

$$\mathcal{R}^d_{s,\log} \le \mathcal{B}^{\log}_{g,s} := \mathcal{R}^d_g + \mathcal{A}_g + \Gamma_T + L_{\mathrm{ctx}} \Xi^{\log}_s. \tag{56}$$

Because h is increasing, the resulting risk-to-drift bound is

$$\boxed{\begin{aligned}D_{S+1} \le\ & \rho^S D_1\\ &+ \sum_{v=1}^{S} \rho^{S-v}\left[L_u L_G h\big(\mathcal{B}^{\log}_{g,v}\big) + \mathbb{E}b_v\right].\end{aligned}} \tag{57}$$

At each aligned evaluation snapshot k, let t_k be its simulation time. The position error of every valid controlled agent is a subvector of the joint-state error. Therefore the scene-macro position-L2 drift in Equation 21 satisfies

$$M^{(\mathrm{L2})} \le \frac{1}{N_{\mathrm{L2}}} \sum_{n=1}^{N_{\mathrm{L2}}} \frac{1}{|\mathcal{K}_n|} \sum_{k \in \mathcal{K}_n} \left\| \hat{\mathbf{s}}_n(t_k) - \mathbf{s}_n^{\log}(t_k) \right\|, \tag{58}$$

Thus, applying Equation 57 at the corresponding snapshots bounds the same time-aligned drift quantity reported in the experiments. With the evaluation distribution, replay residuals, and dynamics constants fixed, lower curriculum recovery risk tightens this finite-horizon bound.

D Supplementary Experimental Results

This section follows the experimental order of the main paper. We first provide supplementary WOMD/Waymax and zero-shot INTERACTION results, then report mechanism and ablation studies, weight sensitivity and speed postprocessing, and finally analyze stochastic closed-loop diversity.

D.1 Supplementary WOMD/Waymax Results

D.1.1 Statistical Reporting

We conduct three independent evaluation runs with different random seeds. Table 11 reports the mean and sample standard deviation of closed-loop quality and drift across the three runs.

Table 11: UniDBO closed-loop quality and aligned 1–8 s drift on the fixed 1,000-scenario WOMD/Waymax set. Values are the mean ± sample standard deviation across three evaluation seeds; lower is better.

(a) Closed-loop quality.

Method	Collision (%)↓	Off-road (%)↓	Wrong-way (%)↓	Kinematic Infeasibility (%)↓	Log Divergence↓
UniDBO	4.773 ± 0.031	3.721 ± 0.073	1.244 ± 0.020	0.0228 ± 0.0015	1.322 ± 0.012

(b) Aligned closed-loop drift.

Method	Position L1 (m)↓	Position L2 (m)↓	Long. (m)↓	Lat. (m)↓	Speed (m/s)↓	Yaw (rad)↓
UniDBO	3.0449 ± 0.0374	2.5638 ± 0.0290	2.3644 ± 0.0283	0.5412 ± 0.0072	0.8426 ± 0.0052	0.0713 ± 0.0012

D.1.2 Additional WOMD/Waymax Closed-Loop Rollouts

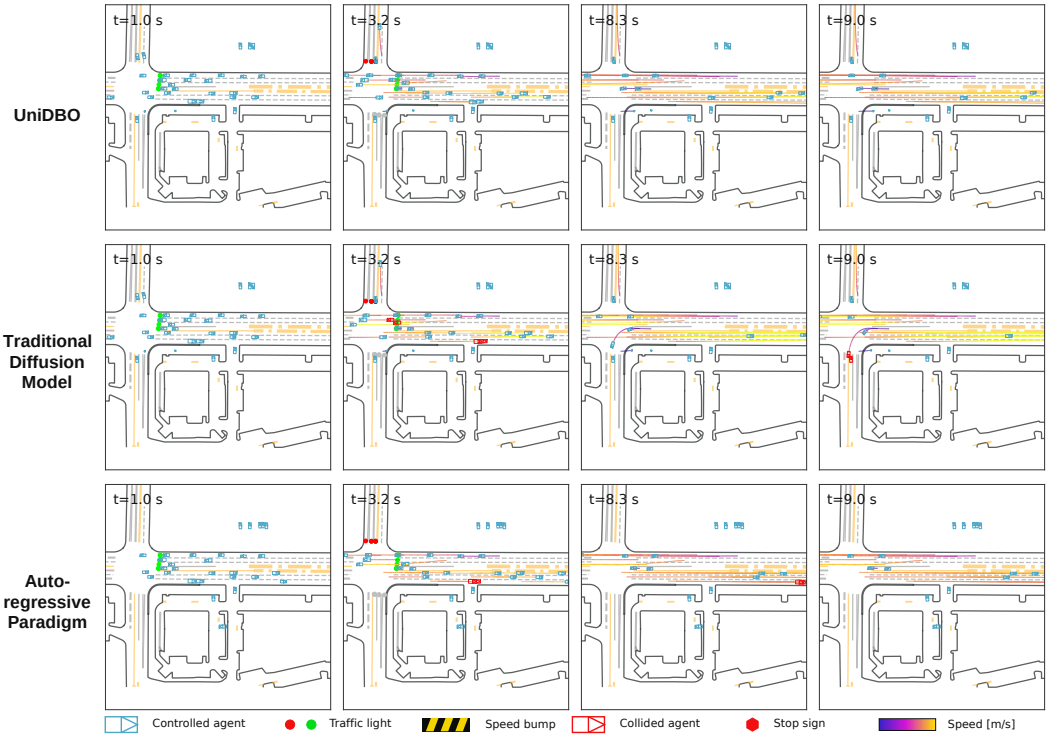

Figure 8: WOMD/Waymax closed-loop rollout in a corridor/T-junction interaction. Visualization conventions follow Figure 3.

