BEYOND ONE PATH: PRESERVING BEHAVIORAL DIVERSITY IN FLOW-BASED END-TO-END DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-based driving policies can generate many trajectories while offering few distinct, useful alternatives. We study how pretraining, candidate generation, and reward fine-tuning shape this diversity at a fixed candidate budget. Our analysis shows that pretraining choices affect the best available candidate and the trajectory chosen by a learned scorer differently. We organize generation and refinement around a shared vocabulary of initial noise vectors, optimized offline for decoded-trajectory coverage across training scenes. Reusing this vocabulary provides persistent candidate identities whose trajectories adapt to each scene. For post-training, we assign each scene to its highest-scoring pretrained candidate and update the generator using local stochastic rollouts from the corresponding noise vector. Reward improvements are compared across scenes assigned to the same identity, with KL regularization toward the pretrained policy. Our evaluated camera-based planner, using a generalized variance-preserving (GVP) schedule and local safety refinement, achieves 94.5 PDMS on NAVSIM v1, 43.4 HD-Score on HUGSIM, and 74.19 Driving Score with 62.86% success on NavSafe-∞. Code and weights will be open-sourced.

1 INTRODUCTION

End-to-end autonomous driving learns to plan vehicle motion from sensor observations, ego state, and navigation commands. Recent systems integrate perception and planning within a learned framework, enabling decisions that account for the surrounding scene (Hu et al., 2023; Jiang et al., 2023; Li et al., 2026b). Driving nevertheless admits multiple reasonable responses: a vehicle approaching an interaction may proceed, yield, or adjust its speed along the same route. Representing these alternatives allows a planner to evaluate different responses before choosing a trajectory for execution.

Diffusion and flow models provide a natural way to represent multimodal trajectory distributions conditioned on the same scene. Recent planners introduce structure into generation: DiffusionDrive uses trajectory anchors to guide denoising (Liao et al., 2025), GoalFlow conditions flow matching on predicted goals (Xing et al., 2025), and MeanFuser combines Gaussian mixture noise with one-step generation and adaptive trajectory reconstruction (Wang et al., 2026). Reinforcement-learning approaches further optimize candidate quality using driving rewards, including ReCogDrive's diffusion-policy refinement and DiffusionDriveV2's intra-anchor and inter-anchor GRPO (Li et al., 2026b; Zou et al., 2025). These advances motivate studying how useful alternatives evolve across both generative training and reward-based refinement.

Flow-based planning raises three challenges. **(1) Learning useful multimodality.** Prediction targets, probability paths, scene conditioning, and training duration shape how strongly initial noise influences trajectories. **(2) Better coverage with limited samples.** Independent noise draws can repeatedly produce similar candidates, leaving useful behaviors unexplored within the inference budget. **(3) Refining behaviors without losing diversity.** Maximizing reward across scenes and noises can draw distinct candidates toward the same high-reward response. Higher average candidate reward alone does not establish better coverage or final selection.

We present *Beyond One Path* (Figure 1) to learn, cover, and refine multimodal driving behaviors. Our pretraining framework combines velocity supervision, the generalized variance-preserving (GVP) schedule, compact scene conditioning, and controlled training duration. We optimize a shared noise

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

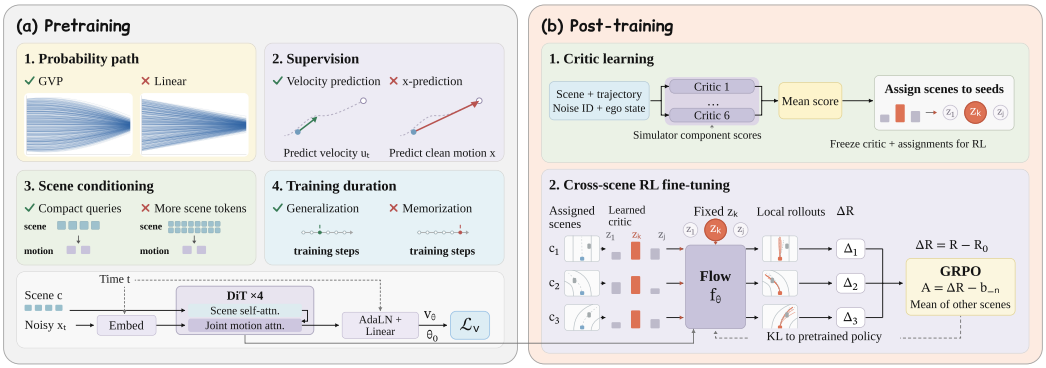

Figure 1: **Overview of Beyond One Path.** (a) Pretraining combines the GVP schedule, velocity supervision, compact scene queries, and training duration selected for candidate coverage. (b) A critic ensemble trained on simulator component scores assigns each scene to its highest-scoring seed in a shared noise vocabulary. Cross-scene GRPO then updates the generator with local stochastic rollouts, keeping the critic and assignments fixed. Advantages center improvements over the pretrained policy across scenes sharing a seed; KL regularization anchors updates to that policy.

vocabulary offline for mode coverage, diversity, and trajectory quality across training scenes. Each noise generates a scene-adapted trajectory, which a scorer ensemble evaluates. For cross-scene post-training, we assign each scene to its highest-scoring pretrained candidate and refine the generator using local stochastic rollouts from that seed. Reward improvements relative to the pretrained candidate are centered across scenes sharing an identity, with KL regularization toward the pretrained policy.

Our methodological contributions are threefold:

- We provide a flow pretraining framework for generating diverse, multimodal trajectories across driving scenarios, supported by a systematic study of supervision, probability paths, scene conditioning, and training duration.
- We formulate a distribution-level optimization problem that jointly promotes behavioral-mode coverage, generation diversity, and trajectory quality through a shared noise vocabulary optimized across training scenes.
- We propose cross-scene post-training to improve the safety of individual behavioral modes while preserving trajectory diversity, using fixed scene assignments, cross-scene reward-improvement advantages, and regularized local refinement.

We evaluate on NAVSIM v1 (Dauner et al., 2024) and two photorealistic closed-loop benchmarks, HUGSIM (Zhou et al., 2024) and NavSafe-∞ (Bao et al., 2026), with broad comparisons against recent end-to-end driving methods. Our camera-based planner with the GVP schedule and local safety refinement achieves 94.5 PDMS, 43.4 HD-Score, and 74.19 Driving Score with 62.86% success, respectively. Component ablations examine pretraining, candidate coverage, trajectory selection, and post-training; Appendix A provides the extended benchmark comparisons.

2 Related Work

Flow-based policy. Generative policies are increasingly used in autonomous driving (Liao et al., 2025; Zou et al., 2025; Zhang et al., 2025b; Wozniak et al., 2026). We adopt flow matching, which learns a vector field along a prescribed probability path (Lipman et al., 2023), and study its planning behavior under practical parameterizations and sampling budgets. Recent methods adapt generative policies with reward optimization, including Flow-GRPO (Liu et al., 2025), GoalFlow (Xing et al., 2025), Epona (Zhang et al., 2025b), and DiffusionDriveV2 (Zou et al., 2025). Unlike these methods, we assign distinct scenario distributions to fixed noise identities and optimize them with an cross-scene improvement baseline.

Reinforcement learning fine-tuning. Imitation learning provides a strong base policy but can still suffer from mode collapse and poor adaptation to rare closed-loop cases. RL fine-tuning addresses these limitations. DSRL optimizes a noise-space policy while freezing the pretrained generator (Wagenmaker et al., 2025), whereas our post-training updates the generator itself. ReCogDrive, DIVER, and AutoVLA use PDMS-based GRPO (Li et al., 2026b; Song et al., 2026; Zhou et al., 2025); TrajHF introduces human trajectory guidance (Li et al., 2025); and MTDrive uses multi-turn GRPO for sparse rewards. We compare continuous noise adaptation, fixed-vocabulary selection, and trajectory-aware scoring under matched decode budgets. NAVSIM provides logged-state evaluation (Dauner et al., 2024), while HUGSIM and NavSafe-∞ support photorealistic closed-loop evaluation (Zhou et al., 2024; Bao et al., 2026).

3 PRELIMINARIES

Problem formulation. An end-to-end driving policy conditions on $c = (i, s, l)$, comprising camera observations i, recent ego-motion history s, and a navigation command l, to sample future ego trajectories:

$$x \sim \pi_\theta(\cdot \mid c). \tag{1}$$

Here, $x \in \mathbb{R}^d$ represents normalized motion over H steps, and $\tau = D(x)$ is the decoded trajectory in the current ego frame.

