PLANWAM: PLANNING-SHAPED FUTURE REPRESENTATIONS FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

World models in end-to-end autonomous driving predict future scene evolution to provide foresight for trajectory planning. Existing methods mainly study how to predict the future and how to use it, but less often ask which future representation is actually most useful for planning. To this end, we propose PlanWAM, a Planning-Shaped World Action Model. The key idea is to let the planning task shape the future-state representation, so that it retains the information most useful for planning. A latent world model then predicts this planning-shaped future latent representation from historical observations and uses it for planning, enabling foresighted planning. Specifically, we first use a Temporal Register Pyramid to compress multi-frame historical information in a recency-aware manner, learning a compact history representation oriented toward future reasoning and planning. We then introduce a privileged future posterior branch that observes ground-truth future frames, and shape its future latent representation with trajectory-planning objectives to obtain a planning-shaped future latent representation. Hindsight-to-Foresight Distillation trains a prior branch that depends only on history to predict this future latent representation. The predicted future latent representation serves as planning context and guides trajectory generation and selection. PlanWAM achieves 93.8 PDMS / 90.9 EPDMS on NAVSIM-v1/v2 navtest and reaches 38.7 HD-Score on closed-loop HUGSIM in a zero-shot setting, demonstrating leading planning performance across both open-loop and closed-loop evaluations. Extensive experiments further demonstrate that planning-shaped future representations provide an effective and deployable form of foresight for world-action models.

1 INTRODUCTION

End-to-end autonomous driving uses a unified model to plan future trajectories directly from raw sensor observations, so that scene representations can be learned for the final planning objective Codevilla et al. (2018); Hu et al. (2023); Jiang et al. (2023); Liao et al. (2025). Reliable driving, however, depends on more than the current observation. A planner must also use history to identify motion trends and interactions among traffic participants, and to infer possible future scene evolution. World models have therefore become an important research direction for end-to-end autonomous driving Wang et al. (2024a); Zheng et al. (2024); Zhou et al. (2025a). By modeling how the scene evolves over time and with vehicle behavior, they supply future information to the planner, shifting trajectory planning from reacting to the current scene toward foresighted planning.

Existing driving world models have gradually evolved in how they represent and use future information, moving from pixel reconstruction to latent prediction and then to future-conditioned planning (Fig. 1). Early methods, such as Epona Zhang et al. (2025) and PWM Zhao et al. (2025), learn environment dynamics by generating future images, videos, or dense scene states, but they often have to model substantial visual detail that is only weakly related to planning, at a high generation cost. Later methods, such as LAW Li et al. (2025a) and Latent-WAM Wang et al. (2026b), turn to latent world modeling and predict future states in a compact feature space. Their future representations mainly serve as prediction targets or auxiliary supervision, so the effect on planning remains indirect. Recent methods, such as DriveFuture Hong et al. (2026) and WorldDrive Gui et al. (2026), further introduce the predicted future latent explicitly into the trajectory planner, so that future information can participate in planning directly. Even so, most existing methods focus on how to predict the future and how to use it, and less often study a more basic question: *which future representation*

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.

is most useful for planning? Existing future representations are usually defined through pixel-level reconstruction or image latent prediction, such as DINO, JEPA, or BEV feature, and may therefore be mismatched with the downstream planning objective: a future representation that can reconstruct the scene does not necessarily retain the key information that affects trajectory decisions.

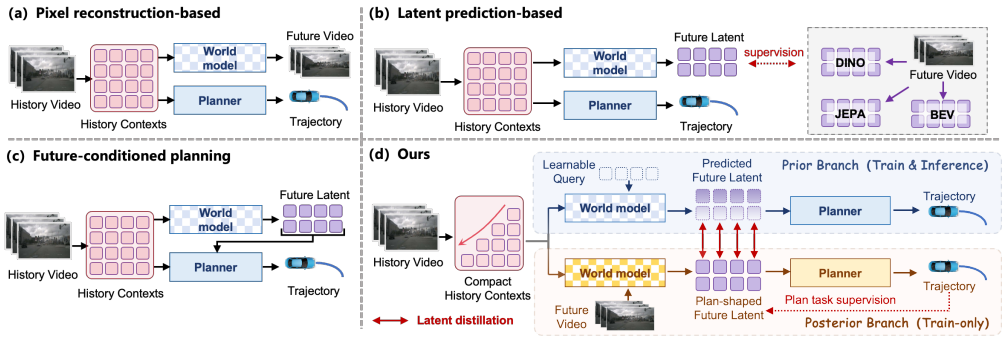

Figure 1: **Four future-modeling paradigms in driving World Action Models (WAMs).** (a) Pixel reconstruction. (b) Latent prediction. (c) Future-conditioned planning. (d) PlanWAM (ours).

We reformulate future modeling in end-to-end driving as a planning-oriented predictive representation learning problem: a world model must not only predict the future, but also learn which future information is worth predicting. For autonomous driving, an effective representation need not fully retain historical observations or reconstruct the future scene, but should focus on information relevant to trajectory planning. The history representation should capture dynamic cues from multi-frame, multi-view observations that are essential for future reasoning and planning, while the future representation should be shaped directly by trajectory planning objectives to prioritize decision-relevant future information. Furthermore, a future representation intended for planning should be predictable from historical observations and serve as foresight for trajectory planning. From this perspective, we unify historical modeling, future prediction, and trajectory planning into a planning-shaped predictive representation learning framework, where planning defines what future information matters, future prediction guides which historical information is retained, and the resulting future representation in turn improves planning.

Building on this perspective, we propose PlanWAM, a Planning-Shaped World Action Model, as illustrated in Fig. 1(d). First, we compress multi-frame, multi-view historical observations in a recency-aware manner and learn a compact history representation under joint future-prediction and planning supervision, preserving key dynamic cues for future reasoning. We then introduce a training-only privileged future posterior branch that leverages ground-truth future observations to construct a future latent representation, whose representation space is shaped by trajectory planning objectives to obtain a planning-shaped future latent representation. Since future observations are unavailable at deployment, Hindsight-to-Foresight Distillation trains a prior branch conditioned only on history to predict this representation, thereby converting privileged hindsight during training into deployable foresight at inference time. The predicted future representation is fused with the history representation as a foresight planning context to guide trajectory generation and selection, enabling Foresight-Conditioned Planning. Our main contributions are as follows:

- We reformulate future modeling in end-to-end driving as a planning-oriented predictive representation learning problem, where the future representation is shaped directly by trajectory planning objectives rather than predefined reconstruction or generic prediction objectives, so that it preferentially encodes decision-relevant future information.
- We propose PlanWAM, which extracts a compact history representation for future reasoning, learns planning-shaped future latents from ground-truth future observations and planning objectives, and uses Hindsight-to-Foresight Distillation to predict this representation from history as foresight for trajectory generation and selection.
- PlanWAM achieves 93.8 PDMS / 90.9 EPDMS on NAVSIM-v1/v2 navtest and 38.7 HD-Score in zero-shot closed-loop evaluation on HUGSIM. Further analyses show that planning-shaped future representations improve trajectory generation and selection while reducing safety-critical failures, including collisions and low-TTC events.

2 Related Work

2.1 End-to-End Autonomous Driving

End-to-end autonomous driving uses a unified model to plan future trajectories directly from raw sensor observations, thereby reducing information loss and cascading errors in traditional modular systems. Early methods, such as UniAD Hu et al. (2023), VAD Jiang et al. (2023), SparseDrive Sun et al. (2025), and TransFuser Chitta et al. (2023), used structured scene representations and multi-task learning to jointly model perception, prediction, and planning. Subsequent methods, such as iPad Guo et al. (2026), DrivoR Kirby et al. (2026), DiffusionDrive Liao et al. (2025), and GoalFlow Xing et al. (2025), adopt multimodal and generative trajectory planning to capture diverse and uncertain driving behaviors. More recent methods, such as ReCogDrive Li et al. (2026c), AutoVLA Zhou et al. (2025b), Orion Fu et al. (2025), and Qwen-Drive-1.0 Zhou et al. (2026b), introduce vision-language and vision-language-action models to strengthen semantic understanding and reasoning. Although these methods keep improving scene representation and trajectory planning, they still decide mainly from current or historical observations and make limited use of future scene evolution. In contrast, PlanWAM uses a world model to predict a future representation useful for planning, and guides trajectory generation and evaluation with that foresight.

2.2 World Models for Autonomous Driving

World models provide foresight for end-to-end driving by modeling how the scene evolves over time and under vehicle behavior. Existing driving world models fall into two lines. One line, such as DriveDreamer Wang et al. (2024a), Drive-WM Wang et al. (2024b), Vista Gao et al. (2024), Epona Zhang et al. (2025), and PWM Zhao et al. (2025), focuses on generating and simulating the future real world. These methods learn traffic evolution by predicting images, videos, or structured scene states such as BEV, and mainly study generation quality, temporal consistency, action controllability, and long-horizon prediction. The other line, such as LAW Li et al. (2025a), Latent-WAM Wang et al. (2026b), World4Drive Zheng et al. (2025), DriveFuture Hong et al. (2026), and WorldDrive Gui et al. (2026), focuses on the role of world modeling in the driving task itself, learning dynamic scene representations by predicting future features or latent states and further assisting or directly participating in trajectory planning. These studies show that future modeling can provide a useful dynamic prior for end-to-end driving. Existing work, however, mainly studies how to predict future features more effectively and how future information is used, and less often asks, from the planning objective itself, what the future representation should encode. In contrast, PlanWAM studies which future representation is most useful for planning and shapes the future latent representation directly with planning objectives, so that the latent preferentially extracts the key future information relevant to planning decisions.

3 Method

3.1 Problem Formulation and Framework Overview

Let $v_{t-n:t}$ denote the historical multi-view camera observations, $c_{t-n:t}$ the corresponding ego states and navigation information, and $\tau_{1:T}$ the future ego trajectory to be planned. A conventional end-to-end planner directly models:

$$p(\tau_{1:T} \mid v_{t-n:t}, c_{t-n:t}).$$

To use future scene evolution explicitly, future-conditioned WAMs (Fig. 1(c)) introduce a future-state representation z_t^+ between historical observations and trajectory planning, and factorize the process as:

$$p(\tau_{1:T}, z_t^+ \mid v_{t-n:t}, c_{t-n:t}) = p(z_t^+ \mid v_{t-n:t}, c_{t-n:t})\, p(\tau_{1:T} \mid v_{t-n:t}, c_{t-n:t}, z_t^+). \tag{1}$$

This factorization lets future information participate in planning, but it also raises a key question: *how should z_t^+ be defined and constructed so that it is more beneficial for trajectory planning?* Existing methods typically define the future representation by pixel reconstruction or image latent prediction, such as DINO, JEPA, or BEV feature, but such a representation is not necessarily the most useful for planning. To this end, we propose PlanWAM, as illustrated in Fig. 2.

