AFFORDDRIVE3D: AFFORDANCE-AWARE WORLD-ACTION MODELING WITH SPATIAL UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models have recently improved autonomous driving by jointly learning future scene prediction and trajectory generation. Most existing approaches model the future primarily through RGB appearance, and recent works have begun to incorporate geometric prediction to improve spatial understanding. However, dense geometry describes the spatial layout of the entire scene without indicating which parts are most relevant to the ego vehicle's action. For driving, the model must also identify and anticipate where it can safely move and which regions may pose collision risks. When action-relevant regions are jointly learned with future geometry, the policy can capture both what matters for driving and the corresponding geometric information. We therefore propose **AffordDrive3D**, an affordance- and geometry-aware world-action model that jointly learns future action-relevant regions and spatial structure. In order to capture the scene semantics and driving context needed for driving affordance prediction, we build AffordDrive3D on a VLM backbone to forecast drivable areas and collision-critical regions that directly affect ego motion, while predicting future geometry from RGB world-model latents. Experiments on NAVSIM show that AffordDrive3D achieves strong performance and demonstrate the effectiveness of jointly modeling future affordance and geometry.

1 INTRODUCTION

Autonomous driving has increasingly shifted toward end-to-end models that directly map sensor observations to future trajectories, allowing scene representations and driving policies to be jointly optimized for motion planning (Hu et al., 2023; Jiang et al., 2023; Liao et al., 2025; Li et al., 2025d;c; Yao et al., 2026b; Li et al., 2026c). More recently, advances in vision-language models (VLMs) have demonstrated strong semantic understanding and reasoning capabilities for complex driving scenes. Vision-language-action (VLA) models further incorporate action prediction into this framework, enabling a unified model to interpret the driving scene, reason about its context, and generate future ego actions (Zhou et al., 2025; 2026a; Renz et al., 2025; Fu et al., 2025).

However, driving decisions depend not only on understanding the current observation, but also on anticipating how the environment may evolve. Motivated by this need, world-action models (WAMs) further couple action learning with future-scene prediction, allowing a shared representation to jointly model how the environment evolves and how the ego vehicle should act (Zhang et al., 2026a; Li et al., 2025b; Hong et al., 2026). One line of work extends VLA models with explicit world modeling, introducing future prediction as additional supervision for action learning (Li et al., 2026b; Wang et al., 2026a; Liu et al., 2026). Another line builds on pretrained video foundation models, adapting either learned temporal representations or generative video priors for joint future prediction and action learning (Shi et al., 2026; Zhao et al., 2026b; Wang et al., 2026d). By supervising the model with future observations in addition to expert trajectories, WAMs encourage the policy to capture scene dynamics that are relevant to future decision making.

Most existing WAMs model the future primarily through RGB appearance or visual features (Wang et al., 2024; Zhao et al., 2026a; Chen et al., 2025; Li et al., 2025a). While such prediction provides rich supervision for scene evolution, appearance alone does not explicitly represent the spatial structure needed for driving. Compared with RGB appearance, geometry provides a more direct representation of scene structure, explicitly capturing relative distance and spatial layout that must

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

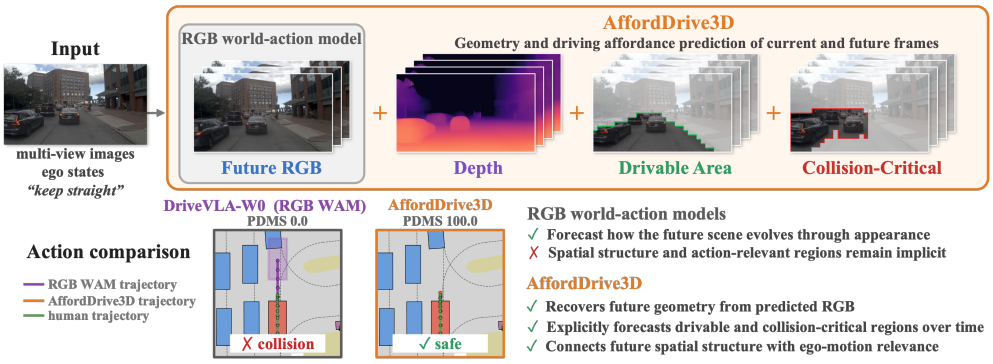


Figure 1: **Overview of AffordDrive3D.** Existing RGB-based world-action models primarily capture future scene evolution through appearance prediction. AffordDrive3D additionally predicts future geometry to model how spatial structure evolves, alongside driving affordances that identify spatial regions critical to the ego agent's actions. Together, these predictions align future world state modeling more closely with planning objectives and improve policy prediction.

otherwise be inferred from visual appearance. Recent approaches have therefore begun to incorporate geometric prediction into future modeling, such as depth (Sheng et al., 2026; Zhang et al., 2026b), point cloud (Lu et al., 2026) and 3D occupancy (Yang et al., 2025; Cheng et al., 2026). However, geometric future modeling remains relatively underexplored in VLA-based driving frameworks (Sheng et al., 2026).

Geometric future modeling provides explicit spatial structure that is missing from appearance-based prediction, yet a key challenge remains: how to use future spatial information to improve policy learning and action prediction. In autonomous driving, this requires identifying which regions of the future scene are relevant to ego motion, such as drivable areas and regions that may pose collision risks. Without capturing their relevance to action, future world state prediction remains insufficiently aligned with the planning objective.

Therefore, we propose **AffordDrive3D**, an affordance- and geometry-aware world-action model that jointly learns future action-relevant regions and spatial structure as shown in Fig. 1. For geometric modeling, we condition depth prediction on the future RGB latents produced by the world model. Because these latents already encode how the scene changes over time, the geometry branch does not need to independently predict future dynamics and can instead focus on recovering the corresponding 3D structure, while keeping the RGB and geometric predictions grounded in the same future scene. Beyond spatial structure, we further model which regions of the scene are important for driving. We refer to these action-relevant regions as ***driving affordances***, including drivable areas and collision-critical regions, and introduce temporal affordance prediction to model their evolution across the current and future scene. To provide the contextual scene understanding needed for affordance prediction, we adopt a VLM backbone and leverage its strong semantic understanding and reasoning capabilities. We jointly learn affordance prediction, future RGB and depth modeling, and trajectory generation through a shared representation. Compared with ExploreVLA (Sheng et al., 2026), which also incorporates future depth prediction, AffordDrive3D improves PDMS by 0.9 points and EPDMS by 1.1 points on the NAVSIM benchmark (Dauner et al., 2024), achieving new state-of-the-art (SOTA) scores of 91.3 PDMS and 89.9 EPDMS.

Our main contributions are:

- We propose **AffordDrive3D**, an affordance- and geometry-aware WAM built on a VLA framework, which extends future world modeling beyond visual appearance to better align it with the planning objective.
- We jointly model future geometry and temporal driving affordances: RGB-conditioned future depth captures the spatial structure of the predicted scene, while affordance forecasting identifies drivable areas and collision-critical regions, connecting future spatial information to ego motion.

- We achieve new SOTA performance with 91.3 PDMS and 89.9 EPDMS on the NAVSIM benchmark. Ablation studies further demonstrate the effectiveness of temporal affordance forecasting and future geometry modeling for trajectory planning.

