# WA-JEPA: Rethinking the Video JEPA Paradigm for World-Action Modeling in Autonomous Driving

**Xinlin Wang**[1 *], **Yujiao Xiang**[1,2 *], **Yuheng Zhou**[1,3 *], **Jingqi Wang**[1 *], **Minqing Huang**[1 *‡], **Jiajie Huang**[1,4], **Dongxu Wei**[1 †], **Tingguang Zhou**[1], **Xiyang Wang**[1], **Gong Chen**[1,5], **Zhi Xu**[1], **Feiyang Tan**[1], **Hangning Zhou**[1], **Mu Yang**[1]

[1]Afari Intelligent Drive [2]University of Electronic Science and Technology of China
[3]Southeast University [4]Beijing University of Posts and Telecommunications [5]Tianjin University

[*]Equal contribution, listed in no particular order. [†]Project lead. [‡]Corresponding author: mqhuang1211@gmail.com.

## Abstract

Video Joint Embedding Predictive Architecture (V-JEPA) learns powerful spatiotemporal representations from video through self-supervised latent feature prediction. However, V-JEPA is built around random-mask completion and deterministic regression, making it fundamentally ill-suited for autonomous driving planning that demands future-directed prediction tightly coupled with action. To address this, we rethink the V-JEPA paradigm and present WA-JEPA, a V-JEPA-native world-action model designed for autonomous driving planning. Instead of random spatiotemporal masking, WA-JEPA employs hybrid future-masked pre-training, where the model infers future latents from observed context. Departing from deterministic regression, we recast future prediction as conditional flow matching over latent futures, which substantially improves the model's ability to generate plausible future latents for downstream planning. Finally, a joint future-action predictor is proposed to denoise future scene tokens and ego trajectories together in a unified spatiotemporal latent space, allowing action supervision to directly shape planning-relevant world representations. Pre-trained on nuPlan videos and fine-tuned on NAVSIM, WA-JEPA reaches 91.7 EPDMS on NAVSIM-v2, surpassing the strongest end-to-end and world-action baselines by 1.6 and 1.3 EPDMS, and, without HUGSIM-specific fine-tuning, attains the best HD-Score of 0.4462 on the closed-loop HUGSIM benchmark under the same evaluation protocol. These results validate V-JEPA-native world-action modeling as a powerful and scalable paradigm for autonomous driving planning. Code is available at https://github.com/AFARI-Research/WA-JEPA.

## Introduction

End-to-end (E2E) autonomous driving distinguishes itself from conventional perception-planning pipelines by learning a unified model that directly maps raw sensor observations to driving actions, thereby eliminating the compounding errors and computational inefficiencies inherent in modular architectures (Hu et al. 2023). Despite this conceptual appeal, traditional E2E methods (Jiang et al. 2023; Sun et al. 2025) heavily rely on domain-specific perception supervision (e.g., object detection, semantic segmentation, and occupancy prediction) and lack explicit reasoning capabilities, fundamentally limiting their robustness in rare and long-tail driving scenarios.

Recently, two emerging paradigms have sought to endow E2E driving models with reasoning abilities that more closely resemble human-like driving behavior. The first is the Vision-Language-Action (VLA) paradigm (Li et al. 2025e; Xu et al. 2025a; Li et al. 2025d; Jia et al. 2026), which leverages the language understanding capacity of Vision-Language Models (VLMs) to make driving decisions from sparse visual observations. In principle, this enables models to reason about complex traffic situations through natural language. However, the majority of VLA methods (Chen, Wang, and Zhang 2025; Li et al. 2025a) simply learn a direct mapping from dense visual inputs to sparse action outputs, suffering from a severe *supervision deficit* (Li et al. 2025d): the vast information bottleneck between rich perceptual signals and minimal action supervision leaves the model poorly constrained.

The second paradigm, World-Action Models (WAMs) (Li et al. 2025d; Xia et al. 2025; Huang et al. 2026; Wang et al. 2026b; Shi et al. 2026; Liu et al. 2026a), takes a fundamentally different approach: instead of relying on language as the medium of reasoning, WAMs harness the visual reasoning capability of video generation models (Wan et al. 2025; HaCohen et al. 2024), i.e., their ability to predict future scene dynamics at pixel level, to inform and guide action decisions. Because future frame prediction naturally provides dense self-supervision without requiring costly perception annotations, WAMs offer a promising solution to the supervision deficit problem (Li et al. 2025d). Critically, this paradigm does not require linguistic reasoning, making it arguably more aligned with the visual foresight involved in driving. However, many video-generation-based WAMs perform world modeling in a latent space compressed by a Variational Autoencoder (VAE). This impoverished representation space with limited spatiotemporal semantics may constrain the model's ability to understand and reason about the evolving driving world. A question thus arises: *can we enrich the spatiotemporal semantics of the latent representations that underpin WAMs, thereby endowing them with stronger spatiotemporal reasoning and, ultimately, superior driving performance?*

In parallel, a major branch of world model research, the Video Joint Embedding Predictive Architecture (V-JEPA) family (Bardes et al. 2024; Assran et al. 2025), has developed feature-level masked modeling for self-supervised learning of spatiotemporal representations from massive video corpora. These representations transfer effectively to a wide range of downstream visual understanding tasks, demon-

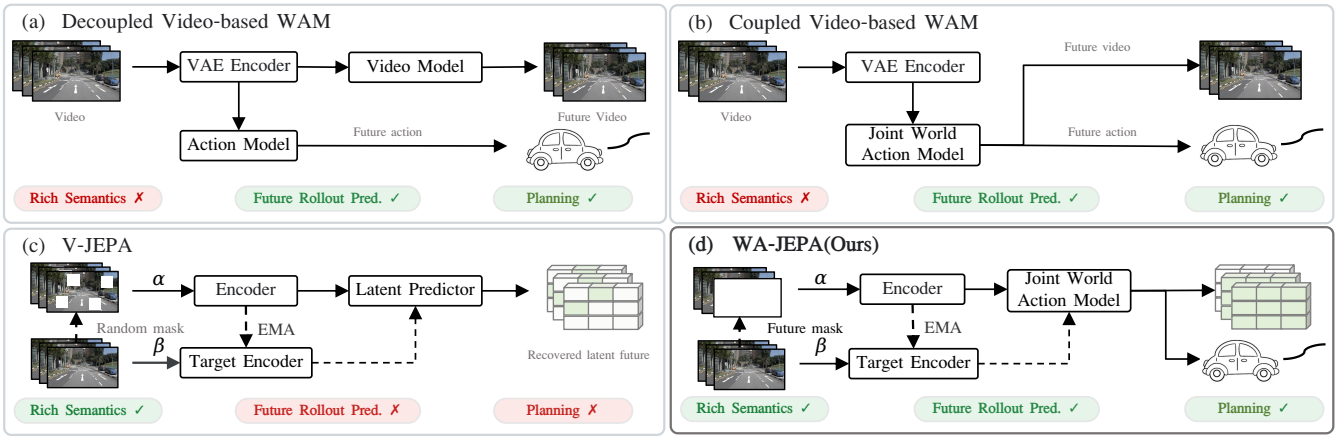


Figure 1: Comparison of V-JEPA, video-based world-action models, and our WA-JEPA. Our method bridges semantic representation learning and predictive world modeling through future-frame masking.

strating remarkable spatiotemporal semantic richness. However, V-JEPA is architecturally ill-suited for planning tasks for three reasons. First, V-JEPA pre-training applies *random* spatiotemporal masking to videos and trains the model to predict the full sequence of features from partially masked context. This is inherently a *completion* objective, which lacks the future-directed predictive capability that is essential for planning. Second, V-JEPA performs this masked prediction via regression, which, while adequate for filling in missing patches within an otherwise observed temporal context, is insufficient for generating entirely unseen future tokens, a task that inherently requires generative modeling. Third, although V-JEPA can be fine-tuned for action-conditioned future prediction (Assran et al. 2025), the gap to actionable planning remains vast: existing approaches require a goal image and rely on Model Predictive Control (MPC) with multi-round optimization to recover actions, falling far short of interactive online planning.

In this paper, we rethink the V-JEPA paradigm for world-action modeling in autonomous driving. Our goal is to preserve and leverage V-JEPA's powerful spatiotemporal representation capacity, while fundamentally extending it with causal future generation abilities, and jointly modeling it with action. This yields a novel, V-JEPA-native World-Action Model paradigm, which we term **WA-JEPA**. Our core technical innovations consist of three key designs: **(1) Pre-training with Hybrid Future Masking.** In contrast to the original V-JEPA, which employs random spatiotemporal masking to train a context-completion model, we introduce a carefully designed hybrid future mask strategy: the model observes past frames and predicts the spatiotemporal features of future frames. This endows V-JEPA with future prediction capability, directly aligning the pre-training objective with the forward-predictive demands of planning. **(2) V-JEPA-based World Modeling with Flow Matching.** Rather than regressing masked features from context as in the original V-JEPA, we reformulate future prediction as a flow-based generation process over the spatiotemporal feature space. This generative formulation substantially improves the model's ability to produce plausible futures, providing a richer evidentiary basis for downstream planning. **(3) V-JEPA-based Joint World-Action Modeling.** Whereas the original V-JEPA models only the visually-observed world, we extend the architecture to jointly model world states and ego actions in a unified spatiotemporal latent space. This tight coupling between visual representations and action representations enables V-JEPA-native future reasoning and decision-making within a single coherent framework.

Our contributions are summarized as follows:

- We propose a V-JEPA-native World-Action Model, WA-JEPA, which harnesses V-JEPA's powerful spatiotemporal representations to strengthen joint world-action modeling, achieving substantial improvements in planning capability for autonomous driving.
- We introduce a suite of architectural innovations—hybrid future masking for pre-training, latent world modeling via flow matching, and joint world-action modeling—that systematically adapt the V-JEPA architecture for planning, while carefully preserving its core representation learning strengths.
- Extensive experiments on the open-loop NAVSIM-v1 and NAVSIM-v2 benchmarks establish a new state of the art, and zero-shot evaluation on the closed-loop simulator HUGSIM (Zhou et al. 2024b) shows that the gain carries over to closed-loop driving.

