PLAN2PLAN: RETHINKING TRAJECTORY GENERATION AS PLAN-TO-PLAN TRANSPORT FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Trajectory proposal-based planning has emerged as a promising paradigm for end-to-end autonomous driving. However, existing generative planners typically produce trajectory candidates from scratch using Gaussian noise or generic trajectory anchors. Such *scratch-to-plan* transport incurs a substantial transport gap, complicating trajectory generation and compromising plan quality. In this work, we propose Plan2Plan, a generative framework that reformulates candidate generation as a *plan-to-plan* transport problem, directly transforming preliminary plans from an off-the-shelf planner (termed *expert priors*) into high-quality driving trajectories. Specifically, we develop an expert prior-guided Shortcut flow-matching model that recasts the velocity field as a scene-conditioned corrective field over preliminary plans. Starting from structured, plausible plans substantially shortens the transport, shifting the generative task from complete trajectory construction to residual plan correction and enabling reliable single-step refinement. Crucially, to supervise this corrective flow matching without collapsing multimodal modes, we avoid uniformly pulling all expert priors toward the single ground-truth demonstration. Instead, we introduce a *geometry-aware multi-expert supervision scheme* that preserves strong priors while steering suboptimal ones toward their geometrically closest higher-quality trajectory, elevating plan quality while safeguarding driving intent. Extensive experiments on NAVSIM-v1, NAVSIM-v2, and closed-loop HUGSIM demonstrate that Plan2Plan outperforms strong generative planning baselines across diverse evaluation settings.

1 INTRODUCTION

End-to-end autonomous driving aims to map sensory observations, ego states, and navigation commands directly to future ego trajectories, reducing error accumulation across conventional modular pipelines (Chen et al., 2024a; Chitta et al., 2023; Hu et al., 2023; Jiang et al., 2023). Nevertheless, end-to-end driving remains challenging because real-world driving is inherently multimodal: given the same traffic scene, multiple driving behaviors can be simultaneously valid, such as following a lead vehicle or changing lanes when appropriate. A single deterministic trajectory can hardly capture such behavioral diversity. Consequently, recent methods increasingly follow a *generate-then-select* paradigm, first generating multiple trajectory candidates and then selecting the most suitable one (Kirby et al., 2026; Liao et al., 2025; Xing et al., 2025; Wang et al., 2026b). The success of this paradigm hinges on candidate generation, specifically, whether the generator can synthesize diverse, kinematically viable, and scene-adaptive trajectories for subsequent evaluation.

Existing generative planners typically follow a *scratch-to-plan* paradigm that constructs those trajectory candidates from scratch. One line of work initializes generation from Gaussian or mixture noise, transforming it into trajectories via iterative denoising or continuous transport (Ho et al., 2020; Lipman et al., 2023; Zheng et al., 2025; Wang et al., 2026b). Another line replaces noise with predefined anchors that capture generic, dataset-level driving patterns (Liao et al., 2025; Wang et al., 2025; Zou et al., 2025; Yin et al., 2026). Despite different starting points, neither initialization provides a concrete driving plan tailored to the current scene: noise priors carry zero spatial context, while static anchors lack adaptivity to instantaneous road geometries and traffic interactions. As such, the generator must bridge a substantial transport gap across the trajectory manifold to construct feasi-

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: **Comparison of trajectory generation paradigms.** Existing generative planners typically start from random noise or generic trajectory anchors and require a long *scratch-to-plan transport* to construct complete trajectory candidates. By contrast, Plan2Plan introduces a novel *plan-to-plan transport* scheme, starting from scene-conditioned expert priors (*i.e.*, preliminary plans from the off-the-shelf planner) and learning a short corrective transport to obtain refined candidates.

ble driving behaviors from an uninformative initialization, thereby complicating trajectory synthesis and potentially compromising candidate quality, as illustrated in Fig. 1.

In this paper, we propose **Plan2Plan**, a generative planning framework that reformulates candidate generation from *scratch-to-plan* synthesis into *plan-to-plan transport*. Instead of starting from noise or generic anchors (Liao et al., 2025; Xing et al., 2025; Wang et al., 2026b), Plan2Plan directly transforms preliminary plans from an off-the-shelf planner (termed *expert priors*) into high-quality driving trajectories. These expert priors already encode plausible driving intentions and scene-specific trajectory structures, substantially reducing the transport gap. Building on this structured initialization, we develop an expert prior-guided Shortcut flow-matching generator that reinterprets the generative velocity field as a scene-conditioned *corrective field* over preliminary plans. Consequently, generation shifts from constructing complete driving behaviors from scratch to learning the residual corrections necessary to improve existing plans, enabling efficient single-step refinement.

Crucially, learning such a corrective transport requires an appropriate destination for each expert prior. Simply pulling all priors toward the single logged trajectory is undesirable: preliminary plans can represent distinct plausible behaviors and vary substantially in quality, while the logged trajectory captures only one realized driving behavior. We therefore introduce a *quality-aware, geometry-compatible target assignment* scheme that preserves strong priors while steering suboptimal ones toward geometrically compatible, higher-quality expert trajectories. In this way, each prior is paired with a meaningful improvement target while its underlying driving intention is retained. Following prior generate-then-select planners (Kirby et al., 2026; Ang et al., 2026), the refined candidates are subsequently ranked by a trajectory scorer for final plan selection. Extensive experiments on NAVSIM-v1, NAVSIM-v2, and closed-loop HUGSIM validate the effectiveness of Plan2Plan. To summarize, our main contributions are as follows:

- We introduce Plan2Plan, a new generative planning paradigm that reformulates trajectory generation from *scratch-to-plan transport* into *plan-to-plan transport*. Instead of constructing trajectories from noise or generic anchors, Plan2Plan starts from scene-conditioned preliminary plans and learns to improve existing planning solutions through short corrective transport.
- We introduce a quality-aware, geometry-compatible *multi-expert supervision* scheme for learning effective plan-to-plan transport. Rather than forcing diverse preliminary plans toward a single logged trajectory, it preserves already strong plans and assigns suboptimal ones to nearby higher-quality expert targets, enabling quality improvement without collapsing distinct driving intents.
- We develop an *expert prior-guided Shortcut flow-matching* formulation that models generation as a scene-conditioned corrective field over preliminary plans. By learning residual plan corrections over structured expert priors, it enables single-step refinement of multimodal trajectory candidates.

2 Related Work

Candidate-Based End-to-End Planning. Candidate-based planning represents multimodal driving decisions with multiple trajectory hypotheses and selects a final plan through scene-conditioned evaluation. Representative approaches model planning uncertainty through probabilistic prediction, multi-target distillation, or expert-guided supervision (Chen et al., 2024b; Li et al., 2024; 2025a).