2 Related Work

2.1 VLA and WAM for Autonomous Driving

VLMs have increasingly been adopted for autonomous driving to integrate perception, reasoning, and decision-making. DriveLM (Sima et al., 2024) introduces graph-structured visual question answering, while DriveVLM (Tian et al., 2024) integrates multimodal reasoning with planning. Recent VLA frameworks, including AutoVLA (Zhou et al., 2025), HybridDriveVLA (Bassole et al., 2026), and ColaVLA (Peng et al., 2026), further connect multimodal reasoning with trajectory generation through action tokenization or visual or latent reasoning. However, reasoning over observed scenes alone does not explicitly model how the environment evolves over time.

World models address this limitation by predicting future observations or structured scene states. Drive-WM (Wang et al., 2024) explores action-conditioned video generation, while Drive-OccWorld (Yang et al., 2025) predicts future occupancy and flow. More recently, WAMs jointly model future dynamics and actions: DriveWAM (Shi et al., 2026) adapts video generative priors to joint video-action generation, WA-JEPA (Wang et al., 2026d) learns latent dynamics with ego trajectories, and SimWAM (Zhao et al., 2026b) as well as SV-WAM (Wang et al., 2026b) use video prediction as training-time supervision. These approaches improve predictive action learning, but often rely on RGB, latent, or generic geometric dynamics without explicitly structuring future states around action-relevant spatial constraints.

2.2 Geometry-Aware Driving Policies

Explicit geometry provides spatial information complementary to images. Drive-OccWorld (Yang et al., 2025) models future occupancy and flow, GenieDrive (Yang et al., 2026) uses 4D occupancy to guide video generation, and SpaceDrive (Li et al., 2026a) incorporates depth-derived 3D information into VLM-based driving. ExploreVLA (Sheng et al., 2026) further predicts future RGB and depth as parallel supervision for trajectory planning, but treats them as parallel prediction objectives.

Recent work also explores geometric dynamics directly within world-action modeling. In particular, GeoWAM (Lu et al., 2026) directly forecasts future point-cloud geometry for action learning. In contrast, our model predicts future depth conditioned on RGB latents from the diffusion world model and jointly learns RGB, depth, affordance, and trajectory objectives, coupling anticipated appearance with geometric structure. Still, geometry alone does not explicitly indicate which regions are feasible or safety-critical for ego motion.

2.3 Affordance-Aware Planning

Affordances provide a task-relevant abstraction between perception and action by emphasizing regions and interactions that directly influence control. In robotics, VLAs use task-conditioned or structured affordances to bridge perception and action (Wang et al., 2026c; Yu et al., 2026). Affordance-aware learning remains less explored in autonomous driving. Recent methods incorporate structured future priors or drivable-area constraints for planning (Zeng et al., 2026; Wang et al., 2026b). Rather than relying on future visual reasoning, structured scene priors, or trajectory-level constraints, we explicitly forecast current and future drivable and collision-critical regions, which provides temporally structured supervision for how action-relevant space evolves over time.

3 Method

3.1 Overview

As shown in Fig. 2, given the current multi-view observations $\mathcal{I}_t = \{I_t^c\}_{c=1}^C$, a navigation command g, and the ego state s, AffordDrive3D predicts a future ego trajectory $\hat{\tau}_{t:t+K}$ while jointly modeling

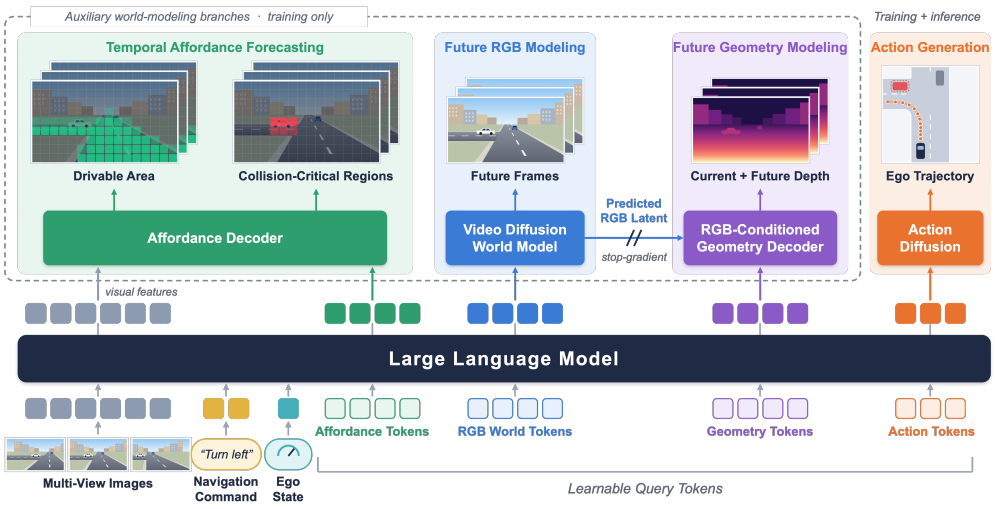


Figure 2: **Overview of AffordDrive3D.** Multi-view images, the navigation command, and the ego state are encoded by a shared VLM backbone together with learnable affordance, RGB world, geometry, and action tokens. The auxiliary branches (dashed box) forecast driving affordances, future RGB, and RGB-conditioned depth to shape the shared representation during training, while only the action branch is used at inference.

driving affordances, scene appearance, and geometry across the current and future scene. We use a shared causal VLM backbone to encode the inputs together with four groups of learned tokens: affordance tokens $\mathbf{Q}^{\text{aff}}$, RGB world tokens $\mathbf{Q}^{\text{rgb}}$, geometry tokens $\mathbf{Q}^{\text{geo}}$, and action tokens $\mathbf{Q}^{\text{act}}$. $\mathbf{Q}^{\text{aff}}$ model current and future drivable and collision-critical regions, $\mathbf{Q}^{\text{rgb}}$ condition the diffusion world model to forecast future visual appearance, and $\mathbf{Q}^{\text{geo}}$ predict current and future scene geometry with guidance from the predicted RGB latent. Finally, the action tokens $\mathbf{Q}^{\text{act}}$ condition an action diffusion model to generate the future ego trajectory.

All components are optimized jointly through the shared backbone, allowing affordance forecasting, visual prediction, and geometry modeling to shape the representation used for planning. During inference, only the action branch is required to generate the ego trajectory. The auxiliary affordance, RGB, and depth outputs do not need to be decoded.

3.2 RGB-Conditioned Future Geometry Modeling

Future RGB modeling. The RGB world tokens $\mathbf{Q}^{\text{rgb}}$ produce hidden states $\mathbf{H}^{\text{rgb}}$ after the shared causal backbone, which condition a video diffusion model to predict future scene appearance in latent space. We encode the current and future RGB frames using a video VAE, obtaining a sequence of T temporal latent groups $\mathbf{Z}^{\text{rgb}} = [\mathbf{Z}_t^{\text{rgb}}]_{t=0}^{T-1}$ after temporal compression, where $\mathbf{Z}_0^{\text{rgb}}$ corresponds to the current-frame anchor and $\{\mathbf{Z}_t^{\text{rgb}}\}_{t=1}^{T-1}$ represent the future sequence.