Flow-based imitation learning. Given scene–motion pairs $(c, x) \sim \mathcal{D}_{\mathrm{IL}}$, flow matching (Lipman et al., 2023) constructs a noise-to-data interpolation

$$x_t = \alpha_t x + \sigma_t z, \qquad z \sim \mathcal{N}(0, I_d), \qquad t \in [0, 1], \tag{2}$$

where $(\alpha_0, \sigma_0) = (0, 1)$ and $(\alpha_1, \sigma_1) = (1, 0)$. The conditional path $p_t(\cdot \mid x) = \mathcal{N}(\alpha_t x, \sigma_t^2 I_d)$ induces the scene-conditioned mixture $p_t(y \mid c) = \int p_t(y \mid x) p_{\mathrm{data}}(x \mid c)\, \mathrm{d}x$, connecting Gaussian noise to the demonstrated motion distribution. A learned velocity field v_θ approximates this transport; inference numerically integrates

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(x_t, t, c), \qquad x_0 = z, \qquad x_1 \sim \pi_\theta(\cdot \mid c). \tag{3}$$

For a fixed scene and solver, the initial noise determines the decoded trajectory $D(x_1)$.

KL-constrained post-training on expert-visited states. Costly high-fidelity rendering limits the scalability of closed-loop fine-tuning for end-to-end driving policies (Karkus et al., 2025). Recent methods optimize trajectory rewards on logged expert contexts, seeking gains that transfer to closed-loop execution (Li et al., 2026b; Zhou et al., 2025; Xu et al., 2026). Let p_{expert} denote the distribution of these contexts. For $c \sim p_{\mathrm{expert}}$ and $x \sim \pi_\theta(\cdot \mid c)$, a constrained formulation is

$$\max_\theta \ \mathbb{E}_{c,x} R(c, D(x)) \quad \text{subject to} \quad \mathbb{E}_c D_{\mathrm{KL}}(\pi_\theta(\cdot \mid c) \,\|\, \pi_{\theta_0}(\cdot \mid c)) \le \kappa. \tag{4}$$

Here, $R(c, \tau)$ evaluates a decoded trajectory, θ_0 denotes the frozen pretrained parameters, and κ bounds the average policy divergence. Our post-training uses a KL penalty on stochastic denoising transitions to regularize toward this reference.

4 ANALYSIS OF DIVERSITY-PRESERVING FLOW PRE-TRAINING

A diverse flow policy maps different initial noises to distinct scene-compatible behavioral modes, such as yielding or proceeding. For a fixed scene c, *conditional multimodality* assigns non-negligible mass to multiple modes, whereas *mode collapse* concentrates mass on fewer modes (Wang et al., 2026). We study how prediction target, probability path, scene conditioning, and training duration affect this diversity, varying each in Figure 2 while fixing the scene, initial noises, and inference settings.

Policy architecture. A common DiT-based backbone (Peebles & Xie, 2023) processes scene context and noisy motion in two interacting streams. The policy uses scene embeddings $c_{\mathrm{emb}} \in \mathbb{R}^{N_{\mathrm{scene}} \times d_{\mathrm{emb}}}$ and noisy trajectory embeddings $x_{\mathrm{emb}} = \mathrm{Embed}_{\mathrm{traj}}(x_t, t) \in \mathbb{R}^{N_{\mathrm{traj}} \times d_{\mathrm{emb}}}$, where N_{scene} and N_{traj} denote the respective token counts and d_{emb} is the shared embedding dimension. A sinusoidal embedding of flow time t conditions the trajectory-token projection and modulates the

output normalization through adaptive layer normalization (AdaLN). Following (Li et al., 2026b), the scene stream uses self-attention, while the action stream, which includes x_{emb}, attends to the concatenated scene and action tokens:

$$\text{Attn}_c = \text{MHA}(\mathbf{Q}_c, \mathbf{K}_c, \mathbf{V}_c), \qquad \text{Attn}_a = \text{MHA}(\mathbf{Q}_a, [\mathbf{K}_c; \mathbf{K}_a], [\mathbf{V}_c; \mathbf{V}_a]), \tag{5}$$

where MHA denotes multi-head attention, $(\mathbf{Q}_j, \mathbf{K}_j, \mathbf{V}_j)$ are the query, key, and value matrices for the scene ($j = c$) or action ($j = a$) stream, and $[\,;\,]$ denotes concatenation along the token dimension.

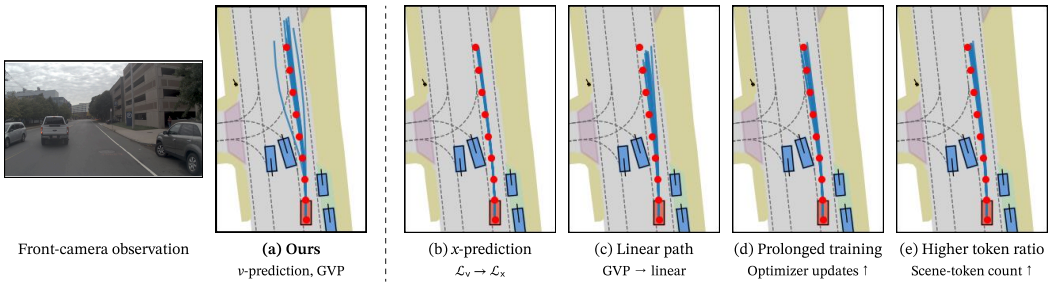

Figure 2: Pretraining choices shape trajectory diversity. The reference policy (a) is compared with four single changes: (b) clean-motion regression, (c) a linear probability path, (d) more optimizer updates, and (e) more scene tokens.

Velocity Supervision. Velocity supervision trains the quantity used to update motion during ODE integration. Velocity (v-) prediction estimates the instantaneous velocity of the interpolation in Equation (2), $u_t = \dot{\alpha}_t x + \dot{\sigma}_t z$, through conditional flow matching (Lipman et al., 2023):

$$\mathcal{L}_v(\theta) = \mathbb{E}\big[\|v_\theta(x_t, t, c) - u_t\|_2^2\big]. \tag{6}$$

The expectation is over $(c, x) \sim \mathcal{D}_{\text{IL}}$, $z \sim \mathcal{N}(0, I_d)$, and $t \sim \mathcal{U}[0, 1]$. Using the same training samples, we compare velocity supervision with x-prediction (Figure 2(b)), which directly estimates the clean trajectory $\hat{x}_\theta(x_t, t, c)$:

$$\mathcal{L}_x(\theta) = \mathbb{E}\big[\|\hat{x}_\theta(x_t, t, c) - x\|_2^2\big]. \tag{7}$$

At inference, x-prediction uses the same ODE sampler by converting the predicted clean trajectory into the corresponding velocity field

$$v_\theta^{(x)}(x_t, t, c) = \frac{\dot{\sigma}_t}{\sigma_t} x_t + \left(\dot{\alpha}_t - \frac{\alpha_t \dot{\sigma}_t}{\sigma_t}\right) \hat{x}_\theta(x_t, t, c), \qquad t < 1. \tag{8}$$

Despite sharing the same training data and ODE sampler, the two objectives produce markedly different candidate distributions. Velocity supervision preserves more distinct trajectories across initial noises, whereas x-prediction concentrates candidates around dominant behaviors (Figure 2(b)). This motivates our use of velocity prediction to preserve behavioral diversity.

Probability-Path Geometry. The probability path determines the intermediate mixtures of motion and noise seen during training. Our proposed planner uses the GVP schedule (Ma et al., 2024), with $(\alpha_t, \sigma_t) = (\sin(\pi t/2), \cos(\pi t/2))$, and we compare it with a linear-schedule ablation, $(\alpha_t, \sigma_t) = (t, 1 - t)$, whose velocity target is $u_t = x - z$ (Figure 2(c)). Although both paths connect the same noise and data endpoints, they induce different intermediate geometries: GVP preserves $\alpha_t^2 + \sigma_t^2 = 1$ throughout the path, whereas the linear path contracts this quantity to $1/2$ at $t = 1/2$. Empirically, the linear path produces more concentrated candidates, while GVP better preserves diversity across initial noises.

Training Duration. Prolonged training improves individual predictions but can reduce diversity across candidates. As shown in Figure 2(d), trajectories generated from different initial noises become increasingly concentrated with training. Accordingly, best-of-64 PDMS peaks at epoch 19 and then declines by 2.0 points despite continued improvement in average prediction quality (Figure 3). We therefore select training duration based on best-of-K performance at the deployment budget, balancing prediction quality against candidate coverage. This behavior is consistent with studies linking prolonged optimization to memorization in generative models (Bonnaire et al., 2025; Rao & Moyer, 2026), where increasingly close fits to observed data can suppress alternative outputs.

Compact Scene Conditioning. Many recent driving policies condition a small set of noisy trajectory tokens on rich pretrained visual or vision-language representations through cross-attention (Li et al., 2026b; Zhao et al., 2026). While these representations provide strong scene semantics, their high capacity can also expose the policy to scene-specific details that are weakly related to the underlying driving behavior. We hypothesize that this imbalance encourages reliance on contextual cues and fitting of recorded responses rather than learning a reusable behavioral prior. We therefore summarize visual features with a compact set of learned scene queries (Kirby et al., 2026), reducing the scene-to-trajectory token ratio $\rho = N_{\text{scene}}/N_{\text{traj}}$. Increasing this ratio produces more concentrated trajectories in Figure 2(e), motivating compact conditioning to preserve the influence of noise-dependent trajectory features.