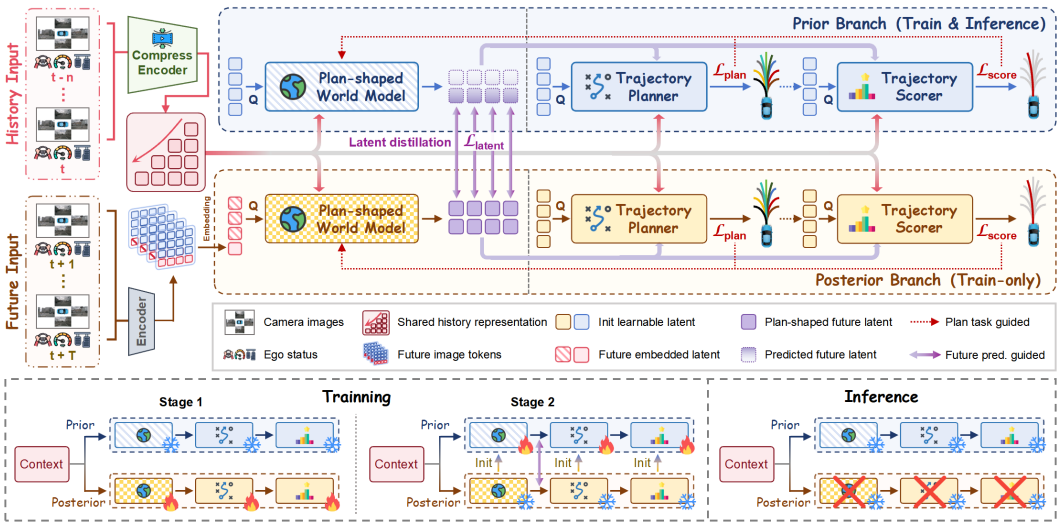

Figure 2: **Overview of PlanWAM**. Historical multi-view observations are compressed into a compact representation for future reasoning. The privileged posterior branch uses ground-truth future observations and planning objectives to learn planning-shaped future latents, which are distilled into a prior branch conditioned only on history. The predicted future latents then serve as foresight for trajectory generation and selection. The posterior is training-only; inference uses only the prior.

We first use a history compression module E_η to compress multi-frame, multi-view historical observations in a recency-aware manner, yielding a compact history representation:

$$h_t = E_\eta(v_{t-n:t}, c_{t-n:t}). \tag{2}$$

During training, we further introduce a privileged posterior branch P_ϕ, hereafter referred to as the posterior, which accesses ground-truth future observations and constructs a future latent representation:

$$z_t^+ = P_\phi\big(h_t,\, v_{t+1:t+T},\, c_{t+1:t+T}\big). \tag{3}$$

Rather than supervising z_t^+ with pixel-level reconstruction or image-latent prediction objectives, we directly condition trajectory generation and selection on z_t^+:

$$\tau_{1:T} \sim \pi_\phi(h_t, z_t^+). \tag{4}$$

The future latent space is therefore shaped jointly by the trajectory generation and trajectory selection, so that z_t^+ preferentially encodes future information that affects planning. We refer to this representation as a *planning-shaped future representation*.

However, the future observations $v_{t+1:t+T}$ and $c_{t+1:t+T}$ are unavailable at deployment. PlanWAM therefore learns a prior branch P_θ, hereafter referred to as the prior, which depends only on the history representation and is trained by Hindsight-to-Foresight Distillation to predict z_t^+:

$$\hat{z}_t^+ = P_\theta(h_t), \qquad \hat{z}_t^+ \approx z_t^+. \tag{5}$$

The predicted planning-shaped future representation $\hat{z}_t^+$ is fused with the compact history representation h_t as a foresight planning context to guide trajectory generation and selection:

$$\tau_{1:T} \sim \pi_\theta(h_t, \hat{z}_t^+). \tag{6}$$

At inference, the planning process of PlanWAM can be expressed as:

$$p(\tau_{1:T}, \hat{z}_t^+ \mid v_{t-n:t}, c_{t-n:t}) = p_\theta(\hat{z}_t^+ \mid h_t)\, p_\theta(\tau_{1:T} \mid h_t, \hat{z}_t^+). \tag{7}$$

3.2 Recency-Aware History Representation

Multi-frame, multi-view historical observations contain substantial temporal redundancy, and information from different temporal ranges has different value for planning. To this end, we propose a

Temporal Register Pyramid that compresses historical scene registers level by level in a recency-aware manner, retaining, under a limited token budget, the dynamic information relevant to future reasoning and trajectory planning, as illustrated in Fig. 3.

Given the historical observations $v_{t-n:t}$ and ego context $c_{t-n:t}$, a shared visual encoder maps the k-th camera observation v_i^k at time i to visual tokens and further extracts scene registers r_i^k; the ego state and navigation information c_i are encoded by an MLP into an ego feature e_i:

$$x_i^k = F_{\text{vis}}(v_i^k), \qquad r_i^k = F_{\text{reg}}(x_i^k), \qquad e_i = F_{\text{ego}}(c_i). \tag{8}$$

We partition the history into L temporal levels $\{\mathcal{G}_l\}_{l=1}^{L}$ according to temporal distance, and assign different query budgets: recent levels use more learnable queries, whereas more distant history is compressed more strongly. For the l-th level, the queries $Q^{(l)}$ are fused with the ego feature as well as time, camera, and compression-level embeddings; the scene registers at the current time are aggregated into a shared current-scene context $\bar{r}_t$:

$$\tilde{Q}^{(l)} = Q^{(l)} + E_{\text{time}}^{(l)} + E_{\text{cam}}^{(l)} + E_{\text{level}}^{(l)} + E_{\text{ego}}^{(l)} + \bar{r}_t. \tag{9}$$

We then apply a Q-Former (Li et al., 2023) to perform compression of the historical information, using $\tilde{Q}^{(l)}$ as Q and the scene registers $R^{(l)}$ of the corresponding level as K and V:

$$h_t^{(l)} = \text{QFormer}\big(Q = \tilde{Q}^{(l)},\, K = R^{(l)},\, V = R^{(l)}\big), \qquad R^{(l)} = \text{Concat}_{i\in\mathcal{G}_l,\,k}\, r_i^k. \tag{10}$$

The compressed tokens from all levels constitute the compact history representation h_t. We do not use an additional historical reconstruction objective; instead, h_t is jointly optimized by the subsequent future-latent distillation objective $\mathcal{L}_{\text{latent}}$ and the trajectory-planning objective $\mathcal{L}_{\text{task}}$, so that it focuses on the information required for future prediction and trajectory planning.

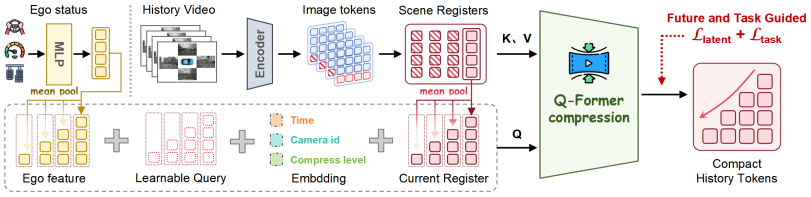

Figure 3: **Temporal Register Pyramid for Recency-Aware History Representation.**

3.3 Planning-Shaped Future Representation

Given the compact history representation h_t from Sec. 3.2, we construct a privileged posterior during training using future observations. For the future observations $v_{t+1:t+T}$, a shared visual encoder extracts scene registers r_i^k for each time and camera in the same way as for historical observations, and adds the corresponding future-time embedding $E_{\text{time}}^{\text{fut}}$ to form future queries:

$$f_t^+ = \text{Concat}_{i,k}\big[r_i^k + E_{\text{time}}^{\text{fut}}(i-t)\big], \qquad i \in \{t+1, \ldots, t+T\}. \tag{11}$$

Rather than using randomly initialized learnable queries, the privileged posterior P_ϕ takes f_t^+ formed from ground-truth future scene registers directly as queries, and uses the history representation h_t as context. The posterior world model $\text{WM}_\phi^{\text{post}}$ aggregates historical evidence and ground-truth future information to obtain the posterior future latent z_t^+:

$$z_t^+ = P_\phi(h_t, f_t^+) = \text{WM}_\phi^{\text{post}}\big(Q = f_t^+,\, K = h_t,\, V = h_t\big). \tag{12}$$

We do not apply an additional generic reconstruction objective to the future latent z_t^+. Instead, we concatenate z_t^+ with the history representation h_t to form a posterior planning context m_t^+, adding type embeddings before concatenation to distinguish historical evidence from future context. The resulting m_t^+ is used for both trajectory planner and trajectory scorer:

$$m_t^+ = \text{Concat}\big(h_t + E_{\text{type}}^{\text{hist}},\, z_t^+ + E_{\text{type}}^{\text{fut}}\big), \quad \tau_{1:T}^+ = G_{\text{plan}}(m_t^+), \quad s^+ = G_{\text{score}}(\tau_{1:T}^+,\, m_t^+). \tag{13}$$

The posterior is supervised by the trajectory generation and trajectory selection objectives:

$$\mathcal{L}_{\text{post}} = \mathcal{L}_{\text{plan}} + \mathcal{L}_{\text{score}}. \tag{14}$$

The planning losses can therefore back-propagate through the planner and the scorer to shape z_t^+ directly. The limited future latent capacity thus preferentially retains future scene information that affects trajectory generation and selection. We refer to this z_t^+ as a *planning-shaped future representation*. The next section uses it as a privileged target to train a history-only prior.

3.4 HINDSIGHT-TO-FORESIGHT PREDICTION AND PLANNING

The planning-shaped future representation z_t^+ in Sec. 3.3 relies on future observations and is therefore unavailable at deployment. We thus construct a prior P_θ conditioned solely on the history representation h_t to predict it. Unlike the posterior, which directly uses ground-truth future scene registers as queries, the prior uses learnable future queries Q^{prior}. To establish a consistent token-wise correspondence between the prior and posterior latents, we arrange the prior queries in the same order over future times, camera views, and register slots, and augment them with the corresponding structured embeddings:

$$Q^{\text{prior}}_{i,k,j} = Q^{\text{prior}}_{0,i,k,j} + E^{\text{fut}}_{\text{time}}(i-t) + E_{\text{cam}}(k) + E_{\text{slot}}(j), \qquad i \in \{t+1, \dots, t+T\}. \tag{15}$$

where i, k, and j index the future time, camera view, and scene-register slot, respectively. The prior world model $\text{WM}^{\text{prior}}_\theta$ then predicts the planning-shaped future representation $\hat{z}_t^+$ from h_t alone:

$$\hat{z}_t^+ = P_\theta(h_t) = \text{WM}^{\text{prior}}_\theta\big(Q = Q^{\text{prior}},\, K = h_t,\, V = h_t\big). \tag{16}$$

To transfer the future information captured by the privileged posterior to the prior, we propose Hindsight-to-Foresight Distillation, which directly aligns the prior prediction $\hat{z}_t^+$ with the posterior target z_t^+. Through this alignment, the prior learns to infer from historical dynamics the planning-shaped representation that the posterior derives from ground-truth future observations, thereby distilling privileged hindsight into foresight:

$$\mathcal{L}_{\text{latent}} = \frac{1}{N_z d}\big\|\hat{z}_t^+ - \text{sg}(z_t^+)\big\|_F^2. \tag{17}$$

where $\text{sg}(\cdot)$ denotes stop-gradient, $\|\cdot\|_F$ denotes the Frobenius norm, and N_z and d denote the number of tokens in the future latent and the feature dimension of each token, respectively. After adding the corresponding type embeddings, we concatenate the predicted $\hat{z}_t^+$ with the history representation h_t to form a prior planning context. This context is provided to both the trajectory planner and scorer, yielding the following Foresight-Conditioned Planning formulation:

$$m_t^- = \text{Concat}\big(h_t + E^{\text{hist}}_{\text{type}},\, \hat{z}_t^+ + E^{\text{fut}}_{\text{type}}\big), \quad \tau_{1:T} = G_{\text{plan}}(m_t^-), \quad s = G_{\text{score}}(\tau_{1:T},\, m_t^-). \tag{18}$$

3.5 TRAINING AND INFERENCE

Training. PlanWAM is trained in two stages. The first stage optimizes only the privileged posterior, which learns the planning-shaped future representation from ground-truth future observations under trajectory generation, trajectory selection, and lightweight regularization objectives:

$$\mathcal{L}_{\text{task}} = \mathcal{L}_{\text{plan}} + \mathcal{L}_{\text{score}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \tag{19}$$

where $\mathcal{L}_{\text{reg}}$ includes diversity regularization to mitigate latent anti-collapse. In the second stage, we freeze the posterior and use its checkpoint to initialize structurally compatible parameters in the prior. We then fine-tune the prior with a lower learning rate using the objective:

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{task}} + \lambda_{\text{latent}}\mathcal{L}_{\text{latent}}. \tag{20}$$

Inference. At inference, we remove the privileged posterior and retain only the prior. Historical observations are first encoded into h_t, after which the prior predicts the planning-shaped future representation $\hat{z}_t^+$. This representation serves as foresight and is combined with h_t for trajectory planner and scorer. Consequently, the entire inference process relies solely on observable history.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and benchmarks. We evaluate PlanWAM on NAVSIM-v1 Dauner et al. (2024), NAVSIM-v2 Cao et al. (2025), and HUGSIM Zhou et al. (2026a) to assess both planning performance and closed-loop generalization. Models are trained on NAVSIM `navtrain` and evaluated on `navtest`. NAVSIM-v1 follows the official evaluation protocol and uses PDM Score (PDMS) as the primary metric; NAVSIM-v2 uses the Extended PDM Score (EPDMS). We further perform zero-shot closed-loop evaluation on the high-fidelity closed-loop simulator HUGSIM, deploying the trained model directly without HUGSIM-specific fine-tuning, and report HUGSIM Driving Score (HD-Score) and Route Completion (RC) to assess closed-loop driving performance and cross-environment generalization.