Following flow-matching, we perturb the full video latent, including the anchor group, as

$$\mathbf{Z}_\sigma^{\text{rgb}} = (1-\sigma)\mathbf{Z}^{\text{rgb}} + \sigma\boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\epsilon}$ is sampled noise and σ denotes the noise level. The video diffusion model is set up as an inpainting model, where the current visual observation is additionally encoded as a conditioning signal $\mathbf{C}_0$. The model predicts the velocity field as

$$\mathbf{v}_\theta = \mathcal{W}_{\text{rgb}}\left(\mathbf{Z}_\sigma^{\text{rgb}}, \sigma; \mathbf{H}^{\text{rgb}}, \mathbf{C}_0\right), \tag{2}$$

where $\mathbf{H}^{\text{rgb}}$ provides scene-level context, while $\mathbf{C}_0$ provides the observed appearance of the current frame for future generation. The RGB world model is trained with the flow-matching objective

$$\mathcal{L}_{\text{rgb}} = \left\|\mathbf{v}_\theta - \left(\boldsymbol{\epsilon} - \mathbf{Z}^{\text{rgb}}\right)\right\|_2^2. \tag{3}$$

From the predicted velocity, we recover the predicted clean RGB latent as

$$\hat{\mathbf{Z}}^{\text{rgb}} = \mathbf{Z}_\sigma^{\text{rgb}} - \sigma\mathbf{v}_\theta, \tag{4}$$

which is subsequently used as future-scene context for geometry prediction.

RGB-conditioned geometry prediction. The RGB world model already predicts how the visual scene evolves over future frames. We use the denoised RGB latent $\hat{\mathbf{Z}}^{\mathrm{rgb}}$ as future-scene context for geometry prediction (Liang et al., 2025), allowing the geometry branch to focus on recovering the 3D structure associated with the predicted visual evolution.

We first map the geometry-token hidden states $\mathbf{H}^{\mathrm{geo}}$ to a coarse temporal-spatial feature grid $\mathbf{G}^{\mathrm{geo}} \in \mathbb{R}^{T_g \times H_g \times W_g \times D}$, where T_g, H_g, and W_g denote the temporal, height, and width dimensions of the grid, respectively, with $T_g H_g W_g = |\mathbf{Q}^{\mathrm{geo}}|$. We then use trilinear interpolation to upsample the coarse geometry feature grid $\mathbf{G}^{\mathrm{geo}}$ to the same temporal-spatial resolution (T, H, W) as the predicted RGB latent $\hat{\mathbf{Z}}^{\mathrm{rgb}}$, yielding $\tilde{\mathbf{G}}^{\mathrm{geo}} \in \mathbb{R}^{T \times H \times W \times D}$.

To condition geometry prediction on the predicted future scene, we apply a 3D convolution ϕ_{rgb} to $\hat{\mathbf{Z}}^{\mathrm{rgb}}$ to map its feature channels to dimension D while preserving its temporal-spatial resolution. The resulting RGB features are fused with $\tilde{\mathbf{G}}^{\mathrm{geo}}$ at corresponding temporal-spatial locations:

$$\mathbf{F}^{\mathrm{geo}} = \mathcal{R}_{\mathrm{3D}}\left(\phi_{\mathrm{fuse}}\left(\left[\tilde{\mathbf{G}}^{\mathrm{geo}}; \phi_{\mathrm{rgb}}\left(\mathrm{sg}\left(\hat{\mathbf{Z}}^{\mathrm{rgb}}\right)\right)\right]\right)\right), \tag{5}$$

where ϕ_{fuse} denotes a $1 \times 1 \times 1$ 3D convolution that fuses RGB and geometry features, and $\mathcal{R}_{\mathrm{3D}}$ denotes a stack of 3D residual blocks for further feature refinement. The stop-gradient operator $\mathrm{sg}(\cdot)$ prevents the geometry loss from back-propagating through the RGB prediction branch.

The fused representation $\mathbf{F}^{\mathrm{geo}}$ retains the anchor–future temporal organization of the RGB latent space. Its anchor component $\mathbf{F}_0^{\mathrm{geo}}$ is formed with RGB context grounded in the observed current frame, whereas the future components $\mathbf{F}_{1:T-1}^{\mathrm{geo}}$ are formed with RGB context predicted by the RGB diffusion model. This yields two distinct geometry prediction settings: constructing the current geometry from observation-grounded visual context and predicting future geometry from predicted visual context. We therefore use separate decoders for the anchor and future depth latents:

$$\hat{\mathbf{Z}}_0^{\mathrm{dep}} = \mathcal{D}_{\mathrm{anc}}\left(\mathbf{F}_0^{\mathrm{geo}}\right), \qquad \hat{\mathbf{Z}}_{1:T-1}^{\mathrm{dep}} = \mathcal{D}_{\mathrm{fut}}\left(\mathbf{F}_{1:T-1}^{\mathrm{geo}}\right). \tag{6}$$

We encode the ground-truth depth sequence with the video VAE to obtain the target depth latent $\mathbf{Z}^{\mathrm{dep}}$, and supervise the predicted depth latent $\hat{\mathbf{Z}}^{\mathrm{dep}}$ with an $\mathcal{L}_1$ loss:

$$\mathcal{L}_{\mathrm{geo}} = \lambda_{\mathrm{anc}} \mathcal{L}_1\left(\hat{\mathbf{Z}}_0^{\mathrm{dep}}, \mathbf{Z}_0^{\mathrm{dep}}\right) + \mathcal{L}_1\left(\hat{\mathbf{Z}}_{1:T-1}^{\mathrm{dep}}, \mathbf{Z}_{1:T-1}^{\mathrm{dep}}\right), \tag{7}$$

where λ_{anc} controls the relative weight of the anchor-depth loss.

3.3 Temporal Affordance Forecasting

We index the affordance tokens by prediction horizon and camera view, such that $\mathbf{Q}^{\mathrm{aff}} = \mathbf{q}_{k,c}^{\mathrm{aff}}$, where $k \in \{0, \ldots, K\}$ denotes the future timestamp for prediction and $c \in \{1, \ldots, C\}$ denotes the camera view. After processing by the shared causal backbone, these tokens produce the corresponding hidden states $\mathbf{H}^{\mathrm{aff}} = \mathbf{h}_{k,c}^{\mathrm{aff}}$.

Dense affordance prediction. Each affordance state $\mathbf{h}_{k,c}^{\mathrm{aff}}$ is decoded into two spatial predictions: a drivable-area map $\hat{\mathbf{Y}}_{k,c}^{\mathrm{da}}$ and a collision-critical map $\hat{\mathbf{Y}}_{k,c}^{\mathrm{col}}$. Both maps are defined over the visual-token grid of camera c, so each map value corresponds to one local patch of the image.

Let $\mathbf{v}_{c,n}$ denote the hidden state of the n-th visual token from camera c. For each affordance type $m \in \{\mathrm{da}, \mathrm{col}\}$, we compute

$$z_{k,c,n}^{m} = \frac{\left\langle f^{m}(\mathbf{h}_{k,c}^{\mathrm{aff}}), g^{m}(\mathbf{v}_{c,n}) \right\rangle}{\sqrt{d}} + \mathrm{bias}^{m}, \tag{8}$$

where f^m and g^m are learnable projections that map the affordance and visual features into a d-dimensional embedding space. The predicted affordance probability at each visual-token location is then given by $\hat{Y}_{k,c,n}^{m} = \mathrm{sigmoid}(z_{k,c,n}^{m})$. The visual features provide current-scene context, while the frame-specific affordance states determine how the affordance predictions evolve over time.

