FORESIGHT ON DEMAND: LEARNING WHAT AND WHEN TO PREDICT FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

World action models predict the future to guide planning, but not all predicted information is useful for planning and not every scene benefits from additional prediction. Foresight can waste computation when history already supports a reliable plan, and can even degrade planning when its predictions are inaccurate. We present FD-WAM, which couples *what to predict* with *when to predict* for better planning, with the added benefit of reducing unnecessary foresight computation. Specifically, a Decision-Relevant Foresight Representation uses planning supervision to shape future slots learned through latent prediction, then compresses predictions from multiple future horizons into a compact set of tokens for planning. Utility-Aware Selective Foresight trains a router with the realized planning-score difference between the same planner's foresight and no-foresight paths, enabling selective prediction from observed history alone. Evaluations on NAVSIM v1, NAVSIM v2, and zero-shot HUGSIM show that FD-WAM achieves competitive planning scores at lower inference cost than state-of-the-art baselines under aligned training scale. Controlled ablations further show that the compact foresight representation and selectively invoked foresight improve planning over the no-foresight baseline with little additional inference cost.

1 INTRODUCTION

End-to-end autonomous driving has made rapid progress by directly mapping visual observations and ego states to future trajectories (Chib & Singh, 2024; Chen et al., 2024). Most existing planners, however, make decisions primarily from past and current observations, leaving future scene evolution only implicitly modeled (Hu et al., 2023; Jiang et al., 2023; 2025). This can be limiting in interactive or partially observed situations, such as merging, cut-ins, and occlusions, where the consequence of a decision may only become clear as the scene evolves. Recent world action models address this limitation by explicitly predicting future images, BEV features, occupancy, or latent representations to support planning (Guan et al., 2024; Feng et al., 2025). These approaches suggest that reasoning about the future can provide useful additional context for driving decisions.

In practice, however, the planning value of future prediction is far from uniform. Many predictable future changes have little influence on the current decision, while a small local interaction may substantially alter the safety or ranking of candidate trajectories (Zhang et al., 2026a; Yang et al., 2026b; Zhong et al., 2026). Moreover, additional foresight does not always improve planning: observation history may already support a reliable decision, while inaccurate predictions may provide little useful guidance or even degrade the plan (Luo et al., 2026; Lu et al., 2026a). More importantly, foresight is useful only when the current plan leaves room for improvement and the predicted future can change the decision for the better. Its value therefore depends not only on whether future reasoning may be needed, but also on whether the learned predictor can provide information that actually improves the current plan. These observations raise two coupled questions: *what* future information should be predicted for planning, and *when* is that prediction worth computing? We argue that both should be determined by the planning utility of foresight.

Based on this view, we introduce FD-WAM, a framework for Foresight on Demand. For *what to predict*, Decision-Relevant Foresight Representation learns a compact foresight representation shaped by planning supervision, predicting future latents across multiple horizons and compressing them into a small set of tokens for downstream planning. This lets the planning objective influence the

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.

future slots while bounding the context size the planner consumes. For *when to predict*, we evaluate the same planner with and without predicted foresight during training. Their planning-score difference supervises a lightweight router that runs before future prediction and uses only observed history. This target reflects the gains the learned predictor actually delivers, including the effects of prediction errors. On NAVSIM v1 and NAVSIM v2, FD-WAM achieves 94.4 PDMS and 91.3 EPDMS, respectively, when trained on the *trainval* split, while zero-shot closed-loop evaluation on HUGSIM yields a 40.2 HD-Score. Across these benchmarks, FD-WAM maintains competitive planning performance while requiring lower inference cost than the compared state-of-the-art methods. Our contributions are threefold:

- We formulate future prediction in world action models from the perspective of *planning utility*, considering both what future information is useful for planning and whether the learned predictor can actually turn it into better decisions.
- We introduce FD-WAM, which unifies *what to predict* and *when to predict* through planning utility. It learns compact future representations shaped by planning supervision and predicts the achievable planning gain of foresight before execution, enabling future computation to be invoked only when it is likely to improve the decision.
- We evaluate FD-WAM on NAVSIM v1, NAVSIM v2, and zero shot HUGSIM. Extensive analyses reveal substantial scene level variation in foresight gains and show that selectively allocating future computation achieves a favorable tradeoff between planning performance and inference cost.

2 Related Work

Predictive Representations for Autonomous Driving World models for autonomous driving learn scene evolution using a range of predictive representations. Visual approaches generate driving videos or multi-view forecasts (Wang et al., 2024a;b), while other methods use compact latent states (Wang et al., 2026b), geometry-aware future tokens in GeoWorldAD[1] (Zhang et al., 2026b), discrete vision-action tokens (Yao et al., 2026b), or temporal residuals (Zhang et al., 2026a). DynVLA similarly learns compact ego-centric and environment-centric dynamics tokens for action reasoning (Shang et al., 2026). These approaches establish compact future modeling as an important design direction. Our study focuses on the relationship between the predictive representation, its contribution to planning, and the decision to incur its inference cost.

Using Future Models for Planning Future modeling can benefit planning through both training supervision and inference-time computation. DriveWorld learns scene representations through world-model pretraining (Min et al., 2024), and Drive-JEPA adapts video predictive learning to driving with multimodal trajectory distillation (Wang et al., 2026c). SimWAM[1] jointly trains video and action experts but removes the video expert at deployment (Zhao et al., 2026). When prediction remains at inference, DriveFuture conditions trajectory generation on predicted future latents (Hong et al., 2026), while DriveWAM and DriveVA couple video and action generation (Shi et al., 2026; Liu et al., 2026a). DA-WAM[1] further shapes predictive representations with planning gradients and scores candidates using their action-conditioned future latents (Zhong et al., 2026). FD-WAM learns a shared, compressed future context from ego-conditioned latent targets and trajectory supervision. Its foresight and no-foresight paths use the same planner, providing a matched reference for assessing the additional value of prediction.

Adaptive Computation for Driving Adaptive computation allocates inference effort according to its usefulness for driving. AdaThinkDrive selects between fast and slow reasoning using rewards that compare trajectory quality across modes (Luo et al., 2026), while IRR-Drive learns adaptive reflection over textual intention and future semantic BEV representations (Chen et al., 2026b). AdaptiveWAM[1] varies video-backbone depth through trajectory-quality-guided early exits (Ang et al., 2026). RISE[1] learns continuation utility from realized planning-score differences and adapts rollout length through sequential Roll/Stop decisions, including zero-rollout decisions (Lu et al., 2026a). FD-WAM decides whether to invoke the future predictor before performing its future computation, us-

[1]Concurrent works first made publicly available after July 17, 2026.