## Related Work

**End-to-end driving and world-action models.** E2E driving maps sensor observations directly to ego motion (Hu et al. 2023; Jiang et al. 2023; Liao et al. 2025), while VLA models incorporate language reasoning or VLM-guided trajectory refinement (Shao et al. 2024; Hwang et al. 2024; Zhou et al. 2026; Wang et al. 2026c). WAMs augment direct planning with learned future-scene prediction (Wang et al. 2024; Li et al. 2025c; Zhu et al. 2026; Wang et al. 2026b; Hong

et al. 2026; Li et al. 2025d). As illustrated in Fig. 1(a), existing WAMs explore different choices of world representation and the coupling between future-scene prediction and action generation. In particular, coupled video-based WAMs jointly model future visual content and actions within a shared generative process, using video-generation priors or video latents (Liu et al. 2026a; Shi et al. 2026), as shown in Fig. 1(b). Although this coupling provides a direct interface between future-scene generation and planning, these methods inherit the limitations of video-generative latent spaces: their representations are primarily optimized for visual generation and reconstruction, which may limit semantic abstraction and the modeling of action-relevant future states.

**Predictive representations for world-action modeling.** Recent world-action models explore predictive representations beyond pixel-level reconstruction. Latent-WAM uses spatially aware latent scene representations for future world modeling and trajectory planning (Wang et al. 2026b), while DINO-WM predicts future features from a pre-trained DINOv2 encoder to support action-conditioned planning (Zhou et al. 2024a). As shown in Fig. 1(c), JEPA-based approaches learn powerful spatiotemporal representations by predicting target embeddings rather than reconstructing (Bardes et al. 2024; Assran et al. 2025). However, V-JEPA primarily provides predictive visual representations and does not directly specify how these representations should be converted into future actions or planning decisions. Drive-JEPA attempts to bridge this gap by introducing a V-JEPA encoder into an end-to-end driving model, but still relies on a separate downstream trajectory planner (Wang et al. 2026a). WA-JEPA instead extends predictive representation learning to jointly predict future world features and ego trajectories within a shared predictive representation.

# Method

## Preliminaries

Our method is greatly inspired by V-JEPA 2 (Assran et al. 2025), which learns predictive representations in latent space instead of reconstructing raw pixels. As shown in Fig. 1 (c), given a masked observation $\alpha$ and a target observation $\beta$, V-JEPA 2 trains a predictor to infer the target latent from the context latent:

$$\min_{\theta,\psi} \left\| P_\psi(E_\theta(\alpha)) - \mathrm{sg}(E_{\bar{\theta}}(\beta)) \right\|_1, \tag{1}$$

where $E_\theta$ is the online encoder, $E_{\bar{\theta}}$ is the target encoder, $P_\psi$ is the latent predictor, and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The target encoder is updated with an exponential moving average (EMA):

$$\bar{\theta} \leftarrow \mu\bar{\theta} + (1-\mu)\theta, \tag{2}$$

where $\mu$ is the EMA momentum coefficient.

## Overview

We formulate end-to-end autonomous driving as conditional future-action generation. Given historical multi-view observations $\mathcal{X}_{1:H}$, ego state $s$, and historical actions $\mathcal{Y}_{1:H}$, the model predicts future actions through the following mapping:

$$f_\theta\left(\mathcal{X}_{1:H}, s, \mathcal{Y}_{1:H}\right) \mapsto \hat{\mathcal{Y}}_{H+1:H+K} = \left\{(\hat{x}_k, \hat{y}_k, \hat{\phi}_k)\right\}_{k=1}^{K}, \tag{3}$$

where $f_\theta$ denotes the world-action model parameterized by $\theta$, $H$ and $K$ represent the number of historical frames and future frames, respectively. Here, $(\hat{x}_k, \hat{y}_k)$ and $\hat{\phi}_k$ denote the predicted ego position and heading at the $k$-th future frame over a planning horizon of $K$ steps.

As shown in Fig. 2, WA-JEPA follows a two-stage training scheme. Stage 1 adapts pre-trained V-JEPA 2 to multi-view driving data by predicting future representations under a hybrid masking strategy. Stage 2 adds action supervision and extends the predictor with an action stream, enabling joint prediction of future scene representations and ego actions.

## Stage 1: Hybrid Future-Masked Pre-training

Stage 1 employs a hybrid objective combining a causal Full-mask branch with a Patch-mask completion branch on synchronized multi-view driving observations, without action supervision. The former learns strictly past-to-future dynamics, while the latter retains partial future context to facilitate representation learning and preserve the partial-masking paradigm of V-JEPA 2.

**Multi-view spatio-temporal encoder.** We adopt a pre-trained V-JEPA 2 ViT-L as the visual backbone. Each training sample contains historical and future frames from $C$ synchronized cameras:

$$\mathcal{X}_{1:H+K} = [\mathcal{X}_{1:H}, \mathcal{X}_{H+1:H+K}], \qquad \mathcal{X}_t = \{X_t^c\}_{c=1}^{C}, \tag{4}$$

where $X_t^c$ represents the frame from the $c$-th camera at the $t$-th time step.

Subsequently, the model treats each camera stream as a separate video and processes all views independently with the shared online encoder to generate visual tokens. During training, history tokens remain visible, while masking is applied only to future tokens. Specifically, we employ a hybrid strategy that combines full future masking with patch-masked future completion. Full masking designates all future tokens as prediction targets and therefore requires prediction from historical context alone. Patch masking retains a subset of future tokens as visible conditions and predicts the remaining masked tokens, reducing the learning difficulty while retaining the partial-masking prediction style of V-JEPA 2.

The outputs derived from historical frames serve as context scene tokens $\mathcal{Z}_{\mathrm{ctx}}$ :

$$\mathcal{Z}_{\mathrm{ctx}} = E_\theta\left(\mathcal{X}_{1:H}\right). \tag{5}$$

For future frames, the mask pattern $M^{(m)}$, where $m \in \{\mathrm{full}, \mathrm{patch}\}$, is applied before the online encoder, such that only visible future patches are provided to $E_\theta$. The model then combines the encoded visible tokens with learnable mask tokens $Z_{\mathrm{mask}}$ to construct the future condition tokens:

$$Z_{\mathrm{cond}}^{(m)} = \Phi^{(m)}\left(Z_{\mathrm{mask}}, E_\theta\left(X_{H+1:H+K}, M^{(m)}\right)\right), \tag{6}$$

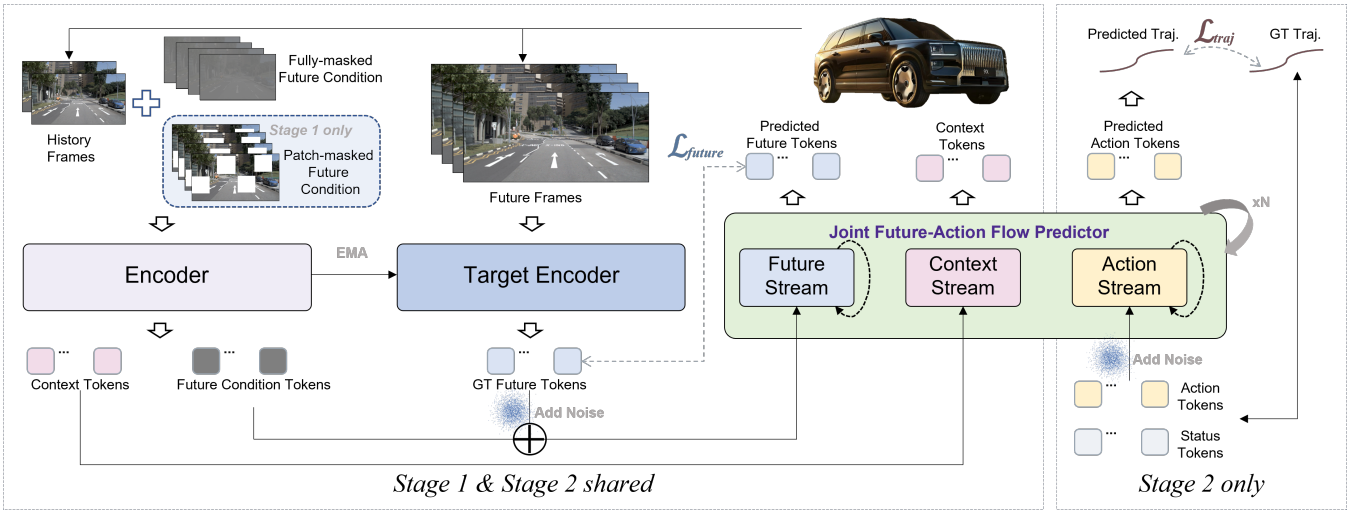


Figure 2: Overview of WA-JEPA. Stage 1 adapts V-JEPA 2 to multi-view driving videos by predicting future representations under full-future and patch-level masks. Stage 2 initializes from this checkpoint and jointly predicts future scene representations and ego actions with the Joint Future-Action Flow Predictor.

where $\Phi^{(m)}(\cdot)$ denotes a mask-aware fill-and-scatter operation. It initializes a full future-token sequence with learnable mask tokens and scatters the encoded visible future tokens back to their original positions according to the mask pattern. Under Full-mask, no future patch is provided to the online encoder, and $Z_{\text{cond}}^{(\text{full})}$ therefore consists entirely of learnable mask tokens.

Meanwhile, the EMA target encoder provides unmasked target representations for the future frames, yielding the clean future scene target $\mathcal{Z}_{\text{future}}^{*}$:

$$\mathcal{Z}_{\text{future}}^{*} = E_{\bar{\theta}}\left(\mathcal{X}_{H+1:H+K}\right). \tag{7}$$

These target representations is used solely to provide latent supervision for future scenes.