Affordance supervision. We construct drivable-area and collision-critical targets that are directly relevant to ego planning, focusing on reachable road regions and objects that may collide with the ego vehicle. The drivable-area target $Y^{\mathrm{da}}_{k,c,n} \in [0,1]$ measures the fraction of verifiable ground within a visual-token region that belongs to the road surface reachable by the ego vehicle, with intermediate values capturing road boundaries. To construct the collision-critical target $Y^{\mathrm{col}}_{k,c,n} \in \{0,1\}$, we select plausible ego trajectories from a predefined trajectory set (Li et al., 2024) and roll them out in the scene. If any selected trajectory results in a collision with an agent or object, the visual-token region containing the collided agent or object is marked as collision-critical. We provide details of the label construction in the Appendix.

For drivable-area prediction, we use weighted soft binary cross-entropy,

$$\mathcal{L}^k_{\mathrm{da}} = \mathrm{WBCE}\left(\mathbf{z}^{\mathrm{da}}_k, \mathbf{Y}^{\mathrm{da}}_k; \mathbf{w}^{\mathrm{da}}_k\right), \tag{9}$$

where $\mathbf{w}^{\mathrm{da}}_k$ down-weights tokens with limited ground evidence, such as those dominated by sky or buildings, and gives higher weight to tokens that contain drivable-area boundaries. In this way, the loss focuses more on reliable ground regions while preserving the fine-grained boundary information encoded by the soft targets.

For collision-critical prediction, we combine class-normalized BCE to address the class imbalance between positive and negative tokens with Dice loss to encourage spatial overlap between the predicted and ground-truth critical regions,

$$\mathcal{L}^k_{\mathrm{col}} = \mathrm{BCE}\left(\mathbf{z}^{\mathrm{col}}_k, \mathbf{Y}^{\mathrm{col}}_k\right) + \lambda_{\mathrm{dice}}\mathrm{Dice}\left(\hat{\mathbf{Y}}^{\mathrm{col}}_k, \mathbf{Y}^{\mathrm{col}}_k\right). \tag{10}$$

Similar to geometry prediction, we separate the affordance of the observed current frame from that of the forecast future frames and aggregate the losses as

$$\mathcal{L}_{\mathrm{aff}} = \lambda_{\mathrm{anc}}\left(\mathcal{L}^0_{\mathrm{da}} + \mathcal{L}^0_{\mathrm{col}}\right) + \frac{1}{K}\sum_{k=1}^{K}\left(\mathcal{L}^k_{\mathrm{da}} + \mathcal{L}^k_{\mathrm{col}}\right). \tag{11}$$

3.4 Action Generation and Joint Learning

We use the hidden states of the action tokens $\mathbf{H}^{\mathrm{act}}$ to condition a DiT for future trajectory generation. The DiT is trained with a flow-matching objective $\mathcal{L}_{\mathrm{act}}$ to predict the velocity field between Gaussian noise and the ground-truth trajectory.

To encourage the RGB and geometry tokens to preserve action-relevant information while modeling future appearance and spatial structure, we additionally apply trajectory supervision to both the RGB and geometry tokens. Specifically, $\mathbf{H}^{\mathrm{rgb}}$ and $\mathbf{H}^{\mathrm{geo}}$ are separately processed by lightweight GRU-based trajectory heads to predict the future ego trajectory. We supervise these predictions with the ground-truth trajectory using L1 loss, denoted as $\mathcal{L}_{\text{rgb-act}}$ and $\mathcal{L}_{\text{geo-act}}$, respectively.

The complete framework is trained end-to-end with

$$\mathcal{L} = \lambda_{\mathrm{aff}}\mathcal{L}_{\mathrm{aff}} + \lambda_{\mathrm{rgb}}\mathcal{L}_{\mathrm{rgb}} + \lambda_{\mathrm{geo}}\mathcal{L}_{\mathrm{geo}} + \lambda_{\mathrm{act}}\mathcal{L}_{\mathrm{act}} + \lambda_{\text{rgb-act}}\mathcal{L}_{\text{rgb-act}} + \lambda_{\text{geo-act}}\mathcal{L}_{\text{geo-act}}, \tag{12}$$

where λ denotes the loss weights.

4 Experiments

4.1 Experimental Setup

Dataset and Metrics. We train and evaluate AffordDrive3D on the NAVSIM benchmark (Dauner et al., 2024), which is built on OpenScene and provides approximately 103K training samples in `navtrain` and 12K evaluation samples in `navtest`. We report both the PDM Score (PDMS) from NAVSIM v1 and the Extended PDM Score (EPDMS) from NAVSIM v2. NAVSIM v1 PDMS combines No-at-fault Collision (NC), Drivable Area Compliance (DAC), Ego Progress (EP), Time-to-Collision (TTC), and Comfort (C). NAVSIM v2 EPDMS further includes Driving Direction Compliance (DDC), Traffic Light Compliance (TLC), Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC). We follow the official NAVSIM evaluation protocol for both metrics.

Table 1: Comparison with baselines on the NAVSIM v1 benchmark. Best results are **bolded**.

Method	World Sup.	NC ↑	DAC ↑	TTC ↑	Comf. ↑	EP ↑	PDMS ↑
DRAMA	-	98.0	93.1	94.8	**100.0**	80.1	85.5
Hydra-MDP	-	98.3	96.0	94.6	**100.0**	78.7	86.5
DiffusionDrive	-	98.2	96.2	94.7	**100.0**	82.2	88.1
AutoVLA	-	98.4	95.6	98.0	99.9	81.9	89.1
ReCogDrive	-	97.9	97.3	94.9	**100.0**	87.3	90.8
WoTE	BEV	98.5	96.8	94.9	99.9	81.9	88.3
ResWorld	BEV	**98.9**	96.5	95.6	**100.0**	83.1	89.0
DrivingGPT	RGB	**98.9**	90.7	94.9	95.6	79.7	82.4
Epona	RGB	97.9	95.1	93.8	99.9	80.4	86.2
PWM	RGB	98.6	95.9	95.4	**100.0**	81.8	88.1
DriveVLA-W0	RGB	98.7	**99.1**	95.3	99.3	83.3	90.2
DriveDreamer-Policy	RGB + Current Depth	98.4	97.1	95.1	**100.0**	83.5	89.2
FSDrive	RGB + CoT	98.2	93.8	93.3	99.9	80.1	85.1
ExploreVLA	RGB + Depth	98.8	98.4	**96.5**	99.9	83.5	90.4
OWMDrive	3D Occupancy	98.2	97.6	94.6	99.8	**87.3**	90.8
Ours	Aff + RGB + Depth	98.6	99.0	96.3	**100.0**	84.7	**91.3**

For temporal affordance prediction, we additionally report average precision (AP). For drivable-area prediction, regions with a ground-truth drivable ratio greater than 0.5 are treated as drivable. Further details on the AP computation are provided in the Appendix.