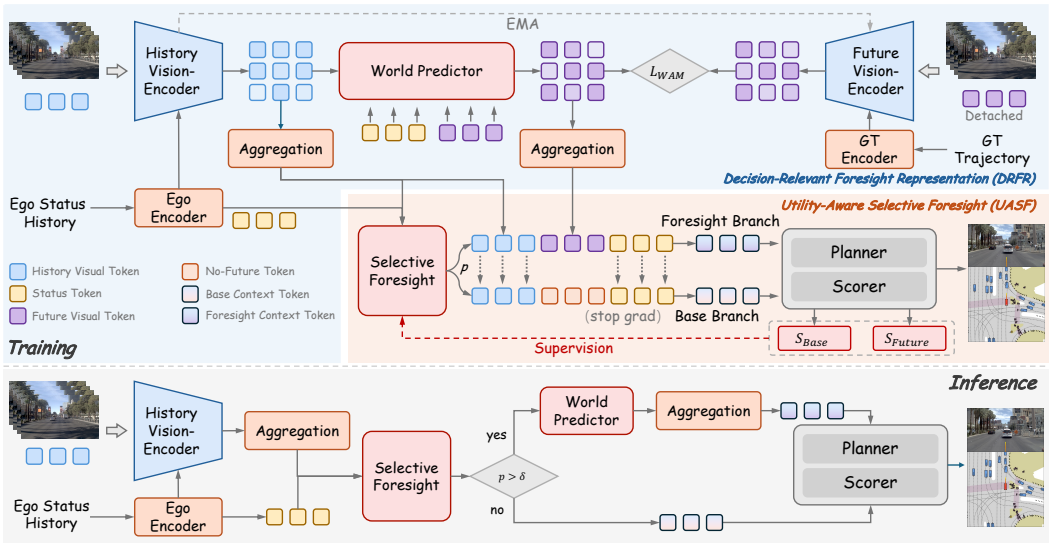


Figure 1: Overview of FD-WAM. Decision Relevant Foresight Representation compresses future slots across multiple horizons into a compact planning context under planning supervision, while Utility Aware Selective Foresight learns from the realized planning gain of the same planner with and without foresight. At inference, the router estimates this gain from observed history before running future prediction, enabling foresight only when it is expected to improve planning.

ing only observed history and supervision from the realized planning gain of the complete foresight context. This couples selective computation with the future representation used by the planner.

3 Method

3.1 Problem Formulation and Overview

Given historical multi view observations $\mathcal{O}_{\leq t}$ and ego states $\mathbf{s}_{\leq t}$, FD-WAM encodes the observed history into scene tokens $\mathbf{H}_t$. We consider two paths that share the same planner. The base path uses a learned null context $\mathbf{C}^0$, while the foresight path uses a compact future context $\mathbf{C}^f$ predicted from observed history:

$$\boldsymbol{\tau}^{\text{base}} = f_{\text{plan}}([\mathbf{H}_t; \mathbf{C}^0], \mathbf{s}_{\leq t}), \qquad \boldsymbol{\tau}^{\text{fore}} = f_{\text{plan}}([\mathbf{H}_t; \mathbf{C}^f], \mathbf{s}_{\leq t}), \tag{1}$$

where $[\cdot;\cdot]$ denotes token concatenation. The null context $\mathbf{C}^0$ preserves the same planning interface without running future prediction, while $\mathbf{C}^f$ uses no ground truth future information at inference. We define the realized planning gain of foresight as:

$$\Delta U = \mathcal{S}(\boldsymbol{\tau}^{\text{fore}}) - \mathcal{S}(\boldsymbol{\tau}^{\text{base}}), \tag{2}$$

where $\mathcal{S}(\cdot)$ denotes the planning score. Thus, ΔU measures the gain delivered by the learned predictor rather than an oracle gain from ground truth future observations.

As illustrated in Fig. 1, this formulation connects two questions: *what to predict for planning* and *when that prediction is worth computing*. The former is addressed by learning the compact future context $\mathbf{C}^f$, while the latter is addressed by estimating its potential planning gain from observed history before future prediction. During training, both planning paths are evaluated to provide supervision for this estimate. At inference, the future branch is executed only when the estimated benefit is sufficient; otherwise, FD-WAM directly uses the base path.

3.2 Decision Relevant Foresight Representation

Compression alone does not ensure that predicted future features are useful for planning. We therefore use ground truth future observations as visual references for future feature learning, while plan-

ning gradients shape the learned representation toward information useful for trajectory evaluation and selection.

Given learnable history registers $\mathbf{R}^h$, we first encode the observed history as:

$$\tilde{\mathbf{R}}^h = \mathbf{R}^h + \psi_h(\mathbf{s}_{\le t}), \qquad \mathbf{H}_t = E_\theta\left(\mathcal{O}_{\le t}, \tilde{\mathbf{R}}^h\right), \tag{3}$$

where E_θ is the online visual encoder and ψ_h projects historical ego states into the register space. We then use learnable future registers $\mathbf{R}^f$ to predict future slots from the observed history:

$$\hat{\mathbf{F}}^f = f_{\text{pred}}\left(\mathbf{H}_t, \mathbf{s}_{\le t}, \psi_q(\mathbf{R}^f)\right), \tag{4}$$

where ψ_q is a learned projection and $\hat{\mathbf{F}}^f = \{\hat{\mathbf{F}}^f_h\}_{h\in\mathcal{H}}$ contains predictions across future horizons $\mathcal{H}$. During training, an EMA teacher encodes ground truth future observations into reference features:

$$\mathbf{F}^\star = E_{\text{ema}}\left(\mathcal{O}_{>t}, \text{sg}(\mathbf{R}^f) + \psi_f(\boldsymbol{\tau}^{\text{gt}})\right), \tag{5}$$

where E_{ema} is the EMA copy of E_θ, ψ_f embeds the ground truth ego trajectory, and $\text{sg}(\cdot)$ denotes stop gradient. The same $\mathbf{R}^f$ defines the slot space for prediction and future reference extraction. Although the teacher does not directly update these registers, they are optimized through the prediction and planning paths, allowing planning supervision to influence the future information queried from the teacher. We align the predicted and reference features using:

$$\mathcal{L}_{\text{wam}} = \frac{1}{|\Omega|}\sum_{(h,k)\in\Omega} \text{SmoothL1}\left(\text{Norm}(\hat{\mathbf{F}}^f_{h,k}), \text{Norm}(\mathbf{F}^\star_{h,k})\right), \tag{6}$$

where h and k index future horizons and slots, respectively, and Ω denotes valid pairs. Finally, the predicted slots are compressed into the foresight context

$$\mathbf{C}^f = \phi_{\text{comp}}(\hat{\mathbf{F}}^f), \tag{7}$$

which is passed to the planner. Since the planning loss backpropagates through $\mathbf{C}^f$, ϕ_{comp}, f_{pred}, and $\mathbf{R}^f$, the learned foresight is shaped jointly by observed future evolution and its downstream planning role.

3.3 Utility Aware Selective Foresight

A foresight representation that improves planning on average need not be useful for every scene. History may already support a reliable decision, while prediction errors may reduce or even reverse the benefit of foresight. We therefore train Utility Aware Selective Foresight (UASF) using the realized planning gain ΔU defined in Eq. equation 2, so that routing reflects the benefit actually delivered by the learned predictor.

The router operates before future prediction and uses only information from the observed history. Following Fig. 1, we aggregate the history tokens $\mathbf{H}_t$ and combine them with the encoded ego states:

$$p_t = \sigma\left(g_{\text{route}}\left(\phi_{\text{agg}}(\mathbf{H}_t), \psi_s(\mathbf{s}_{\le t})\right)\right), \tag{8}$$

where ϕ_{agg} aggregates the history tokens, ψ_s encodes the ego state history, and $p_t \in [0,1]$ denotes the routing score. We label foresight as useful when its realized gain exceeds a margin m:

$$y_t = \mathbb{I}[\Delta U > m], \qquad w_t = \text{clip}\left(\frac{|\Delta U - m|}{\alpha}, 0, 1\right), \tag{9}$$

where m specifies the minimum planning improvement treated as useful and α controls the weighting scale. Samples close to the margin receive lower weight because their routing labels are less decisive. The routing objective is:

$$\mathcal{L}_{\text{rou}} = w_t\,\text{BCE}(p_t, y_t). \tag{10}$$

At inference, future prediction is executed only when $p_t > \delta$. Otherwise, FD-WAM directly follows the base path:

$$\boldsymbol{\tau} = \begin{cases} \boldsymbol{\tau}^{\text{fore}}, & p_t > \delta, \\ \boldsymbol{\tau}^{\text{base}}, & p_t \le \delta. \end{cases} \tag{11}$$

The threshold δ controls the foresight activation rate and therefore the tradeoff between planning performance and computation. Since the router is trained with a binary target, p_t indicates whether the realized gain is likely to exceed m, rather than predicting the magnitude of ΔU.