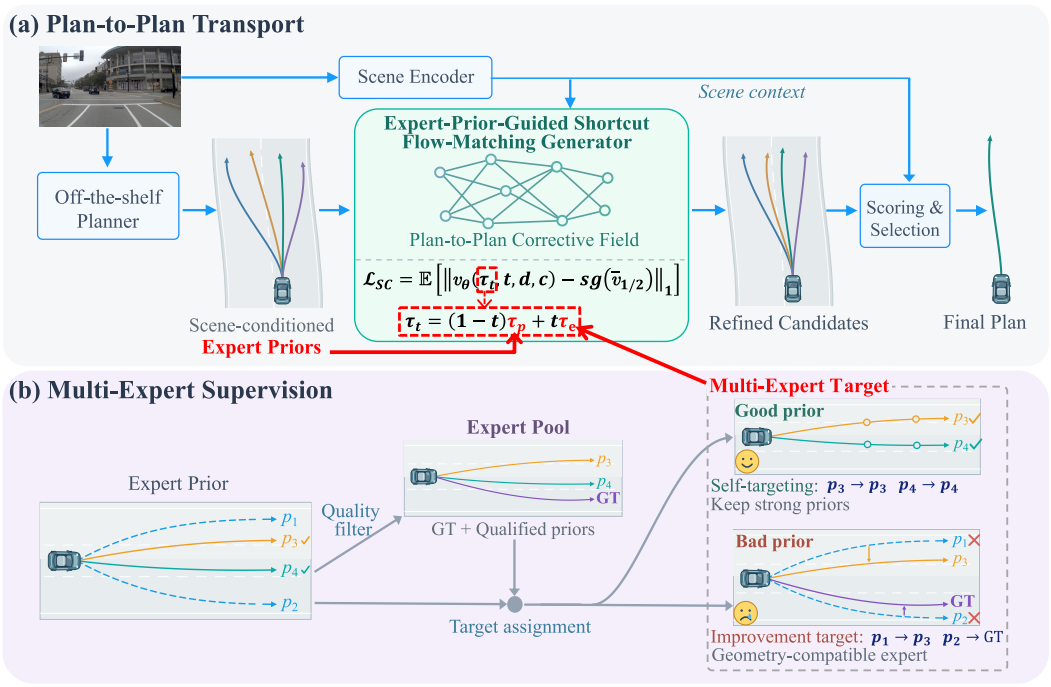


Figure 2: **Overview of Plan2Plan. (a)** Plan2Plan reformulates candidate generation as plan-to-plan transport. Scene-conditioned expert priors produced by an off-the-shelf planner are transformed into refined candidates through expert prior-guided Shortcut flow matching and subsequently scored for final plan selection. **(b)** During training, expert targets are constructed from a scene-specific expert pool through quality-aware, geometry-compatible target assignment.

Another line scores large predefined trajectory vocabularies: SparseDriveV2 improves candidate coverage through factorized trajectory representations, while DriveSuprim further enhances candidate discrimination through coarse-to-fine filtering and hard-sample-oriented training (Sun et al., 2026; Yao et al., 2026). More recent dynamic-proposal methods generate scene-conditioned candidates online, including generalized trajectory scoring in GTRS (Li et al., 2025c), proposal generation and scoring in DrivoR (Kirby et al., 2026), and future-aware selection with a BEV world model in WoTE (Li et al., 2025b). Overall, candidate-based planners are limited either by fixed-vocabulary coverage or the quality of generated proposals; Plan2Plan instead constructs refined candidates from scene-conditioned expert priors before trajectory scoring.

Generative Trajectory Planning. Generative planners typically model multimodal driving trajectories through diffusion or flow-based generation (Ho et al., 2020; Lipman et al., 2023). Noise-initialized methods generate trajectories through iterative denoising or guided sampling (Yang et al., 2024; Zheng et al., 2025), while anchor-based methods start from representative trajectory modes (Liao et al., 2025; Yin et al., 2026). GoalFlow introduces goal-conditioned flow matching (Xing et al., 2025), MeanFuser combines Gaussian-mixture initialization with MeanFlow for efficient one-step generation (Geng et al., 2025; Wang et al., 2026b), and related studies explore Shortcut objectives or planning-reward guidance (Frans et al., 2025; Zou et al., 2025). Most existing methods nevertheless initialize from noise or generic anchors; Plan2Plan instead transports scene-conditioned preliminary plans toward compatible high-quality expert targets.

3 PLAN2PLAN

3.1 BACKGROUND

Given a driving scene observation $\mathcal{O}$, containing onboard sensory inputs, ego-vehicle states, and navigation information, the goal of end-to-end planning is to output an executable future trajectory for the ego vehicle. The trajectory is represented as $\tau = [\mathbf{s}_1, \ldots, \mathbf{s}_T] \in \mathbb{R}^{T\times 3}$, with $\mathbf{s}_t = (x_t, y_t, \psi_t)$ denoting the ego position and heading at the t-th future timestamp.

Most multimodal end-to-end planners instantiate this problem with a *generate-then-select* pipeline (Kirby et al., 2026; Liao et al., 2025; Xing et al., 2025; Wang et al., 2026b). Formally, the planner first encodes the observation into a scene context, $\mathbf{c} = E_{\text{scene}}(\mathcal{O})$, and then decodes a set of future candidate trajectories conditioned on this context,

$$\widehat{\mathcal{T}} = \left\{\hat{\tau}^i = G_\theta(z^i, \mathbf{c})\right\}_{i=1}^{K}, \tag{1}$$

where z^i denotes the initialization of the i-th candidate, such as Gaussian noise or a generic trajectory anchor (Liao et al., 2025; Xing et al., 2025; Wang et al., 2026b). These sources are either unstructured or scene-agnostic, leading to a *scratch-to-plan* generation process. Consequently, a trajectory scorer evaluates the generated candidates together with the scene context and predicts their planning quality, and the highest-scoring trajectory $\hat{\tau}^*$ is selected as the final plan.

3.2 Motivation

Existing generative planners typically construct trajectory candidates from Gaussian noise or generic trajectory anchors (Liao et al., 2025; Xing et al., 2025; Wang et al., 2026b). Although these initializations facilitate multimodal generation, they are not concrete driving plans tailored to the current scene: Gaussian noise contains no planning structure, while predefined anchors only encode generic motion patterns. Consequently, the generator must learn a substantial *scratch-to-plan* transport from a weak initialization, introducing several challenges: **(i) long transport paths** that separate the source from feasible driving trajectories; **(ii) heavy learning burden** from jointly constructing driving intent, trajectory geometry, and scene compliance; and **(iii) inefficient generation**, as accurately traversing such a large gap may require iterative refinement.