Figures 3, 8, and 9 compare closed-loop rollouts at a signalized intersection, a corridor/T-junction, and a curved merge. In these examples, UniDBO maintains coordinated, road-compliant motion, whereas the baselines exhibit local collisions or larger late-horizon deviations, complementing the aggregate results in Table 1.



Figure 9: WOMD/Waymax closed-loop rollout along a curved arterial merge. Visualization conventions follow Figure 3.

D.2 Supplementary Zero-Shot INTERACTION Results

D.2.1 Additional INTERACTION Closed-Loop Rollouts

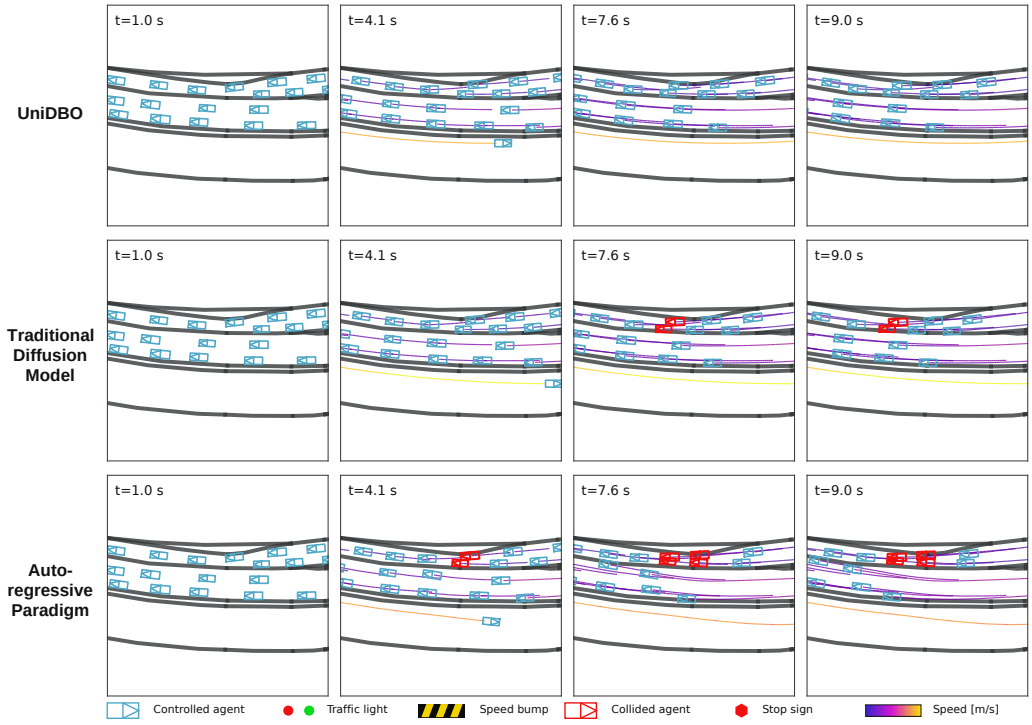

Figure 10: Zero-shot INTERACTION closed-loop rollout at a curved merge. Visualization conventions follow Figure 4.

