# SPARSEWAM: DRIVING WORLD-ACTION MODELS ARE WORTH 16 REGISTER TOKENS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

World Action Models (WAMs) transfer scene-dynamics priors to planning by jointly forecasting future scenes and ego trajectories. However, WAMs built on video generators must generate dense future videos at inference, incurring prohibitive latency, while efficient variants discard the world branch at deployment and thus forgo test-time imagination. Even when dense video latents are replaced with patch-level DINO or JEPA features, these representations still involve long token sequences and encode decision-irrelevant information. We therefore propose SparseWAM, which forecasts in a sparse register space of only 16 tokens per frame, compressed from frozen DINOv2 features and shaped by joint planning-and-scoring pretraining to concentrate decision-relevant information. Within this space, SparseWAM employs a modality-decoupled Mixture-of-Transformers with a causal attention mask that lets action tokens condition on imagined future registers, retaining explicit imagine-then-act reasoning at deployment. Its compactness further enables low-cost multimodal sampling. As a result, our SparseWAM achieves 92.1 PDMS on NAVSIM v1, 89.8 EPDMS on NAVSIM v2, and 40.1 on NavHard at only 260 ms latency, roughly 1/3 that of DriveVA, demonstrating the efficiency of the sparse register space. Code will be released upon acceptance.

## 1 INTRODUCTION

Autonomous driving is inherently interactive, requiring not only perception and understanding of the current scene but also anticipation of surrounding agents' behavior for safe and efficient planning. This requirement has motivated a growing family of World Action Models (WAMs) that jointly model future scene evolution and trajectory planning (Zhao et al., 2026a; Zhang et al., 2025; Xu et al., 2026; Liu et al., 2026; Zhao et al., 2026b; Li et al., 2026a; Xia et al., 2026). Because future-frame prediction provides dense, annotation-free supervision that scales with raw driving logs, these models can transfer scene-dynamics priors to planning, improving interactive reasoning and generalization, and therefore offer a compelling alternative to conventional end-to-end methods (Hu et al., 2023; Jiang et al., 2023; Prakash et al., 2021; Li et al., 2024; Liao et al., 2025) and Vision Language Action models (Hwang et al., 2024; Zhou et al., 2026).

Yet realizing this promise through explicit future prediction introduces a fundamental tension between imagination and efficiency. Inspired by the success of DreamZero in embodied intelligence (Ye et al., 2026), recent driving WAMs build upon pretrained video generators such as Wan2.2-5B (Wan et al., 2025). They jointly denoise future video latents and action tokens within a unified generative process, thereby transferring the scene-dynamics priors acquired during video pretraining to trajectory planning (Liu et al., 2026; Shi et al., 2026). While this design achieves strong performance in *single-trajectory* planning, it requires future videos to be explicitly generated at inference time, resulting in substantial latency that hinders practical on-vehicle deployment. More recent approaches, including SimWAM (Zhao et al., 2026b) and Metis (Li et al., 2026a), improve efficiency by using structured attention masks to decouple action prediction from future-video generation. This allows the video-generation branch to be removed during inference. Their efficiency design, however, fundamentally changes the role of the world model. Rather than enabling the deployed policy to imagine possible futures before acting, it serves primarily as a source of representation shaping and auxiliary supervision during training.



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.

The key may therefore lie not in abandoning test-time forecasting, but in reconsidering what the model is asked to predict. When anticipating how a traffic scene may evolve, human drivers do not reason equally about every visible region. Static background elements such as the sky and roadside buildings generally receive little attention, whereas vehicles, pedestrians, and other entities that may affect the driving decision become the focus of anticipation. Dense future-video prediction does not make this distinction. It requires the model to reconstruct the entire visual scene, devoting substantial capacity and computation to appearance details that have limited relevance to planning. As a result, the dynamics most consequential for decision making must compete with largely irrelevant visual content for modeling capacity and learning signals.

Recent WAMs attempt to reduce this redundancy by predicting future representations alongside action trajectories in semantic latent spaces learned by models such as DINO and JEPA (Oquab et al., 2024; Assran et al., 2025). These representations offer more semantically meaningful prediction targets than pixels or VAE latents Wan et al. (2025), but they do not fully resolve the underlying problem. Their patch-level token sequences can be comparable to, or even longer than, those of VAE latents, so forecasting them at inference time remains computationally demanding. More fundamentally, representations learned through generic visual pretraining are optimized to capture broadly useful visual semantics rather than the particular aspects of a scene that matter for driving decisions. They may therefore continue to encode background appearance and other variations that are visually salient but behaviorally irrelevant. These limitations point to a more fundamental question. ***What representation space would allow a WAM to imagine the future efficiently while remaining tightly aligned with the decisions it must make?***

We address this challenge by learning a decision-aligned sparse register space with only 16 tokens per frame. Learnable register tokens (Darcet et al., 2024) compress the dense feature maps produced by a frozen DINOv2 encoder (Oquab et al., 2024) into a compact representation, which is further shaped through joint planning-and-scoring pretraining. The planning objective encourages the registers to retain scene information that informs trajectory prediction, while the scoring objective complements it by preserving safety-critical cues that may not directly determine the nominal trajectory but remain essential for evaluating its consequences. The resulting registers therefore serve not merely as a compressed visual representation, but as a compact decision-oriented abstraction of the driving scene. We examine this property through counterfactual scene-editing probes that remove lead vehicles, insert pedestrians, or alter weather and illumination. The learned representation responds strongly to interventions that change driving decisions while remaining comparatively insensitive to appearance variations that leave those decisions unchanged. This compact representation makes it possible to preserve explicit test-time imagination without incurring the cost of dense video or feature prediction. Building on it, we introduce **SparseWAM**, a lightweight Mixture-of-Transformers denoiser (Liang et al., 2024) that uses rectified flow (Liu et al., 2022; Lipman et al., 2022) to model future registers together with ego trajectories. Its modality-specific transformer branches reduce interference between world prediction and action generation while still allowing the two processes to exchange information. More importantly, we impose a causal attention structure that prevents future registers from attending to action tokens, while allowing action tokens to condition on denoised future registers. SparseWAM consequently learns a directed generative process in which it first anticipates how the scene may evolve and then plans in light of that imagined future. Unlike approaches that remove world prediction for efficiency, SparseWAM retains explicit imagine-then-act reasoning in the deployed policy.

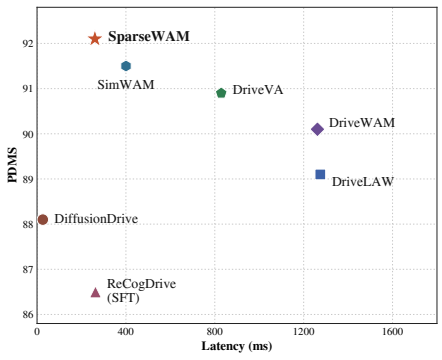



Figure 1: SparseWAM achieves the best PDMS with lowest latency than WAM planners on NAVSIM.

This efficiency extends naturally to multimodal planning. With only 16 tokens per future frame, SparseWAM can efficiently generate diverse future–trajectory pairs by denoising multiple independent noise samples. As shown in Fig.1, it achieves a state-of-the-art PDMS of 92.1 with only 260 ms latency, outperforming the recent DriveVA (90.9 PDMS, 829 ms) with 1/3 inference latency. Even

under single-mode inference, SparseWAM delivers performance comparable to or better than many existing single-trajectory WAMs, with a latency of just 122 ms. Rather than trading imagination for efficiency, SparseWAM makes explicit future reasoning both multimodal and practical.