Implementation details. PlanWAM takes $T{=}4$ historical frames from four cameras (front, left, right, and rear), while the training-only posterior observes future multi-view frames at 1-s intervals over a 4-s horizon. We use a DINOv2 ViT-S backbone with frozen base weights and rank-32 LoRA Hu et al. (2022), extracting 16 scene registers per camera-frame. The Temporal Register Pyramid assigns $[4, 4, 8, 16]$ queries per camera across four temporal levels, producing a 128-token history representation h_t. Four future times, four cameras, and 16 registers per camera yield $N_z{=}256$ posterior future-latent tokens. Over the same horizon, the trajectory planner generates $K{=}64$ candidate trajectories of 8 waypoints, and the trajectory scorer ranks them for final selection. Both stages use AdamW Loshchilov & Hutter (2019) and are trained on 16 NVIDIA A800 GPUs with a per-GPU batch size of 8. The learning rates for Stages 1 and 2 are 1×10^{-4} and 1×10^{-5}, respectively, and the latent-alignment weight is set to $\lambda_{\text{latent}}{=}0.05$. For more details, please refer to the supplementary materials.

4.2 Main Results

Open-loop Planning on NAVSIM. Tables 1 and 2 report the planning performance on the NAVSIM-v1/v2 `navtest` benchmarks. PlanWAM achieves 93.8 PDMS on NAVSIM-v1, outperforming all learned methods compared and improving the DrivoR baseline by 0.7 points; compared with the strongest prior world-model-based planner, DriveFuture, the gain reaches 3.1 points. On the more challenging NAVSIM-v2 benchmark, PlanWAM further achieves 90.9 EPDMS, exceeding the previous strongest learned result by 1.0 point and the reported Human Agent baseline by 0.6 points. PlanWAM also obtains the best learned NC of 99.0 and TTC of 98.5, with stable gains on safety-related planning metrics. These results show that planning-shaped future representations provide effective foresight for trajectory planning on both NAVSIM benchmarks.

Table 1: **Performance on the NAVSIM-v2 `navtest` benchmark.** The best learned result in each column is shown in **bold**.

Method	Reference	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
Human Agent	-	100.0	100.0	99.8	100.0	87.4	100.0	100.0	98.1	90.1	90.3
E2E-based planners											
TransFuser Chitta et al. (2023)	TPAMI'23	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
DiffusionDrive Liao et al. (2025)	CVPR'25	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
MeanFuser Wang et al. (2026a)	CVPR'26	98.3	97.2	99.6	99.8	87.6	97.4	97.3	98.3	**88.2**	89.5
VLM-based planners											
ReCogDrive Li et al. (2026c)	ICLR'26	98.3	95.2	99.5	99.8	87.1	97.5	96.6	98.3	86.5	83.6
SGDrive Li et al. (2026a)	CVPR'26	98.6	94.3	99.5	**99.9**	86.0	97.9	96.1	98.3	85.9	86.2
ExploreVLA Sheng et al. (2026)	ECCV'26	98.8	96.2	99.6	99.8	87.1	98.2	**97.8**	98.3	86.8	88.8
World-model-based planners											
DriveVLA-W0 Li et al. (2026b)	ICLR'26	98.5	99.1	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1
DriveLaW Xia et al. (2026)	CVPR'26	98.7	96.9	99.6	99.8	87.5	98.3	97.6	**98.4**	77.4	88.6
DreamerAD Yang et al. (2026)	ECCV'26	98.0	97.2	99.5	99.8	87.8	97.4	97.5	98.3	72.4	87.7
Latent-WAM Wang et al. (2026b)	arXiv'26	98.1	97.3	99.6	99.8	87.7	97.3	97.6	98.1	87.3	89.3
DriveFuture Hong et al. (2026)	arXiv'26	98.8	99.1	99.6	**99.9**	86.6	98.4	96.4	98.3	74.8	89.9
PlanWAM (**ours**)	-	**99.0**	**99.2**	**99.8**	99.6	**90.8**	**98.5**	96.5	98.1	70.5	**90.9**

Table 2: **Performance on the NAVSIM-v1 `navtest` benchmark.**

Method	NC↑	DAC↑	TTC↑	C↑	EP↑	PDMS↑
Human Agent	100	100	100	99.9	87.5	94.8
E2E-based planners						
DiffusionDrive Liao et al. (2025)	98.2	96.2	94.7	100	82.2	88.1
MeanFuser Wang et al. (2026a)	98.6	97.0	95.0	100	82.8	89.0
iPad Guo et al. (2026)	98.6	98.3	94.9	100	88.0	91.7
DrivoR Kirby et al. (2026)	98.9	98.3	96.2	100	89.1	93.1
VLM-based planners						
AutoVLA Zhou et al. (2025b)	98.4	95.6	**98.0**	99.9	81.9	89.1
ReCogDrive Li et al. (2026c)	97.9	97.3	94.9	100	87.3	90.8
ExploreVLA Sheng et al. (2026)	98.8	98.4	96.5	99.9	83.5	90.4
SGDrive Li et al. (2026a)	98.6	97.8	96.2	100	85.8	91.1
World-model-based planners						
WoTE Li et al. (2025b)	98.5	96.8	94.9	99.9	81.9	88.3
Epona Zhang et al. (2025)	97.9	95.1	93.8	99.9	80.4	86.2
DreamerAD Yang et al. (2026)	98.0	97.2	94.3	100	83.1	88.7
DriveLaW Xia et al. (2026)	**99.0**	97.1	96.7	100	81.3	89.1
DriveVLA-W0 Li et al. (2026b)	98.7	**99.1**	95.3	99.3	83.3	90.2
DriveFuture Hong et al. (2026)	98.8	**99.1**	95.4	100	84.2	90.7
PlanWAM (**ours**)	98.9	98.8	96.3	100	**91.0**	**93.8**

Table 3: **Closed-loop comparison on HUGSIM.** E, M, H, and X denote Easy, Medium, Hard, and Extreme.

Method	E	M	H	X	Avg.
UniAD Hu et al. (2023)	58.6	41.2	40.4	26.0	40.6
VAD Jiang et al. (2023)	38.7	27.0	25.5	23.0	27.9
LTF Chitta et al. (2023)	68.4	40.7	36.9	25.5	41.4
GTRS-Dense Li et al. (2025c)	64.2	**50.0**	20.7	22.3	38.0
Latent-WAM Wang et al. (2026b)	84.2	42.5	30.6	35.5	45.9
PlanWAM (**ours**)	**87.6**	48.3	**51.9**	**37.2**	**53.7**

(a) Route Completion

Method	E	M	H	X	Avg.
UniAD Hu et al. (2023)	48.7	29.5	27.3	14.3	28.9
VAD Jiang et al. (2023)	24.3	9.9	10.4	8.2	12.3
LTF Chitta et al. (2023)	52.8	24.6	19.8	8.1	24.8
GTRS-Dense Li et al. (2025c)	55.5	**39.0**	11.7	14.3	28.6
Latent-WAM Wang et al. (2026b)	72.5	24.0	12.2	18.1	28.9
PlanWAM (**ours**)	**75.3**	32.9	**35.6**	**22.0**	**38.7**

(b) HD-Score

Zero-shot Closed-loop Generalization on HUGSIM. Table 3 reports zero-shot closed-loop results on HUGSIM without simulator-specific fine-tuning. PlanWAM achieves 53.7 average RC and 38.7 HD-Score, outperforming the strongest baselines by 7.8 and 9.8 points, respectively. The gains remain pronounced on the Hard / Extreme subsets, where PlanWAM reaches 51.9 / 37.2 RC and 35.6 / 22.0 HD-Score, respectively. These results demonstrate strong zero-shot closed-loop generalization across varying scenario difficulty.

Qualitative Planning Results. Fig. 4 presents representative qualitative results on NAVSIM and HUGSIM. On NAVSIM, compared with DrivoR, PlanWAM produces trajectories that better account for surrounding vehicles and avoid unsafe interactions in challenging traffic scenarios. On HUGSIM, the model transfers zero-shot across KITTI-360, PandaSet, nuScenes, and Waymo, while maintaining stable closed-loop planning in diverse traffic conditions. These examples qualitatively support the consistent gains observed in the quantitative evaluation.

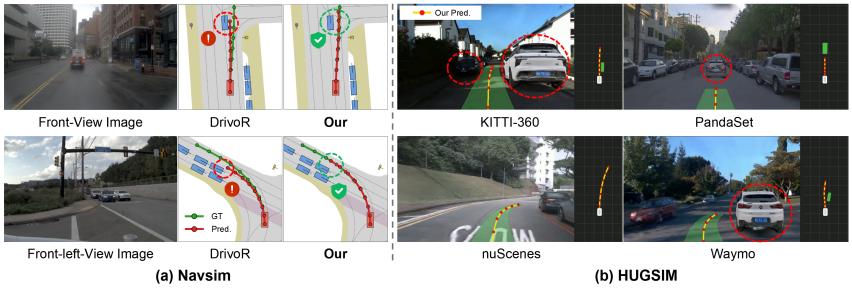

Figure 4: **Qualitative planning results.** (a) Comparison with DrivoR on representative NAVSIM scenarios. (b) Zero-shot closed-loop rollouts on HUGSIM across different driving domains.

4.3 Ablation Studies

We conduct systematic ablations on NAVSIM-v1 `navtest` to analyze the role of each PlanWAM component. Every run follows the full training procedure and changes only the module under study.

Overall Component Evaluation. We add the three core components in sequence, the Temporal Register Pyramid (TRP), Hindsight-to-Foresight Distillation (H2F), and Foresight-Conditioned Planning (FCP), as reported in Table 4. From the base model, TRP raises PDMS from 93.1 to 93.4, showing that a recency-aware history representation extracts more useful dynamics from multi-frame observations. Adding H2F further reaches 93.5, so learning a future latent from privileged future information supplies additional foresight. FCP then uses the predicted future latent as planning context for trajectory generation and selection, and the model reaches 93.8 PDMS.