**flow-based latent future prediction.** We use flow matching for future latent prediction. We use conditional flow matching with a clean-latent ($x$-prediction) parameterization for future latent prediction. First, we sample Gaussian noise $\epsilon_{\text{future}} \sim \mathcal{N}(0, I)$ with the same dimensionality as the future scene tokens and continuous flow time $t$. The noisy future scene tokens $\mathcal{Z}_t$ at time $t$ is obtained via linear interpolation:

$$\mathcal{Z}_t = (1-t)\epsilon_{\text{future}} + t\mathcal{Z}_{\text{future}}^{*}. \tag{8}$$

The future flow predictor receives the historical context $\mathcal{Z}_{\text{ctx}}$, future condition $\mathcal{Z}_{\text{cond}}$, noisy future scene tokens $\mathcal{Z}_t$, and temporal condition $t$, and predicts the clean future scene tokens $\hat{\mathcal{Z}}_{\text{future}}$:

$$\hat{\mathcal{Z}}_{\text{future}} = P_{\psi}^{\text{future}}\left(\mathcal{Z}_{\text{ctx}}, \mathcal{Z}_{\text{cond}}, \mathcal{Z}_t, t\right). \tag{9}$$

The future flow predictor follows an MMDiT (Multimodal Diffusion Transformer)-style (Esser et al. 2024) design, primarily performing joint self-attention between context scene tokens and future scene tokens.

**Training objective.** In Stage 1, the model is pre-trained on multi-view driving observations from the nuPlan dataset. During this stage, the model is optimized using a mean squared error (MSE) loss on the clean future latent prediction:

$$\mathcal{L}_{\text{Stage 1}} = \mathcal{L}_{\text{future}} = \frac{1}{N}\left\|\hat{\mathcal{Z}}_{\text{future}} - \text{sg}\left(\mathcal{Z}_{\text{future}}^{*}\right)\right\|_2^2, \tag{10}$$

where $N$ denotes the number of future scene tokens included in the loss computation. This clean-latent objective corresponds to the $x$-prediction parameterization of conditional flow matching described above.

## Stage 2: Joint World-Action Modeling

Stage 2 introduces an action generation stream, jointly modeling future scene representations and actions within a unified model. The model is initialized using the pre-trained weights from Stage 1 and further adapted to the action planning task through action supervision.

**Joint future-action predictor.** In Stage 2, the joint predictor further incorporates historical actions and the compact ego state. Together, they serve as ego-motion conditions for future actions generation. Different from Stage 1, only full future mask is used in Stage 2 for consistency with the causal nature of driving. Future images are used only to construct supervision signals for the future scene latent prediction and are not provided as input to the student predictor.

For the action stream, we first normalize the ground-truth future actions and construct noisy actions in the normalized action space:

$$\begin{aligned} \tilde{\mathcal{Y}}_t &= (1-t)\epsilon_y + t\bar{\mathcal{Y}}_{H+1:H+K}, \quad \epsilon_y \sim \mathcal{N}(0, I), \\ \bar{\mathcal{Y}}_{H+1:H+K} &= \text{Norm}(\mathcal{Y}_{H+1:H+K}), \end{aligned} \tag{11}$$

where $\mathrm{Norm}(\cdot)$ denotes the action normalization operation, $\bar{\mathcal{Y}}_{H+1:H+K}$ denotes the normalized ground-truth future actions, $\epsilon_y$ denotes Gaussian noise with the same dimensionality as the future actions, and $\tilde{\mathcal{Y}}_t$ denotes the noisy future actions at time $t$.

Subsequently, the noisy future actions $\tilde{\mathcal{Y}}_t$, historical actions $\mathcal{Y}_{1:H}$, and ego state $s$ are separately encoded and concatenated to form the action tokens $\mathcal{T}_{\text{act}}$:

$$\begin{aligned} \mathcal{T}_{\text{act}} &= \text{Concat}\left[\mathcal{T}_n, \mathcal{T}_h, \mathcal{T}_s\right], \\ \mathcal{T}_n = F_n(\tilde{\mathcal{Y}}_t), \quad \mathcal{T}_h &= F_h(\mathcal{Y}_{1:H}), \quad \mathcal{T}_s = F_s(s). \end{aligned} \tag{12}$$

where $F_n$ is a linear projection, and $F_h$ and $F_s$ are MLP encoders.

The joint predictor models interactions among the historical context scene tokens $\mathcal{Z}_{\text{ctx}}$, future scene tokens $\mathcal{Z}_{\text{cond}}$ and noisy future scene tokens $\mathcal{Z}_t$, and action tokens $\mathcal{T}_{\text{act}}$ at time $t$. Its future scene and action output streams are respectively defined as

$$\begin{aligned} \hat{\mathcal{Z}}_{\text{future}} &= P_\psi^{\text{future}}\left(\mathcal{Z}_{\text{ctx}}, \mathcal{Z}_{\text{cond}}, \mathcal{Z}_t, \text{sg}(\mathcal{T}_{\text{act}}), t\right), \\ \hat{\tilde{\mathcal{Y}}}_{H+1:H+K} &= P_\psi^{\text{act}}\left(\mathcal{Z}_{\text{ctx}}, \mathcal{Z}_{\text{cond}}, \mathcal{Z}_t, \mathcal{T}_{\text{act}}, t\right). \end{aligned} \tag{13}$$

Here, $P_\psi^{\text{future}}$ and $P_\psi^{\text{act}}$ denote the future scene and action output streams of the same joint predictor, which also follows an MMDiT-style design, rather than two independent predictors. Under the Full-mask setting used in Stage 2, $Z_{\text{cond}} = Z_{\text{cond}}^{(\text{full})}$ consists entirely of learnable mask tokens, so no future image content is visible to the joint predictor. $\hat{\tilde{\mathcal{Y}}}_{H+1:H+K}$ denotes the predicted normalized future actions. We apply stop-gradient $\text{sg}(\cdot)$ to the action tokens only in the future scene output stream, preventing the future scene prediction loss from updating the action stream. Further details and analysis of this design are provided in Appendix C.

**Training objective.** The future scene stream follows the future scene prediction loss defined in Stage 1. This objective preserves the model's future scene modeling capability during action-supervised fine-tuning. The action stream predicts denoised future actions in the normalized action space and is optimized using the MSE between the predicted and ground-truth actions over all future steps:

$$\begin{aligned} \mathcal{L}_{\text{act}} &= \frac{1}{K}\sum_{k=1}^{K}\left\|\hat{\bar{\mathbf{y}}}_k - \bar{\mathbf{y}}_k\right\|_2^2, \\ \bar{\mathbf{y}}_k &= \text{Norm}\left(x_k, y_k, \phi_k\right), \end{aligned} \tag{14}$$

where $\bar{\mathbf{y}}_k$ denotes the normalized ground-truth action at the $k$-th future frame, $\hat{\bar{\mathbf{y}}}_k$ denotes the corresponding predicted normalized action.

Finally, the overall Stage 2 training objective is

$$\mathcal{L}_{\text{Stage 2}} = \lambda_{\text{future}}\mathcal{L}_{\text{future}} + \lambda_{\text{act}}\mathcal{L}_{\text{act}}. \tag{15}$$

Here, $\lambda_{\text{future}}$ and $\lambda_{\text{act}}$ are the weighting coefficients for the future scene and action losses, respectively.

**Planning inference.** During inference, the model takes only historical multi-view observations, historical actions, and ego state as inputs. The future scene latents and normalized future actions are initialized from Gaussian noise. At each sampling step, the joint predictor estimates their clean endpoints, which are converted into velocities along the linear flow paths and iteratively integrated. Neither future images nor ground-truth future actions are required during this process.

After sampling, the predicted normalized actions are transformed back to the original action space to obtain the final future actions:

$$\hat{\mathcal{Y}}_{H+1:H+K} = \text{Norm}^{-1}\left(\hat{\tilde{\mathcal{Y}}}_{H+1:H+K}\right). \tag{16}$$

Here, $\text{Norm}^{-1}(\cdot)$ denotes the action denormalization operation.

# Experiments

## Experimental Setup

**Datasets, benchmarks, and metrics.** In Stage 1, the model is pre-trained on multi-view driving videos from nuPlan, and Stage 2 is trained on the official NAVSIM `navtrain` split. Evaluation is performed on the held-out `navtest` split under NAVSIM-v1 and NAVSIM-v2 (Dauner et al. 2024; Cao et al. 2025), using the official PDMS and EPDMS scores and their sub-metrics, which are defined in Appendix D.

**Implementation details.** The model takes four historical frames from the left, front, right, and rear cameras at $256 \times 512$ resolution and predicts eight actions at 2 Hz. The visual encoder is initialized from the V-JEPA 2 ViT-L backbone pre-trained in Stage 1. The future scene and action streams are jointly optimized with AdamW, bfloat16 precision, and DeepSpeed ZeRO-2. Stage 1 and Stage 2 use 64 and 32 NVIDIA A800 GPUs, respectively, with a per-GPU batch size of 4. The encoder, scene projector, and joint predictor use learning rates of $1 \times 10^{-5}$, $1 \times 10^{-4}$, and $1.5 \times 10^{-4}$, respectively, with weight decay $0.04$. At inference, the WA-JEPA Joint future-action predictor uses 12 sampling steps. We evaluate its stochastic predictions using a fixed set of 10 seeds and report the mean, while deterministic baselines are evaluated once. Detailed settings are provided in Appendix C.

**Baselines.** We compare WA-JEPA with representative E2E, VLA, and WAM baselines under compatible input modalities and evaluation protocols.

## Comparison with Existing Methods

**Quantitative results.** As shown in Table 1, WA-JEPA achieved an EPDMS of 91.7 on NAVSIM-v2, exceeding the best-performing E2E method, SparseDriveV2, by 1.6 and the best-performing WAM method, Discrete-WAM, by 1.3. On NAVSIM-v1 (Table 3), WA-JEPA attained a PDMS of 91.8.