Our contributions are: **(i)** We construct a driving-oriented sparse register space obtained through joint planning-and-scoring pretraining, and design counterfactual scene-editing probes that verify its sensitivity is concentrated on decision-relevant scene elements rather than appearance-related nuisance factors. **(ii)** We develop a WAM framework operating in the proposed sparse register latent space, which adopts an MoT architecture that prevents cross-modal contamination and employs a causal attention mask. **(iii)** Experiments across multiple NAVSIM benchmarks show that SparseWAM attains 92.1 PDMS at lower latency, surpassing state-of-the-art WAMs, and remains highly competitive even under the single-mode setting, demonstrating the efficiency of the method.

## 2 Related Work

### 2.1 End-to-End Autonomous Driving

End-to-end autonomous driving has progressed along two complementary lines. Classical sensor-to-planner pipelines map multi-sensor observations to trajectories through unified differentiable stacks (Prakash et al., 2021; Hu et al., 2022; 2023; Jiang et al., 2023; Li et al., 2024; Chen et al., 2024a; Guo et al., 2024; Liao et al., 2025; Xing et al., 2025; Guo et al., 2026), typically optimizing planning with auxiliary perception or scoring objectives. In parallel, vision–language–action (VLA) models incorporate language instructions and semantic scene understanding before producing actions (Hwang et al., 2024; Tian et al., 2024; Xu et al., 2024; Shao et al., 2024; Jiang et al., 2024; Wang et al., 2024; Zhou et al., 2026). Both paradigms have substantially advanced closed-loop planning by mapping current observations or semantic instructions directly to actions. However, these approaches mainly infer actions from current observations, leaving future scene evolution implicitly modeled. This motivates WAMs, which explicitly incorporate predictive world dynamics into end-to-end planning.

### 2.2 World-Action Models for Autonomous Driving

WAMs jointly model future scene evolution and ego actions, and their effectiveness depends heavily on the choice of future representation. Video-based WAMs inherit dynamics priors from large-scale video models by coupling future visual prediction with action generation (Chen et al., 2024b; Zhang et al., 2025; Liu et al., 2026; Shi et al., 2026; Xia et al., 2026). These methods demonstrate the value of predictive world dynamics, but their dense video prediction objective inevitably requires modeling substantial visual details beyond those required for driving decisions. To reduce inference cost, SimWAM Zhao et al. (2026b) and Metis Li et al. (2026a) use structured attention masks to decouple action prediction from generated futures, and DriveVLA-W0 Li et al. (2025a) uses future-image prediction as training-time supervision with a lightweight action expert. These designs preserve representation learning during training but forgo test-time imagination. However, these approaches mainly reduce computation by avoiding future generation, rather than reconsidering what future information is necessary for planning. Latent-space WAMs instead forecast future scene states in a learned latent space Wang et al. (2026a); Gui et al. (2026); Wang et al. (2026c). These methods improve semantic abstraction, yet they still rely on visual representations that are not explicitly optimized for driving decisions. This motivates SparseWAM, which forecasts future dynamics in a driving-oriented latent space.

### 2.3 Visual Representations for Autonomous Driving

Self-supervised encoders such as DINOv2, V-JEPA, and MAE learn semantically rich features that transfer across vision tasks (Oquab et al., 2024; Assran et al., 2025; He et al., 2021). Driving-oriented methods further tailor such representations to the driving domain. Drive-JEPA adapts JEPA-style latent prediction to driving videos (Wang et al., 2026b), and DrivoR compresses multi-camera features into a small set of register tokens for trajectory generation and scoring (Kirby et al., 2026). However, these representations mainly serve as inputs to a current-timestep planner rather than as a space for forecasting future scene evolution. SparseWAM instead takes this compressed and driving-adapted space as the prediction target of future scene evolution.

# 3 PRELIMINARIES

**Problem Formulation.** Given a navigation command $l_t$, historical RGB observations $I_{t-n:t}$, and ego state $q_t \in \mathbb{R}^{d_q}$, a world action model (WAM) jointly predicts the future scene and the ego trajectory $a_{t:t+T} \in \mathbb{R}^{T\times 3}$, where each waypoint is $(x, y, \psi)$ in the ego frame and $T$ denotes future horizon. Previous WAMs perform this prediction in dense latent spaces, such as VAE latents (Wan et al., 2025) or DINO/V-JEPA features (Oquab et al., 2024; Assran et al., 2025), with $N_p$ tokens per frame. We instead encode each frame into sparse queries and model their evolution jointly with planning,

$$Q_{t-n:t} = \mathcal{E}\left(I_{t-n:t}\right), \qquad (a_{t:t+T},\, Q_{t+1:t+T}) \sim p_\theta(\cdot \mid Q_{t-n:t},\, q_t,\, l_t)\,, \tag{1}$$

where $Q_i \in \mathbb{R}^{N_r\times d}$ and $N_r = 16 \ll N_p$. We denote the condition by $c = (Q_{t-n:t}, q_t, l_t)$ and the future queries by $Q_f$.

**Rectified Flow.** We model this joint distribution with rectified flow (Liu et al., 2022; Lipman et al., 2022). Given a clean sample $x$ and noise $\epsilon \sim \mathcal{N}(0, I)$, the model regresses the velocity of the linear interpolation path,

$$x^\sigma = (1-\sigma)\, x + \sigma\, \epsilon, \qquad \mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{x,\epsilon,\sigma}\left\|v_\theta(x^\sigma, \sigma, c) - (\epsilon - x)\right\|_2^2\,, \tag{2}$$

where $\sigma \in [0, 1]$. At inference, samples are drawn by integrating $v_\theta$ from $\sigma = 1$ to 0 with an Euler solver. Since $v_\theta$ models the full conditional distribution, varying the initial noise yields distinct yet plausible trajectories under the same condition $c$, naturally supporting multi-modal planning.

# 4 METHOD

## 4.1 PRETRAINING OF THE SPARSE QUERY SPACE

**Register Perceiver.** Self-supervised latents such as DINO Oquab et al. (2024) and V-JEPA Assran et al. (2025) are semantically richer than pixels or VAE latents (Wan et al., 2025), yet their token counts are no smaller. We therefore compress the semantic content of the DINO latent space using register tokens Darcet et al. (2024). Concretely, given a single frame $I \in \mathbb{R}^{3\times H_I\times W_I}$, we concatenate its patch embedding sequence $P = [p_1, \ldots, p_{N_p}]$ with a set of learnable register tokens $R = [r_1, \ldots, r_{N_r}]$ and feed the concatenation to a ViT Dosovitskiy et al. (2020). After the registers have interacted with the image tokens across all Transformer layers, we retain only the register portion of the output sequence as the scene representation,

$$\left[\tilde{P};\, Q\right] = \mathrm{ViT}\big([P;\, R]\big), \qquad Q \in \mathbb{R}^{N_r\times d}, \quad N_r = 16. \tag{3}$$

Because the number of registers is decoupled from the input resolution, the scene token count drops by roughly two orders of magnitude relative to a patch-level representation, which is what makes the lightweight WAM architecture of Section 4.2 feasible. In practice the ViT backbone uses frozen DINOv2 weights, and only the register tokens and LoRA adapters (Hu et al., 2021) are trained, so that generic visual priors are preserved in domain-specific adaptation to driving.