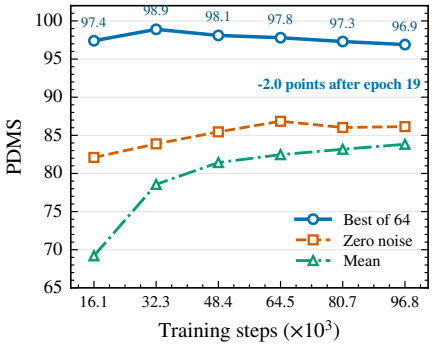

Figure 3: **Training dynamics.** Evolution of best-of-64, zero-noise, and mean-candidate PDMS over training steps.

At inference, we integrate v_θ in Equation (3). Let $f_\theta(z, c) \in \mathbb{R}^d$ denote the terminal normalized motion returned by the fixed ODE solver from $x_0 = z$; its decoded trajectory $\tau^\theta(z, c) = D(f_\theta(z, c))$ supplies the output for noise-vocabulary construction. The next section uses this mapping to choose a fixed set of noises that exposes useful alternatives within the candidate budget.

5 Beyond One Path

Building on the training analysis in Section 4, we organize candidate generation and refinement around a shared noise vocabulary (Figure 1). The vocabulary exposes distinct behaviors at a fixed decode budget, and a separate critic ensemble selects a trajectory for execution. Its persistent noise identities also support mode-specialized post-training on suitable scenarios.

5.1 Optimizing a Shared Noise Vocabulary

With a fixed budget of K candidates, independently sampled noises can decode to similar trajectories, leaving alternative behaviors unexplored. We therefore optimize the initial noise inputs for broad coverage in decoded trajectory space. Context-conditioned diversity samplers generate a latent set for each input context (Yuan & Kitani, 2020). Our goal is to identify *one shared set of initial noises optimized across the data distribution*, while preserving the diversity of the decoded behaviors.

Given scenes $\mathcal{C} = \{c_n\}_{n=1}^N$ drawn from the training distribution, let $Z \in \mathbb{R}^{K\times d}$ have rows $z_k^\top$, with $z_k \in \mathbb{R}^d$. Each vector generates $\tau_{nk} = \tau^{\theta_0}(z_k, c_n)$ in scene c_n. Keeping the pretrained parameters θ_0 and numerical sampler fixed, we pose the general optimization problem

$$Z^\star \in \arg\max_{Z\in\mathcal{Z}_K} \ \mathrm{Div}_{\mathcal{C}}(Z) - \frac{\lambda_C}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K} C(c_n, \tau_{nk}) - \frac{\lambda_Z}{K}\|Z - Z^{(0)}\|_F^2. \tag{9}$$

Here $\mathrm{Div}_{\mathcal{C}}(Z)$ measures decoded-trajectory diversity or coverage across $\mathcal{C}$, and C penalizes low-quality or invalid trajectories. The weights $\lambda_C, \lambda_Z \geq 0$ control quality and anchoring to the initial bank $Z^{(0)}$; $\mathcal{Z}_K$ allows continuous noise vectors or selection from a finite pool.

With differentiable terms, Backpropagation Through Time (BPTT) can optimize continuous seeds through denoising, but repeated forward and backward passes become costly across many scenarios. Discrete combinatorial optimization uses trajectories and costs cached once per seed–scene pair under the frozen deterministic flow; subsequent selection requires no further flow evaluations. Appendices B.1 and B.2 give pseudocode for BPTT and greedy quality-aware log-determinant selection, respectively.

We construct the shared bank offline using the discrete method in Appendix B.2 and reuse it unchanged at inference, generating K candidates with K flow decodes per scene. The persistent noise identities support mode-specialized post-training (Section 5.2).

5.2 Mode-Specialized Post-Training

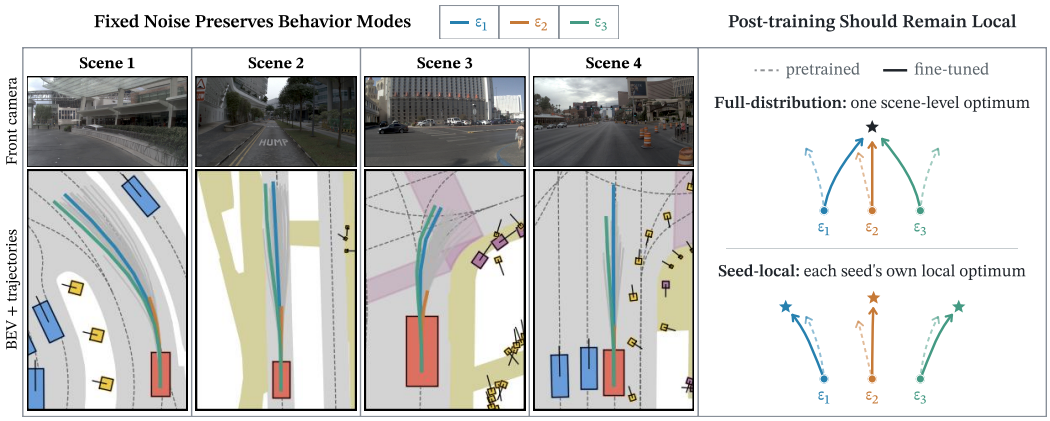

Figure 4: Fixed-noise behaviors and local post-training. Left: the same fixed noises generate scene-adapted trajectories in four scenarios; colors identify noise vectors. Right (c): full-distribution optimization can concentrate distinct decoded behaviors, while seed-local optimization refines each seed's trajectories on suitable scenes under KL regularization.

Recent reinforcement-learning methods refine diffusion and flow policies using rewards averaged over training contexts and samples initialized from a Gaussian prior (Li et al., 2026b; Liu et al., 2025). For our deterministic flow sampler, the reward term in Equation (4) takes the explicit form

$$\max_{\theta} J_{\mathrm{all}}(\theta), \qquad J_{\mathrm{all}}(\theta) = \mathbb{E}_{c\sim p_{\mathrm{expert}}, z\sim\mathcal{N}(0,I_d)} R(c, \tau^{\theta}(z, c)). \tag{10}$$

Optimizing over the entire initial-noise distribution can pull distinct noise-conditioned trajectories toward the same high-reward behavior within each scene, concentrating the policy on fewer modes (Figure 4, right, top). KL regularization discourages departure from the pretrained policy, while the reward term does not explicitly encourage diversity among noise-conditioned trajectories.

Fixed initial noises exhibit reusable behavioral tendencies across scenarios (Figure 4, left), but their reward rankings depend on the scene.

Cross-scene RL objective. For N fine-tuning scenes $\mathcal{C}_{\mathrm{train}}$, let $\mathcal{C}_k$ contain those assigned to seed k. Let $\pi^{\mathrm{loc}}_{\theta,k}(\cdot \mid c)$ be the trajectory distribution induced by local stochastic denoising initialized at z_k (Liu et al., 2025). We maximize the empirical objective

$$J_{\mathrm{cross}}(\theta) = \frac{1}{N}\sum_{k=1}^{K}\sum_{c\in\mathcal{C}_k} \mathbb{E}_{\tau\sim\pi^{\mathrm{loc}}_{\theta,k}(\cdot|c)}[R(c,\tau)] - \beta_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}}(\theta). \tag{11}$$

Each training scene contributes equally. Here $\beta_{\mathrm{KL}} \geq 0$ and $\mathcal{L}_{\mathrm{KL}}$ averages current-to-pretrained denoising-transition KL at frozen rollout-policy states, with matched exploration variance and the same uniform scene weighting.

Critic-based selection. We form $\mathcal{C}_k$ by assigning each training scene to the seed whose pretrained trajectory receives the highest score from the learned critic $s_\phi(c, \tau, k)$. Keeping ϕ fixed, we use the same selection rule during fine-tuning and inference, breaking ties by seed index:

$$\begin{aligned} \hat{k}_\theta(c) &= \arg\max_{1\leq k\leq K} s_\phi(c, \tau^{\theta}(z_k, c), k), \\ \mathcal{C}_k &= \{c \in \mathcal{C}_{\mathrm{train}} : \hat{k}_{\theta_0}(c) = k\}. \end{aligned} \tag{12}$$

The vocabulary and assigned scene sets remain fixed during fine-tuning. At inference, the same critic ranks the post-trained candidates and selects $\tau^{\theta}(z_{\hat{k}_\theta(c)}, c)$ for execution.

Cross-scene GRPO. To optimize Equation (11), we select seed k with probability $|\mathcal{C}_k|/N$, then sample $B \geq 2$ scenes independently and uniformly from $\mathcal{C}_k$, with replacement. Independent local rollouts yield improvements $\Delta R = R(c, \tau) - R(c, \tau^{\theta_0}(z_k, c))$, which we center using the other sampled scenes' mean improvement. Detached advantages drive a clipped update with current-to-rollout-policy likelihood ratios on stochastic transitions, keeping the rollout policy and remaining denoising steps frozen.