3.4 Training Objective

We adopt a two stage training strategy. We first optimize the foresight representation and planner using $\mathcal{L}_{\text{plan}}$ and $\mathcal{L}_{\text{wam}}$. After the foresight branch has learned meaningful future representations, we introduce the routing objective. This ensures that the routing supervision reflects the utility of a learned predictor rather than its early prediction errors. The full objective is:

$$\mathcal{L} = \mathcal{L}_{\text{plan}}^{\text{fore}} + \lambda_{\text{base}}\mathcal{L}_{\text{plan}}^{\text{base}} + \lambda_{\text{wam}}\mathcal{L}_{\text{wam}} + \lambda_{\text{rou}}\mathcal{L}_{\text{rou}}. \tag{12}$$

Here $\mathcal{L}_{\text{plan}}^{\text{fore}}$ and $\mathcal{L}_{\text{plan}}^{\text{base}}$ supervise the two planning paths, while $\mathcal{L}_{\text{wam}}$ trains future prediction and $\mathcal{L}_{\text{rou}}$ supervises selective foresight. At inference, the future teacher and ground truth future observations are removed. The router is evaluated before the future predictor, so future prediction is executed only when $p_t > \delta$.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate FD-WAM on three benchmarks: NAVSIM v1 (Dauner et al., 2024), NAVSIM v2 (Cao et al., 2025), and HUGSIM (Zhou et al., 2026a). NAVSIM v1 evaluates predicted trajectories with non-reactive simulation, using PDMS as the primary metric. NAVSIM v2 reports the extended EPDMS metric. We further evaluate zero-shot closed-loop performance on HUGSIM, where the predicted trajectory is executed in an interactive simulator.

Implementation Details. FD-WAM adopts DINOv2 ViT-S/14 as the visual backbone. The model takes four historical frames from four surrounding cameras and extracts 16 register tokens for each camera.The foresight branch predicts future representations at four horizons and compresses them into 16 condition tokens for planning. The planner operates on 64 trajectory proposals. We optimize the model with AdamW for 20 epochs using a global batch size of 64, an initial learning rate of 2×10^{-4}, and a weight decay of 0.01. The learning rate is linearly warmed up over the first 10% of training steps and then decayed to zero with a cosine schedule. Training uses mixed precision on 8 NVIDIA H800 GPUs without gradient clipping.

4.2 Quantitative Results

Results on NAVSIM. As shown in Tables 1 and 2, FD-WAM achieves strong performance under both *train* and *trainval* settings. On NAVSIM v1, FD-WAM achieves 93.5 PDMS using the *train* split and further improves to **94.4 PDMS** with *trainval*, yielding the best performance among the compared methods. On NAVSIM v2, the corresponding EPDMS increases from 90.5 to **91.3**, approaching the strongest result of 91.7 reported by WA-JEPA. Notably, FD-WAM uses only a ViT-S backbone: on NAVSIM v1, the few methods that surpass our *train* setting, RAP-DINO and ChainFlow-VLA, employ ViT-H+ and ViT-S with InternVL, respectively, while on NAVSIM v2, WA-JEPA relies on a larger ViT-L backbone with additional driving domain pretraining. These results show that FD-WAM achieves strong planning performance with a comparatively lightweight visual backbone and limited training scale.

Results on HUGSIM. Table 3 reports zero-shot closed-loop evaluation on HUGSIM. FD-WAM achieves an average RC of **49.4** and the best average HD-Score of **40.2** among the compared methods. It further ranks first on the Easy and Medium subsets, with HD-Scores of 74.3 and 53.2. Performance is less dominant on Hard and Extreme scenarios, indicating that difficult interactions remain more sensitive to prediction errors and closed-loop accumulation. These results demonstrate effective transfer of the learned foresight from offline planning to interactive closed-loop driving.

Inference Efficiency. As shown in Table 4, FD-WAM activates foresight for only **20%** of scenes, achieving **93.5 PDMS** with a mean latency of **50 ms**. Compared with always using foresight, FD-WAM reduces the mean latency from 55 ms to 50 ms, while improving PDMS from 93.3 to 93.5. It adds only 3 ms over the base planner, yet improves PDMS by 0.7. FD-WAM also surpasses Drive-JEPA in both planning performance and latency, while remaining substantially faster than

Table 1: Comparison on the NAVSIM v1 *navtest* benchmark. All metrics are higher-is-better.

Method	Backbone	NC ↑	DAC ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑
Vision-Based End-to-End Methods							
DriveSuprim† (2026a)	ViT-L	98.6	98.6	91.3	95.5	100.0	93.5
RAP-DINO† (2026)	ViT-H+	99.1	98.9	90.3	96.7	100.0	93.8
DrivoR (2026)	ViT-S	98.9	98.3	89.1	96.2	100.0	93.1
Vision-Language-Action Methods							
DriveWorld-VLA (2026)	R34 + InternVL	99.1	98.2	85.9	96.1	100.0	91.3
DriveFine (2026)	SigLIP-384	98.8	98.6	86.9	96.2	100.0	91.8
Curious-VLA (2026a)	Qwen2.5-VL	99.1	97.1	86.3	95.2	100.0	90.3
DriveVA (2026a)	Wan2.2 VAE	99.2	97.5	83.5	98.7	100.0	90.9
SpanVLA (2026c)	Qwen2.5-VL	99.1	97.1	86.3	95.2	100.0	90.3
ChainFlow-VLA (2026e)	ViT-S + InternVL	98.8	98.6	90.8	96.1	100.0	93.6
World-Model-Based Methods							
Drive-JEPA (2026c)	ViT-L‡	99.1	98.2	90.8	95.9	99.9	93.3
DriveDreamer-Policy (2026b)	Qwen3-VL	98.4	97.1	83.5	95.1	100.0	89.2
DAWN (2026b)	ViT-L	98.7	95.9	84.3	96.0	100.0	89.1
DriveWAM (2026)	Wan2.2	98.3	98.1	84.3	95.2	100.0	90.1
Metis (2026a)	Wan2.2	98.5	97.5	87.9	97.8	99.8	90.3
GeoWorldAD (2026b)	EgoStreamVGGT	99.0	97.8	85.9	95.8	99.9	91.0
Auto-JEPA (2026b)	ViT-L	98.4	98.3	87.1	95.0	100.0	91.3
Adaptive-WAM (2026)	Wan2.2 + S	99.8	98.3	86.6	98.3	100.0	92.6
SimWAM (2026)	Wan2.2	98.4	98.7	86.4	95.5	100.0	91.5
BrainWAM (2026)	Wan2.2 + Qwen3-VL	98.1	97.5	83.8	94.9	100.0	89.5
WA-JEPA (2026d)	ViT-L‡	99.5	98.3	85.0	97.7	100.0	91.8
FD-WAM$_{train}$	ViT-S	99.1	98.4	90.4	96.4	100.0	<u>93.5</u>
FD-WAM$_{trainval}$	ViT-S	98.9	99.0	92.1	96.1	100.0	**94.4**

† Additional training data beyond the standard NAVSIM training split. ‡ Additional driving-domain pretraining of the backbone.

ChainFlow-VLA. These results show that selective foresight retains the planning benefit of future prediction while avoiding unnecessary computation.

4.3 Ablation Studies

Component Ablation. Table 5 shows consistent gains from all components. Future prediction improves PDMS from 92.8 to 93.0, and DRFR further increases it to 93.3 while improving EP and TTC to 90.4 and 96.1. Adding UASF yields the best overall result of **93.5 PDMS**, with NC and TTC reaching 99.1 and 96.4. The three components contribute successive gains of 0.2, 0.3, and 0.2 PDMS, for a total improvement of 0.7 over the base planner.