**Pretraining tasks.** To make the registers carry sufficient scene information while attending as little as possible to driving-irrelevant background, we shape the register encoder with two complementary tasks, planning and scoring. As shown in Stage 1 of Fig. 2, the planning branch initializes a set of trajectory queries $A$, lets them interact with the registers, and decodes a planned trajectory through an MLP. The scoring branch reuses the same trajectory representation, attends to the registers once more, and predicts the PDMS sub-metrics of that trajectory (Dauner et al., 2024). The pretraining loss combines an imitation term with a score-distillation term (Li et al., 2024),

$$\mathcal{L}_{\mathrm{pre}} = \left\|\hat{a} - a^{\mathrm{gt}}\right\|_1 + \lambda_{\mathrm{sc}} \sum_m \mathrm{BCE}\big(\hat{y}_m,\, y_m^{\mathrm{sim}}\big), \tag{4}$$

where $y^{\mathrm{sim}}$ denotes the sub-metric labels provided by the simulator. The two objectives are complementary. The planning objective drives the registers towards scene elements that are highly relevant to trajectory prediction, whereas the scoring objective requires them not to omit elements that need not affect the imitation error against the expert trajectory, yet can decisively change whether a trajectory is feasible (Li et al., 2024; Sun et al., 2026). We will probe the resulting space with counterfactual scene edits find its sensitivity concentrated on decision-relevant scene content rather than on background nuisance factors in Section 5. Once pretraining is complete, the encoder $\mathcal{E}_{\mathrm{reg}}$ is frozen in its entirety and provides stable prediction targets for the WAM.

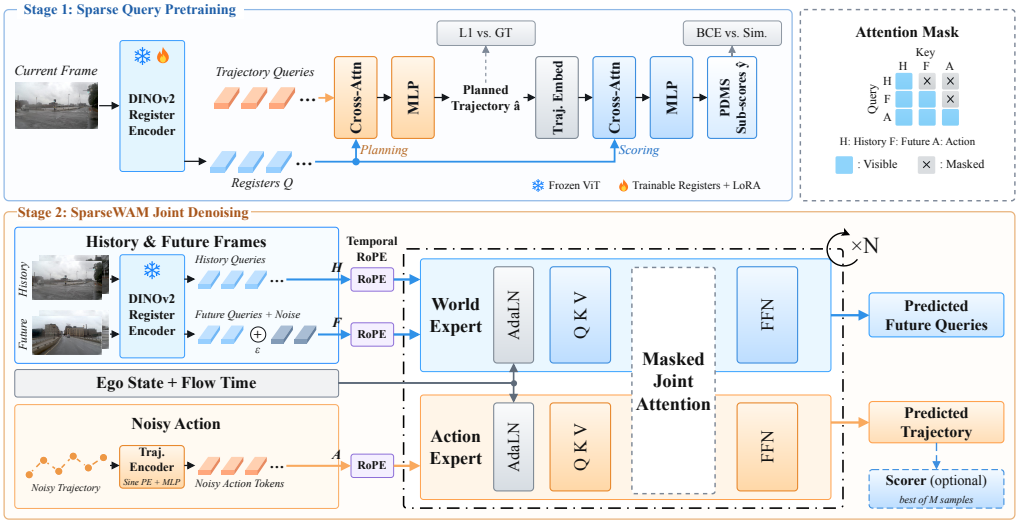



Figure 2: Overview of SparseWAM. Stage 1 pretrains a frozen DINOv2 register encoder with a planning head and a scoring head. Stage 2 encodes history and future frames into sparse queries, perturbs the future queries and the trajectory, and jointly denoises them with a Mixture-of-Transformers under a causal attention mask. An optional scorer selects the best trajectory among $M$ samples.

### 4.2 Efficient MoT Architecture

**MoT architecture.** Scene queries and trajectories are heterogeneous signals. Forcing both through a shared set of weights therefore risks contamination between the two modalities. To keep their representation spaces separate while still allowing sufficient interaction, we adopt a lightweight Mixture-of-Transformers (MoT) architecture (Liang et al., 2024) shown in Fig. 2, comprising a scene expert of width $d_S$ and an action expert of width $d_A$, each with its own QKV projections, output projection, and FFN. The scene expert takes a clean segment $\mathcal{H} = \{Q_{t-1}, Q_t\}$ and a noisy segment $\mathcal{F}$ (the future keyframe queries $\{Q_{t+2}, Q_{t+4}\}$ linearly mixed with Gaussian noise). The action expert reads a single segment $\mathcal{A}$. We represent the trajectory by the increments between consecutive waypoints and standardize each channel with its training-set statistics, so that displacement and heading share the scale of the Gaussian prior, and then perturb the result in the same way. Within each layer, both experts modulate their own hidden states and emit their own queries, keys, and values. These are concatenated along the token dimension, attended jointly under the causal attention mask, and split back so that each expert applies its own output projection and FFN. After the final layer, the two experts output the velocity fields $v_\theta^S$ and $v_\theta^A$.

**Condition injection and positional encoding.** Within both experts, the noise level $\sigma$ of the respective modality, together with the ego state and the navigation command, is injected into the DiT blocks through AdaLN Peebles & Xie (2023). The conditioning segment $\mathcal{H}$ always takes $\sigma = 0$, so that clean observations and the futures to be denoised are distinguished explicitly at the level of conditioning. We also add a one-dimensional rotary positional embedding to $\mathcal{H}$, $\mathcal{F}$, and $\mathcal{A}$ along the same frame time axis, letting scene tokens model temporal relations and action tokens attend to temporally nearby futures, thereby aligning future prediction with planning.

**Causal attention mask.** The Attention mask imposed on the joint attention specifies the direction of information flow among the three segments, as illustrated in Figure 2. The segment $\mathcal{H}$ attends only to itself, so the conditioning representation is never contaminated by the noisy segments. $\mathcal{A}$ may attend to the $\mathcal{F}$ already denoised within the same step, whereas $\mathcal{F}$ does not attend to $\mathcal{A}$. This causal information flow makes the model carry out an explicit imagine-then-act process at inference time. Unlike SimWAM and Metis (Zhao et al., 2026b; Li et al., 2026a), which use masks to sever the dependence of action prediction on future prediction altogether and thereby discard the world model at deployment, our mask retains explicit imagination at test time.