This motivates us to reconsider whether such from-scratch generation is necessary. Modern driving systems can readily provide preliminary plans that already capture scene-specific road geometry and plausible driving behaviors. Rather than discarding this useful planning structure, we treat these preliminary trajectories as *expert priors* and reformulate trajectory generation as a *plan-to-plan* transport problem. Starting from structured and plausible plans substantially shortens the transport path, shifting the generative task from complete behavior construction to residual plan correction. As illustrated in Fig. 2, Plan2Plan reformulates trajectory generation as plan-to-plan transport through expert prior-target construction and expert prior-guided Shortcut flow-matching generation. The resulting refined candidates are ranked by a trajectory scorer for final plan selection.

3.3 Geometry-aware Multi-Expert Plan-to-Plan Supervision

Our plan-to-plan formulation requires specifying both ends of the transport: a structured preliminary plan to start from and an appropriate high-quality plan to transport toward. We therefore construct *expert priors* as the source trajectories and assign each of them a *multi-expert target* as its transport destination. The key is to preserve the useful driving intention already encoded in each prior while correcting its planning deficiencies toward a better solution.

Scene-Conditioned Expert Priors. Given the current observation $\mathcal{O}$, an off-the-shelf planner $\mathcal{P}$ produces K preliminary plans:

$$\mathcal{T}_p = \{\tau_p^i\}_{i=1}^{K} = \mathcal{P}(\mathcal{O}). \tag{2}$$

Unlike Gaussian noise or globally shared trajectory anchors, each τ_p^i is already a concrete planning hypothesis tailored to the current scene. It encodes not only a structured trajectory geometry, but also a particular hypothesis of how the ego vehicle should behave under the surrounding traffic and navigation context. We therefore reinterpret these preliminary plans as *expert priors* and use them directly as the source samples of generative transport, without injecting additional perturbations. Importantly, "expert" does not imply optimality. These priors may vary substantially in planning quality, but they already provide useful structure that would otherwise need to be reconstructed from scratch. Plan2Plan starts from such an existing planning solution and focuses the generative process on the residual correction necessary to improve it.

Geometry-Aware Multi-Expert Targets. Defining a structured source immediately raises a complementary question: *where should each prior be transported?* The logged trajectory τ_g provides one natural target, but it represents only one realized behavior and is not necessarily the appropriate destination for every prior. In particular, different priors can represent distinct yet plausible driving

intentions, and some may already achieve planning quality comparable to or better than the logged trajectory. Forcing all priors toward τ_g would therefore impose an unnecessarily large correction on competent plans and may collapse behaviorally distinct modes. We instead allow high-quality, geometrically compatible expert priors themselves to become alternative expert targets. Specifically, we first construct a scene-specific expert pool as follows:

$$\mathcal{E} = \{\tau_g\} \cup \mathrm{Top}_M \left(\left\{ \tau_p^j \,\middle|\, q(\tau_p^j) \geq q(\tau_g) - \delta \right\} \right), \tag{3}$$

where $q(\cdot)$ measures planning quality, δ specifies the admissible quality gap from the logged trajectory, and Top_M retains at most M highest-quality priors satisfying this criterion. Hence, $\mathcal{E}$ contains not only the demonstrated behavior τ_g, but also alternative high-quality planning solutions discovered by the planner itself. Target assignment then follows two complementary principles: **(i)** *quality improvement* determines which experts are worth transporting toward, while **(ii)** *geometric compatibility* determines which of them requires the least behavioral change. For each prior τ_p^i, we first collect sufficiently better experts as $\mathcal{E}_i^+ = \left\{ \tau \in \mathcal{E} \,\middle|\, q(\tau) \geq q(\tau_p^i) + m \right\}$, where m is a minimum quality-improvement margin. The target of each prior is then assigned as:

$$\tau_e^i = \begin{cases} \tau_p^i, & q(\tau_p^i) \geq q(\tau_g), \\ \arg\min\limits_{\tau \in \mathcal{E}_i^+} D_{\mathrm{xy}}(\tau_p^i, \tau), & q(\tau_p^i) < q(\tau_g),\ \mathcal{E}_i^+ \neq \varnothing, \\ \tau_p^i, & q(\tau_p^i) < q(\tau_g),\ \mathcal{E}_i^+ = \varnothing, \end{cases} \tag{4}$$

where $D_{\mathrm{xy}}(\tau_a, \tau_b) = \frac{1}{T} \sum_{t=1}^{T} \left\| \mathbf{p}_{a,t} - \mathbf{p}_{b,t} \right\|_2$ with $\mathbf{p}_t = (x_t, y_t)$ measures the geometric discrepancy between two trajectories by the mean Euclidean distance between their waypoint positions. The assignment follows a simple principle: *preserve when already strong; otherwise, improve with minimal behavioral deviation*. Strong priors are self-targeted, while weaker ones are transported toward the geometrically closest expert offering meaningful quality improvement; if no such expert exists, the prior is preserved. Each preliminary plan is thus paired with its own transport destination, yielding the multi-expert plan-to-plan supervision pool $\mathcal{C} = \{(\tau_p^i, \tau_e^i)\}_{i=1}^K$. They provide the structured source-target coupling on which our expert prior-guided Shortcut transport is built.

3.4 Expert Prior-Guided Shortcut Flow-Matching Generation

With the expert priors and their corresponding targets established above, we next develop an *expert prior-guided Shortcut flow matching* model to learn the transport between them. Conventional Shortcut formulations (Frans et al., 2025) typically bridge uninformative Gaussian noise and data distributions, which, in the context of motion planning, forces the generator to construct trajectories from scratch toward a uniform demonstration. By contrast, our adaptation rethinks this pipeline in three key dimensions: **(i)** *Source Reformulation*: rather than initializing from random noise, we deploy scene-conditioned expert priors τ_p^i as starting points, significantly shortening the transport path; **(ii)** *Destination Alignment*: instead of pulling all modes toward a single demonstration, we pair each prior with a geometry-compatible multi-expert target τ_e^i, preventing multimodal collapse while retaining distinct driving intents; and **(iii)** *Corrective Flow & Low Latency*: by redefining the velocity field as residual plan corrections over structured trajectories, our formulation allows Shortcut's step-size self-consistency to perform reliable, single-step plan correction at inference time.