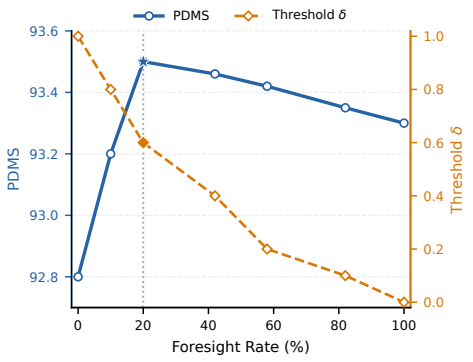


Figure 2: Foresight budget trade-off.

Foresight Representation Ablation. Table 6 compares different future representations under the same prediction horizon and planner. Compact Foresight slightly improves PDMS from 93.0 to 93.1 over Full Foresight, indicating that reducing the future context does not sacrifice planning performance and may help suppress information that is less useful for the current decision. DRFR further increases PDMS to **93.5**, with the best NC, DAC, and TTC of 99.1, 98.4, and 96.4, respectively. These results suggest that the main benefit

Table 2: Comparison on the NAVSIM v2 `navtest` benchmark. All metrics are higher-is-better.

Method	Backbone	NC	DAC	DDC	TLC	EP	TTC	LK	HC	EC	EPDMS
Vision-Based End-to-End Methods											
DiffusionDrive (2025)	ResNet34	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
DiffusionDriveV2 (2025)	ResNet34	97.7	96.6	99.2	99.8	88.9	97.2	96.0	97.8	91.0	87.5
Vision-Language-Action Methods											
DriveVLA-W0 (2026b)	Emu3-8B	98.5	99.1	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1
ExploreVLA (2026)	Show-o	98.8	96.2	99.6	99.8	87.1	98.2	97.8	98.3	86.8	88.8
LWDrive (2026a)	Qwen2.5-VL	98.8	98.4	99.0	99.7	90.3	98.6	96.3	97.9	73.3	89.6
World-Model-Based Methods											
DriveLaW (2026)	LTX-Video-2B	98.7	96.9	99.6	99.8	87.5	98.3	97.6	98.4	77.4	88.6
Drive-JEPA (2026c)	ViT-L‡	98.4	98.6	99.1	99.8	88.4	97.8	97.6	97.9	84.8	87.8
Discrete-WAM† (2026b)	Transformer	98.5	98.2	99.7	99.8	90.5	97.9	97.2	98.3	78.1	90.4
Auto-JEPA (2026b)	ViT-L	98.5	98.7	98.3	99.7	90.5	97.9	94.7	97.8	75.2	89.1
SimWAM† (2026)	Wan2.2	98.6	98.0	99.7	99.9	87.5	98.4	97.9	98.3	84.4	90.2
WA-JEPA† (2026d)	ViT-L‡	99.4	98.2	99.7	99.9	87.8	98.9	98.3	98.3	88.1	**91.7**
GeoWAM (2026c)	ViT-L	98.7	97.7	99.7	99.9	87.0	98.1	97.9	98.3	86.8	90.2
SV-WAM† (2026a)	Wan2.2	98.6	98.8	99.6	99.9	86.8	98.3	97.8	98.4	88.4	91.0
FD-WAM$_{train}$	ViT-S	99.0	98.8	99.6	99.6	90.7	98.5	96.2	97.9	70.1	90.5
FD-WAM$_{trainval}$	ViT-S	99.3	99.2	99.7	99.7	88.3	98.8	96.4	98.2	78.7	<u>91.3</u>

† Additional training data beyond the standard NAVSIM training split. ‡ Additional driving-domain pretraining of the backbone.

Table 3: Zero-shot closed-loop evaluation on the HUGSIM benchmark. E, M, H, and X denote Easy, Medium, Hard, and Extreme scenarios, respectively. All metrics are higher-is-better.

Method	Backbone	RC ↑					HD-Score ↑				
		E	M	H	X	Avg.	E	M	H	X	Avg.
VAD (2023)	ResNet50	51.3	31.1	25.3	26.5	31.4	36.3	9.5	8.0	11.5	13.4
LTF (2023)	ResNet34	67.8	35.1	26.2	40.5	38.9	58.9	18.0	9.8	25.9	23.7
UniAD (2023)	ResNet101	78.4	60.5	33.6	17.8	45.9	64.9	45.8	20.6	6.6	32.7
DrivoR (2026)	ViT-S	80.9	50.5	33.8	47.1	49.8	73.3	34.6	18.8	32.5	35.7
Latent-WAM (2026b)	ViT-B	84.2	42.5	30.6	35.5	45.9	72.5	24.0	12.2	18.1	28.9
MM-Future (2026b)	ViT-S	64.8	51.5	39.5	22.8	44.5	53.8	40.0	27.3	8.6	32.3
GigaPixel (2026)	ViT-S	78.6	60.7	32.7	35.4	50.1	67.4	51.9	19.1	21.6	38.5
FD-WAM	ViT-S	82.4	57.5	29.3	39.5	49.4	**74.3**	**53.2**	17.2	25.6	**40.2**

comes not from compression alone, but from shaping the compact future representation with the planning objective.

Foresight Selection Strategy. Table 7 compares different foresight selection strategies. Random routing achieves 92.9 PDMS, while prioritizing scenes with larger historical dynamics improves it to 93.1. UASF further raises PDMS to **93.5**, outperforming Random and Dynamics by 0.6 and 0.4 points, respectively. These results show that scene dynamics alone are insufficient for deciding when foresight is useful, whereas routing based on realized planning utility better identifies scenes that benefit from future prediction.

Foresight Budget Trade-off. Figure 2 studies planning performance under different foresight budgets. Increasing the foresight rate from 0% to 20% improves PDMS from 92.8 to a peak of **93.5**, whereas further increasing the budget gradually reduces performance. When foresight is applied to all scenes, PDMS drops to 93.2 despite the additional computation. This nonmonotonic trend shows that the benefit of future prediction is concentrated in a subset of scenes, supporting selective allocation of foresight rather than uniform execution.

Table 4: Inference efficiency on NAVSIM v1. Latency is measured on a single NVIDIA H800 GPU with batch size 1. Future Rate denotes the percentage of scenes where the foresight branch is activated. For FD-WAM, we use the default threshold $\delta = 0.8$ selected on the validation set, resulting in a foresight rate of approximately 20%. *Base* denotes the planning-only model without future prediction. *Foresight* denotes the always-on variant with the complete future branch.

Method	Backbone	Future Rate	Latency (ms)		PDMS ↑
			Mean ↓	P99 ↓	
Recent Methods					
Drive-JEPA (2026c)	ViT-L‡	–	57	59	93.3
ChainFlow-VLA (2026e)	ViT-S + InternVL	–	624	627	93.6
FD-WAM Variants					
Base	ViT-S	0%	47	49	92.8
Foresight (w/o DRFR)	ViT-S	100%	66	68	93.0
Foresight	ViT-S	100%	55	58	93.3
FD-WAM	ViT-S	20%	50	52	**93.5**

Table 5: Component ablation of FD-WAM on NAVSIM v1.

Future	DRFR	UASF	NC ↑	DAC ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑
–	–	–	98.8	98.2	90.1	95.6	100.0	92.8
✓	–	–	98.8	98.4	90.1	95.7	100.0	93.0
✓	✓	–	98.9	98.4	90.4	96.1	100.0	93.3
✓	✓	✓	99.1	98.4	90.4	96.4	100.0	**93.5**

Foresight Utility Analysis. Figure 3 compares the routing score p with the realized planning gain. Low scores are mostly associated with small or negative gains, while larger positive gains concentrate at high routing scores. This clear association indicates that UASF can identify scenes where foresight is more likely to improve planning, providing the basis for selective future computation.