**Zero-shot closed-loop generalization.** We compare WA-JEPA with LTF (Chitta et al. 2022), DrivoR (Kirby et al. 2026), UniAD (Hu et al. 2023), and VAD (Jiang et al. 2023) on 436 HUGSIM scenarios (Zhou et al. 2024b). WA-JEPA's

| Method | Backbone | NC↑ | DAC↑ | DDC↑ | TLC↑ | EP↑ | TTC↑ | LK↑ | HC↑ | EC↑ | EPDMS* ↑ | EPDMS↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **End-to-End Methods** | | | | | | | | | | | | |
| TransFuser (Chitta et al. 2022) | RegNetY-3.2GF | 96.9 | 89.9 | 97.8 | 99.7 | 87.1 | 95.4 | 92.7 | 98.3 | 87.2 | 76.7 | – |
| ARTEMIS (Feng et al. 2025) | ResNet-34 | 98.3 | 95.1 | 98.6 | 99.8 | 81.5 | 97.4 | 96.5 | 98.3 | – | 83.1 | – |
| Hydra-MDP++ (Li et al. 2025b) | V2-99 | 98.8 | 97.8 | 99.1 | 100 | 84.0 | 95.3 | 70.1 | – | 96.8 | 84.1 | – |
| DiffusionDrive (Liao et al. 2025) | ResNet-34 | 98.2 | 95.9 | 99.4 | 99.8 | 87.5 | 97.3 | 96.8 | 98.3 | 87.7 | – | 84.5 |
| Drive-JEPA (Wang et al. 2026a) | ResNet-34 | 98.8 | 97.4 | 99.0 | 99.8 | 83.5 | 98.0 | 96.2 | 98.1 | 85.6 | 85.4 | – |
| Drive-JEPA† (Wang et al. 2026a) | ViT-L | 98.4 | 98.6 | 99.1 | 99.8 | 88.4 | 97.8 | 97.6 | 97.9 | 84.8 | 87.8 | – |
| DiffusionDriveV2 (Zou et al. 2025) | ResNet-34 | 97.7 | 96.6 | 99.2 | 99.8 | 88.9 | 97.2 | 96.0 | 97.8 | 91.0 | 85.5 | 87.5 |
| DriveSuprim (Yao et al. 2025) | V2-99 | 97.8 | 97.9 | 99.5 | 99.9 | 90.6 | 97.1 | 96.6 | 98.3 | 77.9 | 86.0 | – |
| SparseDriveV2 (Sun et al. 2026) | ResNet-34 | 98.1 | 98.1 | 99.6 | 99.8 | 91.1 | 97.3 | 96.9 | 98.2 | 78.4 | 86.7 | 90.1 |
| **VLA Methods** | | | | | | | | | | | | |
| ReCogDrive (Li et al. 2025e) | InternVL3 | 98.3 | 95.2 | 99.5 | 99.8 | 87.1 | 97.5 | 96.6 | 98.3 | 86.5 | 83.6 | – |
| WAM-Flow (Xu et al. 2025b) | Janus-1.5B | 98.5 | 94.5 | 99.5 | 99.8 | 86.9 | 96.8 | 97.4 | 97.6 | 73.9 | 84.7 | – |
| WAM-Diff (Xu et al. 2025a) | LLaDA-V | 99.0 | 98.4 | 99.3 | 99.9 | 87.0 | 98.6 | 96.2 | 98.1 | 78.5 | – | 89.7 |
| **World-Action and World-Model Methods** | | | | | | | | | | | | |
| DriveVLA-W0 (Li et al. 2025d) | Emu3-8B | 98.5 | 99.1 | 98.0 | 99.7 | 86.4 | 98.1 | 93.2 | 97.9 | 58.9 | 86.1 | – |
| DriveWorld-VLA (Jia et al. 2026) | InternVL | 98.6 | 99.1 | 99.6 | 99.8 | 87.4 | 97.9 | 97.0 | 97.8 | 78.6 | – | 86.8 |
| CoWorld-VLA (Huang et al. 2026) | Qwen3 | 99.1 | 97.0 | 99.6 | 99.9 | 87.9 | 98.5 | 97.7 | 98.2 | 86.2 | 86.2 | 90.0 |
| DreamerAD (Yang et al. 2026) | Transformer-1.3B | 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) | DINOv2-B | 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) | V2-99 | 98.8 | 99.1 | 99.6 | 99.9 | 86.6 | 98.4 | 96.4 | 98.3 | 74.8 | 86.4 | 89.9 |
| Discrete-WAM (Yao et al. 2026) | Transformer-1B | 98.5 | 98.2 | 99.7 | 99.8 | 90.5 | 97.9 | 97.2 | 98.3 | 78.1 | 87.0 | 90.4 |
| **WA-JEPA (ours)** | ViT-L | 99.4 | 98.2 | 99.7 | 99.9 | 87.8 | 98.9 | 98.3 | 98.3 | 88.1 | **88.0** | **91.7** |

Table 1: Comparison with state-of-the-art methods on NAVSIM-v2 `navtest`. Methods are grouped by modeling paradigm. Backbone lists the primary visual or vision-language backbone used by each method. † marks results using auxiliary simulator-derived supervision. EPDMS* refers to scores obtained before correction of the human-reference penalty-filter aggregation, whereas EPDMS reports the corrected scores.

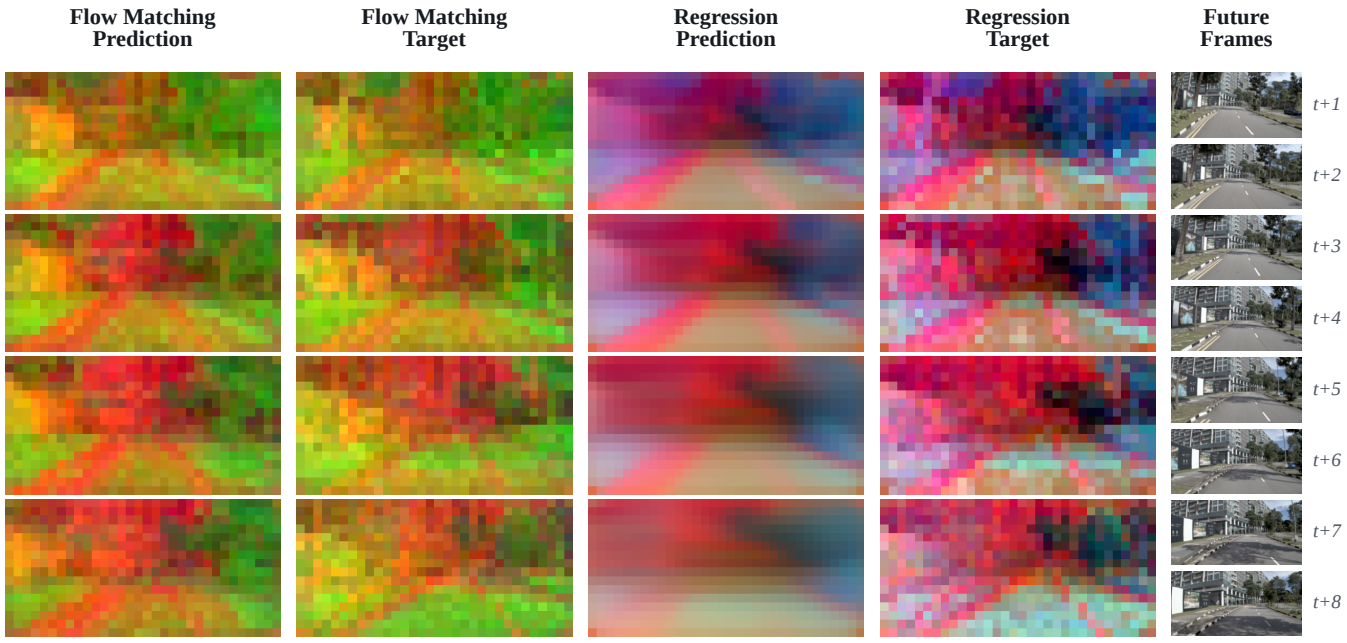


Figure 3: Target-referenced PCA visualization of future latent predictions from flow matching (FM) and direct regression (Reg.). For each method, a separate PCA basis is fitted to its EMA target representations and applied to both target and predicted latents. Each map represents two consecutive future frames.

| | WA-JEPA | LTF | DrivoR | UniAD | VAD |
|---|---|---|---|---|---|
| *All 436 scenarios* | | | | | |
| NC | **0.6856** | 0.4428 | 0.5217 | 0.6555 | 0.4117 |
| DAC | **0.9635** | 0.9275 | 0.9559 | 0.9320 | 0.9028 |
| TTC | **0.6120** | 0.3751 | 0.4620 | 0.5156 | 0.2798 |
| Comf. | 0.6620 | 0.9478 | 0.9390 | 0.6633 | **0.9534** |
| PDMS | **0.5717** | 0.3653 | 0.4475 | 0.4940 | 0.2831 |
| RC | **0.5689** | 0.3804 | 0.4721 | 0.4383 | 0.3006 |
| HD-Score | **0.4462** | 0.2310 | 0.3252 | 0.3124 | 0.1393 |
| *HD-Score by difficulty level* | | | | | |
| Easy ($n$=80) | **0.7977** | 0.6608 | 0.7799 | 0.6395 | 0.4197 |
| Medium ($n$=157) | **0.5563** | 0.1547 | 0.2911 | 0.3718 | 0.0849 |
| Hard ($n$=96) | **0.3060** | 0.1204 | 0.2000 | 0.2099 | 0.0770 |
| Extreme ($n$=103) | 0.1362 | 0.1167 | **0.1407** | 0.0632 | 0.0626 |

Table 2: Zero-shot closed-loop results on HUGSIM. Values are on the $[0, 1]$ scale; higher is better, and the best result in each row is bold.

| Method | NC↑ | DAC↑ | TTC↑ | Comf.↑ | EP↑ | PDMS↑ |
|---|---|---|---|---|---|---|
| **End-to-End Methods** | | | | | | |
| TransFuser | 97.7 | 92.8 | 92.8 | 100 | 79.2 | 84.0 |
| DiffusionDrive | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1 |
| Drive-JEPA | 98.7 | 96.2 | 95.5 | 100 | 82.9 | 89.0 |
| **VLA Methods** | | | | | | |
| ReCogDrive | 97.9 | 97.3 | 94.9 | 100 | 87.3 | 90.8 |
| WAM-Flow | 99.2 | 98.3 | 97.0 | 99.7 | 82.3 | 90.3 |
| WAM-Diff | 99.1 | 98.3 | 96.5 | 99.9 | 84.4 | 91.0 |
| **World-Action and World-Model Methods** | | | | | | |
| CoWorld-VLA | 99.1 | 96.9 | 96.4 | 100 | 83.9 | 89.9 |
| DriveVLA-W0 | 98.7 | 99.1 | 95.3 | 99.3 | 83.3 | 90.2 |
| DriveWorld-VLA | 99.1 | 98.2 | 96.1 | 100 | 85.9 | 91.3 |
| DriveLaW | 99.0 | 97.1 | 96.7 | 100 | 81.3 | 89.1 |
| DriveFuture | 98.8 | 99.1 | 95.4 | 100 | 84.2 | 90.7 |
| **WA-JEPA (ours)** | 99.5 | 98.3 | 97.7 | 100 | 85.0 | **91.8** |

Table 3: Comparison on NAVSIM-v1 `navtest`.

two training stages use neither HUGSIM-rendered observations nor any of its four source datasets; DrivoR is similarly source-disjoint. All methods follow the common protocol detailed in Appendix A, sharing the same scenarios, controller, commands, aggregation, and metric implementation while retaining their native sensor configurations.