Specifically, for each prior-target pair (τ_p^i, τ_e^i), we formulate the plan refinement task as learning a directional residual displacement $\Delta\tau_i = \tau_e^i - \tau_p^i$. The corresponding continuous interpolation path between them is defined as:

$$\tau_t^i = (1-t)\tau_p^i + t\tau_e^i = \tau_p^i + t\Delta\tau_i, \quad t \in [0, 1]. \tag{5}$$

Unlike conventional generative planners that must reconstruct global driving behaviors and road geometries from uninformative noise, the expert prior τ_p^i already grounds the generation on a kinematically viable and scene-adaptive behavioral mode. Consequently, the generative task reduces to learning a scene-conditioned *corrective velocity field* $v_\theta(\tau_t^i, t, d, \mathbf{c})$ that selectively rectifies the residual deficiencies of the preliminary plan while strictly safeguarding its underlying intent. For a finite step size $d \in (0, 1-t]$, the Shortcut state transition is parameterized as $\tau_{t+d}^i = \tau_t^i + d\, v_\theta(\tau_t^i, t, d, \mathbf{c})$.

To ensure that this corrective field remains reliable across diverse correction scales, the network is optimized with two synergistic objectives. First, at the infinitesimal step limit ($d = 0$), a flow-matching objective anchors the velocity field directly to the intended prior-to-target displacement:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{i,t} \left[\left\| v_\theta(\tau_t^i, t, 0, \mathbf{c}) - \Delta\tau_i \right\|_1 \right]. \tag{6}$$

While $\mathcal{L}_{\mathrm{FM}}$ guides the instantaneous transport direction, optimizing this local flow alone does not prevent trajectory drift or compounding discretization errors during large-step transport. To bridge microscopic flow with macroscopic jumps, we enforce the step-size self-consistency of Shortcut (Frans et al., 2025): for any finite step $d > 0$, a direct macroscopic update of length d must strictly match the recursive composition of two consecutive microscopic updates of length $d/2$:

$$\begin{aligned}
\tilde{\tau}^i_{t+d/2} &= \tau^i_t + \frac{d}{2} v_\theta \left(\tau^i_t, t, \frac{d}{2}, \mathbf{c} \right), \\
\bar{v}^{(d/2)}_i &= \frac{1}{2} \left[v_\theta \left(\tau^i_t, t, \frac{d}{2}, \mathbf{c} \right) + v_\theta \left(\tilde{\tau}^i_{t+d/2}, t + \frac{d}{2}, \frac{d}{2}, \mathbf{c} \right) \right], \\
\mathcal{L}_{\mathrm{SC}} &= \mathbb{E}_{i,t,d} \left[\left\| v_\theta(\tau^i_t, t, d, \mathbf{c}) - \mathrm{sg}\left(\bar{v}^{(d/2)}_i \right) \right\|_1 \right],
\end{aligned} \tag{7}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. Here, $\mathcal{L}_{\mathrm{FM}}$ dictates the prior-to-target correction direction, whereas $\mathcal{L}_{\mathrm{SC}}$ flattens the transport dynamics across arbitrary granularities. The overall generation objective is formulated as $\mathcal{L}_{\mathrm{gen}} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{SC}} \mathcal{L}_{\mathrm{SC}}$.

Single-Step Inference. At inference, this step size-consistent corrective field enables direct single-step refinement. Setting $t = 0$ and $d = 1$ gives $\hat{\tau}^i = \tau^i_p + v_\theta(\tau^i_p, 0, 1, \mathbf{c})$, yielding the refined candidate set $\widehat{\mathcal{T}} = \{\hat{\tau}^i\}_{i=1}^K$. Thus, Plan2Plan combines structured expert priors with Shortcut self-consistency to transform trajectory generation into efficient single-step plan-to-plan refinement.

3.5 Trajectory Scoring and Selection

Following DrivoR (Kirby et al., 2026) and CLOVER (Ang et al., 2026), we use a trajectory scorer for candidate selection. Each refined trajectory is detached and re-embedded as a scoring query:

$$\mathbf{e}_i = E_{\mathrm{traj}} \left(\mathrm{sg}(\hat{\tau}^i) \right), \qquad \widehat{\mathbf{Y}} = S_\phi \left(\{\mathbf{e}_i\}_{i=1}^K, \mathbf{c} \right), \qquad \hat{q}_i = \Gamma(\hat{\mathbf{y}}_i), \tag{8}$$

where $\widehat{\mathbf{Y}} = [\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_K]$ contains the predicted benchmark-specific planning sub-metrics for all candidates, and $\Gamma(\cdot)$ denotes the corresponding benchmark composition rule, i.e., PDMS for NAVSIM-v1 and EPDMS for NAVSIM-v2. $\hat{q}_i$ provides the base ranking for candidate selection; NAVSIM-v2 additionally applies the reference-guided selection described in Appendix A.1.

The scorer follows the component-wise metric supervision of DrivoR (Kirby et al., 2026), using evaluator-provided sub-metrics as training targets. During training, the trajectory generator and scorer are jointly optimized in each iteration through a shared scene representation. Following DrivoR, the generated trajectories are detached before entering the scoring branch, such that the scoring objective updates the scorer and shared scene encoder, while the detached trajectory representation prevents scoring gradients from modifying the generator parameters. Meanwhile, the generation objective jointly optimizes the trajectory generator and shared encoder. The overall training objective is formulated as $\mathcal{L} = \mathcal{L}_{\mathrm{gen}} + \lambda_{\mathrm{score}} \mathcal{L}_{\mathrm{score}}$, where λ_{score} controls the contribution of the trajectory-scoring objective.

4 Experiments

4.1 Experimental Settings

Implementation Details. Following DrivoR (Kirby et al., 2026), we adopt a DINOv2 ViT-S backbone with register-based scene representations. We use $K = 64$ preliminary trajectories generated by CLOVER (Ang et al., 2026) as expert priors, each represented by eight waypoints over a 4-second horizon. The expert pool and prior-target assignments are precomputed and fixed throughout training. The generator, DrivoR-style scorer, and shared scene encoder are jointly optimized with AdamW for 20 epochs, using learning rates of 10^{-4}, 2×10^{-4}, and 2×10^{-5}, respectively. At inference, each expert prior is refined with a single Shortcut step before scoring and selection.

Benchmarks and Metrics. We evaluate Plan2Plan on NAVSIM-v1 (Dauner et al., 2024), NAVSIM-v2 (Cao et al., 2025), and HUGSIM (Zhou et al., 2024). On NAVSIM-v1, we report PDMS on navtest. On NAVSIM-v2, we report EPDMS on navtest and the official navhard-two-stage split. On HUGSIM, we report road completion (RC) and HD-Score.

Table 1: Comparison with existing methods on the NAVSIM-v1 navtest split. Plan2Plan is trained on navtrain (85k) instead of trainval (103k). †CLOVER is reproduced by us under the same navtrain setting, as its original paper reports only the trainval result.