Training and Implementation Details. We use Qwen3-VL-2B (Bai et al., 2025) as the VLM backbone and instantiate the RGB diffusion world model with Wan2.1-Fun-1.3B-InP (Wan et al., 2025). We use $C = 3$ camera views and set $K = 8$ at 2 Hz, corresponding to a 4 s prediction horizon and $T = 3$ latent groups after VAE encoding. We use $|\mathbf{Q}^{\text{aff}}| = C(K+1) = 27$ affordance tokens, $|\mathbf{Q}^{\text{rgb}}| = 64$ RGB world tokens, $|\mathbf{Q}^{\text{geo}}| = 64$ geometry tokens, and $|\mathbf{Q}^{\text{act}}| = 8$ action tokens. We set $\lambda_{\text{act}} = \lambda_{\text{rgb}} = \lambda_{\text{rgb-act}} = \lambda_{\text{geo-act}} = 1.0$, and $\lambda_{\text{geo}} = \lambda_{\text{aff}} = 0.1$, with $\lambda_{\text{anc}} = 1.0$ for both the geometry and affordance losses and $\lambda_{\text{dice}} = 1.0$. We obtain pseudo-ground-truth depth from Depth Anything 3 (Lin et al., 2025).

For supervised fine-tuning (SFT), we optimize the model for 14,140 steps with a global batch size of 256 using AdamW, a peak learning rate of 5×10^{-5}, 404 warmup steps, and cosine decay. Following SFT, we further optimize the action diffusion model through reinforcement learning (RL) with GRPO (Shao et al., 2024) for 4,500 steps using a constant learning rate of 3×10^{-6}, while keeping the rest of the model frozen. Detailed hyperparameters and training configurations for both SFT and RL, including the RL reward and rollout settings, are provided in the Appendix.

4.2 Main Results

NAVSIM v1. Results. We evaluate AffordDrive3D on the NAVSIM v1 benchmark against end-to-end, VLA and WAM baselines (Yuan et al., 2024; Li et al., 2024; Liao et al., 2025; Zhou et al., 2025; Xiong et al., 2026; Li et al., 2025b; Zhang et al., 2026a; Chen et al., 2025; Zhang et al., 2025; Zhao et al., 2026a; Li et al., 2026b; Zhou et al., 2026b; Zeng et al., 2026; Sheng et al., 2026; Cheng et al., 2026). In Table 1, AffordDrive3D achieves a PDMS of 91.3, outperforming all baselines in the comparison. In particular, it improves over ExploreVLA (Sheng et al., 2026), which also leverages RGB and depth supervision, by 0.9 PDMS. AffordDrive3D also achieves a strong DAC score of 99.0, together with NC and TTC scores of 98.6 and 96.3, while maintaining an EP score of 84.7. These results suggest that jointly modeling future driving affordances and geometric structure provides the action head with action-relevant spatial cues and their corresponding 3D context, enabling it to better reason about drivable regions and potential collision risks during trajectory planning.

NAVSIM v2. Results. We further compare AffordDrive3D with baseline methods (Zou et al., 2025; Yao et al., 2026a; Li et al., 2026b; Zhou et al., 2026b; Sheng et al., 2026) on NAVSIM v2, which extends the evaluation to a broader set of driving metrics. As shown in Table 2, AffordDrive3D achieves the highest EPDMS of 89.9, improving over ExploreVLA (Sheng et al., 2026) and DriveDreamer-Policy (Zhou et al., 2026b) by 1.1 and 1.2 points, respectively. These results fur-

Table 2: Comparison with baselines on NAVSIM v2 with extended metrics. Best results are **bolded**.

Model	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
DiffusionDriveV2	97.7	96.6	99.2	99.8	88.9	97.2	96.0	97.8	**91.0**	85.5
DriveSuprim	98.4	98.6	99.6	99.8	**90.5**	97.8	97.0	**98.3**	78.6	87.1
DriveVLA-W0	98.5	**99.1**	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1
DriveDreamer-Policy	98.4	97.1	99.5	99.9	87.9	97.7	97.6	**98.3**	79.4	88.7
ExploreVLA	**98.8**	96.2	99.6	99.8	87.1	**98.2**	**97.8**	**98.3**	86.8	88.8
Ours	98.4	98.2	**99.7**	**99.9**	87.9	97.8	97.4	**98.3**	80.4	**89.9**

Table 3: Ablation study of affordance, RGB, and geometry future modeling. For the + *Geometry* setting, the RGB branch provides latent context for depth prediction, but its gradients are stopped from propagating to the shared backbone. Results are reported without RL training.

Setting	Aff.	RGB	Geo.	NC ↑	DAC ↑	TTC ↑	Comf. ↑	EP ↑	PDMS ↑
Action only	–	–	–	97.9	96.4	94.3	**100.0**	82.6	88.1
+ Affordance	✓	–	–	98.3	97.1	95.1	**100.0**	83.5	89.2
+ RGB	–	✓	–	98.3	96.8	**95.3**	**100.0**	83.0	88.9
+ Geometry	–	–	✓	98.2	97.1	95.2	**100.0**	83.2	89.1
Full model	✓	✓	✓	**98.4**	**97.6**	95.2	**100.0**	**83.7**	**89.6**

ther demonstrate the strong planning performance of AffordDrive3D and the effectiveness of jointly modeling future driving affordances and geometric structure.

Qualitative Results. Figure 3 compares AffordDrive3D with an action-only baseline and the RGB-based DriveVLA-W0 (Li et al., 2026b). In the narrow curvy-road example in Fig. 3(a), both baselines go off-road, whereas AffordDrive3D remains near the lane center as its drivable-area maps capture the road structure over time. In the leading-vehicle braking example in Fig. 3(b), both baselines collide, while AffordDrive3D maintains a safe distance. Its collision-critical prediction highlights the leading vehicle across future horizons, and its RGB-consistent depth captures the vehicle's relative distance and surrounding geometry. These examples illustrate the complementary action-relevant cues provided by affordance and geometry forecasting.

4.3 Ablation Studies

Ablation of Affordance and World Modeling. We perform an ablation study to understand how affordance, RGB, and geometry modeling affect planning performance in Table 3. Since our depth prediction is conditioned on the denoised RGB latents, the geometry-only setting still retains RGB prediction to provide the required latent input. However, we stop the RGB diffusion gradients from propagating back to the shared backbone, so the RGB branch only provides the latent context required by the geometry branch and does not directly affect the shared representation used for action prediction. Compared to the action-only baseline at 88.1 PDMS, adding affordance, RGB, or geometry improves performance to 89.2, 88.9, and 89.1 PDMS, respectively. The full model achieves the best performance of 89.6 PDMS, with the highest NC, DAC, and EP, demonstrating the effectiveness of jointly modeling RGB, geometry and affordance.

Temporal Affordance Evaluation. We evaluate the predicted driving affordances across different future horizons in Table 4. AffordDrive3D achieves strong current-frame prediction, with an AP of 0.918 for collision-critical regions and 0.986 for drivable areas. Drivable-area prediction remains highly stable over time, reaching 0.983 AP over the near-term horizon and 0.971 over the longer-term horizon. Compared with drivable-area prediction, collision prediction degrades more noticeably at longer horizons, with AP decreasing from 0.846 to 0.732. This is expected because drivable areas are largely determined by relatively stable road structure, whereas future collision-critical regions depend on anticipating the motion of dynamic agents such as vehicles and pedestrians. Overall, the results show that AffordDrive3D can reliably forecast both where the ego vehicle can safely move and where potential collision risks may emerge over time.