Table 4: **Overall component ablation of PlanWAM.**

ID	TRP	H2F	FCP	NC↑	DAC↑	TTC↑	C↑	EP↑	PDMS↑
0	×	×	×	98.3	**98.9**	94.6	**100**	90.5	93.1
1	✓	×	×	98.8	98.8	95.3	**100**	90.9	93.4
2	✓	✓	×	98.6	98.6	95.5	**100**	**91.5**	93.5
3	✓	✓	✓	**98.9**	98.8	**96.3**	**100**	91.0	**93.8**

Future Representation Learning and Utilization. Table 5 shows that generic future-feature supervision reaches only 93.5 PDMS, so a generic prediction target does not by itself yield planning-relevant information. Shaping the future latent directly with the trajectory planning objectives raises performance to 93.8 PDMS. Table 6 then varies where the predicted latent is used. Auxiliary supervision or conditioning only trajectory generation brings little gain, whereas using it for trajectory selection reaches 93.7 PDMS. Conditioning both generation and selection attains 93.8 PDMS.

History Representation Compression and Future Distillation. Table 7 shows that, under the same 128-token budget, the recency-aware allocation $[4, 4, 8, 16]$ reaches 93.8 PDMS, above both the uniform allocation and the reversed allocation. Raising the token count to 256 yields only 93.6 PDMS, so how the queries are allocated matters more than enlarging the representation. Table 8

shows that a prior alone has a latent similarity of 0.22. Adding latent distillation raises it to 0.92 and PDMS to 93.7. With posterior initialization, similarity reaches 0.93 and PDMS 93.8.

Table 5: **Ablation of Future Representation.**

Future representation	Supervision	PDMS↑
Null future	–	93.4
Generic future latent	$\mathcal{L}_{\text{feature}}$	93.5
Planning-shaped latent (ours)	$\mathcal{L}_{\text{plan}} + \mathcal{L}_{\text{score}}$	**93.8**

$\mathcal{L}_{\text{feature}}$: *reconstruction of future scene registers.*

Table 6: **Ablation of Future Latent Usage.**

Future latent usage	Generation	Selection	PDMS↑
Auxiliary supervision only	×	×	93.5
Generation only	✓	×	93.5
Selection only	×	✓	93.7
Generation + Selection	✓	✓	**93.8**

Table 7: **Ablation of Temporal Allocation.**

Query allocation	Tokens	Latency (ms)↓	PDMS↑
[16, 16, 16, 16]	**256**	380	93.6
[8, 8, 8, 8]	128	343	93.4
[16, 8, 4, 4]	128	346	92.9
[4, 4, 8, 16]	128	342	**93.8**
[2, 4, 8, 16]	120	**335**	93.6

Inference latency on one NVIDIA A800, batch size 1, fp32.

Table 8: **Ablation of future latent distillation.**

Setting	Posterior init.	$\mathcal{L}_{\text{latent}}$	Latent sim.↑	PDMS↑
Prior-only	×	×	0.22	93.4
+ Posterior init.	✓	×	0.46	93.5
+ H2F distillation	×	✓	0.92	93.7
Init. + H2F (ours)	✓	✓	**0.93**	93.8
Privileged posterior (oracle)	–	–	–	**95.0**

Latent sim.: mean cosine similarity between $\hat{z}_t^+$ *and* z_t^+.

4.4 Analysis of Failure Reduction

To further understand how planning-shaped foresight improves planning performance, we analyze the PDMS$=0$ failure cases on NAVSIM-v1 `navtest`. As shown in Fig. 5(a), safety-related failures decrease markedly after introducing PlanWAM (ID 0 to ID 3). In particular, No at-fault Collision (NC) and Time-to-Collision (TTC) failures drop by 62.3% and 52.4%, indicating that the planning-shaped future representation helps the model capture the motion trends of future traffic participants and thereby lowers potential collision risk. Driving Direction Compliance (DDC) failures also drop by 66.7%, further showing that future information improves the modeling of dynamic driving constraints. In contrast, Drivable Area Compliance (DAC) failures show a slight deterioration, indicating that the gains of PlanWAM concentrate on dynamic scene understanding and decision making. As shown in Fig. 5(b), the number of failures falls from 314 to 237, mainly from fewer trajectory select errors (-87). The predicted future latent thus provides effective foresight and improves the reliability of candidate-trajectory evaluation and selection. Proposal quality also improves to some extent, while proposal-collapse issues remain. This analysis indicates that, by learning a planning-shaped future representation, PlanWAM provides effective foresight for trajectory evaluation and selection, thereby reducing planning failures caused by insufficient modeling of future evolution.

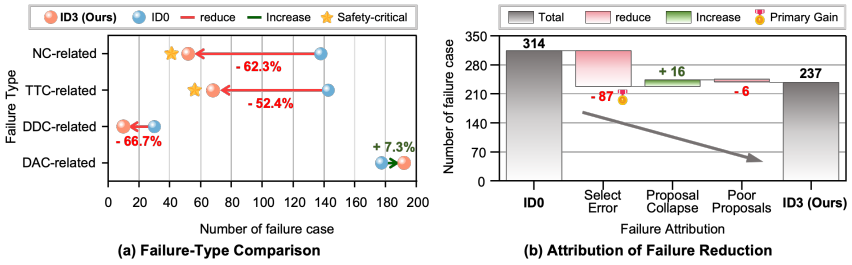

Figure 5: **Analysis of failure reduction enabled by planning-shaped foresight.** (a) Failure Type Distribution. (b) Attribution of Failure Reduction.

5 Conclusion

We presented PlanWAM, a planning-shaped world action model that reformulates future modeling in end-to-end autonomous driving as planning-oriented predictive representation learning. By shaping future latent representations with trajectory planning objectives and distilling them from privileged future observations to a history-only prior, PlanWAM learns deployable foresight for trajectory generation and selection. PlanWAM achieves 93.8 PDMS / 90.9 EPDMS on NAVSIM-v1/v2 navtest and 38.7 HD-Score in zero-shot closed-loop HUGSIM evaluation, demonstrating the effectiveness of planning-shaped future representations for end-to-end autonomous driving.

REPRODUCIBILITY STATEMENT

The method of PlanWAM is specified in Sec. 3, including the Temporal Register Pyramid, the planning-shaped future posterior, Hindsight-to-Foresight Distillation, and the two-stage training and history-only inference procedure. Dataset splits, evaluation protocols, and the main implementation settings are given in Sec. 4.1. Appendix A records the architecture and data flow; Appendix B the training objectives, loss weights, and optimization schedule; Appendix C the PDMS, EPDMS, and HUGSIM scoring protocols used for the reported numbers; Appendix D the additional controlled ablations; and Appendix E further qualitative results. All experiments use the public NAVSIM-v1, NAVSIM-v2, and HUGSIM benchmarks. An anonymized repository containing a project overview, visualization videos, supplementary material, and source code is available at `https://anonymous.4open.science/w/PlanWAM-4BF4/`.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing. This included translation from Chinese notes into English, edits for clarity, grammar, and readability, and formatting of tables and LaTeX. Additionally, we used generative AI tools for retrieval and discovery of related work: they searched for and suggested potentially relevant literature, and the authors verified each item before deciding which works to cite. We have not used generative AI tools to draft sections of the manuscript, to design or execute the research, to generate synthetic datasets, or to formulate or prove mathematical claims, and these uses are not applicable to this work. We have reviewed all AI-assisted text. We take responsibility for the final content of this work, including text and claims produced with the aid of generative AI.

REFERENCES

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric M. Wolff, Alex H. Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. volume abs/2106.11810, 2021. URL `https://arxiv.org/abs/2106.11810`.

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 (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4709–4722. PMLR, 2025. URL `https://proceedings.mlr.press/v305/cao25a.html`.

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.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

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 (NeurIPS)*, 2024.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile

controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 91560–91596, 2024.

Xingtai Gui, Meijie Zhang, Tianyi Yan, Wencheng Han, Jiahao Gong, Feiyang Tan, Cheng-zhong Xu, and Jianbing Shen. Bridging scene generation and planning: Driving with world model via unifying vision and motion representation. *arXiv preprint arXiv:2603.14948*, 2026.

Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. iPad: Iterative proposal-centric end-to-end autonomous driving. *IEEE Robotics and Automation Letters*, 11(10):11142–11149, 2026. doi: 10.1109/LRA.2026.3723334.

Yufeng Hong, Xiaotian Zhou, Yingyan Li, Xiangpo Zhou, Lin Liu, Yadan Luo, Shaoqing Xu, Lei Yang, and Ziying Song. DriveFuture: Future-aware latent world models for autonomous driving. *arXiv preprint arXiv:2605.09701*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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)*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023. URL `https://proceedings.mlr.press/v202/li23q.html`.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *International Conference on Learning Representations (ICLR)*, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2026b.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2026c.

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, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Zihao Sheng, Xin Ye, Jingru Luo, Sikai Chen, and Liu Ren. Explorevla: Dense world modeling and exploration for end-to-end autonomous driving. In *European Conference on Computer Vision (ECCV)*, pp. 266–284, 2026.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. SparseDrive: End-to-end autonomous driving via sparse scene representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8795–8801, 2025. doi: 10.1109/ICRA55743.2025.11128800.

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 (CVPR)*, pp. 17884–17893, June 2026a.

Linbo Wang, Yupeng Zheng, Qiang Chen, Shiwei Li, Yichen Zhang, Zebin Xing, Qichao Zhang, Xiang Li, Deheng Qian, Pengxuan Yang, Yihang Dong, Ce Hao, Xiaoqing Ye, Junyu Han, Yifeng Pan, and Dongbin Zhao. Latent-WAM: Latent world action modeling for end-to-end autonomous driving. *arXiv preprint arXiv:2603.24581*, 2026b.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2024a.

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

Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, et al. Drivelaw: Unifying planning and video generation in a latent driving world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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, 2025.

Pengxuan Yang, Yupeng Zheng, Deheng Qian, Zebin Xing, Qichao Zhang, Linbo Wang, Yichen Zhang, Shaoyu Guo, Zhongpu Xia, Qiang Chen, Junyu Han, Lingyun Xu, Yifeng Pan, and Dongbin Zhao. DreamerAD: Efficient reinforcement learning via latent world model for autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2026.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, et al. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Zhida Zhao, Talas Fu, Yifan Wang, Lijun Wang, and Huchuan Lu. From forecasting to planning: Policy world model for collaborative state-action prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2024.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, Xianpeng Lang, and Dongbin Zhao. World4Drive: End-to-end autonomous driving via intention-aware physical latent world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28632–28642, 2025.

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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4): 4673–4691, 2026a. doi: 10.1109/TPAMI.2025.3647952.

Xin Zhou, Dingkang Liang, Sifan Tu, Xiwu Chen, Yikang Ding, Dingyuan Zhang, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

Xin Zhou, Zongchuang Zhao, Zhibo Yang, Mingsheng Li, Humen Zhong, Shuai Bai, Du Chu, Ruizhe Chen, Zhaohai Li, Jun Tang, Qiuyue Wang, Mingkun Yang, Jiazhao Zhang, Dayiheng Liu, Dingkang Liang, and Xiang Bai. Qwen-Drive-1.0: An initial step towards a vision-language foundation model for autonomous driving. *arXiv preprint arXiv:2609.00111*, 2026b.

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 (NeurIPS)*, 2025b.

The contents of this appendix are as follows:

- Appendix A gives the data flow and tensor shapes of the three transformers, for checking the implementation.
- Appendix B gives the two-stage optimization and the full set of losses.
- Appendix C gives the metric definitions for NAVSIM-v1/v2 and HUGSIM, including how HD-Score and the HUGSIM average are computed.
- Appendix D gives four additional ablations: posterior registers, future duration, the distillation weight, and which modules are trained.
- Appendix E gives additional visualizations of NAVSIM candidate trajectories and HUGSIM closed-loop rollouts.

A Detailed Architecture and Data Flow

Fig. 6 shows the three transformers used at inference: the world model predicts the future latent from prior queries and the history context; the planner decodes candidate trajectories from that latent; the scorer selects one trajectory. The posterior queries that can replace the prior queries are used only while constructing the training target.