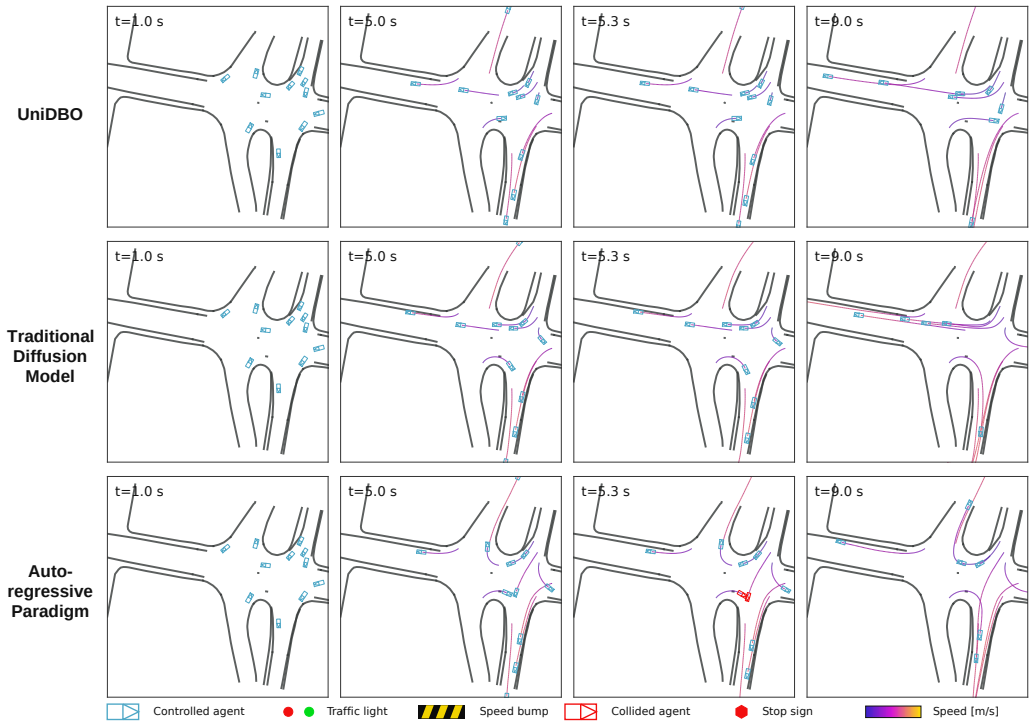

Figure 11: Zero-shot INTERACTION closed-loop rollout at a multi-arm junction. Visualization conventions follow Figure 4.

Figures 4, 10, and 11 show zero-shot INTERACTION rollouts at a roundabout, a curved merge, and a multi-arm junction. In these examples, UniDBO maintains coordinated route continuation without target-domain fine-tuning, whereas the baselines exhibit local conflicts or larger path deviations.

D.3 Mechanism and Ablation Studies

D.3.1 Extended One-Step versus Multi-Step Analysis

Table 12: Closed-loop drift of the Full-range noise model under 50-step and one-step inference on WOMD/Waymax, corresponding to Table 3. Values are scene-macro means over aligned 1–8 s snapshots. Lower is better; best values are **bolded**.

Configuration	NFE	L1 (m)↓	L2 (m)↓	Long. (m)↓	Lat. (m)↓	Speed (m/s)↓	Yaw (rad)↓
Full-range noise	50	5.1278	4.4398	4.0717	0.8838	1.5286	**0.0904**
Full-range noise	1	**4.6612**	**3.9750**	**3.6881**	**0.7705**	**1.2382**	**0.0904**

With the replanning interval held fixed, one-step inference yields lower values for five closed-loop drift measures than 50-step inference, with comparable yaw drift (Table 12). This result indicates that fewer internal denoiser evaluations can still maintain low closed-loop drift under the evaluated protocol, consistent with the conditional recovery-error analysis in Theorem 1.

Multi-step DDPM and DDIM sampler sweep. Table 13 shows that multi-step DDPM and DDIM sampling increases inference latency without improving the five closed-loop quality metrics relative to one-step UniDBO. Removing both input noise and denoising worsens Collision, Off-road, Wrong-way, and Log Divergence. Together, these results support noise-conditioned one-step recovery rather than additional iterative reverse sampling.

Table 13: Sampler comparison on WOMD/Waymax without speed postprocessing. DDPM and DDIM results are the mean $\pm$ sample standard deviation across NFE $\{10, 20, 50\}$; lower is better. Inf. and Total are measured in ms/step.

Sampler	NFE	Collision (%)↓	Off-road (%)↓	Wrong-way (%)↓	Kinematic Infeasibility (%)↓	Log Divergence↓	Inf.↓	Total↓
DDPM [19]	10/20/50	6.919 ± 0.168	6.367 ± 0.132	1.674 ± 0.057	0.3814 ± 0.0326	1.888 ± 0.059	160.985 ± 125.103	327.485 ± 124.886
DDIM [20]	10/20/50	6.972 ± 0.046	6.335 ± 0.162	1.667 ± 0.019	0.3632 ± 0.0454	1.840 ± 0.048	160.719 ± 123.992	326.630 ± 124.489
No noise / no denoising	0	19.525	8.978	3.983	0.0000	2.956	1.692	172.554
UniDBO one-step	1	5.931	3.807	1.233	0.0245	1.334	7.345	179.988