6 EXPERIMENTS

We evaluate our planner on three benchmarks: NAVSIM v1, HUGSIM, and NavSafe-∞. Together, they assess trajectory quality, adaptation to perturbed states, and closed-loop driving. Expanded benchmark comparisons appear in Appendix A.

Experimental setup. Our standard configuration uses velocity prediction with the GVP schedule, generates $K = 64$ candidates with ten Euler steps, and selects a trajectory using the critic ensemble. The main benchmark tables report the pretrained GVP policy and its local safety-RL refinement separately. Table 7 compares the cross-scene GRPO variants from Section 5.2.

Table 1: **NAVSIM v1** on `navtest`. Recent methods and diffusion/flow planners; the full comparison is in Table 8.

Method	Venue	Input	NC	DAC	TTC	Comf.	EP	**PDMS**
Diffusion/flow planners								
DriveX-S (Shi et al., 2025)	ICCV '25	C	97.5	94.0	93.0	**100.0**	79.7	84.5
Epona (Zhang et al., 2025b)	ICCV '25	C	97.9	95.1	93.8	99.9	80.4	86.2
TrajHF (EM) (Li et al., 2025)	arXiv '25	C+L	96.6	96.6	92.1	**100.0**	84.5	87.6
PRIX (Wozniak et al., 2026)	RA-L '26	C	98.1	96.3	94.1	**100.0**	82.3	87.8
DiffusionDrive (Liao et al., 2025)	CVPR '25	C+L	98.2	96.2	94.7	**100.0**	82.2	88.1
DIVER (Song et al., 2026)	arXiv '25	C+L	98.5	96.5	94.9	**100.0**	82.6	88.3
MeanFuser (Wang et al., 2026)	CVPR '26	C	98.6	97.0	95.0	**100.0**	82.8	89.0
GoalFlow (V2-99) (Xing et al., 2025)	CVPR '25	C+L	98.4	98.3	94.6	**100.0**	85.0	90.3
ReCogDrive (Li et al., 2026b)	ICLR '26	C	97.9	97.3	94.9	**100.0**	87.3	90.8
Other methods								
SeerDrive (V2-99) (Zhang et al., 2025a)	NeurIPS '25	C+L	98.8	98.6	95.8	**100.0**	84.2	90.7
DriveVLA-W0 (Li et al., 2026a)	ICLR '26	C	98.7	99.1	95.3	99.3	83.3	90.2
ResAD (V2-99) (Zheng et al., 2026)	CVPR '26	C+L	98.9	97.8	94.9	**100.0**	87.0	90.6
DrivoR (navtrain) (Kirby et al., 2026)	CVPR '26	C	98.9	98.3	96.2	**100.0**	89.1	93.1
Ours (pretrained)	–	C	98.9	99.2	95.4	**100.0**	**92.1**	94.1
Ours (post-trained)	–	C	**99.2**	**99.4**	**96.7**	**100.0**	91.3	**94.5**

C: cameras; L: LiDAR. NC: no at-fault collisions; DAC: drivable-area compliance; TTC: time-to-collision score; Comf.: comfort; EP: ego progress. **Bold**: best; underline: second-best per metric column (ties included).

Table 2: **HUGSIM closed-loop evaluation on 345 scenarios.** Baselines are from DrivoR (Kirby et al., 2026).

	RC					HD-Score				
Method	E	M	H	X	Avg.	E	M	H	X	**Avg.**
VAD (Jiang et al., 2023)	51.3	31.1	25.3	26.5	31.4	36.3	9.5	8.0	11.5	13.4
LTF (Chitta et al., 2023)	67.8	35.1	26.2	40.5	38.9	58.9	18.0	9.8	25.9	23.7
UniAD (Hu et al., 2023)	78.4	60.5	33.6	17.8	45.9	64.9	45.8	20.6	6.6	32.7
DrivoR (Kirby et al., 2026)	80.9	50.5	33.8	**47.1**	49.8	73.3	34.6	18.8	**32.5**	35.7
Ours (pretrained)	**83.0**	59.8	**37.7**	35.9	51.8	**78.5**	52.9	**26.3**	23.8	42.8
Ours (post-trained)	81.9	**62.1**	37.4	36.8	**52.6**	77.4	**54.6**	25.5	25.0	**43.4**

E/M/H/X: Easy/Medium/Hard/Extreme. Avg. averages all scenarios. **Bold**: best; underline: second-best per metric column (ties included).

Table 3: **NavSafe-∞ closed-loop evaluation.** Three-seed means from Bao et al. (2026); the full comparison is in Table 9.

			Overall performance				Event-category success			
Method	Venue	Input	**DS**	SR	DE	Comf.	TC	VRUC	TV	TI
SimWAM (Zhao et al., 2026)	arXiv '26	C	57.53	37.50	160.54	97.19	39.55	35.00	43.25	33.33
SparseDriveV2 (Sun et al., 2026)	ECCV '26	C	65.26	48.78	167.99	93.41	53.55	**43.46**	48.32	41.67
DiffusionDrive (DD) (Liao et al., 2025)	CVPR '25	C	44.77	20.96	165.08	97.68	21.27	11.86	23.65	22.22
DD + SimScale (Tian et al., 2026)	CVPR '26	C	52.19	35.29	189.66	96.92	38.48	17.94	36.81	42.22
ReCogDrive-2B-IL (Li et al., 2026b)	ICLR '26	C	48.98	28.28	166.94	97.58	30.03	17.18	31.16	26.67
ReCogDrive-2B-RL (Li et al., 2026b)	ICLR '26	C	53.64	30.99	173.96	85.10	35.12	18.70	32.59	27.78
DrivoR (Kirby et al., 2026)	CVPR '26	C	64.74	50.69	188.02	87.92	58.61	31.38	52.53	43.33
Ours (pretrained)	–	C	71.65	59.64	201.96	97.07	67.49	39.27	60.73	46.67
Ours (post-trained)	–	C	**74.19**	**62.86**	**204.11**	**98.60**	**69.00**	42.50	**67.27**	**53.33**

DD: DiffusionDrive; C: cameras; NR: evaluation sensors not reported. All scores use a 0–100 scale (↑), except DE, which retains its native scale. **Bold**: best; underline: second-best per metric column (ties included).

6.1 Benchmarks

NAVSIM v1. NAVSIM v1 (Dauner et al., 2024) evaluates four-second trajectories against non-reactive traffic. The vocabulary is constructed on training scenes and reused unchanged on `navtest`. We report the Predictive Driver Model Score (PDMS) and its safety, comfort, and progress components (Tables 1 and 8).

HUGSIM. HUGSIM (Zhou et al., 2024) evaluates closed-loop driving in photorealistic reconstructed scenes with interactive traffic. We report Route Completion (RC) and HUGSIM Driving Score (HD-Score) across its Easy, Medium, Hard, and Extreme difficulty levels (Table 2). HD-Score combines route completion with collision, road-compliance, near-collision, and comfort measures.

NavSafe-∞. NavSafe-∞ (Bao et al., 2026) evaluates event-specific closed-loop safety over 280 scenarios spanning 28 event types. We report Driving Score (DS), Success Rate (SR), Driving Efficiency (DE), and Comfort, together with category-level ability scores for Traffic Crashes (TC), Vulnerable Road User Crashes (VRUC), Traffic Violations (TV), and Traffic Incidents (TI). Each ability score averages event-type success rates within its category (Table 3).

6.2 Main Results

NAVSIM v1. Our camera-based planner achieves 94.1 PDMS after pretraining and 94.5 after post-training (Table 1). The post-trained policy exceeds DrivoR trained on `navtrain` by 1.4 points, and DiffusionDrive, GoalFlow, and MeanFuser by 6.4, 4.2, and 5.5 points, respectively. It also achieves 99.2 NC, 99.4 DAC, and 96.7 TTC, with a comfort score of 100.0.

HUGSIM. On 345 closed-loop scenarios (Table 2), our post-trained planner achieves 52.6 route completion and 43.4 HD-Score, exceeding DrivoR by 2.8 and 7.7 points, respectively. The pre-trained planner already reaches 42.8 HD-Score, demonstrating transfer to interactive closed-loop driving before reinforcement-learning refinement. Both variants exceed the compared baselines in average route completion and HD-Score.

NavSafe-∞. Our post-trained planner achieves 74.19 Driving Score and 62.86% success (Table 3), the highest values among learned policies in the full comparison (Table 9). These exceed GTRS-D + SimScale by 0.65 Driving Score and 1.04 percentage points in success rate. Relative to the pretrained policy, the post-trained policy improves all eight reported metrics, including gains of 2.54 in Driving Score and 3.22 percentage points in success rate.

6.3 Ablation Studies

We conduct ablation studies on NAVSIM v1 with $K = 64$ candidates and report PDMS (↑). Oracle reports the per-scene best-of-K PDMS, mean averages candidate PDMS, and selected reports the trained scorer's choice. The DSRL comparison is in Appendix C.

Initial Noise Construction. Optimizing the bank for diverse, high-quality trajectories yields the highest oracle, mean, and selected PDMS with a frozen generator (Table 4). The 0.8-point oracle

Table 4: Noise construction with the same frozen generator.