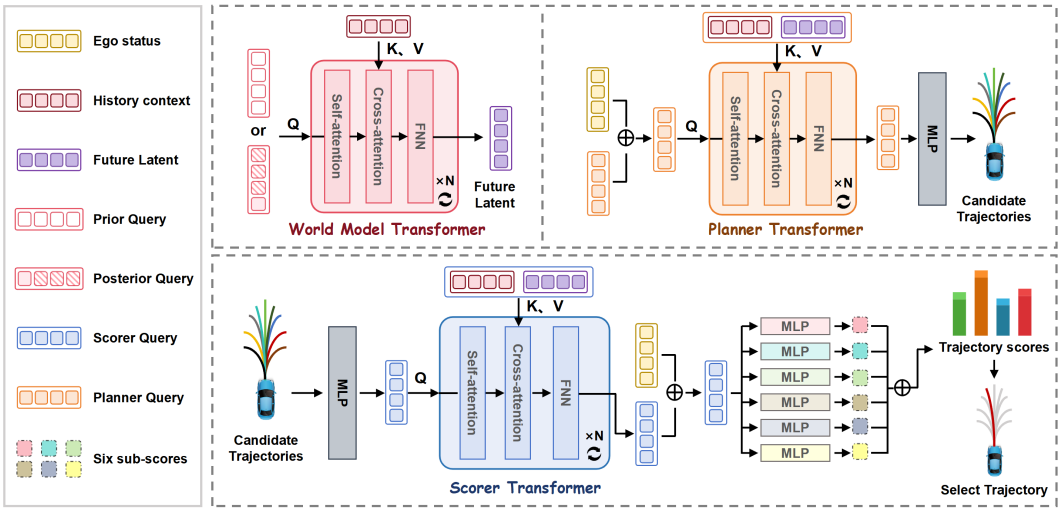

Figure 6: **Detailed transformer architecture and data flow.** The world-model transformer takes either posterior queries or prior queries as Q, and uses the history context as K and V, to produce the future latent. The planner transformer uses planner queries as Q and the combined history–latent memory as K and V; an MLP decodes its outputs into candidate trajectories. The scorer transformer embeds those candidates as Q, cross-attends to the same memory, adds the ego status, and maps the result through six subscore heads to select one trajectory. Posterior queries are used only during training.

A.1 Token Bookkeeping and End-to-End Flow

The visual backbone extracts 16 scene registers from each camera frame. With four historical frames and four cameras, the uncompressed historical stream contains $4 \times 4 \times 16 = 256$ registers. The Temporal Register Pyramid allocates $[4, 4, 8, 16]$ queries per camera from the oldest to the current temporal level, producing $4 \times (4 + 4 + 8 + 16) = 128$ history tokens. The posterior observes four future times, $\{1, 2, 3, 4\}$ seconds, and retains all $4 \times 4 \times 16 = 256$ future slots. The prior uses the same slot layout, so every predicted token has a fixed time–camera–register counterpart in the posterior target. Concatenating history and future tokens gives a 384-token planning memory.

Table 9: **Tensor bookkeeping for the reported model.** B is batch size, d=256, K=64, and J=8.

Quantity	Construction	Shape
Raw history registers	4 times $\times$ 4 cameras $\times$ 16 registers	$B \times 256 \times d$
Compressed history h_t	temporal allocation $[4, 4, 8, 16]$ per camera	$B \times 128 \times d$
Posterior queries f_t^+	4 future times $\times$ 4 cameras $\times$ 16 registers	$B \times 256 \times d$
Prior queries Q^{prior}	learned queries with time/camera/slot embeddings	$B \times 256 \times d$
Planning memory $m_t^{\pm}$	history–future concatenation with type embeddings	$B \times 384 \times d$
Candidate trajectories	K candidates, J poses (x, y, ψ)	$B \times 64 \times 8 \times 3$
Candidate subscores	NC, DAC, TTC, EP, DDC, and comfort	$B \times 64 \times 6$

The data flow is consequently

$$(v_{t-3:t}, c_{t-3:t}) \xrightarrow{E_\eta} h_t \xrightarrow{P_\theta} \hat{z}_t^+ \xrightarrow{\text{concat}(h_t, \hat{z}_t^+)} m_t^- \xrightarrow{G_{\text{plan}}, G_{\text{score}}} \tau^*, \tag{21}$$

whereas Stage 1 replaces P_θ with the privileged posterior $P_\phi(h_t, f_t^+)$ and uses the resulting z_t^+ to shape the latent space through planning supervision.

A.2 World-Model Transformer

As in Fig. 6, the world-model block is shared in structure by the posterior and the prior. Its Q is either the posterior queries, formed from ground-truth future scene registers, or the learnable prior queries. Its K and V are the history context. Each of the four pre-normalized blocks applies self-attention on the queries, cross-attention from the queries to the history memory, and a feed-forward network, with a residual connection around every sublayer. The hidden width is d=256, with four attention heads, an FFN width of 1024, and dropout 0.1. The block output is the future latent. The two branches differ in the query source:

$$z_t^+ = \text{WM}_\phi^{\text{post}}\big(Q = f_t^+, K = h_t, V = h_t\big), \tag{22}$$

$$\hat{z}_t^+ = \text{WM}_\theta^{\text{prior}}\big(Q = Q^{\text{prior}}, K = h_t, V = h_t\big). \tag{23}$$

The prior queries follow the same future-time, camera, and register-slot order as f_t^+, so the token-wise loss in equation 17 compares corresponding tokens. In the reported model, compatible transformer weights are initialized from the Stage 1 posterior checkpoint; the prior query slots remain prior-specific parameters. At inference only the prior queries are used.

A.3 Trajectory Planner

The planner transformer takes planner queries as Q. Its K and V are the planning memory formed from the history context and the future latent, the 384-token concatenation in Table 9. Four decoder blocks refine the K=64 candidate tokens. Each block has one-head self-attention among the candidates, one-head cross-attention to the memory, and a width-1024 FFN, with hidden width 256, projection dropout 0.1, and stochastic depth 0.2. An MLP ($256 \to 1024 \to 24$) then maps each refined token to J=8 poses (x, y, ψ), giving the candidate trajectories drawn on the right of Fig. 6. The final refinement is passed to the scorer.

A.4 Trajectory Scorer and Selection

The scorer follows the bottom path of Fig. 6. Candidate trajectories are embedded by an MLP ($24 \to 1024 \to 256$) and used as Q. The K and V are the same history–latent memory used by the planner. Four decoder blocks, again with width 256, one attention head, and FFN width 1024, mix the candidate tokens and cross-attend to that memory. The ego-status token is added to every candidate feature. Six MLP heads ($256 \to 1024 \to 1$) then predict NC, DAC, TTC, EP, DDC, and comfort. For NAVSIM-v1 selection, candidate k is ranked by

$$\log S_k = \log p_k^{\text{NC}} + \log p_k^{\text{DAC}} + \log\big(5p_k^{\text{TTC}} + 5p_k^{\text{EP}} + 2p_k^{\text{C}}\big), \qquad k^* = \arg\max_k \log S_k. \tag{24}$$

The constant denominator in PDMS does not change the argmax. DDC is supervised, and it has zero weight in this NAVSIM-v1 ranking. The geometry entering the scorer is detached, so the scoring loss updates the scorer without changing the trajectory decoder through that path.

A.5 Architecture Summary

Table 10: **Transformer configurations.** "Heads" refers to attention heads in each block; all FFNs use GELU unless the output head is explicitly described as an MLP above.

Module	Blocks	Width	Heads	FFN	Output
Temporal Register Pyramid	2	256	4	1024	128 history tokens
Posterior world model	4	256	4	1024	256 target tokens z_t^+
Prior world model	4	256	4	1024	256 predicted tokens $\hat{z}_t^+$
Trajectory planner	4	256	1	1024	$64 \times 8 \times 3$ proposals
Trajectory scorer	4	256	1	1024	six scores per proposal

B Training Objectives and Optimization

B.1 Two-Stage Training Protocol

Table 11 gives the complete training schedule used for the reported model. Stage 1 learns the privileged posterior and the shared planning modules from historical and future observations. Stage 2 loads structurally compatible parameters from Stage 1, keeps a separate posterior teacher frozen in evaluation mode, and optimizes the history-only prior. Future images are consumed only by this frozen teacher; they are neither augmented nor exposed to the student. The teacher output is detached before latent alignment. At test time, the teacher and all future-observation loading are removed.

Table 11: **Optimization and implementation settings.** Learning rates are kept consistent with Sec. 4.1; all runs use the same camera, token, and trajectory configuration unless an ablation states otherwise.

Item	Setting
Input images	four historical frames; front, left, right, rear; 1148×672
Training-only future observations	$\{1, 2, 3, 4\}$ s; same four cameras
Visual encoder	DINOv2 ViT-S; frozen base; LoRA rank 32
Planning horizon	4 s, 8 waypoints at 0.5-s intervals
Candidate / latent counts	K=64 trajectories; N_z=256 latent tokens
Optimizer and duration	AdamW; 50 epochs per stage
Learning-rate schedule	linear warm-up for first 10% of steps, then cosine decay to zero
Peak learning rate	Stage 1: 1×10^{-4}; Stage 2: 1×10^{-5}
Hardware and batch	16× NVIDIA A800; 8 samples/GPU; global batch 128
Regularization weights	$\lambda_{\text{norm}} = \lambda_{\text{div}} = 10^{-4}$
Latent-alignment weight	$\lambda_{\text{latent}} = 0.05$

B.2 Trajectory Generation Objective

Let $\tau_k \in \mathbb{R}^{J\times 3}$ be candidate k, τ^* the demonstrated ego trajectory, and J=8. The planner is trained with a best-of-K regression loss,

$$\mathcal{L}_{\text{plan}} = \frac{1}{B}\sum_{b=1}^{B} \min_{1\le k\le K} \frac{1}{J}\sum_{j=1}^{J} \left\|\tau_{b,k,j} - \tau^*_{b,j}\right\|_1 . \tag{25}$$

Thus, only the closest candidate receives direct trajectory-regression supervision for a sample, while the learned candidate tokens and self-attention preserve a multimodal proposal set.

B.3 Trajectory Scoring Objective

Every candidate is evaluated by the official proposal simulator to obtain targets $q_{b,k,m}$ for

$$\mathcal{M} = \{\text{NC}, \text{DAC}, \text{TTC}, \text{EP}, \text{DDC}, \text{C}\}.$$

If $a_{b,k,m}$ denotes the corresponding predicted logit, the scoring loss is

$$\mathcal{L}_{\text{score}} = \sum_{m \in \mathcal{M}} \frac{1}{BK} \sum_{b=1}^{B} \sum_{k=1}^{K} \text{BCEWithLogits}(a_{b,k,m}, q_{b,k,m}). \tag{26}$$

Targets taking the intermediate value 0.5 for NC or DDC are mapped to the conservative negative class for binary training. Undefined TTC targets are masked rather than treated as negatives. This loss teaches proposal selection directly from planning-quality signals instead of using distance to the demonstrated trajectory as a proxy score.