D.3.2 Progressive High-Noise Curriculum and Closed-Loop Drift Analysis

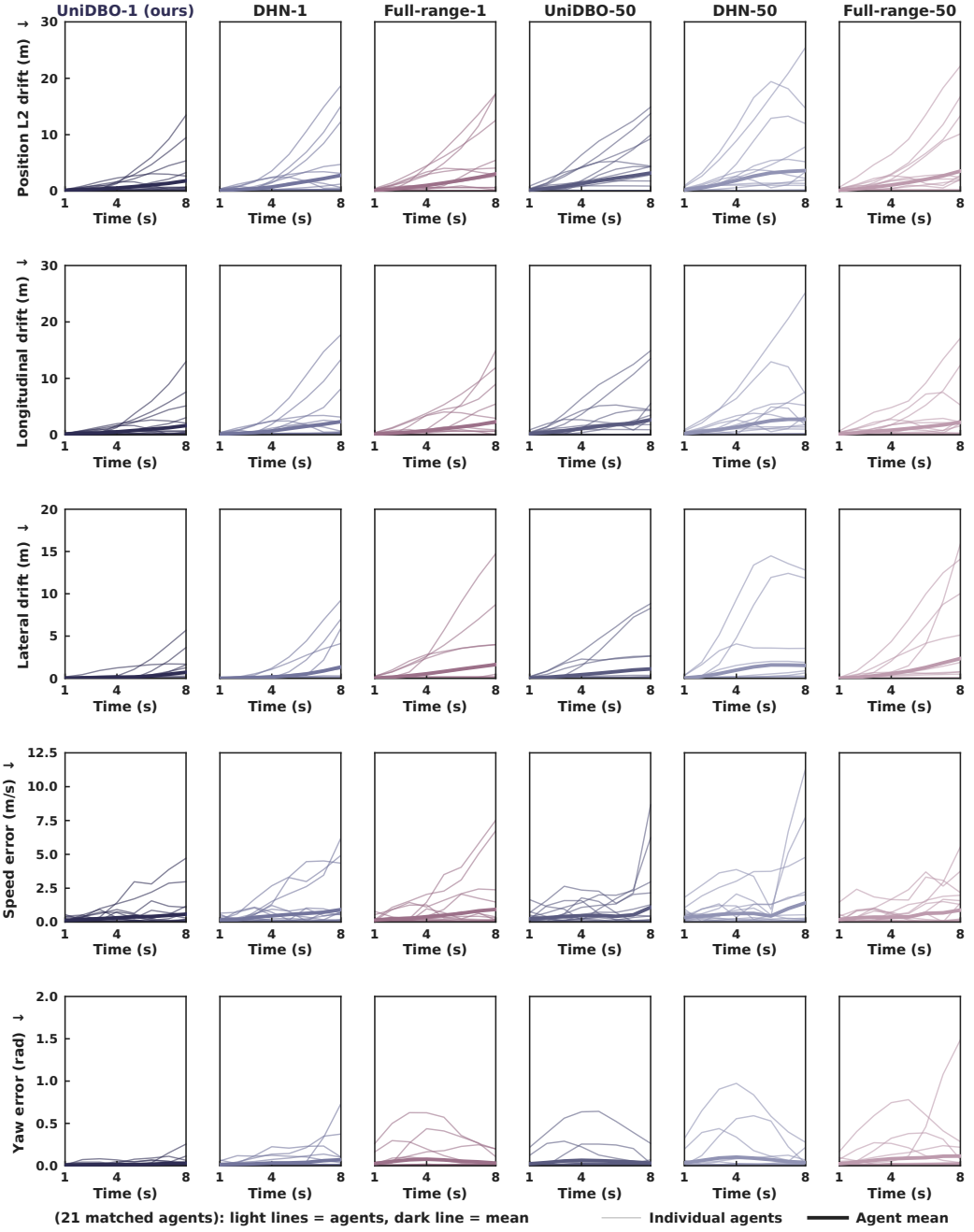

Figure 12: Per-agent drift for 21 matched agents in Case 1 over 1–8 s. Rows show drift measures and columns show six training and sampling configurations; light and dark curves show individual agents and their mean, respectively.

Table 14: One-step closed-loop drift of Full-range noise, DHN, and UniDBO on WOMD/Waymax. Values are scene-macro means over aligned 1–8 s snapshots. Lower is better; best and second-best values are **bolded** and underlined, respectively.

Configuration	L1 (m)↓	L2 (m)↓	Long. (m)↓	Lat. (m)↓	Speed (m/s)↓	Yaw (rad)↓
Full-range noise	4.6612	3.9750	3.6881	0.7705	1.2382	0.0904
DHN	3.0662	2.5856	2.3755	0.5543	0.8475	0.0751
UniDBO	**3.0449**	**2.5638**	**2.3644**	**0.5412**	**0.8426**	**0.0713**

Table 14 shows that DHN lowers the means of all six drift measures relative to Full-range noise, with further reductions from UniDBO; Figure 12 illustrates UniDBO's low-drift behavior in a per-agent case study. These results are consistent with the high-noise recovery-risk and finite-horizon drift bounds in Theorems 2 and 3, supporting direct high-noise recovery as a training proxy for closed-loop robustness.

D.3.3 Joint Dual-Noise Training

Complementary branch supervision. DHN focuses on direct high-noise recovery, while CS provides continuous-scale supervision through the shared scene encoder. Compared with DHN alone, UniDBO improves all five closed-loop quality metrics at NFE$=1$, reducing Collision and Kinematic Infeasibility by 26.6% and 52.9%, respectively (Table 4); auxiliary open-loop recovery also improves in the same table. Closed-loop inference activates only DHN, so joint training adds no denoiser evaluations. UniDBO also retains DHN's low-drift behavior: Figure 5 shows time-aligned differences across all six drift measures, while Table 14 reports further 1–8 s mean reductions of 2.18, 1.11, and 1.31 cm in position-L2, longitudinal, and lateral drift, respectively, and lower means across all six drift measures. These results are consistent with the analyses in Appendix C: high-noise direct recovery serves as a tractable proxy for closed-loop robustness, CS supervision can facilitate DHN optimization under a sufficient gradient-alignment condition, and control-recovery error can propagate into finite-horizon state drift.