4.4 Qualitative Analysis

Figure 4 shows where the scene tokens attend in future frames for two challenging driving cases: dense multi-agent interaction and cut-in. In the dense scene, the attention covers several nearby vehicles and the driving corridor, indicating that the model considers multiple agents whose future motion may affect the ego vehicle. In the cut-in scene, the attention is concentrated on the vehicle moving into the ego lane, which is the most relevant source of future change for planning. The attended regions also change across future horizons, suggesting that FD-WAM captures time-varying cues instead of using a fixed spatial prior. These results qualitatively suggest that the learned foresight representation emphasizes future information that is relevant to planning decisions.

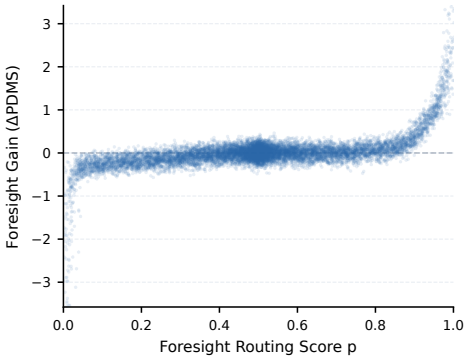


Figure 3: Relationship between predicted foresight utility and realized planning gain on NAVSIM v1.

5 Conclusion

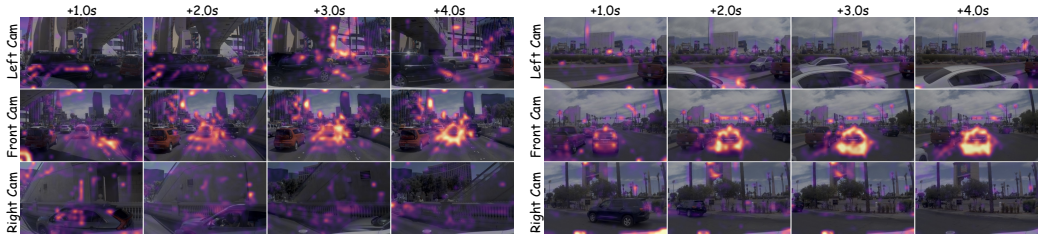
We presented FD-WAM, a framework that studies future prediction through its utility to planning. Instead of treating foresight as uniformly beneficial, FD-WAM jointly considers *what to predict* and

Table 6: Ablation of future representations on NAVSIM v1. All variants use the same future prediction horizon and planner.

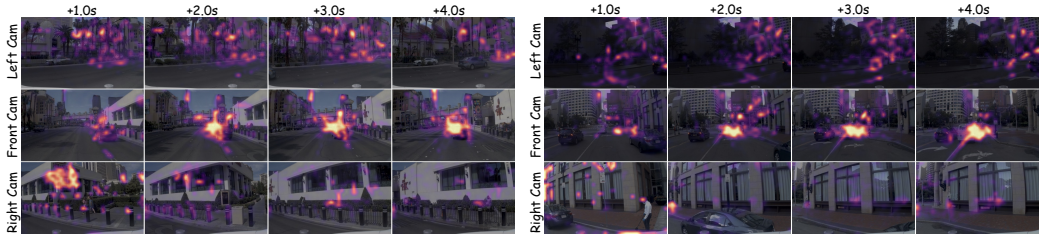
Future Representation	NC ↑	DAC ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑
Full Foresight	98.7	98.2	90.9	95.3	100.0	93.0
Compact Foresight	99.0	98.2	90.4	95.9	100.0	93.1
DRFR (Ours)	99.1	98.4	90.4	96.4	100.0	**93.5**

Table 7: Comparison of selective foresight strategies on NAVSIM v1. Dynamics prioritizes scenes with larger temporal variation in historical visual features.

Routing Strategy	NC ↑	DAC ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑
Random	98.5	98.2	91.2	94.9	100.0	92.9
Dynamics	98.8	98.2	90.6	95.7	100.0	93.1
UASF (Ours)	99.1	98.4	90.4	96.4	100.0	**93.5**



(a) Dense multi-agent interaction



(b) Cut-in interaction

Figure 4: Attention maps of scene tokens over future RGB frames in two representative scenarios: (a) dense multi agent interaction and (b) cut in interaction. Warmer regions indicate stronger attention, which concentrates on interacting vehicles and nearby areas relevant to the ego decision across future horizons.

when to predict. Decision Relevant Foresight Representation learns a compact future context shaped by both future observations and planning supervision, while Utility Aware Selective Foresight invokes future prediction according to the planning benefit it is likely to provide. Experiments on NAVSIM v1, NAVSIM v2, and zero shot HUGSIM show strong planning performance and a favorable balance between planning quality and inference cost. Our analyses further reveal that foresight gains vary substantially across scenes, supporting the view that future modeling should be guided by its contribution to the driving decision rather than prediction completeness alone.

Limitations. Selective foresight still depends on the quality of the learned future predictor and the reliability of the planning score used for routing supervision. The current router also makes a binary decision based on a fixed planning-gain margin, without explicitly estimating gain magnitude or uncertainty. Future work could explore more robust utility estimation and uncertainty-aware routing for challenging or shifted driving conditions.

STATEMENT

AI Use Statement. AI tools were used to assist with research ideation and discussion, literature retrieval and discovery, manuscript organization and drafting, language refinement, and the presentation and interpretation of experimental results. They were also used to assist with assembling and formatting supplementary videos. All underlying video content is derived directly from our original experimental outputs and was not synthetically generated or modified in a way that changes the experimental results. All final methodological choices, implementations, experiments, quantitative results, analyses, and scientific claims were carefully reviewed and verified by the authors. The authors take full responsibility for the accuracy, integrity, and final content of the manuscript and supplementary materials.

Reproducibility Statement We provide the model architecture, training objectives, routing formulation, and inference procedure in the main paper. Implementation details, including the visual backbone, input configuration, optimization settings, training schedule, and evaluation protocols, are specified in the experimental section and appendix. All experiments are conducted on publicly available benchmarks using their official evaluation metrics. Source code and configuration files will be made available upon publication to facilitate reproduction of the reported results.

REFERENCES

Sining Ang, Yuguang Yang, and Yan Wang. Adaptive-WAM: Quality-Guided Early-Exit Planning from Intermediate Video-Diffusion Features. *arXiv preprint arXiv:2608.06008*, 2026.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

Canyu Chen, Yuguang Yang, Zhewen Tan, Yizhi Wang, Ruiyi Zhan, Haiyan Liu, Xuanyao Mao, Jason Bao, Xinyue Tang, Linlin Yang, Bingchuan Sun, Yan Wang, and Baochang Zhang. Devil is in Narrow Policy: Unleashing Exploration in Driving VLA Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, 2026a.

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, 2024.

Zisheng Chen, Yuping Qiu, Jianhua Han, Tao Tang, Xiuwei Chen, Likui Zhang, Ying-Cong Chen, Hang Xu, and Xiaodan Liang. Intend, Reflect, Refine: An Adaptive Multimodal Reflection Framework for Autonomous Driving. *arXiv preprint arXiv:2606.22913*, 2026b.

Pranav Singh Chib and Pravendra Singh. Recent Advancements in End-to-End Autonomous Driving Using Deep Learning: A Survey. *IEEE Transactions on Intelligent Vehicles*, 9(1):103–118, 2024.

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.

Chenxu Dang, Sining Ang, Yongkang Li, Haochen Tian, Jie Wang, Guang Li, Hangjun Ye, Jie Ma, Long Chen, and Yan Wang. DriveFine: Refining-Augmented Masked Diffusion VLA for Efficient and Robust Driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 412–429, 2026.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28706–28719, 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 *International Conference on Learning Representations*, 2026.

Tuo Feng, Wenguan Wang, and Yi Yang. A Survey of World Models for Autonomous Driving. *arXiv preprint arXiv:2501.11260*, 2025.