Construction	Oracle	Mean	Selected
IID Gaussian	98.6	81.1	92.9
Antithetic Gaussian	98.7	81.0	93.1
BPTT	96.3	79.2	91.5
Ours	99.4	82.3	94.1

Table 5: One change from Section 4; fixed noises and sampler.

Variant	Oracle	Selected
Reference recipe (GVP)	98.6	92.9
w/ x-prediction	90.5	89.1
w/ Linear probability path	98.2	93.3
w/ More scene tokens	90.3	88.9
w/ Longer training	96.8	92.4

Table 6: Selected PDMS by critic design; fixed generator and noise bank.

	Number of critics			
Backbone	1	2	4	6
DINOv2-S	92.55	92.78	93.03	93.30
VGGT (GeoUP)	94.30	94.37	94.42	94.47

Table 7: Selected PDMS after GRPO fine-tuning with matched rollout budgets.

Fine-tuning	PDMS ↑
Full GRPO	90.84
Cross-scene GRPO (updated)	94.54
Cross-scene GRPO (fixed)	94.53

gain over IID suggests broader useful coverage; 1.2-point mean and selected gains show that this coverage accompanies stronger candidates and execution.

Pretraining Choices. The GVP reference achieves the highest oracle PDMS (Table 5). Figure 2 gives intuition: velocity prediction, GVP geometry, controlled training duration, and compact conditioning retain distinct noise-dependent trajectories. Altering each choice reduces oracle PDMS. The linear ablation's 0.4-point selected gain despite a 0.4-point oracle loss shows why selected quality alone cannot characterize candidate coverage.

Critic Design. Exploiting coverage requires a strong scorer: one GeoUP critic exceeds six DINOv2-S critics by 1.00 point (Table 6). The single-critic GeoUP entry averages PDMS across six independent runs; the ensemble averages their predictions.

GRPO and Scene Allocation. Cross-scene GRPO reaches 94.54/94.53 PDMS with updated/fixed allocations versus 90.84 for full GRPO (Table 7). The gain supports refining modes on suitable scenes to retain useful alternatives (Figure 4). Fixed allocations differ by 0.01 points and cache critic-based assignments once, avoiding repeated decoding and scoring. Allocation needs no simulator queries; RL still uses simulator rewards.

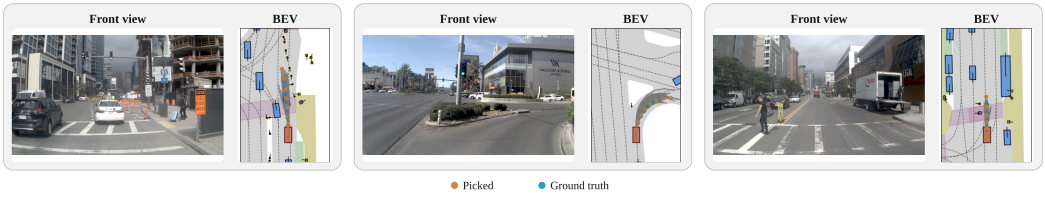

Figure 5: Qualitative examples in three scenes. Front-camera views provide context; bird's-eye views show selected trajectories (orange) and ground truth (blue).

7 Conclusion

Beyond One Path combines GVP-based flow pretraining, distribution-level noise optimization, and cross-scene refinement to improve generative driving. The shared noise vocabulary exposes diverse, high-quality alternatives at a fixed candidate budget and supports behavioral specialization through scene-assigned updates. Results on NAVSIM v1, HUGSIM, and NavSafe-∞ highlight the value of jointly developing candidate coverage, trajectory quality, and effective selection throughout policy learning.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript editing, research brainstorming, and assistance with software development and debugging. We did not use generative AI tools to formulate scientific claims, design the evaluation methodology, or interpret experimental results. We have reviewed all AI-assisted work. We manually reviewed and tested AI-assisted code for correctness. We independently verified all scientific claims, experimental results, and conclusions. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies planning behavior using recorded driving data and simulation-based evaluation. A high benchmark reward or a diverse candidate set does not establish safe behavior in deployment. KL regularization does not enforce safety constraints, and scenario assignment may neglect uncommon situations if suitability estimates are inaccurate. We therefore distinguish privileged evaluation from deployable inputs and report failures and protocol limitations. No real-world deployment or reactive closed-loop safety claim is made.

REPRODUCIBILITY STATEMENT

Sections 4 and 5 specify flow pretraining, vocabulary optimization, and cross-scene post-training. Appendix B describes construction algorithms, the calibration and evaluation protocol, and the artifacts required for an exact rerun; Appendix C details selector training and decode budgets. Vocabulary calibration uses training scenes, and the resulting noise vectors remain fixed during test-scene decoding. A calibration manifest linked to the reported GVP vocabulary is still needed to verify the exact scene split and reproduce that bank.

REFERENCES

Yuxin Bao, Hongwei Ruan, Luobin Wang, Seth Z. Zhao, Ziyang Leng, Zihan Zhang, Yu Zeng, Rowan McAllister, Henrik Christensen, and Bolei Zhou. NavSafe-∞: Benchmarking closed-loop driving safety in photorealistic environments, 2026.

Tony Bonnaire, Raphaël Urfin, Giulio Biroli, and Marc Mézard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy MAP inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023. doi: 10.1109/TPAMI.2022.3200245.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28706–28719. Curran Associates, Inc., 2024. doi: 10.52202/079017-0902.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862, 2023.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8350, 2023.

Peter Karkus, Maximilian Igl, Yuxiao Chen, Kashyap Chitta, Jef Packer, Bertrand Douillard, Ran Tian, Alexander Naumann, Guillermo Garcia-Cobo, Shuhan Tan, et al. Beyond behavior cloning in autonomous driving: a survey of closed-loop training techniques, 2025. Preprint.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, Anh-Quan Cao, Nermin Samet, Tuan-Hung Vu, and Matthieu Cord. Driving on registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32058–32069, 2026.

Derun Li, Changye Li, Yue Wang, Jianwei Ren, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Peng Jia, Xianpeng Lang, Ningyi Xu, and Hang Zhao. Learning personalized driving styles via reinforcement learning from human feedback, 2025.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. DriveVLA-W0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, pp. 7890–7911, 2026a.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. ReCogDrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, 2026b.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. DiffusionDrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems*, volume 38, pp. 40783–40818. Curran Associates, Inc., 2025. doi: 10.52202/085713-1362.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision (ECCV)*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Mingxing Rao and Daniel Moyer. Generalization and memorization in rectified flow. In *European Conference on Computer Vision*, pp. 537–554. Springer, 2026.

Chen Shi, Shaoshuai Shi, Kehua Sheng, Bo Zhang, and Li Jiang. DriveX: Omni scene modeling for learning generalizable world knowledge in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28599–28609, October 2025.

Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Diver: Reinforced diffusion breaks imitation bottlenecks in end-to-end autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. SparseDriveV2: Scoring is all you need for end-to-end autonomous driving. In *European Conference on Computer Vision*, 2026.

Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, Hangjun Ye, Long Chen, and Hongyang Li. SimScale: Learning to drive via real-world simulation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36365–36374, June 2026.

Andrew Wagenmaker, Yunchu Zhang, Mitsuhiko Nakamoto, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 258–282. PMLR, 2025.

Junli Wang, Yinan Zheng, Xueyi Liu, Zebin Xing, Pengfei Li, Kun Ma, Hangjun Ye, Guang Chen, Guang Li, Long Chen, Zhongpu Xia, and Qichao Zhang. MeanFuser: Fast one-step multi-modal trajectory generation and adaptive reconstruction via MeanFlow for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17884–17893, 2026.

Maciej Wozniak, Lianhang Liu, Yixi Cai, and Patric Jensfelt. Prix: Learning to plan from raw pixels for end-to-end autonomous driving. *IEEE Robotics and Automation Letters*, 11(5):6400–6407, 2026.

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. GoalFlow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1602–1611, June 2025.

Yifang Xu, Jiahao Cui, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Feipeng Cai, Neng Zhang, Yaoyi Li, Jia Cai, et al. WAM-Flow: Parallel coarse-to-fine motion planning via discrete flow matching for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24918–24928, 2026.

Ye Yuan and Kris M. Kitani. Diverse trajectory forecasting with determinantal point processes. In *International Conference on Learning Representations*, 2020.

Bozhou Zhang, Nan Song, Jingyu Li, Xiatian Zhu, Jiankang Deng, and Li Zhang. Future-aware end-to-end driving: Bidirectional modeling of trajectory planning and scene evolution. In *Advances in Neural Information Processing Systems*, volume 38, pp. 10204–10229, 2025a. doi: 10.52202/085713-0345.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27220–27230, October 2025b.

Zongchuang Zhao, Xin Zhou, Tianyang Xu, Zhengyang Sun, Kaixuan Zhou, Yu Wu, Honglin Li, Dingkang Liang, and Xiang Bai. SimWAM: A simple world action model for end-to-end autonomous driving, 2026.