Table 15: Branch and noise-sampling ablation on 10% WOMD training data. Single-head CS and DHN variants use full-range random sampling, a progressive high-noise curriculum, or a mixture of both within one recovery head; UniDBO jointly trains separate CS and DHN heads through a shared scene encoder. ADE/FDE measure auxiliary open-loop recovery; C, O, W, K, and L denote closed-loop Collision, Off-road, Wrong-way, Kinematic Infeasibility, and Log Divergence. Lower is better; best and second-best values are **bolded** and underlined, respectively.

Method	Noise schedule	minADE↓	minFDE↓	C (%)↓	O (%)↓	W (%)↓	K (%)↓	L↓
CS random	Full-range random	1.242	3.224	14.715	7.423	3.242	**0.0000**	2.768
CS high-noise	Progressive high-noise	1.718	4.543	13.650	**4.754**	2.384	0.0001	2.636
CS mixed	Full-range + high-noise	1.293	3.516	13.955	5.391	2.686	**0.0000**	2.228
DHN random	Full-range random	2.281	6.462	12.517	5.186	2.132	0.0032	2.045
DHN high-noise	Progressive high-noise	3.281	9.416	8.189	5.146	2.365	0.0515	1.549
DHN mixed	Full-range + high-noise	2.128	5.801	8.599	5.394	1.791	0.0154	1.535
UniDBO	CS full-range + DHN high-noise	**1.122**	**2.787**	**8.156**	5.053	**1.665**	0.0454	**1.492**

Small-scale branch and noise-sampling analysis. Table 15 compares single-head CS and DHN variants under three noise-sampling strategies on 10% of the WOMD training data. Under matched strategies, CS provides more accurate auxiliary open-loop recovery, whereas DHN performs better on most closed-loop metrics. Relative to full-range random sampling, the progressive high-noise curriculum reduces Collision and Log Divergence for both branches but increases open-loop recovery errors. This trade-off indicates that emphasizing severe corruption benefits certain closed-loop outcomes while weakening near-distribution fitting within a single branch.

Mixing full-range and high-noise samples partly mitigates this trade-off. For DHN, mixed sampling improves open-loop recovery and Wrong-way over the high-noise curriculum, but slightly increases Collision. Because both noise regimes still train one recovery head, broader noise coverage alone does not provide branch-specific specialization. UniDBO instead shares the scene encoder while assigning the two regimes to separate recovery heads. Among the evaluated variants, it achieves the

lowest auxiliary open-loop errors, Collision, Wrong-way, and Log Divergence, supporting a better overall open-/closed-loop balance than the tested single-head schedules.

D.3.4 Comparisons with Alternative Training Designs

To assess alternative training designs, we compare UniDBO with Continuous DHN, Diffusion Forcing, and Self-Forcing on the same 10% WOMD training subset.

Continuous DHN versus UniDBO. Table 16 shows that UniDBO outperforms Continuous DHN on four of the five closed-loop metrics. At the terminal high-noise condition $t^\star = 49$, little clean-control signal remains, so recovery relies more strongly on noise conditioning and scene representation. Continuous DHN uses interpolated continuous noise conditions, whereas UniDBO's DHN repeatedly trains at a finite set of discrete high-noise timesteps that includes $t^\star$, providing repeated supervision at a fixed signal-to-noise ratio for each timestep. This may stabilize recovery under high-noise conditions. Beyond noise conditioning, UniDBO uses CS supervision across continuous noise scales to enrich the shared scene representation for DHN's high-noise recovery.

Table 16: Closed-loop performance of UniDBO and alternative training designs on WOMD/Waymax under NFE$=1$. Best values are **bolded**, and second-best values are underlined.

Method	Collision (%)↓	Off-road (%)↓	Wrong-way (%)↓	Kinematic Infeasibility (%)↓	Log Divergence↓
Continuous DHN	13.851	7.721	3.347	**0.0000**	2.177
Diffusion Forcing [61]	<u>8.592</u>	<u>5.264</u>	<u>1.810</u>	<u>0.0015</u>	<u>1.658</u>
Self-Forcing [62]	11.127	5.350	2.644	0.0143	1.895
UniDBO	**8.156**	**5.053**	**1.665**	0.0454	**1.492**

Comparison with Sequential Training Methods. Table 16 compares UniDBO with task-adapted Diffusion Forcing [61] and Self-Forcing [62]. These methods differ in how they model the full-horizon multi-agent control sequence.

Analysis of the Alternative Designs. Diffusion Forcing samples independent timesteps t_j for control blocks and uses causal multi-agent attention, so block j cannot access later blocks $j' > j$. Heterogeneous noise conditions and restricted cross-block interaction can limit global coordination. Self-Forcing instead feeds decoded predictions into subsequent histories, but its stop-gradient operation implies

$$\left.\frac{\partial \mathcal{L}_{>j}}{\partial \hat{\mathbf{x}}^{(j)}}\right|_{\text{context path}} = 0, \tag{59}$$

where $\mathcal{L}_{>j}$ collects later-block losses. Early predictions therefore affect subsequent contexts, but later losses cannot correct them through this path. UniDBO instead jointly generates full-horizon controls for all agents in one recovery call, avoiding repeated generated-history updates within a single prediction and supporting coordination across agents and time.