Table 1: **Comparison with state-of-the-art planners on the NAVSIM `navtest` benchmark.** C denotes camera and L denotes LiDAR. The best learned result in each column is shown in **bold**.

| Method | Reference | Sensors | NC↑ | DAC↑ | EP↑ | TTC↑ | C↑ | PDMS↑ |
|---|---|---|---|---|---|---|---|---|
| Human Agent | – | – | 100.0 | 100.0 | 87.5 | 100.0 | 99.9 | 94.8 |
| *Traditional E2E planners* | | | | | | | | |
| UniAD Hu et al. (2023) | CVPR'23 | 6×C | 97.8 | 91.9 | 78.8 | 92.9 | 100.0 | 83.4 |
| TransFuser Prakash et al. (2021) | TPAMI'22 | 3×C+L | 97.7 | 92.8 | 79.2 | 92.8 | 100.0 | 84.0 |
| DiffusionDrive Liao et al. (2025) | CVPR'25 | 3×C+L | 98.2 | 96.2 | 82.2 | 94.7 | 100.0 | 88.1 |
| WoTE Li et al. (2025b) | ICCV'25 | 3×C+L | 98.5 | 96.8 | 81.9 | 94.9 | 99.9 | 88.3 |
| DiffusionDriveV2 Zou et al. (2025) | arXiv'25 | 3×C+L | 98.3 | 97.9 | 87.5 | 94.8 | 99.9 | 91.2 |
| SparseDriveV2 Sun et al. (2026) | arXiv'26 | 3×C | 98.5 | 98.4 | **88.6** | 95.0 | 99.9 | 92.0 |
| *VLM-based planners* | | | | | | | | |
| OneDrive Zhang et al. (2026) | arXiv'26 | 1×C | 98.4 | 95.2 | 81.1 | 94.9 | 100.0 | 86.8 |
| ReCogDrive Li et al. (2025c) | ICLR'26 | 1×C | 97.9 | 97.3 | 87.3 | 94.9 | 100.0 | 90.8 |
| AutoVLA Zhou et al. (2025) | NeurIPS'25 | 3×C | 98.4 | 95.6 | 81.9 | 98.0 | 99.9 | 89.1 |
| DriveVLA-W0 Li et al. (2025a) | ICLR'26 | 1×C | 98.7 | **99.1** | 83.3 | 95.3 | 99.3 | 90.2 |
| SGDrive Li et al. (2026b) | CVPR'26 | 1×C | 98.6 | 97.8 | 85.8 | 96.2 | 100.0 | 91.1 |
| ExploreVLA Sheng et al. (2026) | ECCV'26 | 1×C | 98.8 | 98.4 | 83.5 | 96.5 | 99.9 | 90.4 |
| *World-model-based planners* | | | | | | | | |
| Epona Zhang et al. (2025) | ICCV'25 | 1×C | 97.9 | 95.1 | 80.4 | 93.8 | 99.9 | 86.2 |
| PWM Zhao et al. (2026a) | NeurIPS'25 | 1×C | 98.6 | 95.9 | 81.8 | 95.4 | 100.0 | 88.1 |
| DriveLaW Xia et al. (2026) | CVPR'26 | 1×C | 99.0 | 97.1 | 81.3 | 96.7 | 100.0 | 89.1 |
| DriveVA Liu et al. (2026) | ECCV'26 | 1×C | **99.2** | 97.5 | 83.5 | **98.7** | 100.0 | 90.9 |
| SparseWAM (**ours**) | – | 1×C | **99.2** | 98.5 | 86.0 | 97.2 | 100.0 | **92.1** |

### 4.3 Training Objective and Multi-modal Inference

**Training objective.** Given the condition $c = (Q_{t-n:t}, q_t, l_t)$, we supervise the velocity field predicted for each modality with its rectified-flow target and sum the two terms,

$$\mathcal{L} = \mathbb{E}\Big[\big\|v_\theta^S - (\epsilon_S - Q_f)\big\|_2^2\Big] + \lambda_A(\rho)\,\mathbb{E}\Big[\big\|v_\theta^A - (\epsilon_A - \tilde{a})\big\|_2^2\Big], \tag{5}$$

where $v_\theta^S$ and $v_\theta^A$ are the velocities predicted by the scene and action experts, $\epsilon_S$ and $\epsilon_A$ are independent Gaussian noises drawn for the two branches, and $\tilde{a}$ is the normalized trajectory. The action weight $\lambda_A$ is fixed to 1 for the first half of training and then increases linearly to 2 with the training progress $\rho \in [0, 1]$, so that the model first learns scene dynamics before shifting its emphasis towards trajectory accuracy.

**Multi-modal inference.** Since flow matching models a conditional distribution rather than its mean, SparseWAM obtains multi-modal plans by repeated sampling. At inference, $\mathcal{H}$ is encoded once, and $M$ independent noise samples are denoised in parallel, yielding $M$ candidate trajectories. The frozen pretraining scorer from Section 4.1 then selects the highest-scoring candidate (Li et al., 2025b; Sun et al., 2026). As it operates on the current-frame registers and the ego state, it adds no image-encoding cost. Owing to the lightweight denoiser, multi-modal inference only double the latency relative to single-mode inference while substantially improving planning performance (Section 5).

## 5 Experiment

### 5.1 Experimental Setup

**Datasets and Metrics.** NAVSIM (Dauner et al., 2024) is a planning-oriented benchmark built upon OpenScene, a 2 Hz redistribution of nuPlan (Caesar et al., 2021). We train on the navtrain split with 103,288 samples and evaluate on the navtest split with 12,146 samples. Performance is measured by the PDM Score (PDMS), which combines five closed-loop submetrics, namely No Collision (NC), Drivable Area Compliance (DAC), Ego Progress (EP), Time-to-Collision (TTC), and Comfort (C). NAVSIM v2 (Cao et al., 2026) adopts a two-stage pseudo-simulation. Both stages are scored with the Extended PDM Score (EPDMS). Compared with PDMS, EPDMS adds driving direction compliance (DDC) and traffic light compliance (TLC) as penalties, and lane keeping (LK), history comfort (HC), and extended comfort (EC) as weighted terms.

**Implementation details.** All experiments use an input resolution of $1148 \times 672$. The Register Perceiver uses DINOv2 ViT-S/14 (Oquab et al., 2024) with rank-32 LoRA (Hu et al., 2021)

Table 2: **Performance on the NAVSIM-v2 `navtest` benchmark.** * indicates training with reinforcement learning. The best learned result in each column is shown in **bold**.

| Method | Reference | NC↑ | DAC↑ | DDC↑ | TLC↑ | EP↑ | TTC↑ | LK↑ | HC↑ | EC↑ | EPDMS↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Human Agent | – | 100.0 | 100.0 | 99.8 | 100.0 | 87.4 | 100.0 | 100.0 | 98.1 | 90.1 | 90.3 |
| *Traditional E2E planners* | | | | | | | | | | | |
| TransFuser Prakash et al. (2021) | TPAMI'22 | 96.9 | 89.9 | 97.8 | 99.7 | 87.1 | 95.4 | 92.7 | 98.3 | 87.2 | 76.7 |
| DiffusionDrive Liao et al. (2025) | CVPR'25 | 98.2 | 95.9 | 99.4 | 99.8 | 87.5 | 97.3 | 96.8 | 98.3 | **87.7** | 84.5 |
| *VLM-based planners* | | | | | | | | | | | |
| ReCogDrive* Li et al. (2025c) | ICLR'26 | 98.3 | 95.2 | 99.5 | 99.8 | 87.1 | 97.5 | 96.6 | 98.3 | 86.5 | 83.6 |
| SGDrive Li et al. (2026b) | CVPR'26 | 98.6 | 94.3 | 99.5 | **99.9** | 86.0 | 97.9 | 96.1 | 98.3 | 85.9 | 86.2 |
| DriveFine* Dang et al. (2026) | arXiv'26 | 98.7 | 97.3 | 99.5 | 99.8 | 88.7 | 97.8 | **97.7** | **98.4** | 83.8 | 89.7 |
| *World-model-based planners* | | | | | | | | | | | |
| DriveVLA-W0 Li et al. (2025a) | ICLR'26 | 98.5 | **99.1** | 98.0 | 99.7 | 86.4 | 98.1 | 93.2 | 97.9 | 58.9 | 86.1 |
| DriveLaW Xia et al. (2026) | CVPR'26 | 98.7 | 96.9 | **99.6** | 99.8 | 87.5 | 98.3 | 97.6 | **98.4** | 77.4 | 88.6 |
| SparseWAM (**ours**) | – | **98.8** | 98.6 | 99.4 | 99.8 | **88.8** | **98.8** | 96.5 | 98.2 | 73.3 | **89.8** |