Method	Source	NC ↑	DAC ↑	TTC ↑	Comf. ↑	EP ↑	PDMS ↑
LTF (Chitta et al., 2023)	TPAMI'23	97.4	92.8	92.4	100.0	79.0	83.8
UniAD (Hu et al., 2023)	CVPR'23	97.8	91.9	92.9	100.0	78.8	83.4
VADv2 (Chen et al., 2024b)	arXiv'24	97.2	89.1	91.6	100.0	76.0	80.9
DiffusionDrive (Liao et al., 2025)	CVPR'25	98.2	96.2	94.7	100.0	82.2	88.1
GoalFlow (Xing et al., 2025)	CVPR'25	98.4	98.3	94.6	100.0	85.0	90.3
WoTE (Li et al., 2025b)	ICCV'25	98.5	96.8	94.9	99.9	81.9	88.3
DiffusionDriveV2 (Zou et al., 2025)	arXiv'25	98.3	97.9	94.8	99.9	87.5	91.2
DiffRefiner (Yin et al., 2026)	AAAI'26	98.6	98.4	95.8	100.0	84.5	90.7
DriveSuprim (Yao et al., 2026)	AAAI'26	98.6	98.6	95.5	100.0	91.3	93.5
DrivoR (Kirby et al., 2026)	CVPR'26	98.9	98.3	96.2	100.0	89.1	93.1
MeanFuser (Wang et al., 2026b)	CVPR'26	98.6	97.0	95.0	100.0	82.8	89.0
ResAD (Zheng et al., 2026)	CVPR'26	98.9	97.8	94.9	100.0	87.0	90.6
SparseDriveV2 (Sun et al., 2026)	ECCV'26	98.5	98.4	95.0	99.9	88.6	92.0
BeyondDrive (Wang et al., 2026a)	ECCV'26	98.4	97.9	95.0	100.0	83.7	89.7
CLOVER† (Ang et al., 2026)	arXiv'26	98.7	98.2	95.9	100.0	90.1	92.9
Plan2Plan (Ours)		98.8	98.4	95.9	100.0	**91.6**	**93.7**

Table 2: Comparison on NAVSIM-v2 navtest. EPDMS* uses the original evaluation before the human-behavior filtering fix, while EPDMS uses the updated official implementation. Higher is better; "–" denotes unreported results.

Method	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS* ↑	EPDMS↑
Ego-MLP (Cao et al., 2025)	93.1	77.9	92.7	99.6	86.0	91.5	89.4	98.3	85.4	64.0	–
Transfuser (Chitta et al., 2023)	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7	–
ARTEMIS (Feng et al., 2026b)	98.3	95.1	98.6	99.8	81.5	97.4	96.5	98.3	98.3	83.1	–
DiffusionDriveV2 (Zou et al., 2025)	97.7	96.6	99.2	99.8	88.9	97.2	96.0	97.8	91.0	85.5	87.5
Hydra-MDP++ (Li et al., 2025a)	98.4	98.0	99.4	99.8	87.5	97.7	95.3	98.3	77.4	85.1	–
DriveSuprim (Yao et al., 2026)	98.4	98.6	99.6	99.8	90.5	97.8	97.0	98.3	78.6	87.1	–
MeanFuser (Wang et al., 2026b)	98.3	97.2	99.6	99.8	87.6	97.4	97.3	98.3	88.2	–	89.5
SparseDriveV2 (Sun et al., 2026)	98.1	98.1	99.6	99.8	91.1	97.3	96.9	98.2	78.4	86.7	90.1
CLOVER (Ang et al., 2026)	99.4	99.3	99.5	99.8	86.9	98.9	95.2	98.2	75.5	87.2	90.4
Plan2Plan (Ours)	98.8	98.4	**99.7**	99.7	89.5	98.3	96.5	98.3	87.0	**87.8**	**91.4**

4.2 Comparison with Existing Methods

Evaluation on NAVSIM-v1. As shown in Table 1, Plan2Plan achieves 93.7 PDMS on NAVSIM-v1 navtest and surpasses all compared methods trained on the standard navtrain (85k) split. Compared with representative generative planners, Plan2Plan improves over GoalFlow (90.3) and DiffusionDriveV2 (91.2), while also outperforming strong trajectory-selection methods such as DrivoR (93.1) and DriveSuprim (93.5). Plan2Plan further achieves an EP of 91.6 while maintaining strong safety-related metrics, demonstrating balanced planning performance under the same setting.

Evaluation on NAVSIM-v2 navtest. As shown in Table 2, Plan2Plan achieves 87.8 EPDMS* and 91.4 EPDMS, surpassing all compared methods under both evaluation protocols. Under the corrected official implementation, Plan2Plan improves over CLOVER (90.4) and SparseDriveV2 (90.1) by 1.0 and 1.3 points, respectively. The consistent improvements under both evaluation protocols show that the performance advantage is preserved after the official metric correction.

Evaluation on NAVSIM-v2 navhard-two-stage. As shown in Table 3, Plan2Plan achieves 43.6 EPDMS with the adopted DrivoR-style scorer, outperforming GTRS-Dense (41.7) and GTRS-Aug (42.1). To disentangle candidate quality from the choice of scoring backend, we further evaluate the same Plan2Plan candidates using the public GTRS-Dense scorer (Li et al., 2025c) under the same reference-guided selection protocol. This yields 49.8 EPDMS, surpassing DrivoR (48.3) by 1.5 points. Since the candidate set remains unchanged, the improvement demonstrates that Plan2Plan provides high-quality candidates that can be further exploited by stronger scorers.

Closed-loop Evaluation on HUGSIM. We further evaluate the zero-shot closed-loop generalization of Plan2Plan on all 436 HUGSIM scenarios (Zhou et al., 2024), without HUGSIM-specific training. As shown in Table 4, Plan2Plan + GTRS-Dense achieves an HD-Score of 39.8 and an RC of 48.9. Under the same frozen GTRS-Dense scoring backend, Plan2Plan achieves 39.8 HD-Score, compared with 38.1 for DrivoR + GTRS-Dense and 35.0 for the standalone GTRS-Dense policy.

Table 3: Comparison on NAVSIM-v2 `navhard-two-stage`. Stage 1 and Stage 2 report the corresponding sub-metrics, and EPDMS denotes the final two-stage score.