D.4 Weight Sensitivity

Loss-weight sensitivity. With $w_a = 0.05$ fixed, we vary the branch weights without speed postprocessing (Table 17). Equal branch weights of 1.25 provide the best balance between open-loop trajectory fitting and closed-loop robustness. At a fixed total weight of 2.5, increasing the CS weight marginally improves ADE but degrades closed-loop performance, whereas increasing the DHN weight provides no overall benefit. We therefore use $w_c = w_d = 1.25$.

Speed postprocessing. Prior traffic-simulation work has used simulation-time constraints, trajectory smoothing, and speed adaptation [6, 8, 63]. At each replan, UniDBO checks for predicted conflicts over the next second and processes at most three conflicting agent pairs. It seeks the smallest path-preserving deceleration for the involved controlled agents to resolve the earliest conflict without introducing new ones; if no feasible adjustment is found, it retains the original prediction. The UniDBO results in Tables 1 and 2 use this postprocessing. The baselines use analogous speed postprocessing. Table 18 compares UniDBO's closed-loop quality and time-aligned drift with and without this postprocessing.

Table 17: CS/DHN loss-weight sensitivity. Panel (a) varies equal main weights; panel (b) varies their ratio at fixed total main weight 2.5. ADE and FDE are measured in meters. Lower is better; best and second-best values are **bolded** and underlined, respectively.

(a) Equal CS/DHN main weights.

Weight	minADE	minFDE	Collision (%)	Off-road (%)	Wrong-way (%)	Kin. Inf. (%)	Log Div.
0.75	0.790	1.929	6.585	**3.471**	1.313	**0.0156**	1.337
1.00	0.780	1.918	6.428	3.902	1.389	0.0211	1.319
1.25	**0.778**	**1.907**	**5.989**	3.828	**1.167**	0.0239	**1.304**
1.50	0.793	1.938	6.045	3.835	1.254	0.0253	1.332

(b) CS/DHN relative weights.

w_c/w_d	minADE	minFDE	Collision (%)	Off-road (%)	Wrong-way (%)	Kin. Inf. (%)	Log Div.
0.75	0.799	1.949	6.030	3.850	1.206	**0.0208**	1.331
1.00	0.778	**1.907**	**5.989**	**3.828**	**1.167**	0.0239	**1.304**
1.25	**0.776**	1.911	6.674	4.068	1.451	0.0247	1.344

Table 18: Effect of evaluation-time speed postprocessing. Panel (a) reports closed-loop quality, and panel (b) reports scene-macro drift over aligned 1–8 s snapshots. Postprocessed results are the final three-seed means; lower is better.

(a) Closed-loop quality.

Postprocessing	Collision (%)↓	Off-road (%)↓	Wrong-way (%)↓	Kinematic Infeasibility (%)↓	Log Divergence↓
None	5.931	3.807	1.233	0.0245	1.334
Speed postprocessing	4.773	3.721	1.244	0.0228	1.322

(b) Aligned closed-loop drift.

Postprocessing	Position L1 (m)↓	Position L2 (m)↓	Long. (m)↓	Lat. (m)↓	Speed (m/s)↓	Yaw (rad)↓
None	3.0834	2.5935	2.3916	0.5512	0.8458	0.0727
Speed postprocessing	3.0449	2.5638	2.3644	0.5412	0.8426	0.0713

D.5 Stochastic Diversity and Multimodality

To assess whether stochastic queries preserve closed-loop performance while generating diverse futures, we perform eight closed-loop realizations on the same ordered WOMD/Waymax scenario set using independently sampled Gaussian-query sequences.

Table 19: Closed-loop quality (a) and aligned drift (b) across eight stochastic UniDBO query realizations. Values are the eight-query mean $\pm$ sample standard deviation; lower is better.

(a) Closed-loop quality.

Statistic	Collision (%)↓	Off-road (%)↓	Wrong-way (%)↓	Kinematic Infeasibility (%)↓	Log Divergence↓
Eight-query mean $\pm$ std.	4.811 ± 0.039	3.713 ± 0.043	1.247 ± 0.018	0.0233 ± 0.0015	1.318 ± 0.008

(b) Aligned closed-loop drift.

Statistic	Position L1 (m)↓	Position L2 (m)↓	Long. (m)↓	Lat. (m)↓	Speed (m/s)↓	Yaw (rad)↓
Eight-query mean $\pm$ std.	3.028 ± 0.025	2.550 ± 0.020	2.350 ± 0.020	0.541 ± 0.004	0.839 ± 0.004	0.0708 ± 0.0007

Quantitative and qualitative diversity analysis. Table 19 shows small standard deviations in closed-loop quality and aligned drift across queries, while Figure 13 visualizes their temporal drift deviations. Despite this aggregate stability, the generated trajectories remain distinct: mean pairwise positional separation is 0.250 m and reaches 0.685 m at 8 s, and 38.9% of matched agent endpoints have a maximum pairwise separation exceeding 1 m. Figure 14 further shows variations in turning geometry, interaction evolution, and endpoint locations across six scenes, with trajectories remaining road-consistent and coordinated. Together, these results demonstrate stable closed-loop performance and diverse joint multi-agent futures.