Yanchen Guan, Haicheng Liao, Zhenning Li, Jia Hu, Runze Yuan, Yunjian Li, Guohui Zhang, and Chengzhong Xu. World Models for Autonomous Driving: An Initial Survey. *IEEE Transactions on Intelligent Vehicles*, pp. 1–17, 2024. Early Access.

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.

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

Feiyang Jia, Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, and Long Chen. DriveWorld-VLA: Unified Latent-Space World Modeling with Vision-Language-Action for Autonomous Driving. *arXiv preprint arXiv:2602.06521*, 2026.

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

Hao Jiang, Zhipeng Zhang, Yu Gao, Zhigang Sun, Yiru Wang, Yuwen Heng, Shuo Wang, Jinhao Chai, Zhuo Chen, Hao Zhao, Hao Sun, Xi Zhang, Anqing Jiang, and Chuan Hu. FlowDrive: Energy Flow Field for End-to-End Autonomous Driving. *arXiv preprint arXiv:2509.14303*, 2025.

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

Jingyu Li, Zhe Liu, Dongnan Hu, Junjie Wu, Zipei Ma, Wenxiao Wu, Chao Han, Zhihui Hao, Zhikang Liu, Kun Zhan, Jiankang Deng, Xiatian Zhu, and Li Zhang. Metis: A Generalizable and Efficient World-Action Model for Autonomous Driving and Urban Navigation. *arXiv preprint arXiv:2606.15869*, 2026a.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. DriveVLA-W0: World Models Amplify Data Scaling Law in Autonomous Driving. In *International Conference on Learning Representations*, 2026b.

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

Mengmeng Liu, Diankun Zhang, Jiuming Liu, Jianfeng Cui, Hongwei Xie, Guang Chen, Hangjun Ye, Michael Ying Yang, Francesco Nex, and Hao Cheng. DriveVA: Video Action Models are Zero-Shot Drivers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 315–335, 2026a.

Shuai Liu, Hechangle Gong, Hao Jiang, Runlin He, Junxiang Zhan, Kai Huang, Sheng Yang, and Shaoqing Ren. MM-Future: Multi-Mode Joint World-Action Modeling for Autonomous Driving. *arXiv preprint arXiv:2609.20377*, 2026b.

Hongbo Lu, Liang Yao, Chenghao He, Hao Han, Fan Liu, Wenlong Liao, Tao He, and Pai Peng. RISE: Adaptive Imagination for World Action Models. *arXiv preprint arXiv:2608.20430*, 2026a.

Hongbo Lu, Liang Yao, Chenghao He, Haoyu Wang, Xiang Gu, Xianfei Li, Wenlong Liao, Tao He, and Pai Peng. The DAWN of World-Action Interactive Models. *arXiv preprint arXiv:2605.11550*, 2026b.

Yiren Lu, Xin Ye, Jiaming Liu, Philip Jacobson, Jin Yao, Yi-chung Chen, Liam Merino, Dhruva Dixith Kurra, Min Cai, Tom Lampo, Yu Yin, Danhua Guo, and Burhan Yaman. GeoWAM: Visual Geometry World Action Models for Autonomous Driving. *arXiv preprint arXiv:2608.23486*, 2026c.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, Long Chen, Bing Wang, and Zhi-xin Yang. AdaThinkDrive: Adaptive Thinking via Reinforcement Learning for Autonomous Driving. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, Liping Jing, Yiming Nie, and Bin Dai. DriveWorld: 4D Pre-trained Scene Understanding via World Models for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15522–15533, 2024.

Luke Rowe, Roger Girgis, Rodrigue de Schaetzen, Daphne Cornelisse, Alaap Grandhi, Felix Heide, Eugene Vinitsky, Christopher Pal, and Liam Paull. Scaling Self-Play for End-to-End Driving. *arXiv preprint arXiv:2606.19641*, 2026.

Shuyao Shang, Bing Zhan, Yunfei Yan, Yuqi Wang, Yingyan Li, Yasong An, Xiaoman Wang, Jierui Liu, Lu Hou, Lue Fan, Zhaoxiang Zhang, and Tieniu Tan. DynVLA: Learning World Dynamics for Action Reasoning in Autonomous Driving. In *International Conference on Machine Learning*, 2026.

Zihao Sheng, Xin Ye, Jingru Luo, Sikai Chen, and Liu Ren. ExploreVLA: Dense World Modeling and Exploration for End-to-End Autonomous Driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 266–284, 2026.

Chen Shi, Jinrui Xu, Shaoshuai Shi, Kehua Sheng, Bo Zhang, and Li Jiang. DriveWAM: Video Generative Priors Enable Scalable World-Action Modeling for Autonomous Driving. *arXiv preprint arXiv:2605.28544*, 2026.

Jinyang Wang, Shiwei Li, Junjian Wang, Zhiqiang Deng, Jianbin Gao, Yihang Zhao, Liu Liu, Yongjia Zhao, Jinlong Chen, Huirui Xu, Yifeng Pan, Kangwei Liu, Fan Ren, Ji Tao, and Minghao Yang. SV-WAM: An Efficient Surround-View World-Action Model for End-to-End Autonomous Driving. *arXiv preprint arXiv:2609.03602*, 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.

Linhan Wang, Zichong Yang, Chen Bai, Guoxiang Zhang, Xiaotong Liu, Xiaoyin Zheng, Xiao-Xiao Long, Chang-Tien Lu, and Cheng Lu. Drive-JEPA: Video JEPA Meets Multimodal Trajectory Distillation for End-to-End Driving. *arXiv preprint arXiv:2601.22032*, 2026c.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards Real-World-Drive World Models for Autonomous Driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 55–72, 2024a.

Xinlin Wang, Yujiao Xiang, Yuheng Zhou, Jingqi Wang, Minqing Huang, Jiajie Huang, Dongxu Wei, Tingguang Zhou, Xiyang Wang, Gong Chen, Zhi Xu, Feiyang Tan, Hangning Zhou, and Mu Yang. WA-JEPA: Rethinking the Video JEPA Paradigm for World-Action Modeling in Autonomous Driving. *arXiv preprint arXiv:2608.20974*, 2026d.

Xiyang Wang, Xinlin Wang, Tingguang Zhou, Gong Chen, Xingtai Gui, Zhi Xu, Xiaolei Wu, Feiyang Tan, Hangning Zhou, and Mu Yang. ChainFlow-VLA: Causal Flow Planning with Vision-Language Models. *arXiv preprint arXiv:2605.23270*, 2026e.

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, Wenyu Liu, and Xinggang Wang. 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)*, pp. 39701–39712, 2026.

Chen Yang, Yuhao Wei, Ze Xu, Ziheng Zou, Shuang Liang, Delin Ouyang, Lingfeng Qi, Jie Li, and Guofa Li. LWDrive: Layer-Wise World-Model-Guided Vision-Language Model Planning for Autonomous Driving. *arXiv preprint arXiv:2606.29879*, 2026a.

Jiwei Yang, Zhengxian Chen, Chaosheng Huang, and Jun Li. Auto-JEPA: A Latent World Model of Continuous Intent for End-to-End Autonomous Driving. *arXiv preprint arXiv:2607.29031*, 2026b.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(14):11910–11918, 2026a.

Ziyang Yao, Haochen Liu, Yuncheng Jiang, Zeyu Zhu, Zibin Guo, Jingru Wang, Tianle Liu, Jianwei Cui, Kuiyuan Yang, Hongwei Xie, Jingwei Zhao, Guang Chen, and Hangjun Ye. DiscreteWAM: Unified Discrete Vision-Action Token Editing for World-Policy Learning. *arXiv preprint arXiv:2606.05645*, 2026b.