As shown in Table 2, WA-JEPA achieves the best NC, DAC, TTC, PDMS, RC, and HD-Score, improving HD-Score from 0.3252 to 0.4462. This result demonstrates that joint world–action pre-training transfers effectively to closed-loop control, with the largest gains on the medium and hard levels.

**Qualitative evaluation.** Qualitative results are provided in Appendix B, including closed-loop HUGSIM rollouts and NAVSIM trajectory comparisons.

## Ablation Studies

**Vision encoder initialization.** Table 4(a) isolates the contribution of the visual encoder. Every variant is trained directly in Stage 2 without any Stage 1 pre-training, so the models share the same Stage 2 architecture, training data, and optimization protocol and differ only in how the encoder is initialized. The publicly released V-JEPA 2 weights reach an EPDMS of 89.5, exceeding the strongest alternatives, MAE and DINOv3, by 5.7 EPDMS, whereas the three image-level self-supervised and vision–language initializations lie within 0.7 EPDMS of one another. The gap therefore tracks the V-JEPA 2 pre-training objective rather than the choice among image-level objectives, which motivates adopting a V-JEPA 2 encoder as the backbone of our world-action model.

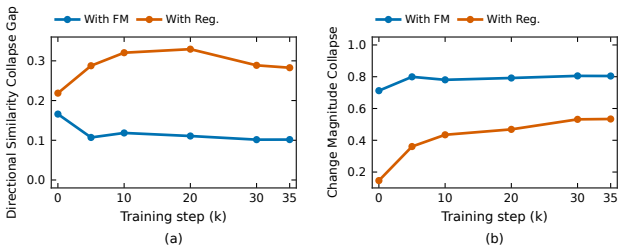


Figure 4: Temporal representation preservation with flow matching (FM) and direct regression (Reg.). A lower directional-similarity collapse gap and a change-magnitude collapse closer to 1 indicate better preservation of the target temporal dynamics.

| Encoder | EPDMS↑ |
|---|---|
| MAE | 83.8 |
| SigLIP2 | 83.1 |
| DINOv3 | 83.8 |
| V-JEPA 2 | **89.5** |

**(a) Encoder**

| Patch-mask | Full-mask | EPDMS↑ |
|---|---|---|
| | | 89.5 |
| ✓ | | 91.0 |
| | ✓ | 91.3 |
| ✓ | ✓ | **91.7** |

**(b) Stage 1 Pre-training**

| Joint | Future Pred. | | EPDMS↑ |
|---|---|---|---|
| | FM | Reg. | |
| | | | 89.9 |
| | ✓ | | 90.8 |
| ✓ | | | 91.1 |
| ✓ | | ✓ | 90.7 |
| ✓ | ✓ | | **91.7** |

**(c) Stage 2 Component**

Table 4: Ablations on NAVSIM-v2 `navtest`. (a) Vision encoder selection. (b) Masking strategy in Stage 1. The first row represents the baseline that skips Stage 1 pre-training and directly uses the original pre-trained V-JEPA 2 for Stage 2 training. (c) Joint future–action predictor and future-prediction methods in Stage 2. FM and Reg. denote flow matching and direct regression, respectively. The first row denotes an action-only Stage 2 baseline initialized from the Stage 1 checkpoint and fine-tuned with aciton supervision, without joint scene-action modeling or future prediction.

**Stage 1 masking strategies.** Table 4(b) ablates the masking design in Stage 1. The first row directly uses the original pretrained V-JEPA 2 checkpoint without our Stage 1 pretraining on nuPlan, serving as the no-Stage 1 baseline with an EPDMS of 89.5. Applying Patch-mask alone improves EPDMS to 91.0 by providing partial future context for representation learning, while Full-mask achieves 91.3 by enforcing strictly causal past-to-future prediction. Combining both strategies yields the best performance of 91.7, outperforming the individual variants by 0.7 and 0.4, respectively. This demonstrates that Patch-mask and Full-mask provide complementary training signals.

**Stage 2 scene–action coupling and future prediction.** Table 4(c) ablates scene–action modeling and future-latent supervision in Stage 2. The first cascaded baseline uses only historical latent features through cross-attention and obtains 89.9 EPDMS. Adding a separate flow-based future predictor and injecting its predicted future latents through the same cross-attention design improves EPDMS to 90.8. Under joint modeling, the baseline without explicit future-latent supervision achieves 91.1. Adding direct regression reduces EPDMS to 90.7, whereas flow-based future prediction yields the best result of 91.7. These results show that future-scene prediction and joint scene–action modeling are complementary, while the choice of prediction objective also matters.

**Future representation analysis.** To examine how the two prediction objectives affect future representations, Fig. 4 reports two target-referenced metrics. The directional similarity collapse gap measures excessive cross-step similarity relative to the target representations (lower is better), while the change-magnitude collapse measures the predicted temporal variation relative to the targets (closer to 1 is better). Detailed definitions are provided in Appendix E. Compared with direct regression, flow matching reduces the directional similarity collapse gap from $0.30$ to $0.10$ and increases the change-magnitude collapse from $0.45$ to $0.80$. Figure 3 provides complementary qualitative evidence: flow matching retains clearer spatial structures over the prediction horizon, whereas direct regression becomes progressively smoother.

## Conclusion

We presented WA-JEPA, a V-JEPA-native world-action model that extends V-JEPA 2 from visual representation learning to future-predictive planning for autonomous driving. By combining hybrid future-masked pre-training, prediction of future latents via flow matching, and joint future–action modeling, WA-JEPA learns future scene dynamics and actions within a shared spatiotemporal latent space. WA-JEPA achieves 91.7 EPDMS on NAVSIM-v2 `navtest` and a 0.4462 HD-Score on 436 HUGSIM closed-loop scenarios, demonstrating strong open-loop planning and zero-shot closed-loop generalization.

## References


Assran, M.; Bardes, A.; Fan, D.; Garrido, Q.; Howes, R.; Komeili, M.; Muckley, M.; Rizvi, A.; Roberts, C.; Sinha, K.; Zholus, A.; Arnaud, S.; Gejji, A.; Martin, A.; Hogan, F. R.; Dugas, D.; Bojanowski, P.; Khalidov, V.; Labatut, P.; Massa, F.; Szafraniec, M.; Krishnakumar, K.; Li, Y.; Ma, X.; Chandar, S.; Meier, F.; LeCun, Y.; Rabbat, M.; and Ballas, N. 2025. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning. arXiv:2506.09985.

Bardes, A.; Garrido, Q.; Ponce, J.; Chen, X.; Rabbat, M.; LeCun, Y.; Assran, M.; and Ballas, N. 2024. Revisiting Feature Prediction for Learning Visual Representations from Video. arXiv:2404.08471.

Cao, W.; Hallgarten, M.; Li, T.; Dauner, D.; Gu, X.; Wang, C.; Miron, Y.; Aiello, M.; Li, H.; Gilitschenski, I.; Ivanovic, B.; Pavone, M.; Geiger, A.; and Chitta, K. 2025. Pseudo-Simulation for Autonomous Driving. arXiv:2506.04218.

Chen, Y.; Wang, Y.; and Zhang, Z. 2025. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 26890–26900.

Chitta, K.; Prakash, A.; Jaeger, B.; Yu, Z.; Renz, K.; and Geiger, A. 2022. TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. doi:10.1109/TPAMI.2022.3200245.

Dauner, D.; Hallgarten, M.; Li, T.; Weng, X.; Huang, Z.; Yang, Z.; Li, H.; Gilitschenski, I.; Ivanovic, B.; Pavone, M.; Geiger, A.; and Chitta, K. 2024. NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking. arXiv:2406.15349.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Feng, R.; Xi, N.; Chu, D.; Wang, R.; Deng, Z.; Wang, A.; Lu, L.; Wang, J.; and Huang, Y. 2025. ARTEMIS: Autoregressive End-to-End Trajectory Planning with Mixture of Experts for Autonomous Driving. arXiv:2504.19580.

HaCohen, Y.; Chiprut, N.; Brazowski, B.; Shalem, D.; Moshe, D.; Richardson, E.; Levin, E.; Shiran, G.; Zabari, N.; Gordon, O.; Panet, P.; Weissbuch, S.; Kulikov, V.; Bitterman, Y.; Melumian, Z.; and Bibi, O. 2024. LTX-Video: Realtime Video Latent Diffusion. *arXiv preprint arXiv:2501.00103*.

Hong, Y.; Zhou, X.; Li, Y.; Zhou, X.; Liu, L.; Luo, Y.; Xu, S.; Yang, L.; and Song, Z. 2026. DriveFuture: Future-Aware Latent World Models for Autonomous Driving. arXiv:2605.09701.

Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; Lu, L.; Jia, X.; Liu, Q.; Dai, J.; Qiao, Y.; and Li, H. 2023. Planning-Oriented Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17853–17862.

Huang, M.; Xiang, Y.; Liang, Z.; Huang, J.; Wang, J.; Xu, Z.; Tan, F.; Zhou, H.; Yang, M.; and Che, G. 2026. Coworld-vla: Thinking in a multi-expert world model for autonomous driving. *arXiv preprint arXiv:2605.10426*.