B.4 Latent Regularization and Hindsight-to-Foresight Distillation

For a latent tensor $z \in \mathbb{R}^{B \times N_z \times d}$, we use two lightweight anti-collapse terms:

$$\mathcal{L}_{\text{norm}}(z) = \frac{1}{BN_z} \sum_{b,i} \left[1 - \frac{\|z_{b,i}\|_2}{\sqrt{d}} \right]_+^2, \tag{27}$$

$$\mathcal{L}_{\text{div}}(z) = \frac{1}{BN_z(N_z - 1)} \sum_{b} \sum_{i \neq j} \left(\frac{z_{b,i}^\top z_{b,j}}{\|z_{b,i}\|_2 \|z_{b,j}\|_2} \right)^2. \tag{28}$$

The first term penalizes vanishing token norms; the second penalizes redundant latent slots. Written without the shorthand in equation 19, the task objective is

$$\mathcal{L}_{\text{task}}(z) = \mathcal{L}_{\text{plan}} + \mathcal{L}_{\text{score}} + 10^{-4}\mathcal{L}_{\text{norm}}(z) + 10^{-4}\mathcal{L}_{\text{div}}(z). \tag{29}$$

Stage 1 substitutes $z = z_t^+$. Stage 2 substitutes $z = \hat{z}_t^+$ and adds the token-aligned teacher loss

$$\mathcal{L}_{\text{Stage 2}} = \mathcal{L}_{\text{task}}(\hat{z}_t^+) + 0.05 \underbrace{\frac{1}{N_z d} \left\| \hat{z}_t^+ - \text{sg}(z_t^+) \right\|_F^2}_{\mathcal{L}_{\text{latent}}}. \tag{30}$$

The latent loss is computed in FP32 even under mixed-precision training. The stop-gradient operation and frozen teacher ensure that Stage 2 improves the causal prior rather than moving the privileged target toward the student.

C Benchmarks and Evaluation Metrics

C.1 Benchmark Summary

Table 12: **Evaluation benchmarks and their roles.** NAVSIM results use the official `navtest` evaluator; HUGSIM receives the NAVSIM-trained model without simulator-specific adaptation.

Benchmark	Data and protocol	Reported metrics
NAVSIM-v1 Dauner et al. (2024)	OpenScene/nuPlan-derived real driving logs; non-reactive 4-s proposal simulation; train on `navtrain` (103,288 scenes: 85k train / 18k validation), evaluate on `navtest` (12,146).	NC, DAC, TTC, EP, C, PDMS
NAVSIM-v2 Cao et al. (2025)	Same split used in this paper, evaluated with the extended official proposal scorer and additional rule/smoothness criteria.	NC, DAC, DDC, TLC, EP, TTC, LK, HC, EC, EPDMS
HUGSIM Zhou et al. (2026a)	Closed-loop rollouts on 436 public scenarios from nuScenes, KITTI-360, Waymo, and PandaSet; zero-shot transfer of the NAVSIM-trained model.	RC and HD-Score on Easy, Medium, Hard, Extreme, and the scene-weighted average

NAVSIM is derived from OpenScene, a compact distribution of nuPlan logs Caesar et al. (2021). It evaluates planned trajectories by rolling the ego vehicle forward while other actors follow recorded

trajectories, which makes large-scale evaluation deterministic and non-reactive. HUGSIM complements this protocol with closed-loop observations: the ego action changes the next rendered view and simulated state, and surrounding actors can interact with the rollout. The cross-domain HUGSIM test therefore probes compounding planning errors and distribution shift that are not present in a single NAVSIM proposal rollout.

C.2 NAVSIM-v1: PDM Score

For each proposal, the official Predictive Driver Model scorer returns no-at-fault collision (NC), drivable-area compliance (DAC), time-to-collision (TTC), normalized ego progress (EP), and comfort (C). NC and DAC are multiplicative safety gates, whereas TTC, EP, and C form a weighted quality term:

$$\mathrm{PDMS} = \mathrm{NC} \cdot \mathrm{DAC} \cdot \frac{5\,\mathrm{TTC} + 5\,\mathrm{EP} + 2\,\mathrm{C}}{12}. \tag{31}$$

NC penalizes at-fault collisions along the simulated ego rollout; DAC checks whether the ego footprint stays inside the drivable area; TTC detects imminent projected collisions; EP measures progress along the route centerline and is normalized within the proposal set; and C checks acceleration, jerk, and yaw-related comfort bounds. Component values and PDMS are first computed per scene and then averaged. Consequently, the aggregate component means in Table 2 need not reproduce the aggregate PDMS exactly when substituted into equation 31.

C.3 NAVSIM-v2: Extended PDM Score

NAVSIM-v2 adds driving-direction compliance (DDC), traffic-light compliance (TLC), lane keeping (LK), history comfort (HC), and extended comfort (EC). DDC and TLC join NC and DAC as multiplicative penalties. LK measures lane-relative stability, HC applies the standard within-trajectory dynamics constraints, and EC measures temporal consistency between consecutive plans. Before aggregation, the reported evaluator applies a human-penalty filter to every subscore. Let $\mathcal{M}_{\mathrm{pen}} = \{\mathrm{NC}, \mathrm{DAC}, \mathrm{DDC}, \mathrm{TLC}\}$ and $\mathcal{M}_{\mathrm{avg}} = \{\mathrm{EP}, \mathrm{TTC}, \mathrm{LK}, \mathrm{HC}, \mathrm{EC}\}$, with $w_{\mathrm{EP}} = w_{\mathrm{TTC}} = 5$ and $w_{\mathrm{LK}} = w_{\mathrm{HC}} = w_{\mathrm{EC}} = 2$. If the human trajectory scores 0 on a subscore, that term is set to 1; otherwise the agent's score is used:

$$\begin{aligned} \mathrm{EPDMS} &= \left(\prod_{m \in \mathcal{M}_{\mathrm{pen}}} \mathrm{filter}_m \right) \cdot \frac{\sum_{m \in \mathcal{M}_{\mathrm{avg}}} w_m \,\mathrm{filter}_m}{\sum_{m \in \mathcal{M}_{\mathrm{avg}}} w_m}, \\ \mathrm{filter}_m &= \begin{cases} 1 & \text{if } m(\mathrm{human}) = 0, \\ m(\mathrm{agent}) & \text{otherwise.} \end{cases} \end{aligned} \tag{32}$$

As for PDMS, the benchmark computes equation 32 at the scene level before averaging; all values in the main tables are displayed on a 0–100 scale.

C.4 HUGSIM: Route Completion and HD-Score

HUGSIM evaluates closed-loop driving on reconstructed scenes from nuScenes, KITTI-360, Waymo, and PandaSet Zhou et al. (2026a). The public set used here has 436 scenarios, distributed as in Table 13. Easy inserts no extra vehicle. Medium typically inserts one constant-velocity vehicle. Hard uses more obstructive vehicles, including some attackers. Extreme uses attack planners that intercept the predicted ego future, with more inserted vehicles. We deploy the NAVSIM-trained prior directly, without HUGSIM-specific fine-tuning.

Table 13: **HUGSIM public scenario split.** Counts are the 436 released scenarios.

Source	Scenarios	E	M	H	X
nuScenes	88	18	34	18	18
Waymo	108	22	42	22	22
KITTI-360	113	25	36	26	26
PandaSet	127	15	45	30	37
Total	436	80	157	96	103

Route completion $R_c \in [0, 1]$ is the fraction of the prescribed route completed before termination. At each evaluated timestamp t, HUGSIM scores no collision, drivable-area compliance, TTC, and comfort. For an episode of T_e timestamps,

$$\text{HD-Score} = R_c \cdot \frac{1}{T_e} \sum_{t=1}^{T_e} \left[\text{NC}_t \, \text{DAC}_t \frac{5 \, \text{TTC}_t + 2 \, \text{COM}_t}{7} \right]. \tag{33}$$

The product of NC and DAC keeps an off-road or colliding rollout from being offset by comfort, and R_c scales the result by how much of the route is finished. Each difficulty column in Table 3 is the mean of its scenarios, shown on a 0–100 scale. The Avg. column is not $(E + M + H + X)/4$, and it is not an equal average of the four source datasets. It weights the four difficulty means by their scenario counts,

$$\text{Avg.} = \frac{80 \, E + 157 \, M + 96 \, H + 103 \, X}{436}, \tag{34}$$

which is the mean over all 436 scenarios. Medium contains the most scenarios, so it pulls Avg. below the unweighted mean of the four difficulties.

D Additional Controlled Ablations

The ablations below are reported on NAVSIM-v1 `navtest`. This section first compares how the posterior-only model reads future-register tokens, and those scores are posterior PDMS. It then compares the length of the observed future, the weight on latent distillation, and, finally, the distillation loss together with which modules are updated. Each subsection changes the factor named in its title. The highlighted row is the reported setting.

D.1 Future Registers in the Posterior

Table 14 compares three ways to use the scene-register tokens from the four future frames, on the posterior-only branch. These scores are posterior PDMS, not the deployed prior. Taking the future registers directly as queries is the reported posterior and reaches 95.0. Using the same tokens as keys and values instead drops the score to 94.6. Compressing them with an MLP to 128 tokens and then using those tokens as queries drops it further, to 94.2. The uncompressed future registers are therefore the stronger queries for the privileged posterior.

Table 14: **How the posterior uses future-register tokens.** Scores are posterior-only PDMS on NAVSIM-v1 `navtest`.

Future-register role	Tokens	PDMS↑
Keys and values	256	94.6
Queries (ours)	256	**95.0**
MLP compression, then queries	128	94.2

D.2 Observed-Future Duration

Table 15 varies how many future times the model observes. Each future time contributes one frame from each of the four cameras, and each camera-frame keeps 16 scene registers, so one future second is 64 tokens. Using only the 1 s future therefore supplies 64 tokens and reaches 93.5 PDMS. Extending the observation to $\{1, 2, 3, 4\}$ s raises the latent to 256 tokens and the score to 93.8, which is the reported setting. The gain from the longer horizon shows that the additional future information is useful for planning.

Table 15: **Ablation of future duration.** Each future second contributes 4×16 register tokens. Scores are PDMS on NAVSIM-v1 `navtest`.

Future times	Tokens	PDMS↑
1 s	64	93.5
$\{1, 2, 3, 4\}$ s (ours)	256	**93.8**

D.3 Latent-Distillation Weight

Table 16 varies the weight on token-wise latent distillation. With $\lambda_{\text{latent}}=0$, the prior is not trained to match the posterior target: latent similarity stays at 0.46 and PDMS is 93.5. A small weight of 0.01 raises similarity to 0.88 and PDMS to 93.6. The reported weight, 0.05, reaches similarity 0.93 and the highest PDMS, 93.8. Increasing the weight further to 0.10 makes the latents still closer, with similarity 0.95, but PDMS falls back to 93.7. Tighter alignment therefore does not by itself produce a better plan. The selected weight is the one at which the planning score peaks.

Table 16: **Sensitivity to the latent-distillation weight.**

λ_{latent}	Latent sim.↑	PDMS↑
0	0.46	93.5
0.01	0.88	93.6
0.05 (ours)	**0.93**	**93.8**
0.10	0.95	93.7

D.4 Distillation Loss and Trainable Modules

Table 17 varies the distillation loss and the set of trainable modules. The first two rows train only the prior world model. The history encoder, planner, scorer, and ViT stay frozen, and the planner and scorer are copied from the posterior-only model. Under $\mathcal{L}_{\text{latent}}$ alone, latent similarity is 0.96 and PDMS is 93.0. Adding $\mathcal{L}_{\text{plan}}$ and $\mathcal{L}_{\text{score}}$, with gradients flowing only into the world model, changes these values to 0.94 and 93.1. Latent alignment alone therefore leaves the planning score low. Whether making the latent still closer would raise PDMS is measured with the modules held fixed, in Table 16: increasing λ_{latent} from 0.05 to 0.10 raises similarity from 0.93 to 0.95, while PDMS falls from 93.8 to 93.7. The planning and scoring losses, not a smaller latent gap, determine the planning score.