Table 3: **Performance on the NAVSIM-v2 `navhard` benchmark.** S1/S2 denote the per-stage results of the official two-stage protocol, and EPDMS is the overall score. The best learned result in each column is shown in **bold**.

| Method | Reference | Stage | NC↑ | DAC↑ | DDC↑ | TLC↑ | EP↑ | TTC↑ | LK↑ | HC↑ | EC↑ | EPDMS↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Traditional E2E planners* | | | | | | | | | | | | |
| TransFuser Prakash et al. (2021) | TPAMI'22 | S1 | 96.2 | 79.5 | 99.1 | 99.5 | 84.1 | 95.1 | 94.2 | 97.5 | 79.1 | 23.1 |
| | | S2 | 77.7 | 70.2 | 84.2 | 98.0 | 85.1 | 75.6 | 45.4 | 95.7 | **75.9** | |
| DiffusionDrive Liao et al. (2025) | CVPR'25 | S1 | 96.8 | 86.0 | 98.8 | 99.3 | 84.0 | 95.8 | 96.7 | 97.6 | **79.6** | 27.5 |
| | | S2 | 80.1 | 72.8 | 84.4 | 98.4 | 85.9 | 76.6 | 46.4 | 96.3 | 72.8 | |
| *VLM-based planners* | | | | | | | | | | | | |
| SGDrive Li et al. (2026b) | CVPR'26 | S1 | 95.8 | 87.6 | 97.8 | **99.8** | 84.4 | 94.7 | 92.9 | **97.8** | 28.9 | 25.5 |
| | | S2 | 79.4 | 65.4 | 79.1 | **98.9** | 88.9 | 75.3 | 42.7 | 96.4 | 29.6 | |
| ReCogDrive Li et al. (2025c) | ICLR'26 | S1 | 96.4 | 78.9 | 98.7 | **99.8** | 82.6 | 95.6 | 94.4 | 97.6 | 74.2 | 25.7 |
| | | S2 | 80.2 | 65.0 | 82.4 | 98.7 | 85.2 | 76.9 | 43.8 | 96.6 | 71.8 | |
| DriveFine Dang et al. (2026) | arXiv'26 | S1 | 97.6 | 90.0 | 99.1 | 99.3 | **84.9** | 96.7 | **97.3** | 97.6 | 72.0 | 30.5 |
| | | S2 | 82.1 | 71.3 | 84.8 | 98.4 | 88.1 | 74.3 | 47.2 | **96.8** | 72.8 | |
| *World-model-based planners* | | | | | | | | | | | | |
| DriveVLA-W0 Li et al. (2025a) | ICLR'26 | S1 | 96.8 | 83.3 | 99.0 | 99.6 | 84.6 | 95.3 | 96.4 | 97.6 | 78.2 | 24.4 |
| | | S2 | 76.8 | 64.3 | 79.9 | 98.3 | **89.2** | 75.0 | 46.8 | 95.8 | 53.1 | |
| DriveLaW Xia et al. (2026) | CVPR'26 | S1 | 97.3 | 89.1 | **99.2** | 99.6 | 84.3 | 97.1 | 96.2 | **97.8** | 67.6 | 30.6 |
| | | S2 | 82.5 | 67.6 | 83.5 | 98.1 | 84.8 | 78.5 | 45.8 | 96.4 | 57.3 | |
| SparseWAM (**ours**) | – | S1 | **98.8** | **93.6** | 99.0 | 99.3 | 82.9 | **98.4** | 96.0 | **97.8** | 59.1 | **40.1** |
| | | S2 | **85.5** | **81.1** | **87.2** | 98.4 | 83.9 | **83.6** | **48.1** | 96.7 | 60.3 | |

and is pretrained on navtrain plus 65K SimScale samples (Tian et al., 2026). The WAM predicts $\{Q_{t+2}, Q_{t+4}\}$ from $\{Q_{t-1}, Q_t\}$ (subscripts denote time in seconds), and $\mathcal{A}$ holds one token per waypoint for $T = 8$ waypoints over 4 s. Our MoT has 5 layers with $d_S = 512$ and $d_A = 256$, and is trained on navtrain for 50 epochs using AdamW (Loshchilov, 2017) with a learning rate of $10^{-4}$ and a cosine annealing schedule with linear warm-up. At inference, we use a 10-step Euler solver and draw $M = 64$ samples for multi-modal inference.

### 5.2 Main Results

As shown in Tab. 1, with only a single front camera, SparseWAM achieves SOTA PDMS of 92.1 on NAVSIM navtest. It outperforms all world-model-based planners, exceeding the strongest one, DriveVA, by 1.2 PDMS. Notably, SparseWAM improves EP by 2.5 (86.0 vs. 83.5) while matching DriveVA in NC, indicating that imagining the future allows the model to anticipate the behavior of surrounding agents and drive more efficiently without compromising safety. On the more comprehensive NAVSIM-v2 benchmarks (Tab. 2 and Tab. 3), SparseWAM achieves the SOTA EPDMS on both navtest and navhard, outperforming DriveLaW by 1.2 and 9.5 EPDMS, respectively. These results show that future imagination of sparse register space remains advantageous under more comprehensive evaluations that better approximate closed-loop driving.

### 5.3 Ablations Studies

**Ablation on key designs.** To validate the contribution of each design in SparseWAM, we incrementally add components on NAVSIM navtest, with results reported in Tab. 4. To rule out the

Table 4: Ablation of future prediction and MoT on NAVSIM `navtest`.

| Method | PDMS↑ | NC↑ | DAC↑ | EP↑ | TTC↑ | C↑ |
|---|---|---|---|---|---|---|
| Action-only | 87.9 | 98.3 | 96.1 | 82.1 | 94.8 | 100.0 |
| + Future prediction | 88.5 | 98.6 | 96.4 | 82.5 | 95.3 | 100.0 |
| + MoT (**ours**) | **89.1** | **98.7** | **96.7** | **83.1** | **95.6** | 100.0 |

Table 5: Model size and latency on NAVSIM `navtest`. † means single-mode inference.

| Method | Params (train)↓ | Params (test)↓ | PDMS↑ | Latency (ms)↓ |
|---|---|---|---|---|
| DriveVA | 5.72B | 5.72B | 90.9 | 829.0 |
| SimWAM (no RL) | 6.02B | 1.02B | 90.3 | 400.4 |
| SparseWAM † | **51.8M** | **51.8M** | 89.1 | **122.0** |
| SparseWAM | **51.8M** | 57.9M | **92.1** | 260.4 |

Table 6: Ablation on the attention mask between future imagination and action.

| Attention | PDMS↑ | NC↑ | DAC↑ | EP↑ | TTC↑ | Comfort↑ |
|---|---|---|---|---|---|---|
| Isolated | 88.7 | 98.4 | **96.7** | 82.8 | 95.2 | **100** |
| Bidirectional | 88.7 | 98.5 | 96.6 | 82.8 | 95.2 | **100** |
| Causal (**ours**) | **89.1** | **98.7** | **96.7** | **83.1** | **95.6** | **100** |