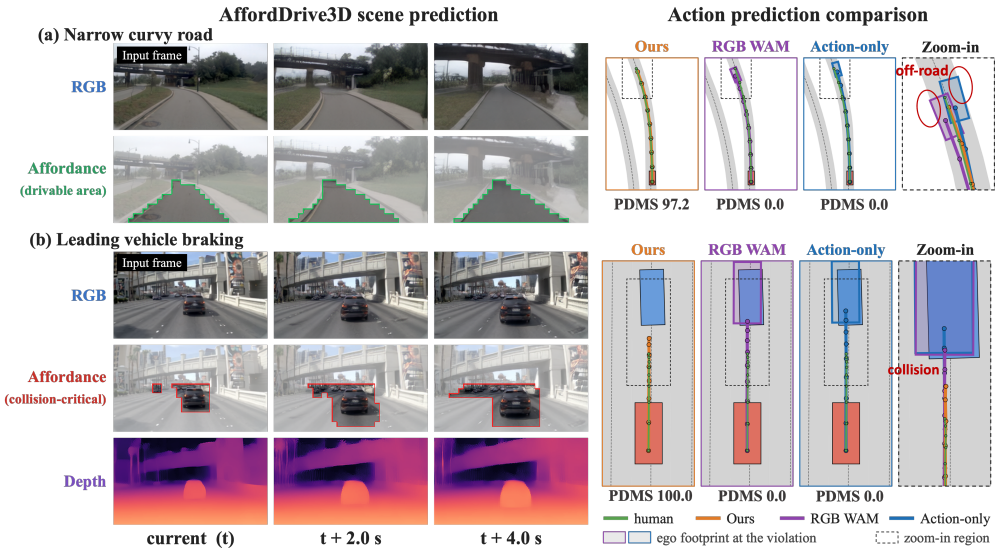


Figure 3: **Qualitative results of AffordDrive3D in two challenging scenarios.** (a) The narrow curvy-road scenario: AffordDrive3D forecasts the road structure and remains on-road, while both baselines deviate from the lane. (b) The leading-vehicle braking scenario: AffordDrive3D captures the potential collision risk from the leading vehicle together with RGB-consistent future geometry, maintaining a safe trajectory while both baselines collide.

Table 4: Affordance prediction across horizons.

t	$AP_{Collision}$	$AP_{Drivable\ Area}$
Current	0.918	0.986
0–2 s (avg.)	0.846	0.983
2–4 s (avg.)	0.732	0.971

Table 5: Current-frame affordance prediction using frozen and unfrozen VLM.

	$AP_{Collision}$	$AP_{Drivable\ Area}$
Frozen VLM	0.782	0.931
Unfrozen VLM	**0.912**	**0.985**

Affordance Information in the VLM Representation. We adopt a VLM backbone to provide the contextual scene understanding needed for affordance prediction. To assess whether the original VLM representation already contains information useful for identifying regions relevant to ego motion, we remove all world-modeling and action components and attach two lightweight MLP heads directly to the VLM visual tokens to predict current-frame drivable-area and collision-critical maps. We compare a frozen setting, where only the prediction heads are trained, with an unfrozen setting, where the VLM backbone is jointly optimized with the affordance objectives. As shown in Table 5, the frozen setting achieves APs of 0.782 and 0.931 for collision-critical and drivable-area prediction, respectively. In the unfrozen setting, the APs increase to 0.912 and 0.985, with a particularly large gain for collision-critical prediction. These results indicate that the original VLM representation already contains useful information for identifying driving-relevant regions, while explicit affordance supervision further enriches the backbone with affordance-relevant understanding that can be used to support trajectory planning.

5 Conclusion

We presented **AffordDrive3D**, an affordance- and geometry-aware world-action model for autonomous driving. Beyond RGB-based future prediction, AffordDrive3D models future geometry to provide explicit spatial structure and forecasts temporal driving affordances, including drivable and collision-critical regions, to support trajectory planning. By jointly learning future appearance, geometry, affordance, and action within a shared VLM backbone, AffordDrive3D more closely aligns future world modeling with the planning objective. AffordDrive3D achieves 91.3 PDMS and 89.9 EPDMS on NAVSIM, while ablation and qualitative results validate the benefits of geometry modeling and temporal affordance forecasting.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of the manuscript, including checking grammar, clarity, and readability. We reviewed all AI-suggested edits and take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies autonomous-driving planning using publicly available benchmark data and does not involve human-subject experiments or collection of personal data. As autonomous driving is a safety-critical application, the proposed model should not be considered deployment-ready based solely on benchmark performance.

REPRODUCIBILITY STATEMENT

We provide the implementation, training, and evaluation details necessary to reproduce our results in the main paper and appendix. All experiments are conducted on publicly available datasets and benchmarks. We plan to release the code upon acceptance.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Yipene Cedric Francois Bassole, Sungwoo Kim, Jiwoo Jung, and Yunsick Sung. Hybriddrivevla: Vision-language-action model with visual cot reasoning and tot evaluation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32421–32430, 2026.

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26890–26900. IEEE, 2025.

Junjie Cheng, Ruiqi Song, Ye Wu, Nanxing Zeng, Ximiao Li, and Yunfeng Ai. Owmdrive: Causality-aware end-to-end autonomous driving via 4d occupancy world model. *arXiv preprint arXiv:2606.30421*, 2026.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24823–24834. IEEE, 2025.

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

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8306–8316. IEEE, 2023.

Peizheng Li, Zhenghao Zhang, David Holtz, Hang Yu, Yutong Yang, Yuzhi Lai, Rui Song, Andreas Geiger, and Andreas Zell. Spacedrive: Infusing spatial awareness into vlm-based autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40096–40107, 2026a.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *International Conference on Learning Representations*, volume 2025, pp. 42942–42959, 2025a.

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

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 7890–7911, 2026b.

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.

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Jingde Chen, Nadine Chang, Maying Shen, Jingyu Song, Zuxuan Wu, Shiyi Lan, et al. Ztrs: Zero-imitation end-to-end autonomous driving with trajectory scoring. *arXiv preprint arXiv:2510.24108*, 2025c.

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

Zhenxin Li, Nadine Chang, Xinglong Sun, Jingde Chen, Wenhao Yao, Zi Wang, Maying Shen, Yu-Gang Jiang, Zuxuan Wu, Shiyi Lan, et al. Large discrete policy: Advancing explicit behavior modeling with stochastic iterative scoring. *arXiv preprint arXiv:2609.07049*, 2026c.

Dingkang Liang, Dingyuan Zhang, Xin Zhou, Sifan Tu, Tianrui Feng, Xiaofan Li, Yumeng Zhang, Mingyang Du, Xiao Tan, and Xiang Bai. Unifuture: A 4d driving world model for future generation and perception. *arXiv preprint arXiv:2503.13587*, 2025.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047. IEEE, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, Long Chen, et al. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving. *arXiv preprint arXiv:2602.06521*, 2026.

Yiren Lu, Xin Ye, Jiaming Liu, Philip Jacobson, Jin Yao, Yi-chung Chen, Liam Merino, Dhruva Dixith Kurra, Min Cai, Tom Lampo, et al. Geowam: Visual geometry world action models for autonomous driving. *arXiv preprint arXiv:2608.23486*, 2026.