Bing Zhan, Shuyao Shang, Jiahao Gu, Shuo Lu, Yuan Xu, Zhao Wang, Yida Wang, Xueyang Zhang, Kun Zhan, Lue Fan, and Zhaoxiang Zhang. BrainWAM: Action-Space Coordination of Semantic Priors and Predictive Dynamics for Autonomous Driving. *arXiv preprint arXiv:2608.12854*, 2026.

Jinqing Zhang, Zehua Fu, Zelin Xu, Wenying Dai, Qingjie Liu, and Yunhong Wang. ResWorld: Temporal Residual World Model for End-to-End Autonomous Driving. In *International Conference on Learning Representations*, 2026a.

Songyan Zhang, Jinyuan Tian, Hanbing Li, Daqi Liu, Hao Chen, Wenhui Huang, Fang Li, Guang Chen, Hangjun Ye, Long Chen, Kuiyuan Yang, and Chen Lv. GeoWorldAD: Geometry World Action Model for Autonomous Driving. *arXiv preprint arXiv:2607.17521*, 2026b.

Zongchuang Zhao, Xin Zhou, Tianyang Xu, Zhengyang Sun, Kaixuan Zhou, Yu Wu, Honglin Li, Dingkang Liang, and Xiang Bai. SimWAM: A Simple World Action Model for End-to-End Autonomous Driving. *arXiv preprint arXiv:2608.07468*, 2026.

Ruiguo Zhong, Benshan Ma, Xiaolong Chen, Lang Zhang, Mingyue Feng, Yaonong Wang, Pei Liu, and Jun Ma. DA-WAM: Decision-Aligned Future Latents for Driving World Models. *arXiv preprint arXiv:2608.19085*, 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4):4673–4691, 2026a.

Yang Zhou, Xiaofeng Wang, Hao Shao, Letian Wang, Guosheng Zhao, Jiangnan Shao, Jiagang Zhu, Tingdong Yu, Zheng Zhu, Guan Huang, and Steven L. Waslander. DriveDreamer-Policy: A Geometry-Grounded World-Action Model for Unified Generation and Planning. *arXiv preprint arXiv:2604.01765*, 2026b.

Zewei Zhou, Ruining Yang, Xuewei (Tony) Qi, Yiluan Guo, Sherry X. Chen, Tao Feng, Kateryna Pistunova, Yishan Shen, Lili Su, and Jiaqi Ma. SpanVLA: Efficient Action Bridging and Learning from Negative-Recovery Samples for Vision-Language-Action Model. *arXiv preprint arXiv:2604.19710*, 2026c.

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

A.1 Evaluation Metrics

NAVSIM v1. NAVSIM v1 (Dauner et al., 2024) evaluates planning with the Planning Data Metric Score (PDMS), which combines no-collision (NC), drivable-area compliance (DAC), time-to-collision (TTC), ego progress (EP), and comfort (C):

$$\mathrm{PDMS} = \mathrm{NC} \cdot \mathrm{DAC} \cdot \frac{5\mathrm{TTC} + 5\mathrm{EP} + 2\mathrm{C}}{12}. \tag{13}$$

NC and DAC act as multiplicative penalties, while TTC, EP, and C form a weighted score.

NAVSIM v2. NAVSIM v2 (Cao et al., 2025) adopts the Extended Planning Data Metric Score (EPDMS). Compared with PDMS, EPDMS adds driving-direction compliance (DDC), traffic-light compliance (TLC), lane keeping (LK), history comfort (HC), extended comfort (EC), and a false-positive penalty filtering mechanism. Let $\mathcal{M}_{\mathrm{mul}} = \{\mathrm{NC}, \mathrm{DAC}, \mathrm{DDC}, \mathrm{TLC}\}$ and $\mathcal{M}_{\mathrm{avg}} = \{\mathrm{TTC}, \mathrm{EP}, \mathrm{HC}, \mathrm{LK}, \mathrm{EC}\}$. Following the official definition,

$$\mathrm{EPDMS} = \prod_{m \in \mathcal{M}_{\mathrm{mul}}} \mathrm{filter}_m(\mathrm{agent}, \mathrm{human}) \cdot \frac{\sum_{m \in \mathcal{M}_{\mathrm{avg}}} w_m \mathrm{filter}_m(\mathrm{agent}, \mathrm{human})}{\sum_{m \in \mathcal{M}_{\mathrm{avg}}} w_m}, \tag{14}$$

where $w_{\mathrm{TTC}} = w_{\mathrm{EP}} = 5$ and $w_{\mathrm{HC}} = w_{\mathrm{LK}} = w_{\mathrm{EC}} = 2$. The filtering function is

$$\mathrm{filter}_m(\mathrm{agent}, \mathrm{human}) = \begin{cases} 1.0, & \text{if } m(\mathrm{human}) = 0, \\ m(\mathrm{agent}), & \text{otherwise.} \end{cases} \tag{15}$$

HUGSIM. HUGSIM (Zhou et al., 2026a) evaluates driving in a closed-loop simulator. At each simulation step, NC and DAC are treated as driving-policy items, while TTC and comfort (COM) are combined as contributory items:

$$\text{HD-Score}_t = \prod_{m \in \{\mathrm{NC}, \mathrm{DAC}\}} \mathrm{score}_{m,t} \cdot \frac{\sum_{m \in \{\mathrm{TTC}, \mathrm{COM}\}} \alpha_m \mathrm{score}_{m,t}}{\sum_{m \in \{\mathrm{TTC}, \mathrm{COM}\}} \alpha_m}. \tag{16}$$

The final HD-Score further accounts for route completion R_c:

$$\text{HD-Score} = R_c \cdot \frac{1}{T} \sum_{t=1}^{T} \text{HD-Score}_t, \tag{17}$$

where T is the number of simulation steps. Unlike NAVSIM, HUGSIM evaluates the planner after rolling out its actions in closed loop.

A.2 Additional Implementation Details

Training setup. Unless otherwise specified, FD-WAM is trained with AdamW using an initial learning rate of 2×10^{-4} and cosine learning-rate decay. We use mixed-precision training on 8 NVIDIA H800 GPUs with a per-GPU batch size of 8. For NAVSIM experiments, we report both the train and trainval settings following the data split used in the main paper. The HUGSIM evaluation is conducted in a zero-shot manner without finetuning on HUGSIM.

Inference protocol. At inference time, the router is evaluated before the future prediction branch. If the predicted future-need probability is below the threshold δ, FD-WAM directly uses the no-foresight branch; otherwise, it executes the future prediction branch and uses the foresight-conditioned planner. We select the default threshold on the validation set and use the same threshold for the reported test results. Latency is measured on a single NVIDIA H800 GPU with batch size 1, and we report both mean and P99 latency.

Training and inference objective hyperparameters. To improve reproducibility, we summarize the main hyperparameters used in the FD-WAM training and inference objective in Table 8. The table also expands the branch-level planning loss into trajectory imitation and proposal scoring terms, and further lists the sub-losses used for proposal scoring.

Table 8: Training and inference objective hyperparameters of FD-WAM. We report the effective loss weights and auxiliary hyperparameters used for training and inference.