Table 7: Performance with different visual encoders on NAVSIM `navtest`.

| Method | PDMS↑ | NC↑ | DAC↑ | EP↑ | TTC↑ | C↑ |
|---|---|---|---|---|---|---|
| DINOv2 | 87.8 | 98.2 | 96.2 | 82.4 | 94.1 | 100.0 |
| V-JEPA 2.1 | 87.8 | 98.1 | 96.3 | 82.3 | 94.2 | 100.0 |
| SparseWAM (**ours**) | **89.1** | **98.7** | **96.7** | **83.1** | **95.6** | 100.0 |

influence of scorer selection, all variants are evaluated in the single-mode setting. The *action-only* baseline, a DiT that denoises only the trajectory conditioned on historical registers, reaches 87.9 PDMS. Next, Jointly denoising future-keyframe registers in the same DiT raises it to 88.5, with gains in NC and TTC, indicating that modeling the future scene helps anticipate surrounding agents, thereby reducing collisions and potential collision risks. Finally, we split the shared-parameter denoiser into a scene expert and an action expert following the MoT architecture, which further improves the PDMS to 89.1, indicating modality-specific parameters reduce cross-modal interference and better promote positive transfer from future prediction to planning.

**Model size and inference latency.** Tab. 5 compares model size and latency, measured on a single H20 GPU with batch size 1. Because WAMs built on video generation models denoise in a dense VAE latent space, they are large and slow: SimWAM still retains 1.02B parameters at deployment, and DriveVA requires 829.0 ms per inference. In contrast, SparseWAM operates on only 16 tokens per frame, requiring 51.8M parameters and 122 ms, which is $3.3\times$ and $6.8\times$ faster than SimWAM and DriveVA, respectively, while still imagining the future at test time. With multi-modal planning, SparseWAM further raises PDMS to 92.1 with only 6.1M extra parameters, at roughly one third of DriveVA's latency (260.4 ms). Although dense WAMs could in principle also perform multi-modal inference by drawing multiple samples, their already high per-sample cost would be further multiplied, making it impractical for deployment. In contrast, our compact register space keeps multi-sample inference cheap.

**Ablation on the causal mask.** As shown in Tab. 6, the *Isolated* mask adopted by SimWAM and Metis (Zhao et al., 2026b; Li et al., 2026a) blocks action tokens from attending to future registers, so future prediction serves only as a training-time auxiliary objective (88.7 PDMS). The *Bidirectional* mask lets actions read the imagined future, but also exposes it to noisy action tokens, which corrupts the imagination and yields no gain (88.7 PDMS). The *Causal* mask removes this interference while keeping the future accessible to actions, achieving the best PDMS of 89.1, confirming that the gain stems from test-time imagination.

### 5.4 Comparison and Visualization of Representations

**Representation ablation and visualization.** We replace the register space in our architecture with alternative representation spaces while keeping the MoT architecture unchanged (more details in appendix). As shown in Tab. 7, the WAMs built on DINOv2 and V-JEPA 2.1 both get PDMS of 87.8, yet fall below the single-sample performance of the register-based variant trained with the same recipe. This further demonstrates that our representation space is better suited for WAM prediction and yields stronger positive transfer to planning. We further visualize the register-to-image attention maps in the appendix, which show that the registers concentrate on planning-critical elements such as lead vehicles and road signs while largely ignoring the background, explaining why such a compact space suffices for future prediction and planning.

**Element sensitivity of representations.** To further evaluate how sensitive different representation spaces are to scene-level background versus planning-relevant objects, we adopt a counterfactual image editing protocol. Using rule-based filtering on 3D ground-truth boxes, we first mine 400 daytime *car-following* frames and 400 daytime *open-road* frames (no object ahead) from navtest. We then edit these base images with FLUX.2-klein-4B Labs (2025) under four categories, which are two planning-relevant edits $\mathcal{P}$ (removing the lead vehicle in car-following scenes and inserting a pedestrian in open-road scenes), and two background edits $\mathcal{B}$ (converting the whole image to nighttime and to rainy weather). Each category contains 400 pairs of original and edited images. Then, Each pair $(x, x')$ is fed into the frozen encoder under evaluation to obtain flattened representation vectors $z(x)$ and $z(x')$. Intuitively, a larger representation shift under a given edit type indicates that the representation is more sensitive to that type of element.

**Metric.** The sensitivity $S$ of each pair is defined as the relative $L_2$ displacement computed over the entire flattened vector:

$$S(x, x') = \frac{\|z(x') - z(x)\|_2}{\|z(x)\|_2 + \epsilon}. \tag{6}$$

Averaging $S$ over the pairs in $\mathcal{P}$ and $\mathcal{B}$ yields $S_{\text{plan}}$ and $S_{\text{bg}}$, and we define the foreground-to-background sensitivity ratio FBSR $= S_{\text{plan}}/S_{\text{bg}}$. A higher FBSR indicates that the representation is more sensitive to planning-relevant objects relative to appearance changes. As shown in Tab. 8, we find that the Sparse Register space is the only representation whose FBSR exceeds 1, demonstrating that it responds more strongly to planning-relevant object changes than to lighting and weather changes, whereas all other representations are dominated by appearance variations. We also find that DINO and V-JEPA achieve higher FBSR than the VAE encoder of Wan-2.2 owing to their richer semantic content.

Table 8: Token budget and counterfactual sensitivity of scene representations.

| Encoder | Tokens↓ | $S_{\text{plan}}\uparrow$ | $S_{\text{bg}}\downarrow$ | FBSR↑ |
|---|---|---|---|---|
| DINOv2 | 3936 | 0.297 | 0.692 | 0.43 |
| V-JEPA 2.1 | 3024 | 0.189 | 0.551 | 0.34 |
| Wan2.2 VAE | 756 | 0.206 | 1.061 | 0.19 |
| Sparse Register (**ours**) | **16** | **0.736** | **0.520** | **1.41** |

### 5.5 Qualitative Results

As shown in Fig. 3, we compare the planning results of SparseWAM and DriveVA Liu et al. (2026) in two representative scenarios. The first row shows a car-following scenario with a large truck ahead of the ego vehicle. SparseWAM attends to the motion of the leading vehicle and plans a decelerating trajectory, braking in time to avoid a rear-end collision. DriveVA plans too aggressively, and its trajectory extends into the leading vehicle, which results in a collision. The second row shows a left turn at an intersection. SparseWAM produces a smooth and accurate turning trajectory that stays within the drivable area. DriveVA's turning trajectory hits the curb and leaves the drivable area.




Figure 3: Qualitative comparison between DriveVA and SparseWAM.

## 6 Conclusion

We presented SparseWAM, a world action model that imagines the future in a decision-aligned sparse register space. Built by compressing a frozen DINOv2 encoder into 16 registers per frame and shaped by joint planning-and-scoring pretraining, this space is markedly more sensitive to planning-relevant objects relative to appearance changes. On top of it, a lightweight Mixture-of-Transformers with a causal attention mask jointly denoises future registers and trajectories, retaining explicit imagine-then-act reasoning at deployment. Our SparseWAM achieves SOTA PDMS of 92.1 on NAVSIM at 260 ms while remaining competitive on NAVSIM-v2, showing that decision-aligned sparsity is a possible practical path towards efficient WAMs. Extending it to multi-view inputs and closed-loop evaluation is left for future work.