Zhiyu Zheng, Shaoyu Chen, Haoran Yin, Xinbang Zhang, Jialv Zou, Xinggang Wang, Qian Zhang, and Lefei Zhang. ResAD: Normalized residual trajectory modeling for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3729–3739, 2026.

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. HUGSIM: A real-time, photo-realistic and closed-loop simulator for autonomous driving, 2024.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. AutoVLA: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 27920–27956, 2025.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

A Extended Benchmark Comparisons

We report expanded NAVSIM v1 and NavSafe-∞ comparisons. Table 2 includes the complete HUGSIM comparison from DrivoR.

A.1 NAVSIM v1

Table 8: **Full NAVSIM v1 comparison** on `navtest`. All distinct baseline variants in DrivoR and MeanFuser (Kirby et al., 2026; Wang et al., 2026).

Method	Venue	Input	NC	DAC	TTC	Comf.	EP	**PDMS**
Reference policies								
PDM-Closed	CoRL '23	GT	94.6	99.8	89.9	86.9	99.9	89.1
Human driver	NeurIPS '24	–	100.0	100.0	100.0	99.9	87.5	94.8
Ego-status MLP	NeurIPS '24	S	93.0	77.3	83.6	**100.0**	62.8	65.6
Camera-only planners								
UniVLA	arXiv '25	C	96.9	91.1	91.7	96.7	76.8	81.7
DrivingGPT	ICCV '25	C	98.9	90.7	94.9	95.6	79.7	82.4
UniAD	CVPR '23	C	97.8	91.9	92.9	**100.0**	78.8	83.4
LTF	TPAMI '23	C	97.4	92.8	92.4	**100.0**	79.0	83.8
PARA-Drive	CVPR '24	C	97.9	92.4	93.0	99.8	79.3	84.0
DriveX-S	ICCV '25	C	97.5	94.0	93.0	**100.0**	79.7	84.5
World4Drive	ICCV '25	C	97.4	94.3	92.8	**100.0**	79.9	85.1
DRAMA	ISRR '24	C	98.0	93.1	94.8	**100.0**	80.1	85.5
VADv2	ICLR '26	C	98.1	94.8	94.3	**100.0**	80.6	86.2
Epona	ICCV '25	C	97.9	95.1	93.8	99.9	80.4	86.2
PRIX	RA-L '26	C	98.1	96.3	94.1	**100.0**	82.3	87.8
MeanFuser	CVPR '26	C	98.6	97.0	95.0	**100.0**	82.8	89.0
AutoVLA	NeurIPS '25	C	98.4	95.6	**98.0**	99.9	81.9	89.1
DriveVLA-W0	ICLR '26	C	98.7	99.1	95.3	99.3	83.3	90.2
ReCogDrive	ICLR '26	C	97.9	97.3	94.9	**100.0**	87.3	90.8
Hydra-MDP++ (V2-99)	arXiv '25	C	98.6	98.6	95.1	**100.0**	85.7	91.0
iPad	RA-L '26	C	98.6	98.3	94.9	**100.0**	88.0	91.7
DrivoR (navtrain)	CVPR '26	C	98.9	98.3	96.2	**100.0**	89.1	93.1
DriveSuprim	AAAI '26	C	98.6	98.6	95.5	**100.0**	91.3	93.5
Camera–LiDAR planners								
VADv2 (R34)	ICLR '26	C+L	97.2	89.1	91.6	**100.0**	76.0	80.9
TransFuser	TPAMI '23	C+L	97.7	92.8	92.8	**100.0**	79.2	84.0
GoalFlow (R34, no scorer)	CVPR '25	C+L	98.3	93.8	94.3	**100.0**	79.8	85.7
DistillDrive	arXiv '25	C+L	98.1	94.6	93.6	**100.0**	81.0	86.2
Hydra-MDP (R34)	arXiv '24	C+L	98.3	96.0	94.6	**100.0**	78.7	86.5
TrajHF (EM)	arXiv '25	C+L	96.6	96.6	92.1	**100.0**	84.5	87.6
DiffusionDrive	CVPR '25	C+L	98.2	96.2	94.7	**100.0**	82.2	88.1
WoTE	ICCV '25	C+L	98.5	96.8	94.9	99.9	81.9	88.3
DIVER	arXiv '25	C+L	98.5	96.5	94.9	**100.0**	82.6	88.3
Hydra-MDP (V2-99)	arXiv '24	C+L	98.0	97.8	93.9	**100.0**	86.5	90.3
GoalFlow (V2-99)	CVPR '25	C+L	98.4	98.3	94.6	**100.0**	85.0	90.3
ResAD (V2-99)	CVPR '26	C+L	98.9	97.8	94.9	**100.0**	87.0	90.6
SeerDrive (V2-99)	NeurIPS '25	C+L	98.8	98.6	95.8	**100.0**	84.2	90.7
Additional training data								
DrivoR (navtrain+navval)	CVPR '26	C	99.0	98.9	96.7	**100.0**	90.0	93.7
RAP-DINO (10× navtrain)	ICLR '26	C	99.1	98.9	96.7	**100.0**	90.3	93.8
DrivoR (+65k SimScale)	CVPR '26	C	99.1	99.0	96.9	**100.0**	90.3	94.0
DrivoR (+134k SimScale)	CVPR '26	C	99.1	99.2	96.9	**100.0**	91.6	**94.6**
Ensembling								
Hydra-MDP-C (V2-99)	arXiv '24	C+L	98.7	98.2	95.0	**100.0**	86.5	91.0
Test-time training								
Centaur	arXiv '25	C+L	**99.5**	98.9	**98.0**	**100.0**	85.9	92.6
Our method								
Ours (pretrained)	–	C	98.9	99.2	95.4	**100.0**	**92.1**	94.1
Ours (post-trained)	–	C	99.2	**99.4**	96.7	**100.0**	91.3	94.5

C: cameras; L: LiDAR; S: ego state; GT: ground-truth scene information; R34: ResNet-34. Groups distinguish training data and inference settings. Human and privileged policies are unranked. Metrics follow Table 1 (↑). **Bold**: best; underline: second-best per metric column (ties included).

A.2 NAVSAFE-∞

Table 9 reports all 17 learned-policy configurations and both reference policies in the NavSafe-∞ evaluation. Each row includes the eight metrics defined in Section 6.1.

Table 9: **Full NavSafe-∞ comparison.** All evaluated policies in Bao et al. (2026); entries are means over three seeds.

			Overall performance				Event-category success			
Method	Venue	Input	**DS**	**SR**	**DE**	**Comf.**	**TC**	**VRUC**	**TV**	**TI**
Reference policies										
Human reference	–	–	100	100	316.52	75.15	100	100	100	100
PDM-Closed	CoRL '23	Priv.	90.17	81.29	305.27	82.49	88.97	71.62	84.24	57.78
Imitation learning										
LTF	NeurIPS '24	NR	53.04	27.71	163.71	97.81	31.81	24.81	28.35	15.56
DiffusionDrive	CVPR '25	NR	44.77	20.96	165.08	97.68	21.27	11.86	23.65	22.22
DrivoR	CVPR '26	C	64.74	50.69	188.02	87.92	58.61	31.38	52.53	43.33
RAP	ICLR '26	C	47.70	27.32	220.78	48.38	32.36	14.12	28.93	22.22
SparseDriveV2	ECCV '26	C	65.26	48.78	167.99	93.41	53.55	43.46	48.32	41.67
GTRS-D	arXiv '25	NR	64.27	49.45	163.88	95.90	52.04	40.00	54.21	36.67
ReCogDrive-2B-IL	ICLR '26	C	48.98	28.28	166.94	97.58	30.03	17.18	31.16	26.67
MTDrive-SFT	arXiv '26	C	48.78	31.79	173.77	94.51	30.40	15.20	35.65	44.44
Reinforcement learning										
ReCogDrive-2B-RL	ICLR '26	C	53.64	30.99	173.96	85.10	35.12	18.70	32.59	27.78
MTDrive-mtGRPO	arXiv '26	C	55.86	40.69	**277.62**	93.94	40.35	17.04	46.46	52.22
AutoVLA	NeurIPS '25	C	51.05	29.52	186.02	65.37	32.89	14.76	29.87	36.67
Perturbed-state training										
DD + SimScale	CVPR '26	C	52.19	35.29	189.66	96.92	38.48	17.94	36.81	42.22
DrivoR + SimScale	CVPR '26	C	57.66	42.50	176.73	85.88	44.00	17.50	48.18	50.00
LTF + SimScale	CVPR '26	C	52.65	34.18	192.09	97.41	34.78	17.50	38.48	36.67
GTRS-D + SimScale	CVPR '26	NR	73.54	61.82	141.50	87.69	**69.11**	**50.00**	64.19	43.33
DD + BeyondDrive	ECCV '26	NR	46.61	21.78	172.79	97.24	22.65	10.00	25.14	22.22
World-model training										
SimWAM	arXiv '26	C	57.53	37.50	160.54	97.19	39.55	35.00	43.25	33.33
Ours (pretrained)	–	C	71.65	59.64	201.96	97.07	67.49	39.27	60.73	46.67
Ours (post-trained)	–	C	**74.19**	**62.86**	204.11	**98.60**	69.00	42.50	**67.27**	**53.33**

C: cameras; Priv.: privileged inputs; NR: evaluation sensors not reported; DD: DiffusionDrive. GTRS-D denotes GTRS-Dense with V2-99. DrivoR + SimScale uses 134k synthetic scenes. Scores use a 0–100 scale (↑), except DE, which retains its native scale. Human and privileged policies are unranked. **Bold**: best; underline: second-best per metric column (ties included).