Hwang, J.-J.; Xu, R.; Lin, H.; Hung, W.-C.; Ji, J.; Choi, K.; Huang, D.; He, T.; Covington, P.; Sapp, B.; Zhou, Y.; Guo, J.; Anguelov, D.; and Tan, M. 2024. EMMA: End-to-End Multimodal Model for Autonomous Driving. arXiv:2410.23262.

Jia, F.; Liu, L.; Song, Z.; Jia, C.; Ye, H.; Hao, X.; and Chen, L. 2026. DriveWorld-VLA: Unified Latent-Space World Modeling with Vision–Language–Action for Autonomous Driving. arXiv:2602.06521.

Jiang, B.; Chen, S.; Xu, Q.; Liao, B.; Chen, J.; Zhou, H.; Zhang, Q.; Liu, W.; Huang, C.; and Wang, X. 2023. VAD: Vectorized Scene Representation for Efficient Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 8306–8316.

Kirby, E.; Boulch, A.; Xu, Y.; Yin, Y.; Puy, G.; Zablocki, E.; Bursuc, A.; Gidaris, S.; Marlet, R.; Bartoccioni, F.; Cao, A.-Q.; Samet, N.; Vu, T.-H.; and Cord, M. 2026. Driving on Registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, J.; Zhang, B.; Jin, X.; Deng, J.; Zhu, X.; and Zhang, L. 2025a. ImagiDrive: A Unified Imagination-and-Planning Framework for Autonomous Driving. *arXiv preprint arXiv:2508.11428*.

Li, K.; Li, Z.; Lan, S.; Xie, Y.; Zhang, Z.; Liu, J.; Wu, Z.; Yu, Z.; and Alvarez, J. M. 2025b. Hydra-MDP++: Advancing End-to-End Driving via Expert-Guided Hydra-Distillation. arXiv:2503.12820.

Li, Y.; Fan, L.; He, J.; Wang, Y.; Chen, Y.; Zhang, Z.; and Tan, T. 2025c. Enhancing End-to-End Autonomous Driving with Latent World Model. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Li, Y.; Shang, S.; Liu, W.; Zhan, B.; Wang, H.; Wang, Y.; Chen, Y.; Wang, X.; An, Y.; Tang, C.; Hou, L.; Fan, L.; and Zhang, Z. 2025d. DriveVLA-W0: World Models Amplify Data Scaling Law in Autonomous Driving. arXiv:2510.12796.

Li, Y.; Xiong, K.; Guo, X.; Li, F.; Yan, S.; Xu, G.; Zhou, L.; Chen, L.; Sun, H.; Wang, B.; Ma, K.; Chen, G.; Ye, H.; Liu, W.; and Wang, X. 2025e. ReCogDrive: A Reinforced Cognitive Framework for End-to-End Autonomous Driving. arXiv:2506.08052.

Liao, B.; Chen, S.; Yin, H.; Jiang, B.; Wang, C.; Yan, S.; Zhang, X.; Li, X.; Zhang, Y.; Zhang, Q.; and Wang, X. 2025. DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12037–12047.

Liu, M.; Zhang, D.; Liu, J.; Cui, J.; Xie, H.; Chen, G.; Ye, H.; Yang, M. Y.; Nex, F.; and Cheng, H. 2026a. Driveva: Video action models are zero-shot drivers. *arXiv preprint arXiv:2604.04198*.

Shao, H.; Hu, Y.; Wang, L.; Song, G.; Waslander, S. L.; Liu, Y.; and Li, H. 2024. LMDrive: Closed-Loop End-to-End Driving with Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15120–15130.

Shi, C.; Xu, J.; Shi, S.; Sheng, K.; Zhang, B.; and Jiang, L. 2026. DriveWAM: Video Generative Priors Enable Scalable World-Action Modeling for Autonomous Driving. *arXiv preprint arXiv:2605.28544*.

Sun, W.; Lin, X.; Chen, K.; Pei, Z.; Li, X.; Shi, Y.; and Zheng, S. 2026. SparseDriveV2: Scoring is All You Need for End-to-End Autonomous Driving. arXiv:2603.29163.

Sun, W.; Lin, X.; Shi, Y.; Zhang, C.; Wu, H.; and Zheng, S. 2025. SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 8795–8801.

Wan, T.; Wang, A.; Ai, B.; Wen, B.; Mao, C.; Xie, C.-W.; Chen, D.; Yu, F.; Zhao, H.; Yang, J.; et al. 2025. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*.

Wang, L.; Yang, Z.; Bai, C.; Zhang, G.; Liu, X.; Zheng, X.; Long, X.-X.; Lu, C.-T.; and Lu, C. 2026a. Drive-JEPA: Video JEPA Meets Multimodal Trajectory Distillation for End-to-End Driving. arXiv:2601.22032.

Wang, L.; Zheng, Y.; Chen, Q.; Li, S.; Zhang, Y.; Xing, Z.; Zhang, Q.; Li, X.; Qian, D.; Yang, P.; Dong, Y.; Hao, C.; Ye, X.; Han, J.; Pan, Y.; and Zhao, D. 2026b. Latent-WAM: Latent World Action Modeling for End-to-End Autonomous Driving. arXiv:2603.24581.

Wang, X.; Wang, X.; Zhou, T.; Chen, G.; Gui, X.; Xu, Z.; Wu, X.; Tan, F.; Zhou, H.; and Yang, M. 2026c. ChainFlow-VLA: Causal Flow Planning with Vision-Language Models. arXiv:2605.23270.

Wang, X.; Zhu, Z.; Huang, G.; Chen, X.; Zhu, J.; and Lu, J. 2024. DriveDreamer: Towards Real-World-Drive World Models for Autonomous Driving. In *Computer Vision – ECCV 2024*, 55–72.

Xia, T.; Li, Y.; Zhou, L.; Yao, J.; Xiong, K.; Sun, H.; Wang, B.; Ma, K.; Chen, G.; Ye, H.; et al. 2025. Drivelaw: Unifying planning and video generation in a latent driving world. *arXiv preprint arXiv:2512.23421*.

Xu, M.; Cui, J.; Cai, F.; Shang, H.; Zhu, Z.; Luan, S.; Xu, Y.; Zhang, N.; Li, Y.; Cai, J.; and Zhu, S. 2025a. WAM-Diff: A Masked Diffusion VLA Framework with MoE and Online Reinforcement Learning for Autonomous Driving. arXiv:2512.11872.

Xu, Y.; Cui, J.; Cai, F.; Zhu, Z.; Shang, H.; Luan, S.; Xu, M.; Zhang, N.; Li, Y.; Cai, J.; and Zhu, S. 2025b. WAM-Flow: Parallel Coarse-to-Fine Motion Planning via Discrete Flow Matching for Autonomous Driving. arXiv:2512.06112.

Yang, P.; Zheng, Y.; Xing, Z.; Zhang, Q.; Qian, D.; Wang, L.; Zhang, Y.; Guo, S.; Xia, Z.; Chen, Q.; Han, J.; Xu, L.; Pan, Y.; and Zhao, D. 2026. DreamerAD: Efficient Reinforcement Learning via Latent World Model for Autonomous Driving. arXiv:2603.24587.

Yao, W.; Li, Z.; Lan, S.; Wang, Z.; Sun, X.; Alvarez, J. M.; and Wu, Z. 2025. DriveSuprim: Towards Precise Trajectory Selection for End-to-End Planning. arXiv:2506.06659.

Yao, Z.; Liu, H.; Jiang, Y.; Zhu, Z.; Guo, Z.; Wang, J.; Liu, T.; Cui, J.; Yang, K.; Xie, H.; Zhao, J.; Chen, G.; and Ye, H.

2026. Discrete-WAM: Unified Discrete Vision-Action Token Editing for World-Policy Learning. arXiv:2606.05645.

Zhou, G.; Pan, H.; LeCun, Y.; and Pinto, L. 2024. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*.

Zhou, H.; Lin, L.; Wang, J.; Lu, Y.; Bai, D.; Liu, B.; Wang, Y.; Geiger, A.; and Liao, Y. 2024. HUGSIM: A Real-Time, Photo-Realistic and Closed-Loop Simulator for Autonomous Driving. *arXiv preprint arXiv:2412.01718*.

Zhou, X.; Han, X.; Yang, F.; Ma, Y.; Tresp, V.; and Knoll, A. 2026. OpenDriveVLA: Towards End-to-End Autonomous Driving with Large Vision Language Action Model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(16): 13782–13790.

Zhu, J.; Jia, Z.; Gao, T.; Deng, J.; Li, S.; Zhang, L.; Liu, F.; Jia, P.; and Lang, X. 2026. Other Vehicle Trajectories Are Also Needed: A Driving World Model Unifies Ego-Other Vehicle Trajectories in Video Latent Space. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(16): 13934–13942.

Zou, J.; Chen, S.; Liao, B.; Zheng, Z.; Song, Y.; Zhang, L.; Zhang, Q.; Liu, W.; and Wang, X. 2025. DiffusionDriveV2: Reinforcement Learning-Constrained Truncated Diffusion Modeling in End-to-End Autonomous Driving. arXiv:2512.07745.

## A. HUGSIM Closed-Loop Evaluation

**Protocol.** The 436 scenarios span four source datasets and four difficulty levels; the difficulty distribution is reported in Table 2 of the main text and the dataset distribution in Table 6. All methods use the same scenarios, ground-truth driving commands, aggregation procedure, and HUGSIM controller and evaluator. Specifically, we use HUGSIM at commit `ead17f2`, which applies the trajectory-to-heading coordinate-order correction introduced in PR #57. Sensor configurations follow the original methods: LTF uses three front-view cameras, while the others use four.

We rescore all evaluated methods using this fixed code snapshot, ensuring consistent controller and metric implementations. DrivoR was originally evaluated on an earlier HUGSIM release containing 345 scenarios, whereas our evaluation uses the current 436-scenario set. Its published scores are therefore not directly comparable to those reported here. For DrivoR, we report the variant trained on the combined NAVSIM training and validation sets without the additional simulated training data introduced in SimScale. In contrast, Stage 2 of WA-JEPA uses only the NAVSIM training set.