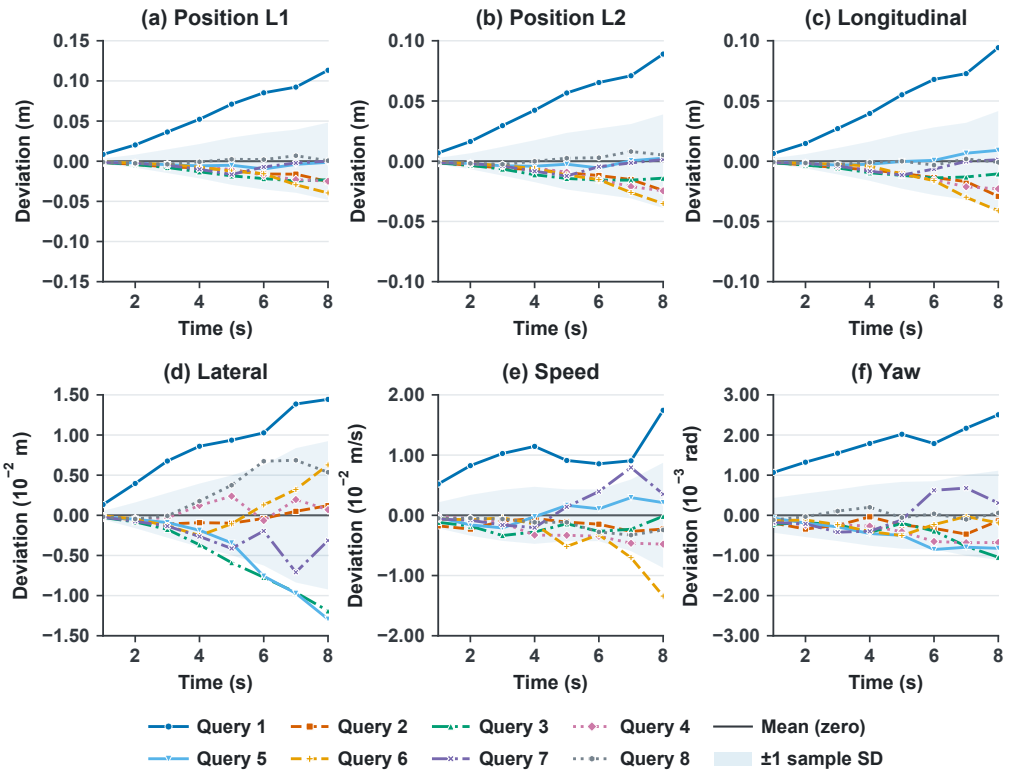

Figure 13: Cross-query deviations in aligned 1–8 s closed-loop drift. Each curve shows one query's deviation from the eight-query mean at the same timestamp; the black zero line marks the mean. Colors, line styles, and markers distinguish queries, and the shaded band shows ± 1 sample standard deviation. Absolute drift is reported in Table 19.

E Broader Impact

This work targets simulation-side autonomous-driving scenario generation for training, testing, and safety validation. Interactive traffic simulation supports scalable closed-loop evaluation of AV systems [6, 8]. Given fixed maps, agent sets, and initial states, UniDBO generates plausible, interactive background traffic for repeatable evaluation of external ego planners. The OnSite experiment assesses this use case on 800 unseen scenarios under a system-level evaluation protocol [2, 52]. UniDBO does not explicitly target long-tail safety-critical events; controllable rare-event generation remains future work.

Potential risks remain. First, generated scenarios may still deviate from real-world traffic distributions; if simulation results are used in isolation, this can lead to overestimation of autonomous-driving robustness. Second, benchmark and data biases (e.g., geography, traffic style, and interaction patterns) may limit generalization in underrepresented settings. Third, scenario-generation methods could be misused to construct adversarial or misleading traffic cases beyond intended safety-evaluation contexts.

To mitigate these risks, we recommend using generated scenarios as a complement rather than a replacement for real-world and rule-based evaluation; reporting multi-dimensional safety outcomes together with random-seed uncertainty; and maintaining transparent protocol documentation (including data splits, runtime/timing settings, and evaluation assumptions). In addition, the results in this paper should not be interpreted as sufficient evidence for direct deployment of autonomous-driving policies; our contribution is focused on simulation-side scenario-generation methodology and evaluation capability.