B NOISE VOCABULARY OPTIMIZATION

We detail two ways to construct the shared vocabulary in Equation (9): continuous optimization of the noise coordinates and discrete selection from a large candidate pool.

Calibration and test-scene reuse. Our protocol constructs one vocabulary on training scenes and holds its noise vectors fixed when evaluating test scenes. Scene conditioning changes the decoded trajectories, and the critic selects among them; neither operation reoptimizes the vocabulary. The NAVSIM noise-construction ablation therefore assesses reuse across scenes within one benchmark. HUGSIM and NavSafe-∞ evaluate the complete planner, including perception, generation, selection, and post-training; their scores do not isolate the vocabulary's contribution to cross-domain transfer.

Stronger generalization test. A vocabulary-specific transfer test should freeze the bank, generator, critic, and selection rule before evaluation on driving logs or a benchmark excluded from training, calibration, and method selection. At the same decode budget, it should compare the optimized bank with IID and antithetic noises, report oracle, mean, and selected quality together with trajectory diversity, and quantify variability across calibration subsets and random seeds. Such a test would distinguish a transferable vocabulary benefit from dependence on a particular calibration set or evaluation-driven recipe choice; it is not established by the whole-planner benchmark comparisons alone.

Artifacts for exact reproduction. An exact rerun requires calibration token and log identifiers, the eligible scene-pool snapshot and sampling seed, the candidate pool and selected noise vectors, the generator and critic checkpoints, and the solver and selection settings. These records should identify the code revision and artifact hashes and allow token- and log-level overlap checks against evaluation data. The current manuscript package does not contain a verified calibration manifest linked to the reported GVP bank. Consequently, the construction algorithms specify the procedure, while exact reconstruction of that bank and an independent overlap audit remain pending.

B.1 DIFFERENTIABLE NOISE OPTIMIZATION THROUGH FLOW DENOISING

For continuous Z, a fixed-step flow sampler is a differentiable composition of denoising updates. Unrolling these updates and applying BPTT gives gradients with respect to the initial noises while keeping the pretrained parameters θ_0 fixed. The gradient path includes the motion-to-trajectory decoder D. This requires differentiable diversity and quality terms: we use $\widetilde{C} = C$ when the cost is differentiable, or a differentiable surrogate $\widetilde{C}$ when the original cost comes from a simulator. In the latter case, the gradient updates optimize the surrogate cost.

Algorithm 1 takes the scene-wise form $\mathrm{Div}_{\mathcal{C}}(Z) = N^{-1} \sum_n \mathcal{D}(\tau_{n1}, \ldots, \tau_{nK})$, where $\mathcal{D}$ is differentiable, so uniformly sampled scene minibatches give unbiased gradients of this empirical objective. The Euler discretization follows the noise-to-data time convention in Equation (3); other differentiable fixed-step solvers can replace it. The projection $\Pi_{\mathcal{Z}_K}$ enforces any continuous noise constraints, including $z_1 = 0$ when retaining zero, and is the identity for unconstrained noises.

Algorithm 1 Shared noise optimization by BPTT

Require: Scenes $\mathcal{C}$, initial bank $Z^{(0)} \in \mathbb{R}^{K\times d}$, frozen flow v_{θ_0};
decoder D, smooth diversity $\mathcal{D}$, cost $\widetilde{C}$, weights λ_C, λ_Z;
L denoising steps, T updates, learning rate η, feasible set $\mathcal{Z}_K$

Cache scene encodings; set $Z \leftarrow Z^{(0)} \in \mathcal{Z}_K$; freeze θ_0
for $r = 1, \dots, T$ **do**
  Sample a uniform scene minibatch $\mathcal{B} \subseteq \mathcal{C}$
  for all $n \in \mathcal{B}, k \in \{1, \dots, K\}$ **in parallel do**
    $x_{nk}^{(0)} \leftarrow z_k$ ▷ Retain gradients to the shared bank
    for $\ell = 0, \dots, L-1$ **do**
      $t_\ell \leftarrow \ell / L$
      $x_{nk}^{(\ell+1)} \leftarrow x_{nk}^{(\ell)} + L^{-1} v_{\theta_0}(x_{nk}^{(\ell)}, t_\ell, c_n)$
    end for
    $\tau_{nk} \leftarrow D(x_{nk}^{(L)})$
  end for
  $d_{\mathcal{B}} \leftarrow |\mathcal{B}|^{-1} \sum_{n\in\mathcal{B}} \mathcal{D}(\tau_{n1}, \dots, \tau_{nK})$
  $c_{\mathcal{B}} \leftarrow (|\mathcal{B}|K)^{-1} \sum_{n\in\mathcal{B}} \sum_{k=1}^{K} \widetilde{C}(c_n, \tau_{nk})$
  $\widehat{J} \leftarrow d_{\mathcal{B}} - \lambda_C c_{\mathcal{B}} - (\lambda_Z/K)\|Z - Z^{(0)}\|_F^2$
  $g \leftarrow \nabla_Z \widehat{J}$ by BPTT through the decoder and all denoising steps
  $Z \leftarrow \Pi_{\mathcal{Z}_K}(Z + \eta g)$ ▷ Update noises; keep θ_0 fixed
end for
return Z

B.2 Discrete K-Seed Selection with Quality-Aware Log-Determinant Maximization

Restricting the bank to K members of a fixed pool gives a combinatorial instance of Equation (9) with $\lambda_Z = 0$. A large pool can be formed from antithetic pairs $\{u_j, -u_j\}_{j=1}^{m}$, $u_j \sim \mathcal{N}(0, I_d)$, together with zero, giving $M = 2m + 1$ candidates. Selection measures diversity after decoding across a fixed calibration set. For each candidate i, cache $f_i = N^{-1/2} \operatorname{concat}_{n=1}^{N} \operatorname{xy}(\tau_{ni})$ and $\bar{C}_i = N^{-1} \sum_n C(c_n, \tau_{ni})$. The cost can be the original non-differentiable simulator metric. Define a positive-definite trajectory kernel and the subset objective by

$$
\begin{aligned}
A_{ij} &= \exp\!\left(-\frac{\|f_i - f_j\|_2^2}{2\ell^2}\right) + \varepsilon \mathbf{1}[i = j], \qquad \ell > 0, \quad \varepsilon > 0, \\
\max_{\substack{S\subseteq\{1,\dots,M\} \\ |S|=K,\ 1\in S}} \quad & J(S) = \log\det A_{S,S} - \frac{\lambda_C}{K} \sum_{i\in S} \bar{C}_i, \qquad z_1 = 0.
\end{aligned}
\tag{13}
$$

Algorithm 2 approximates this objective by greedy selection with incremental Cholesky updates (Chen et al., 2018). The residual $r_i = A_{ii} - A_{iS} A_{S,S}^{-1} A_{Si}$ gives the exact marginal gain $\log r_i - (\lambda_C/K)\bar{C}_i$. Thus, with $q_i = \exp[-\lambda_C \bar{C}_i/(2K)]$ and $Q = \operatorname{diag}(q)$, the objective equals $\log\det(QAQ)_{S,S}$, a quality-weighted K-DPP MAP objective. The objective is generally non-monotone; greedy selection returns exactly K seeds as a heuristic for the subset optimum.