The last three rows keep $\mathcal{L}_{\text{latent}}+\mathcal{L}_{\text{plan}}+\mathcal{L}_{\text{score}}$ and train a larger set of modules. With the posterior planner and scorer held fixed, PDMS remains 93.0–93.1. Training the prior world model together with the scorer raises it to 93.4, and training the planner as well raises it to 93.5. Training the remaining modules, with the ViT base frozen and adapted by LoRA, reaches 93.8. Those copied planner and scorer weights were learned for the privileged posterior latent. The prior has to relearn the downstream weights for the latent it predicts.

Table 17: **Distillation loss and trainable modules.** The first two rows train only the prior world model and freeze every other module, including LoRA. Scores are PDMS on NAVSIM-v1 `navtest`.

Trainable modules	Loss	Latent sim.↑	PDMS↑
Prior world model only	$\mathcal{L}_{\text{latent}}$	**0.96**	93.0
Prior world model only	$\mathcal{L}_{\text{latent}}+\mathcal{L}_{\text{plan}}+\mathcal{L}_{\text{score}}$	0.94	93.1
Prior world model and scorer	$\mathcal{L}_{\text{latent}}+\mathcal{L}_{\text{plan}}+\mathcal{L}_{\text{score}}$	0.93	93.4
Prior world model, scorer, and planner	$\mathcal{L}_{\text{latent}}+\mathcal{L}_{\text{plan}}+\mathcal{L}_{\text{score}}$	0.93	93.5
All except the ViT base (ours)	$\mathcal{L}_{\text{latent}}+\mathcal{L}_{\text{plan}}+\mathcal{L}_{\text{score}}$	0.93	**93.8**

E Additional Qualitative Results

We provide two complementary qualitative views. The NAVSIM figures expose the complete candidate set and therefore illustrate generation and selection in a single frame. The HUGSIM figures show consecutive closed-loop observations, revealing whether the selected plan remains stable after its actions alter the subsequent input. These examples are presented without cherry-picked metric annotations; the quantitative conclusions remain those of Sec. 4.2.

E.1 NAVSIM Candidate Generation and Selection

In Figs. 7–10, each row contains the front-view image, ground-truth trajectory in the BEV scene, and PlanWAM output. Gray curves are all 64 candidates, the red curve is the trajectory selected

by PlanWAM, and the green curve is the recorded trajectory. The cases cover curved roads, turns, intersections, and dense multi-agent traffic.

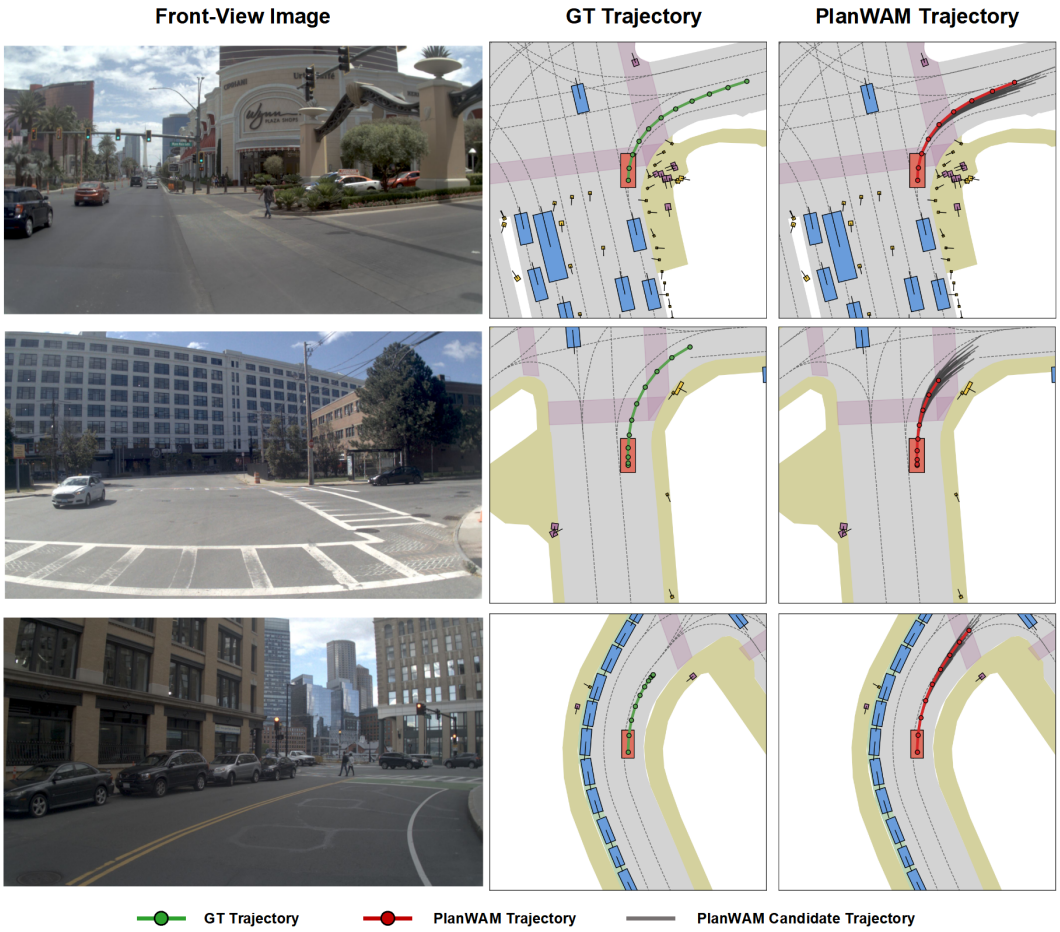

Figure 7: **Additional NAVSIM results (set 1).** Front view, ground-truth trajectory, and PlanWAM candidates.

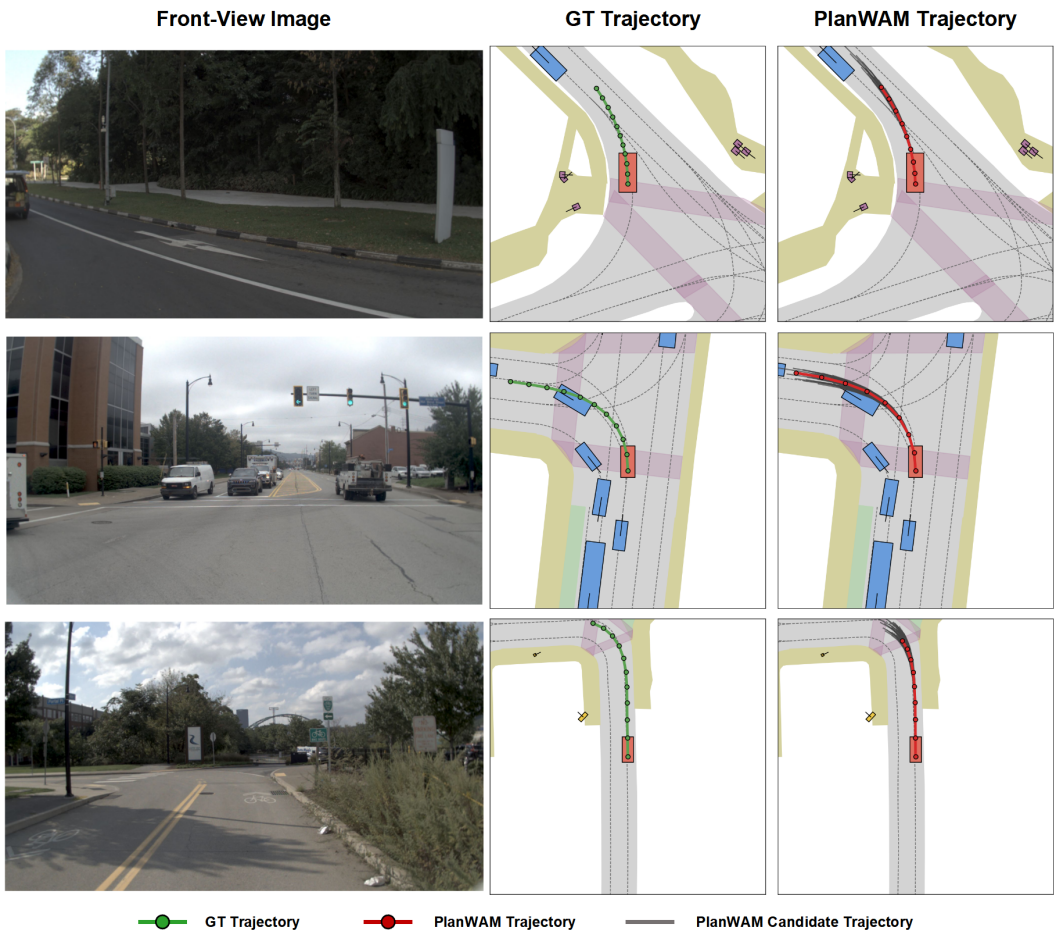

Figure 8: **Additional NAVSIM results (set 2).** Curved roads and intersections.

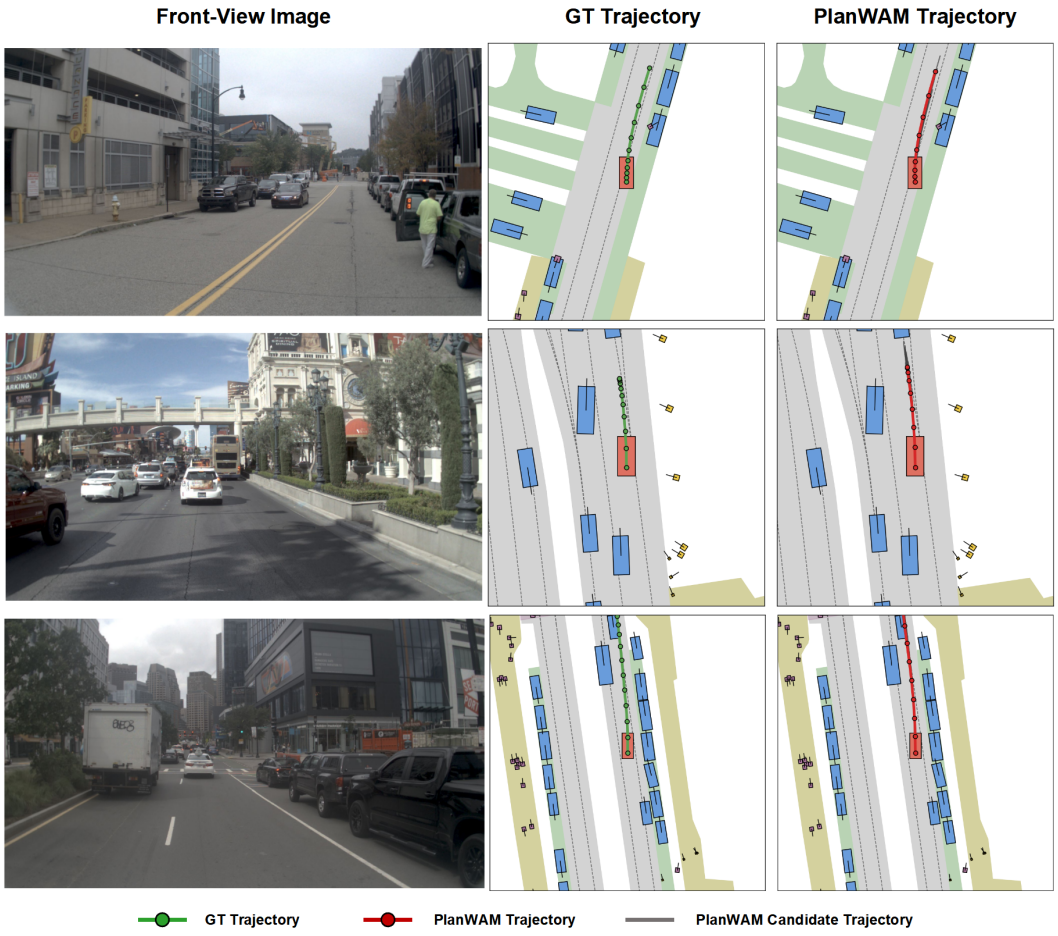

Figure 9: **Additional NAVSIM results (set 3).** Urban scenes with parked and moving traffic.