Method	Stage	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
LTF (Chitta et al., 2023)	Stage 1	96.2	79.5	99.1	99.5	84.1	95.1	94.2	97.5	79.1	23.1
	Stage 2	77.7	70.2	84.2	98.0	85.1	75.6	45.4	95.7	75.9	
RAP-DINO (Feng et al., 2026a)	Stage 1	97.1	94.4	98.8	99.8	83.9	96.9	94.7	96.4	66.2	39.6
	Stage 2	83.2	83.9	87.4	98.0	86.9	80.4	52.3	95.2	52.4	
GTRS-Dense (Li et al., 2025c)	Stage 1	98.7	95.8	99.4	99.3	72.8	98.7	95.1	96.9	40.4	41.7
	Stage 2	91.4	89.2	94.4	98.8	69.5	90.1	54.6	94.1	49.7	
GTRS-Aug (Li et al., 2025c)	Stage 1	98.9	95.1	99.2	99.6	76.1	99.1	94.7	97.6	54.2	42.1
	Stage 2	87.9	88.8	89.6	98.8	80.3	86.0	53.5	97.1	56.1	
DrivoR (Kirby et al., 2026)	Stage 1	98.8	95.1	98.9	100.0	72.6	98.7	94.0	97.6	73.3	48.3
	Stage 2	90.2	88.4	91.9	98.6	70.0	88.0	50.1	98.5	76.2	
GuideFlow (Liu et al., 2026)	Stage 1	96.6	80.5	96.3	99.3	82.3	94.9	91.5	97.7	67.8	27.1
	Stage 2	87.3	76.7	88.8	99.2	84.3	85.1	49.7	93.1	44.5	
GuideFlow + GTRS-Dense (Liu et al., 2026)	Stage 1	98.8	95.5	99.1	99.5	76.0	99.1	94.4	97.5	52.4	43.0
	Stage 2	88.8	89.4	89.4	98.8	80.3	86.7	52.9	96.9	56.9	
Plan2Plan (Ours)	Stage 1	**99.1**	94.0	**99.6**	99.8	75.0	98.9	94.7	97.6	48.9	43.6
	Stage 2	**91.5**	86.1	92.5	98.7	71.5	89.4	53.2	97.6	51.2	
Plan2Plan + GTRS-Dense (Ours)	Stage 1	98.9	**98.0**	**99.6**	99.8	74.4	98.4	**96.7**	97.6	53.8	**49.8**
	Stage 2	91.3	**89.7**	94.0	**99.2**	71.3	**90.3**	54.3	97.7	56.3	

Table 4: Comparison on HUGSIM under zero-shot closed-loop evaluation. RC and HD-Score are reported across four difficulty levels and their overall averages; higher is better. E, M, H, and X denote Easy, Medium, Hard, and Extreme, respectively.

Method	RC					HD-Score				
	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
GTRS-Dense (Li et al., 2025c)	74.4	51.8	26.5	9.3	40.3	69.8	46.6	20.9	3.5	35.0
DrivoR (Kirby et al., 2026)	77.0	41.2	28.3	22.3	40.4	70.6	29.6	17.6	10.7	30.0
DrivoR + GTRS-Dense (Kirby et al., 2026)	79.9	58.8	26.5	10.9	44.3	75.0	52.9	19.8	3.9	38.1
Plan2Plan (Ours)	**83.8**	33.1	**32.4**	29.7	41.5	73.2	15.9	15.4	12.9	25.6
Plan2Plan + GTRS-Dense (Ours)	82.6	**62.3**	30.4	19.4	**48.9**	**76.3**	52.0	**21.7**	9.7	**39.8**

This shared-scorer comparison further indicates that the performance gain stems from the quality of the refined Plan2Plan candidate set rather than a specialized scoring module.

4.3 Ablation Studies and Analysis

Unless otherwise specified, all ablations use the same 10-epoch training budget and default configuration, and are evaluated on NAVSIM-v1 `navtest`. Mean and Oracle denote the average and best candidate PDMS per scene, respectively; Selected denotes the PDMS of the scorer-selected trajectory. Div. denotes trajectory diversity following DIVER (Song et al., 2025).

Effect of Trajectory Priors. We compare four 64-trajectory prior sets in Table 5: globally shared Anchor64, its Gaussian-perturbed variant Noisy Anchor64, and scene-conditioned DrivoR64 and CLOVER64. Scene-conditioned priors achieve higher Selected PDMS than static anchors. Gaussian perturbation increases Oracle PDMS from 82.4 to 92.3 but leaves Mean PDMS nearly unchanged (38.6 vs. 39.0), indicating that occasional strong candidates do not improve the candidate set.

Effect of Expert Target Assignment. Table 6 compares three target assignment strategies under the same CLOVER priors. Prior preservation improves both candidate quality and diversity over Single GT, while Multi-Expert Targets performs best overall, reaching 93.4 Selected PDMS and 0.30 diversity. This supports assigning weaker priors to compatible higher-quality targets. A qualitative comparison is provided in Appendix A.2.

Effect of Flow Matching and Shortcut Consistency. Table 7 compares direct residual regression, flow matching, and flow matching with Shortcut consistency using the same CLOVER priors, target assignments, and single-step inference. Flow matching improves Oracle and Selected PDMS from 99.2/92.7 to 99.6/93.1, while adding Shortcut consistency further increases them to 99.7/93.4. This supports the use of explicit prior-to-target transport with Shortcut-based one-step correction.

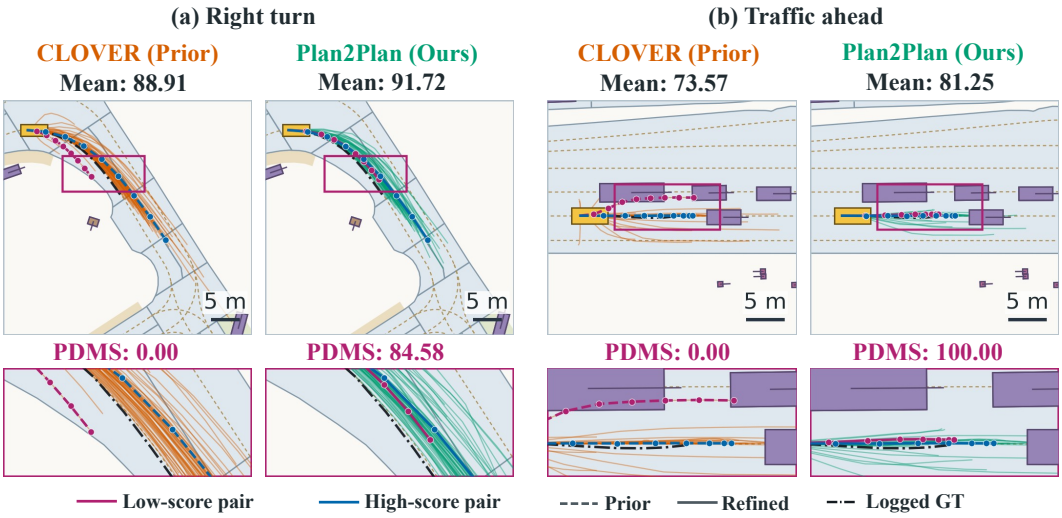


Figure 3: **Qualitative visualization of Plan2Plan generation.** CLOVER expert priors and their Plan2Plan outputs are shown in orange and green, respectively. Magenta and blue highlight low- and high-quality candidate pairs, while black dash-dotted curves denote the logged trajectories.