Algorithm 2 Quality-aware selection of K seeds from an antithetic pool

Require: Scenes $\mathcal{C}$, frozen deterministic flow τ^{θ_0}, cost C;
pool size $M = 2m + 1 \geq K$, bandwidth ℓ, jitter $\varepsilon > 0$, weight λ_C

Draw $u_1, \ldots, u_m \overset{\text{iid}}{\sim} \mathcal{N}(0, I_d)$
$(z_1, \ldots, z_M) \leftarrow (0, u_1, -u_1, \ldots, u_m, -u_m)$
for all $i \in \{1, \ldots, M\}$ **in batches do**
    Decode $\tau_{ni} \leftarrow \tau^{\theta_0}(z_i, c_n)$ for all n without gradient tracking
    Cache $f_i \leftarrow N^{-1/2} \operatorname{concat}_{n=1}^{N} \operatorname{xy}(\tau_{ni})$
    Cache $\bar{C}_i \leftarrow N^{-1} \sum_{n=1}^{N} C(c_n, \tau_{ni})$
end for
$S \leftarrow \varnothing$; $V \leftarrow 0_{K \times M}$; $r_i \leftarrow 1 + \varepsilon$ for all i; $\beta \leftarrow \lambda_C / K$
for $t = 1, \ldots, K$ **do**
    if $t = 1$ **then**
        $j \leftarrow 1$ ▷ Retain the zero seed
    else
        $j \leftarrow \arg\max_{i \notin S} \{\log r_i - \beta \bar{C}_i\}$
    end if
    $S \leftarrow S \cup \{j\}$
    if $t < K$ **then**
        Compute $a_i \leftarrow A_{ij}$ on demand from f_i, f_j for all $i \notin S$
        $e_i \leftarrow \big(a_i - \sum_{s=1}^{t-1} V_{si} V_{sj}\big) / \sqrt{r_j}$ for all $i \notin S$
        $V_{ti} \leftarrow e_i$ and $r_i \leftarrow r_i - e_i^2$ for all $i \notin S$
    end if
end for
return $(z_j)_{j \in S}$ in selection order

C DSRL: Noise Steering versus Trajectory Scoring

Diffusion steering via reinforcement learning (DSRL) learns a policy over the initial noise of a frozen generative policy (Wagenmaker et al., 2025). We adapt this idea to the flow planner and compare three ways to choose a trajectory: predicting a continuous noise vector, selecting an entry from the fixed noise vocabulary, and scoring decoded trajectories. This appendix details the recorded pre-refinement selectors, their training objectives, and their decode budgets. In Table 10, "Before" and "After" refer to generator RL fine-tuning; the continuous noise actor is already reward-trained in the Before column.

C.1 Continuous Noise Steering

Let $\bar{\theta}$ denote a frozen generator checkpoint and $p_\psi(z \mid c)$ a scene-conditioned noise policy. Steering changes the distribution of z while retaining the mapping $\tau^{\bar{\theta}}(z, c)$ defined in Section 4. Our NAVSIM adaptation is a one-step contextual bandit: a logged scene supplies c, the actor supplies z, and the simulator evaluates the decoded trajectory. The reward and actor distribution are

$$r_{\bar{\theta}}(c, z) = R(c, \tau^{\bar{\theta}}(z, c)), \qquad p_\psi(z \mid c) = \mathcal{N}\big(\mu_\psi(c), \operatorname{diag}(\sigma_\psi^2(c))\big). \tag{14}$$

Here R is NAVSIM v1 PDMS on a $[0, 1]$ scale during training. Both the visual encoder and flow generator remain frozen; only the noise actor and its critics are trained.

Actor–critic objective. The continuous result uses a SAC-style adaptation with direct reward regression and Gaussian-prior regularization. Two noise critics Q_{η_1}, Q_{η_2} predict simulator reward from the scene and initial noise. With replay buffer $\mathcal{B}$, their loss and the actor loss are

$$\mathcal{L}_Q(\eta) = \mathbb{E}_{(c,z,r) \sim \mathcal{B}} \left[\frac{1}{2} \sum_{j=1}^{2} \big(Q_{\eta_j}(c, z) - r\big)^2 \right], \tag{15}$$

$$\mathcal{L}_{\text{actor}}(\psi) = \mathbb{E}_c \left[\beta_z D_{\text{KL}}(p_\psi(\cdot \mid c) \,\|\, \mathcal{N}(0, I_d)) - \mathbb{E}_{z \sim p_\psi(\cdot|c)} \min_{j=1,2} Q_{\eta_j}(c, z) \right], \tag{16}$$

with $\beta_z = 0.003$ and the KL summed over all 24 noise coordinates. Critic targets are observed trajectory rewards, so training requires no next-state or temporal-difference target. Eight reparameterized antithetic noise samples per scene estimate the actor expectation; actor gradients pass through the noise critics. The Gaussian-prior KL controls the noise distribution, whereas the KL in our generator post-training regularizes denoising transitions.

Architecture and training. The actor reads frozen scene tokens and ego-motion/navigation features using two cross-attention layers with four heads. Each critic uses four such layers; the shared token width is 256. The actor outputs a 24-dimensional initial noise for eight trajectory steps, with its mean clipped elementwise to $[-3, 3]$. Standard deviations are configured in $[0.25, 0.5]$ and remain at the upper bound in the reported run. Each batch contains 32 training scenes, with 32 actor samples, eight standard-normal samples, and the actor mean evaluated per scene. A FIFO buffer stores 16,384 scenes and their scored noises. After 100 critic-only warm-up batches, each outer step performs one actor update, one critic update on the fresh batch, and seven critic updates on replay batches of 64 scenes. Separate AdamW optimizers use learning rates 10^{-4} for the actor and 3×10^{-4} for the critics, zero weight decay, and gradient-norm clipping at 1.0.

For this baseline, the 91.09 result uses a separate demonstration-fine-tuned velocity-prediction generator with a linear probability path, three actor-training epochs on `navtrain`, training seed 2, and the final checkpoint. At inference, we decode $\mu_\psi(c)$ with ten deterministic Euler steps to produce one trajectory. The noise critics are used during training only.

C.2 Discrete Noise Selection

The discrete noise-space baseline scores the $K = 64$ fixed vocabulary entries before decoding. Its selection rule is

$$\hat{k}(c) = \arg\max_{1 \le k \le K} s_\phi^{\text{noise}}(c, z_k, k), \qquad \hat{\tau}(c) = \tau^{\bar{\theta}}(z_{\hat{k}(c)}, c). \tag{17}$$

The recorded 91.48 result is a supervised noise-only selector. Each candidate token combines its vocabulary identity and raw noise vector; three attention layers combine candidate tokens, frozen GeoUP scene tokens, and ego state. Six output heads predict the PDMS submetrics. Three independently trained models, with seeds 0, 1, and 2, are ensembled by averaging their composed candidate scores.

Training uses cached simulator labels from 103,288 `navtrain` scenes, each containing the 64 vocabulary candidates and 124 additional Gaussian/grid candidates. Logs are held out for validation. The loss combines submetric binary cross-entropy, score distillation from a trajectory-conditioned GeoUP ensemble, and a contrastive alignment term between noise-conditioned candidate features and their decoded trajectories, with weights 1, 1, and 0.3, respectively. Distillation uses per-scene standardized scores with temperature 0.5; alignment uses temperature 0.1. Decoded trajectories supply this auxiliary training supervision. At inference the selector reads only scene/state features, identities, and noise vectors, then decodes its chosen entry once. We train for 12 epochs with batch size 64, AdamW learning rate 2×10^{-4}, weight decay 0.01, and dropout 0.2, selecting checkpoints by held-out selected PDMS.

C.3 Comparison Protocol and Results

All scores in Table 10 are reported on the 0–100 PDMS scale. The continuous actor and discrete noise selector each require one base trajectory decode at deployment; the trajectory scorer decodes all 64 vocabulary entries before selecting. The table therefore compares complete selection systems with different representations and decode budgets. The noise-only entry uses three ensembled models. The recorded pre-refinement noise-selection evaluations cover the full `navtest` split of 12,146 scenes.

Table 10: **Noise steering and trajectory scoring.** NAVSIM v1 PDMS before and after generator RL fine-tuning. Base decodes count trajectories generated by the pre-refinement selection rule, each with ten Euler steps.

Method	Space	Choice	Base decodes	Before	After
DSRL (SAC-style actor)	Noise	Cont.	1	91.09	91.85
Noise-only selector	Noise	Disc.	1	91.48	92.31
Trajectory scorer	Traj.	Disc.	64	**94.17**	**94.53**

Selecting noises or decoded trajectories. Trajectory scoring achieves the highest PDMS in Table 10 both before and after generator fine-tuning. To examine candidate input more directly, we compare single selectors with the same frozen policy register features, submetric supervision, and additional training candidates. Replacing raw noise with decoded waypoints improves selected PDMS from 89.48 to 90.73; both runs use seed 0 and omit distillation and alignment. This comparison more closely isolates the information provided by the decoded trajectory.

Direct classification diagnostic. A separate baseline maps scene/state features directly to 64 logits and minimizes the exact negative expected reward, $-\mathbb{E}_c \sum_k p_\phi(k \mid c) R(c, \tau^{\bar{\theta}}(z_k, c))$, using cached reward tables from an initial 30,000-scene pool. This classifier converges to the zero-noise entry and scores 87.11 PDMS. It has no per-candidate noise inputs or submetric heads, so its collapse concerns this direct-classification formulation; the stronger noise-only selector above achieves 91.48.

Single-trajectory steering and candidate coverage. An additional diagnostic applies the same continuous-actor recipe to a plain imitation-trained generator using the linear-schedule ablation at epoch 19. The deployed actor mean reaches 90.13 PDMS, compared with 85.42 for zero initial noise, using one decode in each case. With 64 candidates and the same frozen trajectory scorer, a pool comprising the actor mean and 63 actor samples reaches 92.63, compared with 93.16 for the fixed vocabulary. The actor pool has higher mean candidate PDMS (87.83 versus 73.12) but lower oracle PDMS (95.18 versus 97.23). These measurements use the same generator and selector in the two 64-candidate evaluations, with training seed 2; candidate means and oracles use joint scoring within each proposal set. They show that improving a single trajectory and improving the alternatives available to a selector are distinct objectives.