**Aggregation robustness.** Within each difficulty level we average scenarios uniformly inside a source dataset and then average the four datasets uniformly, as in the HUGSIM benchmark. The single overall HD-Score is obtained by weighting the four difficulty levels by their scenario counts $(80/157/96/103)$, following subsequent work on this benchmark. Table 5 reports two alternative rules, uniform averaging over the four datasets and uniform averaging over all 436 scenarios. The method ranking in the main text is unchanged under both.

**Per-dataset results.** Table 6 reports the HD-Score separately for each HUGSIM source dataset. None of the four datasets is used for either Stage 1 or Stage 2 training. WA-JEPA achieves the highest HD-Score on every dataset, indicating consistent source-disjoint generalization across diverse visual domains.

| Aggregation | WA-JEPA | LTF | DrivoR | UniAD | VAD |
|---|---|---|---|---|---|
| Primary | **0.4462** | 0.2310 | 0.3252 | 0.3124 | 0.1393 |
| Dataset-uniform | **0.4483** | 0.2300 | 0.3246 | 0.3085 | 0.1304 |
| Scenario-uniform | **0.4464** | 0.2243 | 0.3194 | 0.3082 | 0.1266 |

Table 5: HD-Score under three aggregation rules. *Primary* is the rule described above; the other two rows average uniformly over the four datasets and over all 436 scenarios, respectively. The best result in each row is bold.

| Dataset | $n$ | WA-JEPA | LTF | DrivoR | UniAD | VAD |
|---|---|---|---|---|---|---|
| nuScenes | 88 | **0.4725** | 0.3334 | 0.3830 | 0.3405 | 0.2069 |
| KITTI-360 | 113 | **0.2963** | 0.0969 | 0.2175 | 0.0550 | 0.0272 |
| Waymo | 108 | **0.5542** | 0.2478 | 0.4025 | 0.4372 | 0.1376 |
| PandaSet | 127 | **0.4702** | 0.2419 | 0.2955 | 0.4012 | 0.1500 |

Table 6: Per-dataset HD-Scores on the 436 HUGSIM scenarios. $n$ denotes the number of scenarios. The best result in each row is bold.

| Statistic | EPDMS |
|---|---|
| Number of seeds | 10 |
| Mean | 91.7014 |
| Standard deviation | 0.0531 |
| Standard error | 0.0168 |
| 95% $t$-confidence interval | [91.6634, 91.7393] |
| Median | 91.6960 |
| Range | [91.6294, 91.8070] |

Table 7: Seed-level EPDMS variability for the main WA-JEPA experiment over a fixed set of ten seeds.

## B. Qualitative Results

**HUGSIM closed-loop rollouts.** Figure 5 visualizes zero-shot closed-loop rollouts across turning, oncoming-vehicle encounters, and overtaking. WA-JEPA follows the drivable corridor and maintains lateral clearance from reactive agents across all four source datasets despite the visual domain gap between HUGSIM renderings and the training data.

**NAVSIM trajectory predictions.** Figure 6 shows representative Stage 2 predictions on NAVSIM. Across turning, fork and gateway navigation, stopping, and straight-driving scenarios, the predicted trajectories agree with the references in maneuver direction and overall geometry, with only minor local deviations.

## C. Additional Experimental Details

**Details of stop-gradient design.** Although the scene output stream conditions on the action tokens during the forward pass, gradients from the scene prediction loss are blocked at the action-token interface and therefore do not update the action stream. In contrast, the action output stream attends to differentiable historical context and future scene tokens, allowing action supervision to shape the learned scene representations through the joint interaction module. This asymmetric gradient design preserves future scene prediction while encouraging the scene representations to capture future dynamics that are relevant to ego planning.

**Inference details and seed-level variability.** The WA-JEPA flow predictor uses 12 sampling steps. For all experiments involving stochastic noise initialization, we use a fixed set of ten seeds. For each seed, we reinitialize the sampling noise and run the full evaluation while keeping the model parameters, scenarios, and inference settings unchanged. All sub-metrics and EPDMS are computed independently for each seed, with EPDMS following Eq. (21). We report their arithmetic mean before rounding.

As shown in Table 7, the main WA-JEPA experiment achieves a mean EPDMS of 91.7014, which is reported as 91.7 in the main text after rounding. Methods without stochastic noise initialization are evaluated once, while all reproduced baselines use their native inference procedures.

## D. NAVSIM Evaluation Metrics

Evaluation follows the official NAVSIM protocol, with all scores computed by the pseudo-simulator on `navtest`. This

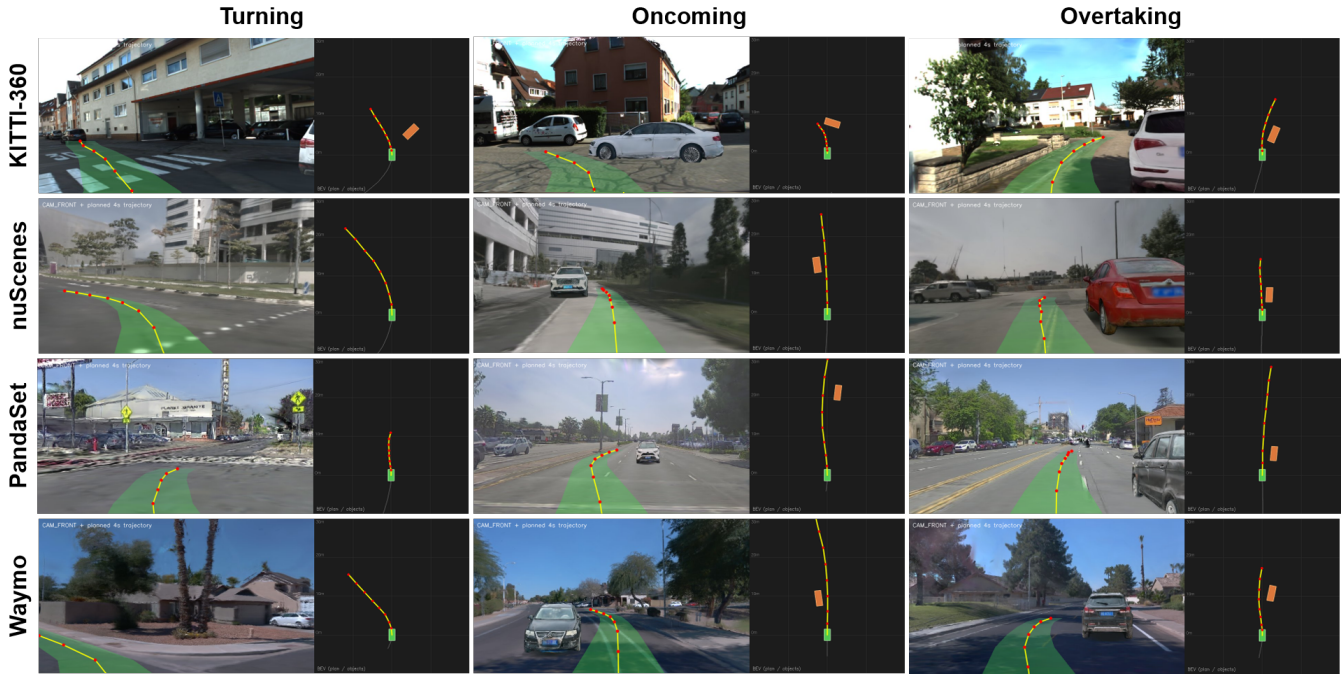


Figure 5: Zero-shot closed-loop rollouts of WA-JEPA on HUGSIM. Rows correspond to the four source datasets and columns to three scenario types. Each pair shows the front camera with the projected 4 s plan (left) and the BEV view with the planned trajectory and detected objects (right). The green region marks the drivable corridor, the yellow–red curve the planned trajectory, the green box the ego vehicle, and orange boxes other agents.

section defines each sub-metric and the aggregation rules used in Tables 1 and 3 of the main text.

**Sub-metrics.** NAVSIM-v1 uses five sub-metrics: *no-at-fault collision* (NC), which is 0 if the ego vehicle causes a collision and 1 otherwise (collisions for which the ego is not responsible, e.g. being rear-ended, are excluded); *drivable-area compliance* (DAC), which is 0 if any part of the ego footprint leaves the drivable area and 1 otherwise; *time-to-collision* (TTC), which is 1 when the minimum time-to-collision along the rollout stays above a safety threshold under a constant-velocity projection of the ego and surrounding agents; *comfort* (Comf.), which is 1 when longitudinal and lateral accelerations, jerk, and yaw rate remain within the human-driving bounds estimated from the dataset; and *ego progress* (EP), the ratio of the ego's traveled distance along the route centerline to that of the privileged PDM-Closed planner, clipped to $[0, 1]$.

NAVSIM-v2 additionally introduces: *driving-direction compliance* (DDC), which penalizes driving against the nominal direction of the current lane, with a graded penalty depending on the magnitude of the violation; *traffic-light compliance* (TLC), which is 0 if the ego crosses a stop line under a red light and 1 otherwise; *lane keeping* (LK), which measures whether the ego stays within its assigned lane corridor over the horizon; *history comfort* (HC), which evaluates comfort bounds jointly over the past trajectory and the planned trajectory, thereby penalizing abrupt transitions at the planning boundary; and *extended comfort* (EC), which measures the consistency of the kinematic profile across the two stages of the pseudo-simulation, so that trajectories that change sharply between rollouts are penalized.

**Aggregation.** Sub-metrics are grouped into multiplicative penalties $\mathcal{M}_{\text{pen}}$ and a weighted average $\mathcal{M}_{\text{avg}}$. The NAVSIM-v1 Predictive Driver Model Score (PDMS) is

$$\text{PDMS} = \underbrace{\text{NC} \cdot \text{DAC}}_{\mathcal{M}_{\text{pen}}} \cdot \frac{5\,\text{TTC} + 2\,\text{Comf.} + 5\,\text{EP}}{12}, \tag{17}$$

so that a single safety violation drives the score to zero, while the remaining terms trade off safety margin, comfort, and progress.