### AI USE STATEMENT

In this work, we used generative AI tools for image editing, i.e., we applied the image editing model FLUX.2-klein-4B to real `navtest` frames to construct the counterfactual pairs used in the representation sensitivity analysis (Appendix A.1). We have not used generative AI tools for generating synthetic training datasets, theoretical models or conceptual frameworks, mathematical claims or proofs, hypotheses, research methodology or experiment design, method implementation, translation, dataset cleaning, or interpreting results. Additionally, we used generative AI tools to aid and polish the writing of the paper. We have reviewed all AI-assisted work. All edited images were manually inspected and filtered according to the criteria in Appendix A.1 and are used only for analysis, not for training, and all AI-polished text was checked and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

All experiments use publicly available datasets under their licenses, and no human subjects are involved. We note that SparseWAM is evaluated only in open-loop and pseudo-simulation benchmarks and has not been tested on a real vehicle; strong benchmark results do not guarantee safe real-world driving, and the method should not be deployed without thorough real-world validation.

### REPRODUCIBILITY STATEMENT

We describe the Register Perceiver pretraining, the MoT world–action model, and the training objective in Sec. 4, and provide the datasets, evaluation metrics, and implementation details, including architecture sizes, optimizer, learning-rate schedule, and inference settings, in Sec. 5.1. The counterfactual scene-editing protocol and the implementation of the dense-representation baselines are detailed in Appendix A.1 and Appendix A.3, respectively. All datasets used in this work are publicly available, and the code, configurations, and model checkpoints will be released upon acceptance.

## REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning, 2025. URL `https://arxiv.org/abs/2506.09985`.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving, 2026. URL `https://arxiv.org/abs/2506.04218`.

Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024a.

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. *arXiv preprint arXiv:2412.18607*, 2024b.

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 precise and robust driving, 2026. URL `https://arxiv.org/abs/2602.14577`.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International conference on learning representations*, volume 2024, pp. 2632–2652, 2024.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Xingtai Gui, Meijie Zhang, Tianyi Yan, Wencheng Han, Jiahao Gong, Feiyang Tan, Cheng zhong Xu, and Jianbing Shen. Bridging scene generation and planning: Driving with world model via unifying vision and motion representation, 2026. URL https://arxiv.org/abs/2603.14948.

Mingzhe Guo, Zhipeng Zhang, Yuan He, Ke Wang, and Liping Jing. End-to-end autonomous driving without costly modularization and 3d manual annotation. *arXiv preprint arXiv:2406.17680*, 2024.

Mingzhe Guo, Yixiang Yang, Chuanrong Han, Rufeng Zhang, Shirui Li, Ji Wan, and Zhipeng Zhang. Flowad: Ego-scene interactive modeling for autonomous driving. *arXiv preprint arXiv:2603.13399*, 2026.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*. Springer, 2022.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

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. *arXiv preprint arXiv:2303.12077*, 2023.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

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, 2026. URL https://arxiv.org/abs/2601.05083.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Jingyu Li, Zhe Liu, Dongnan Hu, Junjie Wu, Zipei Ma, Wenxiao Wu, Chao Han, Zhihui Hao, Zhikang Liu, Kun Zhan, et al. Metis: A generalizable and efficient world-action model for autonomous driving and urban navigation. *arXiv preprint arXiv:2606.15869*, 2026a.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving, 2026b. URL https://arxiv.org/abs/2601.05640.

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, 2025a. URL https://arxiv.org/abs/2510.12796.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27137–27146, 2025b.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving, 2025c. URL https://arxiv.org/abs/2506.08052.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12037–12047, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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 *European Conference on Computer Vision*, pp. 315–335. Springer, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

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

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15120–15130, 2024.

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

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. Sparsedrivev2: Scoring is all you need for end-to-end autonomous driving. *arXiv preprint arXiv:2603.29163*, 2026.

Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, Hangjun Ye, Tieniu Tan, Long Chen, and Hongyang Li. Simscale: Learning to drive via real-world simulation at scale, 2026. URL https://arxiv.org/abs/2511.23369.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Chenxu Hu, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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, 2026a. URL https://arxiv.org/abs/2603.24581.

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, 2026b. URL https://arxiv.org/abs/2601.22032.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv preprint arXiv:2405.01533*, 2024.

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, 2026c. URL https://arxiv.org/abs/2608.20974.

Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, et al. Drivelaw: Unifying planning and video generation in a latent driving world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39701–39712, 2026.

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

Jiawei Xu, Zhizhou Zhong, Zhijian Shu, Mingkai Jia, Mingxiao Li, Jia-Wang Bian, Qian Zhang, Kaicheng Zhang, Jin Xie, Jian Yang, et al. Eponav2: Driving world model with comprehensive future reasoning. *arXiv preprint arXiv:2605.14696*, 2026.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

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

Yiwei Zhang, Xuesong Chen, Jin Gao, Hanshi Wang, Fudong Ge, Weiming Hu, Shaoshuai Shi, and Zhipeng Zhang. Onedrive: Unified multi-paradigm driving with vision-language-action models. *arXiv preprint arXiv:2604.17915*, 2026.

Zhida Zhao, Talas Fu, Yifan Wang, Lijun Wang, and Huchuan Lu. From forecasting to planning: Policy world model for collaborative state-action prediction. *Advances in Neural Information Processing Systems*, 38:134585–134611, 2026a.

Zongchuang Zhao, Xin Zhou, Tianyang Xu, Zhengyang Sun, Kaixuan Zhou, Honglin Li, Dingkang Liang, and Xiang Bai. Simwam: A simple world action model for end-to-end autonomous driving. *arXiv preprint arXiv:2608.07468*, 2026b.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13782–13790, 2026.

Zewei Zhou, Tianhui Cai, Seth Z. Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning, 2025. URL `https://arxiv.org/abs/2506.13757`.

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

# A Appendix

## A.1 Details of Counterfactual Scene Editing

**Base-frame mining.** All base frames are front-camera images from NAVSIM `navtest`. We first keep daytime frames and then apply rule-based filtering on the 3D ground-truth boxes in the ego frame. A frame is labeled *car-following* if a vehicle lies in the ego lane ($|y| < 1.5\,\mathrm{m}$) with a longitudinal distance of 5–30 m, and *open-road* if no annotated agent lies within $40\,\mathrm{m}$ ahead and 3 m laterally. We sample 400 frames of each type.

**Editing protocol.** Each base image is edited with FLUX.2-klein-4B Labs (2025) at its original resolution using the instructions in Tab. 9. Car-following frames receive the *remove-lead-vehicle* edit, open-road frames receive the *insert-pedestrian* edit, and the background edits (*night* and *rain*) are applied to car-following frames, so that each of the four categories contains 400 original–edited pairs. Examples are shown in Fig. 4.

**Human review.** Generative editing can fail silently, e.g., by leaving the target partly visible, altering unrelated agents or the road layout, or introducing visible artifacts. To ensure that each pair differs *only* in the intended factor, every edited image was manually inspected by two annotators against three criteria: (i) the target edit is fully realized; (ii) the road geometry, other agents, and camera viewpoint remain unchanged; and (iii) the result is photorealistic without noticeable artifacts. Pairs failing any criterion were discarded and regenerated with a new random seed, or replaced by another base frame, until each category contained 400 valid pairs. Overall, 1.82% of the initial edits were rejected.