Table 5: Trajectory priors (NAVSIM-v1).

Source	PDMS ↑		
	Mean	Oracle	Selected
Anchor64	38.6	82.4	73.6
Noisy Anchor64	39.0	92.3	78.1
DrivoR64	79.4	98.6	90.6
CLOVER64	**79.7**	**99.7**	**93.4**

Table 6: Expert target assignment.

Target assignment	PDMS ↑			
	Mean	Oracle	Selected	Div. ↑
Single GT	81.0	94.3	88.0	0.10
GT + Prior Preservation	82.0	99.3	92.8	0.17
Multi-Expert Targets	**83.1**	**99.6**	**93.4**	**0.30**

Table 7: Generation objectives.

Training objective	PDMS ↑	
	Oracle	Selected
Direct residual regression	99.2	92.7
Flow matching	99.6	93.1
Flow matching + Shortcut	**99.7**	**93.4**

Table 8: Quality improvement through Plan2Plan.

Prior source	PDMS ↑			
	Mean	Oracle	Lower	Higher
DrivoR64	**79.5**→**79.6**	99.1→98.9	**55.8**→**56.5**	97.5→97.3
CLOVER64	**80.3**→**83.1**	99.7→99.6	**60.0**→**66.6**	98.2→97.6

Quality Improvement through Plan2Plan Generation. Table 8 reports the quality of DrivoR and CLOVER priors before and after refinement, grouped by their initial PDMS relative to the logged trajectory. Plan2Plan improves the Mean PDMS of CLOVER priors from 80.3 to 83.1 and raises the quality of its weaker priors from 60.0 to 66.6. The higher-quality group and Oracle PDMS remain largely stable, while DrivoR priors show the same but smaller trend. These results indicate that Plan2Plan primarily improves weaker priors while preserving already strong candidates.

Qualitative Analysis. Figure 3 visualizes Plan2Plan generation in representative driving scenarios. Starting from CLOVER expert priors, Plan2Plan corrects suboptimal candidates while leaving high-quality trajectories largely unchanged. Additional qualitative comparisons of different expert target assignments are provided in Appendix A.2.

5 Conclusion

Plan2Plan reframes candidate generation as plan-to-plan transport from scene-conditioned preliminary plans. Geometry-aware multi-expert supervision preserves strong priors and improves weaker ones through higher-quality targets, while expert prior-guided Shortcut flow matching enables single-step refinement. Experiments on NAVSIM-v1, NAVSIM-v2, and zero-shot closed-loop HUGSIM demonstrate strong performance, with ablations confirming improved candidate quality, especially for weaker priors, while largely preserving strong candidates and diversity.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and polishing of the manuscript, including improving grammar, clarity, concision, and readability, and refining wording for more precise academic presentation. All AI-assisted edits were reviewed by the authors to ensure that the intended technical meaning was preserved. The authors take full responsibility for the final content of the manuscript.

ETHICS STATEMENT

We acknowledge and adhere to the ICLR Code of Ethics throughout this work. This research studies autonomous driving planning using existing research benchmarks and does not involve human-subject experiments or the collection of new personally identifiable or sensitive personal data. The reported results are obtained under benchmark and simulation-based evaluation and should not be interpreted as evidence of readiness for real-world deployment; practical deployment would require additional safety validation and testing.

REPRODUCIBILITY STATEMENT

We have made efforts to facilitate the reproducibility of our work. Section 3 provides the formulation of expert-prior construction, multi-expert target assignment, Shortcut flow-matching generation, and trajectory scoring. Section 4.1 describes the implementation settings, benchmarks, evaluation metrics, and training configuration, while Section 4.3 provides controlled ablations of the main design choices. Appendix A.1 specifies the reference-guided candidate-selection rules and parameters used for NAVSIM-v2 evaluation.

REFERENCES

Sining Ang, Yuguang Yang, Canyu Chen, and Yan Wang. CLOVER: Closed-loop value estimation and ranking for end-to-end autonomous driving planning. *arXiv preprint arXiv:2605.15120*, 2026.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4709–4722, 2025.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024a.

Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024b.

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.

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 *NeurIPS*, pp. 28706–28719, 2024.

Lan Feng, Yang Gao, Éloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. RAP: 3d rasterization augmented end-to-end planning. In *ICLR*, 2026a.

Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. ARTEMIS: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *IEEE Robotics and Automation Letters*, 11(1): 226–233, 2026b.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *ICLR*, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pp. 6840–6851, 2020.

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 *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 *ICCV*, pp. 8340–8350, 2023.

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 *CVPR*, pp. 32058–32069, 2026.

Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M. Alvarez. Hydra-MDP++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025a.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via BEV world model. In *ICCV*, pp. 27137–27146, 2025b.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, Yu-Gang Jiang, and Jose M. Alvarez. Hydra-MDP: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M. Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025c.

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 *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 *ICLR*, 2023.

Lin Liu, Caiyan Jia, Guanyi Yu, Ziying Song, Junqiao Li, Feiyang Jia, Peiliang Wu, Xiaoshuai Hao, and Yadan Luo. GuideFlow: Constraint-guided flow matching for planning in end-to-end autonomous driving. In *CVPR*, pp. 3719–3728, 2026.

Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Breaking imitation bottlenecks: Reinforced diffusion powers diverse trajectory generation. *arXiv preprint arXiv:2507.04049*, 2025.

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 *ECCV*, pp. 446–463, 2026.

Bin Wang, Pingjun Li, Jinkun Liu, Jun Cheng, Hailong Lei, Yinze Rong, Huan ang Gao, Kangliang Chen, Xing Pan, and Weihao Gu. HMAD: Advancing E2E driving with anchored offset proposals and simulation-supervised multi-target scoring. *arXiv preprint arXiv:2505.23129*, 2025.

Junli Wang, Zhihua Hua, Xueyi Liu, Zebin Xing, Wei Zhang, Kun Ma, Guang Chen, Hangjun Ye, Long Chen, and Pengxuan Yang. Beyond imitation: Learning safe end-to-end autonomous driving from hard negatives. In *ECCV*, pp. 224–241, 2026a.

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 *CVPR*, pp. 17884–17893, 2026b.

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 *CVPR*, pp. 1602–1611, 2025.

Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-ES: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In *CVPR*, pp. 15342–15353, 2024.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M. Alvarez, and Zuxuan Wu. DriveSuprim: Towards precise trajectory selection for end-to-end planning. In *AAAI*, pp. 11910–11918, 2026.