Qihang Peng, Xuesong Chen, Chenye Yang, Shaoshuai Shi, and Hongsheng Li. Colavla: Leveraging cognitive latent reasoning for hierarchical parallel trajectory planning in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17809–17819, 2026.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11993–12003. IEEE, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 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.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pp. 256–274. Springer, 2024.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, XianPeng Lang, and Hang Zhao. DriveVLM: The convergence of autonomous driving and large vision-language models. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=928V4Umlys.

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.

Guoqing Wang, Pin Tang, Xiangxuan Ren, Guodongfang Zhao, Bailan Feng, and Chao Ma. Learning vision-language-action world models for autonomous driving. *arXiv preprint arXiv:2604.09059*, 2026a.

Jinyang Wang, Shiwei Li, Junjian Wang, Zhiqiang Deng, Jianbin Gao, Yihang Zhao, Liu Liu, Yongjia Zhao, Jinlong Chen, Huirui Xu, et al. Sv-wam: An efficient surround-view world-action model for end-to-end autonomous driving. *arXiv preprint arXiv:2609.03602*, 2026b.

Runze Wang, Yuqian Fu, Yu Li, Tao Lin, Tianwen Qian, Mohamed Elhoseiny, Bo Zhao, Yanwei Fu, Yu-Gang Jiang, and Xiangyang Xue. Afford-vla: Action-aligned visual planning via internalized affordance. *arXiv preprint arXiv:2605.24203*, 2026c.

Xinlin Wang, Yujiao Xiang, Yuheng Zhou, Jingqi Wang, Minqing Huang, Jiajie Huang, Dongxu Wei, Tingguang Zhou, Xiyang Wang, Gong Chen, et al. Wa-jepa: Rethinking the video jepa paradigm for world-action modeling in autonomous driving. *arXiv preprint arXiv:2608.20974*, 2026d.

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

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026.

Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9327–9335, 2025.

Zhenya Yang, Zhe Liu, Yuxiang Lu, Liping Hou, Chenxuan Miao, Siyi Peng, Bailan Feng, Xiang Bai, and Hengshuang Zhao. Geniedrive: Towards physics-aware driving world model with 4d occupancy guided video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35680–35690, 2026.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M Alvarez, and Zuxuan Wu. Drivesuprim: Towards precise trajectory selection for end-to-end planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11910–11918, 2026a.

Wenhao Yao, Xinglong Sun, Zhenxin Li, Shiyi Lan, Zi Wang, Jose M Alvarez, and Zuxuan Wu. Had: Combining hierarchical diffusion with metric-decoupled rl for end-to-end driving. *arXiv preprint arXiv:2604.03581*, 2026b.

Qize Yu, Jiadi You, Yuran Wang, Jiaqi Liang, Bowen Ping, Yang Tian, Yue Chen, Minghong Cai, Zeying Gong, Ruihai Wu, et al. Affordancevla: A vision-language-action model empowering action generation through affordance-aware understanding. *arXiv preprint arXiv:2606.06155*, 2026.

Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024.

Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, and Xing Wei. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *Advances in Neural Information Processing Systems*, 38:67299–67318, 2026.

Jinqing Zhang, Zehua Fu, Zelin Xu, Wenying Dai, Qingjie Liu, and Yunhong Wang. Resworld: Temporal residual world model for end-to-end autonomous driving. *arXiv preprint arXiv:2602.10884*, 2026a.

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.

Songyan Zhang, Jinyuan Tian, Hanbing Li, Daqi Liu, Hao Chen, Wenhui Huang, Fang Li, Guang Chen, Hangjun Ye, Long Chen, et al. Geoworldad: Geometry world action model for autonomous driving. *arXiv preprint arXiv:2607.17521*, 2026b.

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, 2026a.

Yang Zhou, Xiaofeng Wang, Hao Shao, Letian Wang, Guosheng Zhao, Jiangnan Shao, Jiagang Zhu, Tingdong Yu, Zheng Zhu, Guan Huang, et al. Drivedreamer-policy: A geometry-grounded world-action model for unified generation and planning. *arXiv preprint arXiv:2604.01765*, 2026b.

Zewei Zhou, Tianhui Cai, Yun Zhao, Seth Z.and 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. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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 Affordance Label Construction

Both targets are defined on the visual-token grid of the VLM: each of the $C = 3$ cameras is resized to 1024×576, and each merged Qwen3-VL token covers a 32×32 pixel patch $\mathcal{P}_{c,n}$, giving an 18×32 grid per camera. Targets are generated at $t_k = 0.5k$ s for $k = 0, \ldots, 8$. As for the future geometry, the target of frame k is defined in the logged frame at t_k, using that frame's images, camera calibration, ground-truth ego pose, and box annotations. Figure 4 shows an example.

Drivable area. We derive the region $\mathcal{R}_k$ reachable by the ego from the NAVSIM map. Starting from the roadblocks that contain the ego pose at t_k, we follow the directed outgoing edges of the roadblock graph for up to six hops and take the union of the roadblock polygons within 80 m ahead of the ego. $\mathcal{R}_k$ thus covers all lanes of the ego carriageway and every lawful branch at intersections, excludes the oncoming carriageway and sidewalks, and is independent of the navigation command. A pixel is *verifiable ground* if its camera ray hits the ground plane within 80 m and its monocular depth from Depth Anything 3 (Lin et al., 2025) is at least 0.7 times the distance to that plane along the ray. Pixels failing the depth test observe a vertical surface (buildings, vegetation) rather than the ground. The ground-plane height is estimated per frame from the same depth, and ground occluded by annotated vehicles is kept. With $\mathcal{G}_{k,c,n} \subseteq \mathcal{P}_{c,n}$ the verifiable ground pixels of a token and $\pi_k(p)$ the ground point of pixel p, the target and the evidence share are

$$Y^{\mathrm{da}}_{k,c,n} = \frac{|\{p \in \mathcal{G}_{k,c,n} : \pi_k(p) \in \mathcal{R}_k\}|}{|\mathcal{G}_{k,c,n}|}, \qquad s_{k,c,n} = \frac{|\mathcal{G}_{k,c,n}|}{|\mathcal{P}_{c,n}|}. \tag{13}$$

Y^{da} is 1 on the reachable road, 0 on non-drivable ground, and fractional at road boundaries. Tokens with $s_{k,c,n} < 0.15$, such as sky or buildings, are treated as unknown and supervised as 0 with a low weight. The WBCE weight in the main text is

$$w^{\mathrm{da}}_{k,c,n} = \mathbf{1}[\mathcal{R}_k \neq \emptyset] \cdot \mathrm{clip}(s_{k,c,n}/0.3,\, 0.1,\, 1) \cdot \left(1 + 2 \cdot \mathbf{1}\left[0 < Y^{\mathrm{da}}_{k,c,n} < 1\right]\right), \tag{14}$$

which ignores frames without mapped road, down-weights tokens with little ground evidence, and up-weights boundary tokens.