Table 9: Editing instructions for the four counterfactual categories.

| Type | Category | Instruction |
|---|---|---|
| $\mathcal{P}$ | Remove lead vehicle | "Remove the vehicle directly in front of the camera and fill the region with the road surface, keeping everything else unchanged." |
| $\mathcal{P}$ | Insert pedestrian | "Add a pedestrian walking on the road in front of the camera, keeping everything else unchanged." |
| $\mathcal{B}$ | Night | "Convert the scene to nighttime with street lights on, keeping the layout and all objects unchanged." |
| $\mathcal{B}$ | Rain | "Convert the scene to heavy rain with a wet road surface, keeping the layout and all objects unchanged." |

**Feature extraction.** All encoders are frozen and receive the same resized input. We use the last-layer patch tokens of DINOv2 ViT-S/14 without the CLS and register tokens ($82 \times 48 = 3936$ tokens), the image-mode patch tokens of V-JEPA 2.1 ViT-B with the input width resized to $1152$ ($72 \times 42 = 3024$ tokens), the latent of the Wan2.2 VAE ($756$ tokens), and the $16$ output registers of our Register Perceiver. Each representation is flattened into a single vector before computing the sensitivity.

### A.2 Visualization of Register Attention

To see which image regions the registers absorb, we take the attention weights from the register tokens to all patch tokens in the last block of the DINOv2 backbone, average them over heads and registers, and reshape them to the $82 \times 48$ patch grid. The map is bilinearly upsampled to the input resolution and overlaid on the RGB image, with low-attention regions left transparent.

As shown in Fig. 5, the attention is sparse and concentrates on planning-critical elements rather than on salient but driving-irrelevant content. In the first row, it focuses almost exclusively on the distant traffic lights and road signs, while the large wall and vegetation receive little attention. In the second row, it spreads along the road ahead and the markings on the road surface. In the third row, it concentrates on the vehicles ahead in the ego and adjacent lanes, while ignoring the large billboard and the roadside sign despite their high visual saliency. In the fourth row, it highlights the lead vehicle, the cyclist on the left, and the traffic lights, while the sky and the building facades are largely ignored. These observations are consistent with the high FBSR of the register space in Tab. 8 and explain why only $16$ tokens per frame suffice for future prediction and planning.

### A.3 Implementation of Dense Representation Baselines

For the representation ablation in Tab. 7, we replace the learnable register queries of the Register Perceiver with a frozen off-the-shelf visual encoder followed by spatial pooling, and train the same MoT world–action model on top of it. The two variants differ only in the encoder; the WAM architecture and training recipe are shared.

**Scene tokenization.** Each front-camera frame at $1148 \times 672$ is encoded independently; history and future frames share the same encoder.

- *DINOv2*: the official ViT-S/14 (Oquab et al., 2024) with patch size $14$. We discard the CLS token and its $4$ register tokens and keep the last-layer patch features, yielding a dense map of $384 \times 48 \times 82$ ($C \times H \times W$).
- *V-JEPA 2.1*: the ViT-B encoder (Assran et al., 2025) in image mode, i.e., a single frame ($T = 1$) through the image patch embedding. The input width is bilinearly resized from $1148$ to $1152$ to be divisible by the patch size $16$, yielding a dense map of $768 \times 42 \times 72$.

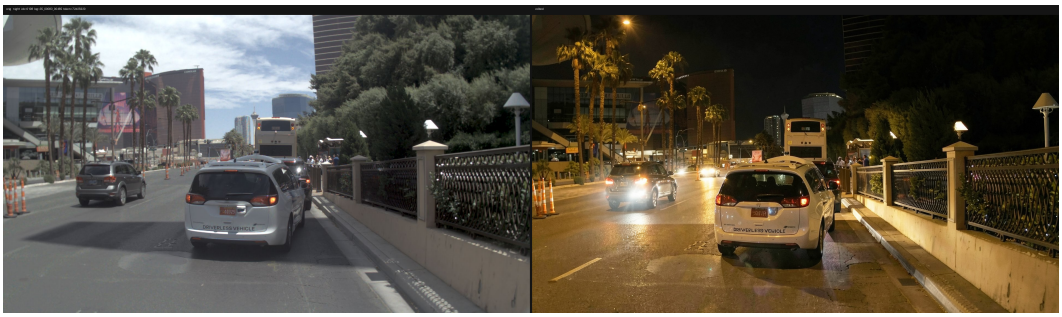
The dense map is compressed by `AdaptiveAvgPool2d` to a $10 \times 10$ grid, i.e., $100$ tokens per frame, and projected to $256$ dimensions by a linear neck. Both the encoder and the neck are kept frozen in evaluation mode throughout training, so that the WAM predicts a fixed target space, analogous to the frozen $\mathcal{E}_{\text{reg}}$ in SparseWAM.



(a) $\mathcal{P}$: remove lead vehicle



(b) $\mathcal{P}$: insert pedestrian



(c) $\mathcal{B}$: nighttime



(d) $\mathcal{B}$: rainy weather

Figure 4: Examples of counterfactual edits. In each pair, the left image is the original `navtest` frame and the right image is the edited one. Planning-relevant edits $\mathcal{P}$ change a decision-critical object while keeping the rest of the scene intact, whereas background edits $\mathcal{B}$ drastically alter the appearance while keeping all objects and the layout unchanged.

**Joint sequence and attention.** The joint sequence is $[\mathcal{H} \mid \mathcal{F} \mid \mathcal{A}]$. The history segment $\mathcal{H}$ contains the clean tokens of the frames at $-1$ s and $0$ s (200 tokens) with the flow time fixed to $0$; the future segment $\mathcal{F}$ contains the noisy tokens of the frames at $+2$ s and $+4$ s (200 tokens); and the action segment $\mathcal{A}$ contains $8$ noisy waypoints. The attention mask is asymmetric: $\mathcal{H}$ attends only to $\mathcal{H}$, $\mathcal{F}$




Figure 5: Register-to-image attention in the last DINOv2 block. Left: input front-camera frame. Right: attention averaged over registers and overlaid on the input frame.

attends to $\mathcal{H}$ and $\mathcal{F}$ but not to actions, and $\mathcal{A}$ attends to all tokens. Temporal positions are encoded with a shared 1D RoPE whose coordinate is the frame index at 2 Hz; the $k$-th waypoint is assigned index $k{+}1$, so the $+2$ s future frame is aligned with the fourth waypoint. Spatial positions within the $10 \times 10$ grid use fixed 2D sinusoidal embeddings.

**MoT and training.** The MoT consists of a scene expert of width 512 and an action expert of width 256, whose tokens are concatenated for joint attention with 12 heads of dimension 64. Both branches are trained with rectified flow: a shared uniform sample $u$ is drawn per example and mapped to the flow time of each branch by a separate SNR shift. The loss is the MSE between the predicted and target velocities on $\mathcal{F}$ and $\mathcal{A}$ only, where a Beta-distributed timestep weighting is applied to the action term. The action loss weight is fixed to 1 for the first half of training and increases linearly to 2 afterwards, as in SparseWAM. At inference, both segments start from pure Gaussian noise and

are integrated with a 10-step Euler solver, and only the trajectory is output. For a fair comparison, all variants in Tab. 7 are evaluated with single-sample inference without the scorer.