Liuhan Yin, Runkun Ju, Guodong Guo, and Erkang Cheng. DiffRefiner: Coarse-to-fine trajectory planning via diffusion refinement with semantic interaction for end-to-end autonomous driving. In *AAAI*, pp. 12009–12017, 2026.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-based planning for autonomous driving with flexible guidance. In *ICLR*, 2025.

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 *CVPR*, 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. *arXiv preprint arXiv:2412.01718*, 2024.

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 Appendix

A.1 Reference-Guided Candidate Selection

Plan2Plan generates $K = 64$ candidate trajectories for each observation. For NAVSIM-v2, we use reference-guided selection in two settings: navtest with the native Plan2Plan scorer, and navhard-two-stage with the frozen GTRS-Dense scorer. The native Plan2Plan NAVHARD result uses its own scorer without reference guidance. We empirically set the reference-guided selection parameters on a small development subset drawn from the training data and keep them fixed for all reported evaluations. All settings select a final plan from the trajectories generated by Plan2Plan.

We use the 53rd generated candidate as the reference, $\tau_{\text{ref}} = \hat{\tau}^{53}$. Its branch index is fixed across scenes and planning steps, while its trajectory is generated anew from the current observation. Thus, the reference is a scene-dependent candidate rather than a fixed path. Selection uses the current candidate set and does not require the plan chosen at the preceding step.

For each candidate $\hat{\tau}^i$, we measure its geometric deviation from the reference position and heading:

$$d_i = 0.1\frac{d_{\text{XY}}(\hat{\tau}^i, \tau_{\text{ref}})}{5} + 0.25\frac{d_\theta(\hat{\tau}^i, \tau_{\text{ref}})}{0.35}. \tag{9}$$

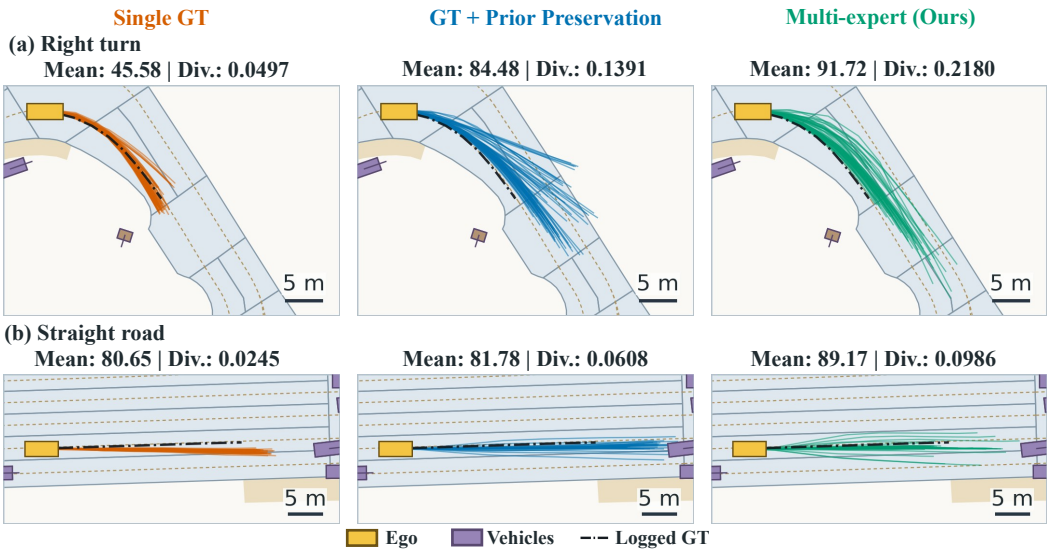


Figure 4: **Qualitative comparison of expert target assignment.** Candidate sets generated from identical CLOVER priors under Single GT, GT + Prior Preservation, and Multi-Expert Targets. Mean and Div. denote candidate quality and diversity, respectively; black dash-dotted curves denote the logged trajectories.

For two trajectories τ_a and τ_b with T waypoints, the distance components are

$$d_{\mathrm{XY}}(\tau_a, \tau_b) = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \|\mathbf{p}_{a,t} - \mathbf{p}_{b,t}\|_2^2},$$
$$d_{\theta}(\tau_a, \tau_b) = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \mathrm{wrap}\left(\psi_{a,t} - \psi_{b,t}\right)^2}. \tag{10}$$

Here $\mathbf{p}_t = (x_t, y_t)$ and $\mathrm{wrap}(\varphi) = \mathrm{atan2}(\sin\varphi, \cos\varphi)$ accounts for the periodicity of heading angles. Both selection rules use the same distance definition.

NAVSIM-v2 navtest. Let s_i denote the native Plan2Plan scorer's log-space selection score for candidate i. We incorporate the reference distance as a soft penalty:

$$i^* = \arg\max_{i \in \{1,\ldots,K\}} \left(s_i - \alpha d_i\right), \qquad \alpha = 1.25. \tag{11}$$

All K candidates remain eligible. A candidate farther from the reference can still be selected if its predicted score compensates for the larger geometric penalty.

NAVSIM-v2 navhard-two-stage. For the Plan2Plan + GTRS-Dense variant, the frozen GTRS-Dense scorer (Li et al., 2025c) provides a predicted log-score s_i for each Plan2Plan candidate. We first retain candidates whose scores are within δ of the highest predicted score, then select the candidate closest to the reference:

$$\mathcal{E}_\delta = \left\{ i \in \{1,\ldots,K\} \;\middle|\; s_i \geq \max_j s_j - \delta \right\}, \qquad \delta = 0.02,$$
$$i^* = \arg\min_{i \in \mathcal{E}_\delta} d_i. \tag{12}$$

If only one candidate satisfies the score criterion, it is selected directly. Otherwise, reference distance resolves the choice among candidates with similar predicted scores. Equal distances are resolved by the higher predicted score and then by the lower candidate index. The margin δ applies to predicted log-scores, rather than to ground-truth EPDMS.

This variant uses the original GTRS-Dense score composition. GTRS-Dense uses its trajectory vocabulary as scoring context, while the trajectory remains one of the 64 Plan2Plan candidates. Neither reference-guided rule uses ground-truth metrics to select a trajectory at inference time.

A.2 Additional Qualitative Analysis

Figure 4 compares three target assignment strategies in a right-turn scene and a straight-road scene, starting from identical CLOVER priors. Single GT concentrates the generated candidates around the logged trajectory, whereas GT + Prior Preservation produces a wider candidate spread. Multi-Expert Targets produces distinct trajectories while achieving the highest displayed candidate Mean PDMS and diversity in both scenes. These illustrative examples show the quality–diversity pattern quantified across NAVSIM-v1 navtest in Table 6.