The NAVSIM-v2 Extended PDMS (EPDMS) additionally applies a human-reference penalty filter before aggregating the sub-metrics. Let $s_{i,m}^{\text{agent}}$ and $s_{i,m}^{\text{human}}$ denote the agent and human-reference scores, respectively, for metric $m$ in scenario $i$. For original first-stage pseudo-simulation scenarios, the corrected sub-metric is defined as

$$\tilde{s}_{i,m} = \text{filter}_m\left(s_{i,m}^{\text{agent}}, s_{i,m}^{\text{human}}\right) = \begin{cases} 1, & \substack{m \in \mathcal{M}_{\text{filt}},\\ s_{i,m}^{\text{human}}=0,} \\ s_{i,m}^{\text{agent}}, & \text{otherwise,} \end{cases} \tag{18}$$

where $\mathcal{M}_{\text{filt}} = \{\text{NC}, \text{DAC}, \text{DDC}, \text{TLC}, \text{EP}, \text{TTC}, \text{LK}, \text{HC}\}$. Synthetic second-stage pseudo-simulation scenarios do not use this filter, and EC is computed subsequently from the paired rollouts. Thus, the filter does not remove scenarios;

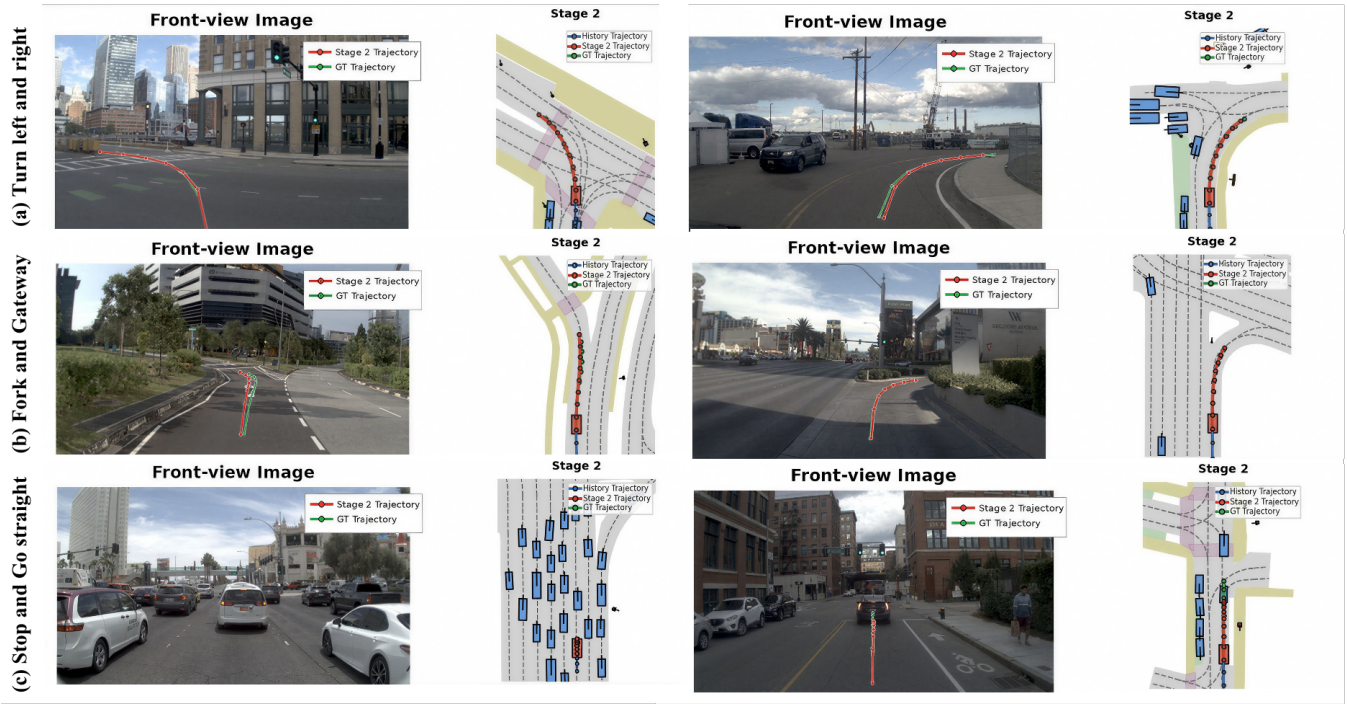


Figure 6: Trajectory predictions on representative NAVSIM scenarios: (a) left and right turns, (b) fork and gateway navigation, and (c) stopping and straight driving.

it suppresses a metric-specific penalty when the human reference incurs the same violation.

Using the filtered sub-metrics, the per-scenario extended score is

$$q_i = \tilde{s}_{i,\mathrm{NC}}\tilde{s}_{i,\mathrm{DAC}}\tilde{s}_{i,\mathrm{DDC}}\tilde{s}_{i,\mathrm{TLC}} \times \frac{5\tilde{s}_{i,\mathrm{EP}} + 5\tilde{s}_{i,\mathrm{TTC}} + 2\tilde{s}_{i,\mathrm{LK}} + 2\tilde{s}_{i,\mathrm{HC}} + 2\tilde{s}_{i,\mathrm{EC}}}{16}. \tag{19}$$

The final benchmark EPDMS follows the official two-stage NAVSIM-v2 aggregation. Let $\mathcal{G}$ denote the official mapping groups, $\mathcal{I}_{g,b}^{(r)}$ the scenarios assigned to branch $b \in \{1,2\}$ at pseudo-simulation stage $r \in \{1,2\}$, and $\alpha_i$ the provided scene weight. Define

$$\bar{q}_{g,b}^{(r)} = \frac{\sum_{i\in\mathcal{I}_{g,b}^{(r)}} \alpha_i q_i}{\sum_{i\in\mathcal{I}_{g,b}^{(r)}} \alpha_i}. \tag{20}$$

The reported score is

$$\mathrm{EPDMS} = \frac{100}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}\frac{1}{2}\sum_{b=1}^{2}\bar{q}_{g,b}^{(1)}\bar{q}_{g,b}^{(2)}. \tag{21}$$

**EPDMS$^*$ versus EPDMS.** All corrected NAVSIM-v2 results are computed using the official NAVSIM devkit at commit `359c7f7`, which recomputes the multiplicative and weighted terms after applying the human-reference filter. EPDMS$^*$ denotes scores obtained with the pre-fix evaluator, whereas EPDMS denotes scores obtained with this corrected protocol. The two settings are therefore reported separately and compared only within the same evaluation protocol.

## E. Temporal Representation Metrics

We evaluate whether predicted future scene tokens preserve the temporal variation of the corresponding EMA target tokens. Consistent with the main paper, the comparison uses two target-referenced metrics: the *directional similarity collapse gap* (lower is better) and the *change-magnitude collapse* (closer to 1 is better). Both are computed in the projected scene-token space.

**Dynamic-token selection.** Let $\widehat{\mathbf{z}}_{i,r,t}$ and $\mathbf{z}_{i,r,t}$ denote the predicted and target features for prediction instance $i$, camera–spatial location $r$, and future token step $t$. To prevent static regions from dominating the statistics, we rank locations by the target's mean adjacent-step change,

$$s_{i,r} = \frac{1}{F-1}\sum_{t=0}^{F-2}\left\|\mathbf{z}_{i,r,t+1} - \mathbf{z}_{i,r,t}\right\|_2, \qquad \mathcal{A}_i = \mathrm{TopK}_r(s_{i,r}). \tag{22}$$

The same target-selected set $\mathcal{A}_i$ is used for both methods and both metrics. Selection is performed independently for each prediction instance.

**Directional similarity collapse gap.** For a feature sequence $\mathbf{q}$, define the mean cosine similarity over all ordered

pairs of distinct future steps at the selected locations as

$$C(\mathbf{q}) = \frac{1}{N_{\cos}} \sum_{i} \sum_{r \in \mathcal{A}_i} \sum_{t \neq u} \cos\left(\mathbf{q}_{i,r,t}, \mathbf{q}_{i,r,u}\right),$$
$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^{\mathsf{T}}\mathbf{b}}{\max(\|\mathbf{a}\|_2, \epsilon)\max(\|\mathbf{b}\|_2, \epsilon)}, \quad \epsilon = 10^{-6}. \tag{23}$$

where $N_{\cos}$ is the number of valid terms. The reported metric is

$$\Delta_{\mathrm{dir}} = C(\widehat{\mathbf{z}}) - C(\mathbf{z}). \tag{24}$$

A positive value indicates that predicted future steps are more mutually similar than the targets and therefore exhibit additional directional collapse. Accordingly, lower values indicate better target-relative temporal preservation, as stated in the main text.

**Change-magnitude collapse.** We also compare the mean adjacent-step feature change,

$$D(\mathbf{q}) = \frac{1}{N_{\Delta}} \sum_{i} \sum_{r \in \mathcal{A}_i} \sum_{t=0}^{F-2} \left\|\mathbf{q}_{i,r,t+1} - \mathbf{q}_{i,r,t}\right\|_2,$$
$$R_{\Delta} = \frac{D(\widehat{\mathbf{z}})}{\max(D(\mathbf{z}), \epsilon)}. \tag{25}$$

Although referred to as change-magnitude collapse in the main text, $R_{\Delta}$ is a ratio: $R_{\Delta} = 1$ means that the prediction preserves the target's average temporal change, while values below $1$ indicate under-variation. Thus, values closer to $1$ are better for the comparisons reported in the main text.

**Evaluation protocol.** Both objectives use the same projected feature space, target-selected locations, and global averaging over all valid prediction instances and token pairs. For flow matching, the metric is computed from the one-step $x$-prediction at the sampled training flow time, without running the multi-step inference sampler; regression is evaluated from its direct latent prediction. The summary values in the main text are arithmetic means of the raw logged metrics over the common 0–36k training interval. For the standard setting, eight future frames with tubelet size 2 give $F = 4$ future token steps, and $K = 64$ target-dynamic locations are selected from the four-camera scene-token grid.