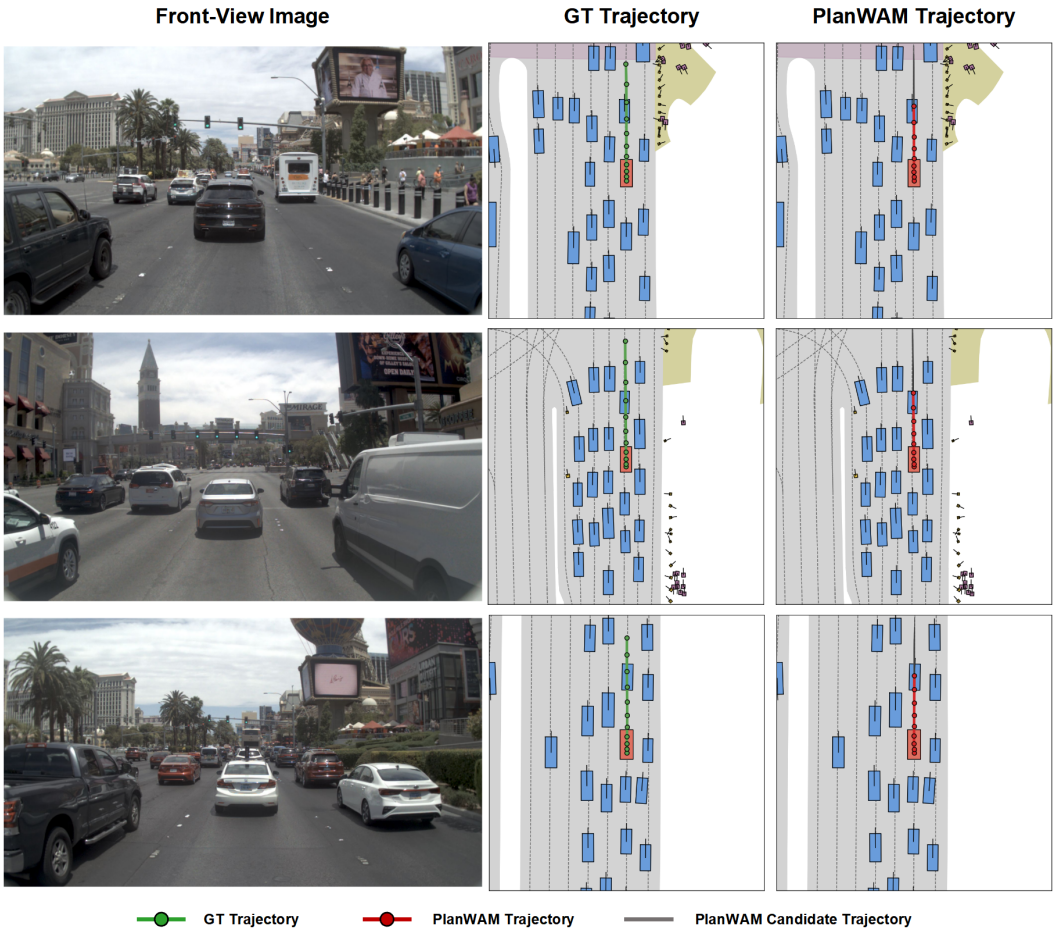

Figure 10: **Additional NAVSIM results (set 4).** Dense multi-lane traffic. Gray curves are candidates; red is the selected plan.

E.2 Zero-Shot Closed-Loop Rollouts on HUGSIM

Figures 11–14 show two six-step rollouts for each of the four HUGSIM source domains. The overlaid corridor, centerline, and waypoints visualize the current PlanWAM plan. The model is unchanged across domains and receives no HUGSIM-specific training.



Figure 11: **Zero-shot HUGSIM rollouts on nuScenes.** Two closed-loop cases are shown over six consecutive planning steps.



Figure 12: **Zero-shot HUGSIM rollouts on KITTI-360.** Two closed-loop cases are shown over six consecutive planning steps.

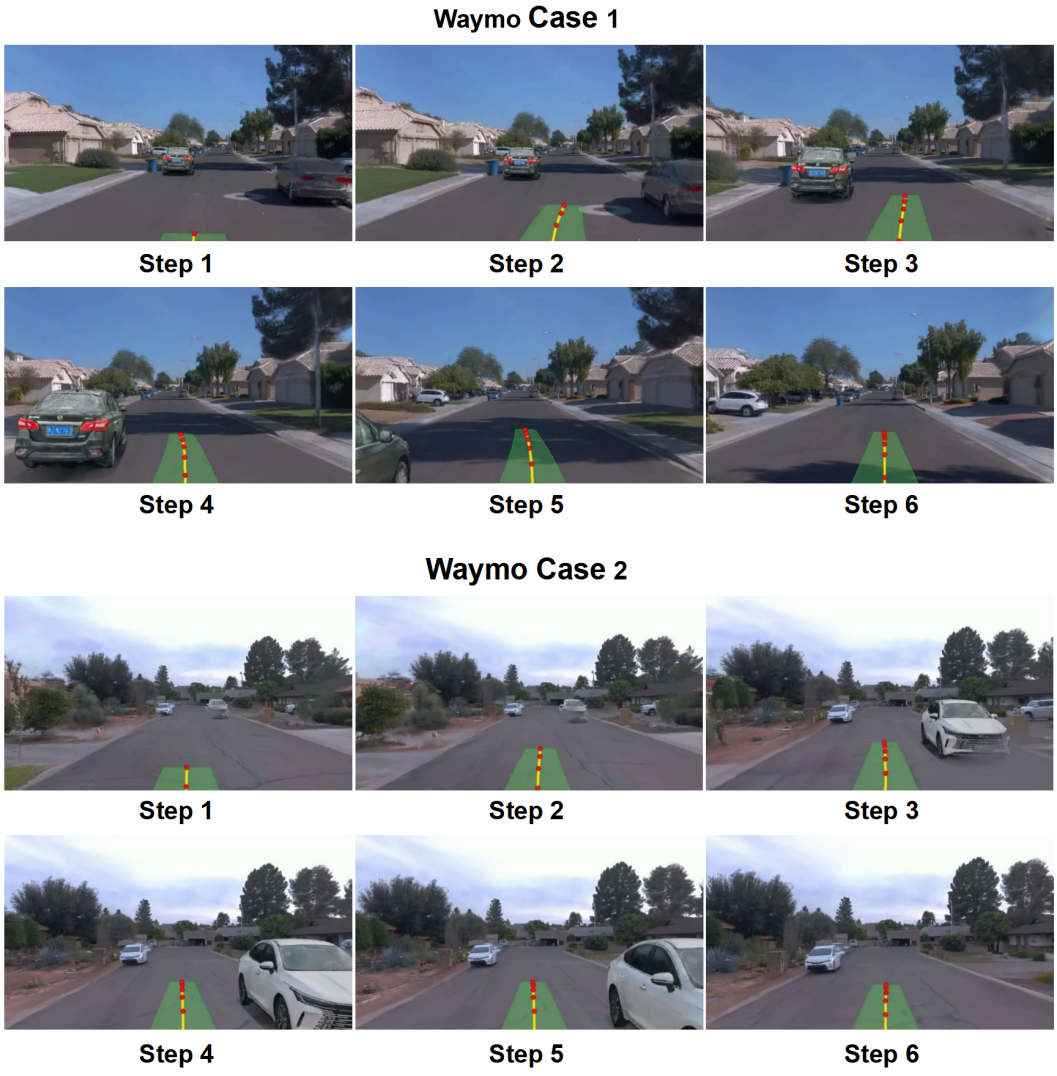

Figure 13: **Zero-shot HUGSIM rollouts on Waymo.** Two closed-loop cases are shown over six consecutive planning steps.

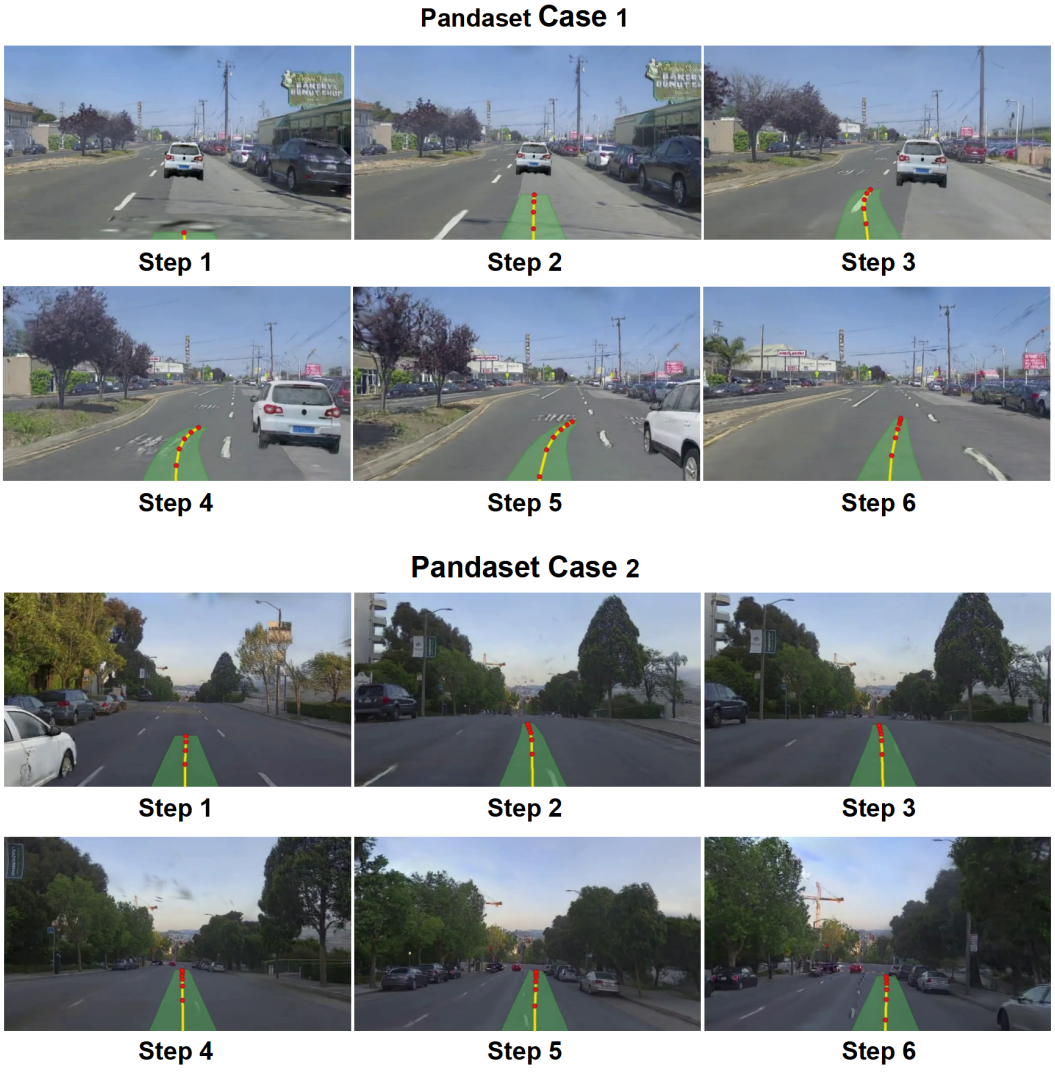

Figure 14: **Zero-shot HUGSIM rollouts on PandaSet.** Two closed-loop cases are shown over six consecutive planning steps.