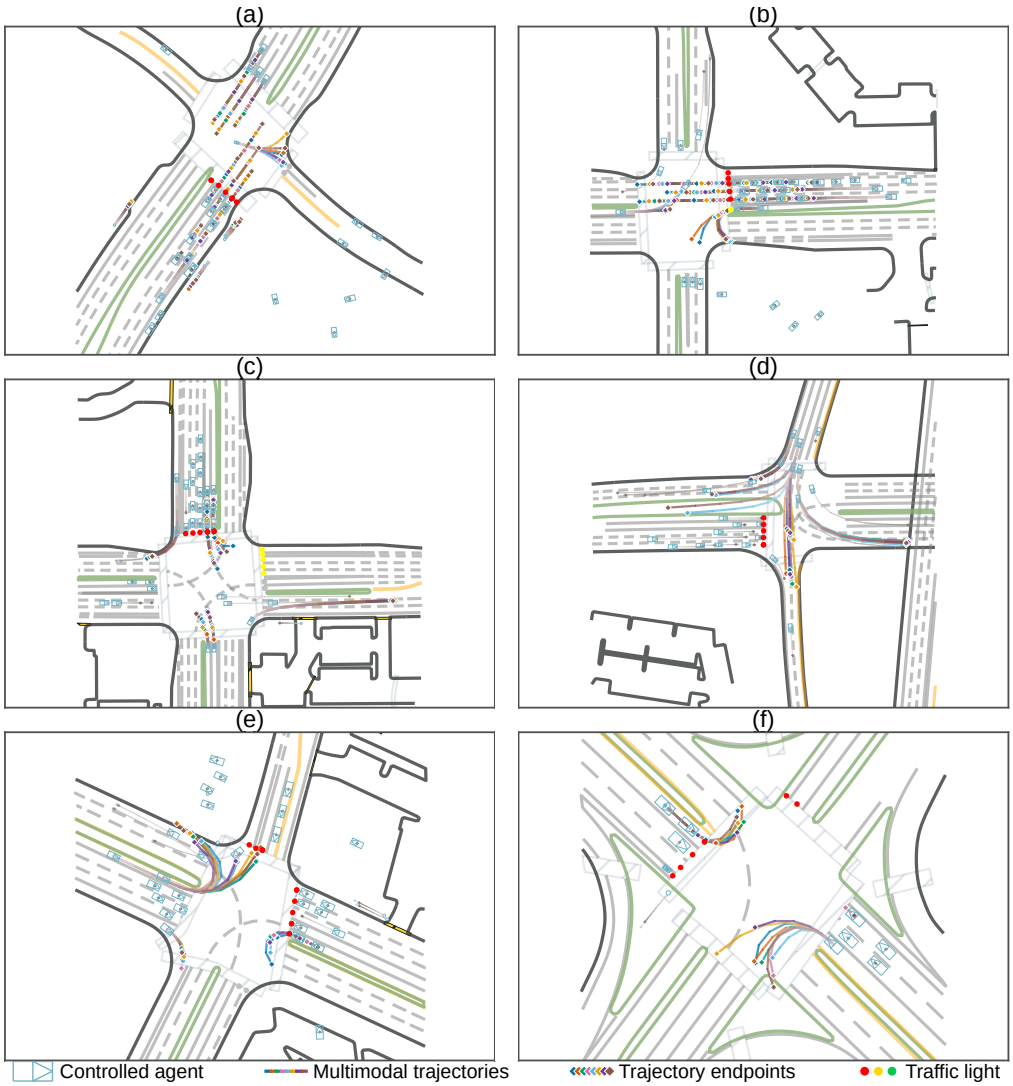

Figure 14: Same-scene stochastic multi-agent futures on WOMD/Waymax. Panels (a)–(f) each overlay eight UniDBO closed-loop realizations from one initial scene. Cyan outlines mark controlled agents; trajectory colors identify joint query realizations across agents; diamonds and colored circles denote endpoints and traffic lights, respectively. The map, observed history, initial state, model parameters, and simulator settings remain fixed within each panel.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are supported by the method, ablation studies, and experiments on WOMD/Waymax, INTERACTION, and OnSite.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations paragraph following the conclusion discusses planar dynamics, fixed evaluation protocols, residual collision risk, and future directions. Appendices B.2 and C explain the evaluation and theoretical limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Appendix C states the assumptions and derives the direct-versus-iterative recovery decomposition, curriculum-to-query risk bound, local gradient-transfer conditions, and finite-horizon control-to-state error bound.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 3 and Appendix A describe the model and training. Section 4 and Appendix B specify the data, protocols, metrics, and baselines.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The abstract links the code repository, and Appendix B.1 links the official data sources. The repository supplies environment, conversion, training, and evaluation instructions; raw datasets are not redistributed.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3 and Appendix A specify the model, losses, masks, training settings, weight selection, and algorithms. Section 4 summarizes the evaluation; Appendix B details datasets, metrics, runtime accounting, and baselines.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix D.1.1 reports means and sample standard deviations of closed-loop quality and drift across three evaluation seeds. Appendix D.5 reports means and sample standard deviations across eight independent Gaussian-query realizations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.2 reports training hardware, wall-clock time, and the joint model's parameter count. Appendix B.2 defines runtime accounting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses established public/benchmark datasets and simulator-based evaluation, and reports limitations and safety-relevant trade-offs without overstating deployment guarantees.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix E discusses potential testing benefits, over-reliance on simulation, data and benchmark biases, misuse, and mitigation recommendations.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The submission does not release a high-risk foundation model, scraped dataset, or deployable autonomous-driving controller.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper documents the usage terms and provenance of WOMD/Waymax, INTERACTION, OnSite, and the Apache-2.0-licensed VBD codebase in Appendix B.1, and cites the original sources.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new dataset, benchmark, or released model package.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing tasks or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: No human-subject study, intervention, or participant recruitment is conducted in this work.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as an important, original, or non-standard component of the proposed UniDBO method or evaluation pipeline.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.