Category	Item	Symbol	Weight / Value
Overall objective			
Foresight planning	Branch planning loss	$\mathcal{L}_{\text{plan}}^{\text{fore}}$	1.0
Future-token prediction	WAM loss	$\mathcal{L}_{\text{wam}}$	$\lambda_{\text{wam}} = 0.5$
No-foresight planning	Base-branch planning loss	$\mathcal{L}_{\text{plan}}^{\text{base}}$	$\lambda_{\text{base}} = 0.3$
Gate supervision	Gate loss	$\mathcal{L}_{\text{rou}}$	$\lambda_{\text{rou}} = 0.1$
Planning objective			
Trajectory imitation	Proposal trajectory loss	$\mathcal{L}_{\text{traj}}$	$\lambda_{\text{traj}} = 1.0$
Trajectory scoring	Proposal score loss	$\mathcal{L}_{\text{score}}$	$\lambda_{\text{score}} = 1.0$
Proposal scoring objective			
Drivable area	Score sub-loss	$\mathcal{L}_{\text{DAC}}$	1.0
Time to collision	Score sub-loss	$\mathcal{L}_{\text{TTC}}$	1.0
No collision	Score sub-loss	$\mathcal{L}_{\text{NC}}$	1.0
Ego progress	Score sub-loss	$\mathcal{L}_{\text{EP}}$	1.0
Driving direction	Score sub-loss	$\mathcal{L}_{\text{DDC}}$	1.0
Comfort	Score sub-loss	$\mathcal{L}_{\text{C}}$	1.0
Auxiliary hyperparameters			
Score-gain margin	Gate target margin	m	0.0
Gain confidence scale	Gate confidence scale	α	0.02
Positive gate weight	BCE positive weight	–	3.0
EMA decay	Teacher update decay	μ	0.99
Inference threshold	Gate threshold	δ	0.8

Table 9: Ablation on the number of future visual tokens. We vary the number of tokens used to summarize predicted future information while keeping all other settings unchanged.

N_t	NC ↑	DAC ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑
4	98.7	98.1	90.8	95.5	100.0	93.0
8	99.0	98.2	90.4	96.0	100.0	93.1
16	99.1	98.4	90.4	96.4	100.0	**93.5**
32	99.1	98.7	89.8	96.5	100.0	93.4

B Appendix B

B.1 Additional Ablation Results

Number of Future Condition Tokens. Table 9 studies the size of the compact foresight context while keeping the backbone, prediction horizons, routing strategy, and training objective unchanged. Increasing the number of condition tokens from 4 and 8 to 16 improves PDMS from 93.0 and 93.1 to **93.5**, respectively. Further increasing the context to 32 tokens yields no additional gain, with PDMS slightly decreasing to 93.4 despite small improvements in DAC and TTC. These results indicate that 16 tokens provide sufficient capacity for the foresight context in our setting, while a larger planning context does not necessarily improve overall performance.

Number of Future Horizons. Table 10 studies the number of future horizons while keeping the backbone, condition token number, routing strategy, and training objective fixed. Using four horizons improves PDMS from 93.1 with two horizons to **93.5**, while increasing the number to eight yields 93.3 PDMS. Although eight horizons slightly improve TTC, the lower EP offsets this gain. Overall, the results show that increasing the number of predicted horizons does not monotonically improve planning, with four horizons providing the best overall performance in our setting.

No-future tokens. We ablate the design of no-future tokens in Table 11. In the base branch, future prediction is skipped, and the planner needs a corresponding representation for the no-future state. We compare three designs: removing the tokens, using zero padding, and using learnable no-

Table 10: Ablation on the number of future horizons. All variants cover the same future range while varying the temporal sampling density.

N_h	Interval (s)	Max Horizon (s)	NC ↑	DAC ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑
2	2.0	4.0	99.0	98.2	90.4	96.0	100.0	93.1
4	1.0	4.0	99.1	98.4	90.4	96.4	100.0	**93.5**
8	0.5	4.0	99.0	98.3	90.2	96.5	100.0	93.3

future tokens. The results show that learnable no-future tokens achieve the best overall performance. Removing the tokens leads to 93.0 PDMS and a high Future Rate of 40%, indicating that the base branch becomes less stable when its input format differs from the future branch. Zero padding keeps the input length unchanged and slightly improves PDMS to 93.1, but still activates future prediction in 35% of the scenes. In contrast, learnable no-future tokens improve PDMS to 93.5 while reducing the Future Rate to 20%. This suggests that a learnable no-future representation provides a better-calibrated base branch, allowing the model to avoid unnecessary future prediction while preserving planning performance.

Table 11: Ablation on no-future tokens. We compare different designs for representing the no-future tokens of base branch while keeping all other settings unchanged.

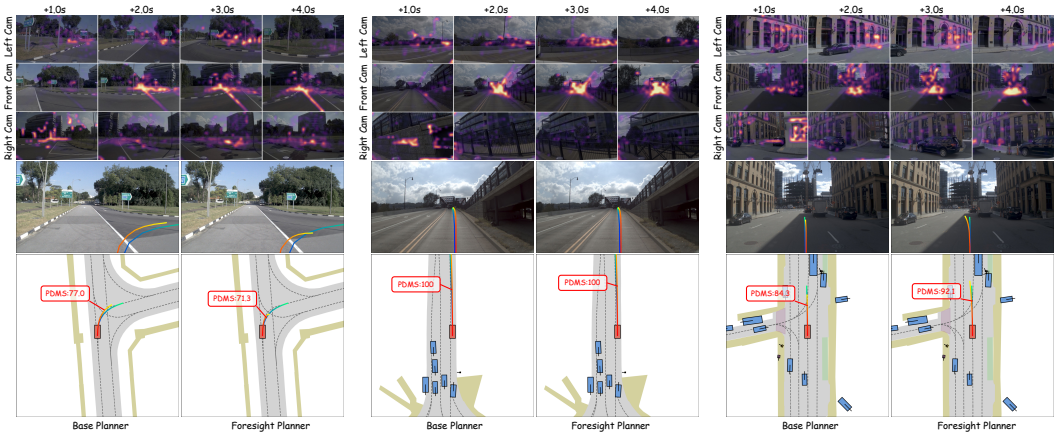
No-future tokens	NC ↑	DAC ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑	Future Rate ↓
No tokens	98.9	98.2	90.0	95.9	100.0	93.0	40%
Zero padding	98.8	98.2	90.6	95.7	100.0	93.1	35%
Learnable tokens	99.1	98.4	90.4	96.4	100.0	**93.5**	**20%**

C Appendix C

C.1 Additional Qualitative Results

Representative Driving Cases. Figure 5 presents three representative outcomes of using foresight. In Fig. 5a, the foresight path produces a lower planning score than the base path. In Fig. 5b, the two paths yield nearly identical scores and similar trajectories, indicating little additional benefit from foresight. In Fig. 5c, the foresight path achieves a higher planning score and produces a different trajectory in response to the surrounding interaction. These examples are consistent with our quantitative finding that the planning value of foresight varies across scenes, motivating selective rather than uniform future prediction.

Closed-Loop Visualization on HUGSIM. Figure 6 presents representative closed-loop rollouts of FD-WAM in the HUGSIM simulator. Different from the open-loop examples in Fig. 5, these visualizations show how the planner behaves over consecutive simulation steps when its actions are fed back into the environment. We highlight two typical interactive scenarios. In Fig. 6a, the ego vehicle performs a lane change to overtake a slower vehicle ahead. In Fig. 6b, the ego vehicle yields in response to an oncoming vehicle and adjusts its motion accordingly. Across both cases, FD-WAM maintains coherent behavior over time and reacts to surrounding traffic in a stable manner. These examples provide additional qualitative evidence that the learned foresight representation remains useful in closed-loop driving.



(a) Base $>$ Foresight (b) Base $\approx$ Foresight (c) Base $<$ Foresight

Figure 5: Representative driving cases. Foresight can hurt, have little effect, or improve planning depending on the scene, motivating utility-aware selective foresight. Ground-truth and planned trajectory are visualized with the *winter* and *autumn* color maps, respectively.



(a) Lane change for overtaking. (b) Yielding to an oncoming vehicle.

Figure 6: Closed-loop visualization on HUGSIM. Two rollout examples of FD-WAM in HUGSIM. Rows show consecutive simulation steps from t to $t{+}3$. The two cases illustrate lane-change overtaking and yielding to an oncoming vehicle, respectively.