Collision-critical regions. We start from a vocabulary of 16,384 ego trajectories (4 s at 10 Hz) following Hydra-MDP (Li et al., 2024) and keep the plausible ones, i.e., those that (i) stay within the route lanes, dilated by 0.25 m, at all 40 poses, (ii) keep their heading within 60° of the nearest route-lane direction, and (iii) comply with the driving direction (NAVSIM DDC $= 1$). Each plausible trajectory is rolled out with the NAVSIM v1 simulator, which tracks it with an LQR controller and a kinematic bicycle model while other agents replay their logged motion (Dauner et al., 2024). Following the NAVSIM v1 scoring rules, an object is *causal* if at least one plausible trajectory has an at-fault collision with it (a frontal collision, a collision with a stationary object, or a lateral collision while the ego is off its lane). The causal set $\mathcal{O}$ is determined once from the rollouts at t_0 and shared across all timestamps. For frame k, the annotated 3D box of each causal object $o \in \mathcal{O}$ at t_k is clipped at the near plane and projected into each camera of that frame, and every token overlapping a projection is marked:

$$Y^{\mathrm{col}}_{k,c,n} = \mathbf{1}\left[\exists\, o \in \mathcal{O} :\ \mathrm{Proj}_{k,c}(o) \cap \mathcal{P}_{c,n} \neq \emptyset\right]. \tag{15}$$

An object that is not annotated at t_k contributes no positive tokens at that timestamp. If the rollouts at t_0 produce collisions but no causal object is visible in any camera at t_0, the current timestamp and every future timestamp without a visible causal object are treated as unknown and excluded from both the loss and AP evaluation.

A.2 Training Details

Supervised fine-tuning. We train for 14,140 steps (35 epochs) with a per-device batch size of 2 on 128 Nvidia H100 GPUs, giving a global batch size of 256. We use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 10^{-3}) with gradient clipping at 1.0, DeepSpeed ZeRO-2, and bf16 mixed precision. A single peak learning rate of 5×10^{-5} is shared by all trainable parameter groups, with 404 warmup steps and cosine decay to 2.5×10^{-7}. The VAE, the CLIP encoders, and the VLM vision encoder remain frozen.

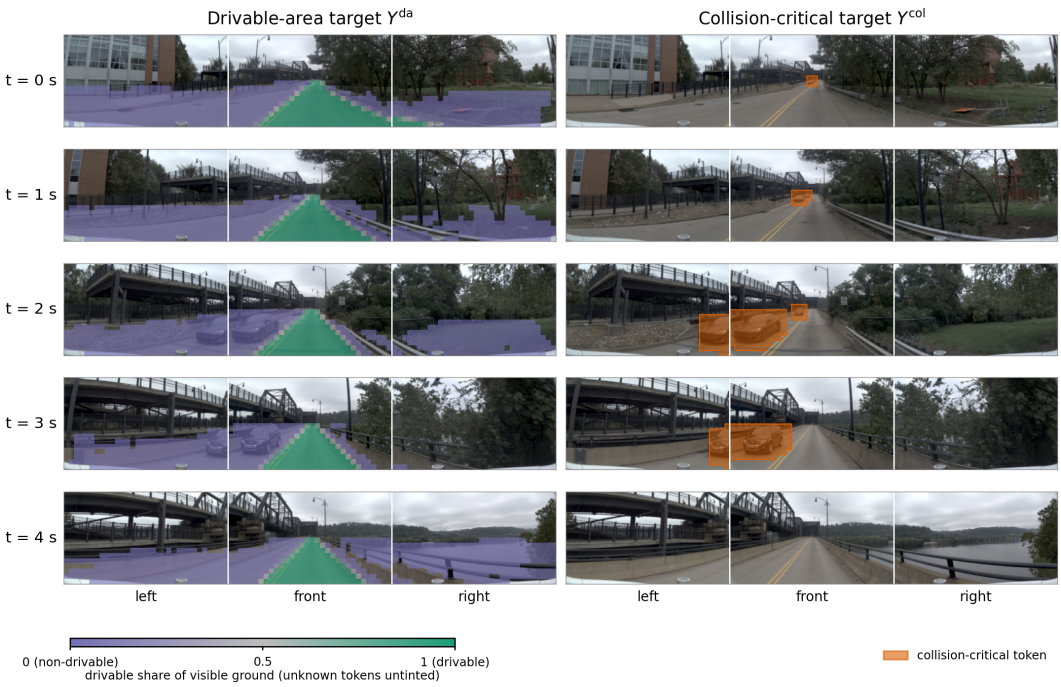


Figure 4: Example of ground-truth affordance targets at $t = 0$–4 s.

Reinforcement learning setup. After SFT we optimize only the action DiT with GRPO (Shao et al., 2024) and keep the VLM, the world model, the query banks, and all auxiliary heads frozen. We train RL on 8 Nvidia H100 GPUs.

We optimize the benchmark metric itself, evaluated per sampled trajectory with the official NAVSIM simulator on `navtrain`. For NAVSIM v1 the reward is the PDMS. For NAVSIM v2, the Extended Comfort (EC) term compares two consecutive frames and is therefore undefined for a trajectory in isolation, so we use

$$r = \mathrm{NC} \cdot \mathrm{DAC} \cdot \mathrm{DDC} \cdot \mathrm{TLC} \cdot \frac{5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{LK} + 2\,\mathrm{HC}}{14} - \lambda_{\mathrm{ec}}\,\mathbf{1}[\mathrm{EC\ violated}]\,, \qquad (16)$$

with $\lambda_{\mathrm{ec}} = 0.2$, where the EC flag is evaluated against the deterministic trajectory that the current policy produces for the preceding frame (82.5% of `navtrain` scenes have such a frame).

At each step we draw 128 scenes and sample $G = 16$ trajectories per scene, giving 2,048 scored trajectories per step. Trajectories are sampled with a drift-corrected SDE variant of the flow-matching sampler whose noise scale $\sigma_k = \eta\sqrt{1 - t_k}$ preserves the marginals of the deterministic sampler. η is adapted online (initialized at 0.3, clipped to $[0.1, 0.6]$) so that roughly half of the scene groups remain informative. Advantages are mean-centered within each scene and clipped to ± 5. We do not divide by the group standard deviation, and we skip groups whose reward range falls below 0.02. The objective is the clipped policy-gradient loss ($\epsilon = 0.2$) plus a KL penalty to the frozen SFT policy with coefficient $\beta = 0.01$. We use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, no weight decay), gradient clipping at 1.0, and a constant learning rate of 3×10^{-6} after 50 warmup steps. We train for 4,500 steps with either reward on 8 Nvidia H100 GPUs.

A.3 Affordance AP Computation

AP is computed separately for each head $m \in \{\mathrm{col}, \mathrm{da}\}$ and each timestamp t_k (t_0 is the current frame) on `navtest`. All token predictions at timestamp t_k, over all scenes and the $C = 3$ cameras, are pooled into one set $\mathcal{C}_k^m$. Each element i has the predicted probability $\hat{Y}_i$ (no threshold is applied) and a binary label y_i. For collision, $\mathcal{C}_k^{\mathrm{col}}$ contains all tokens of the scenes whose label at t_k is known, and $y_i = Y_i^{\mathrm{col}}$. For the drivable area, $\mathcal{C}_k^{\mathrm{da}}$ contains all tokens of every scene, without any ground-evidence filter, and $y_i = \mathbf{1}[Y_i^{\mathrm{da}} > 0.5]$. Unknown tokens ($s < 0.15$) keep their training target

$Y^{\mathrm{da}} = 0$. Let $\tau_1 > \cdots > \tau_M$ be the distinct predicted values in $\mathcal{C}_k^m$. Then AP is computed by

$$\mathrm{AP}_k^m = \sum_{j=1}^{M} (R_j - R_{j-1})\, P_j, \tag{17}$$

where

$$P_j = \frac{\sum_i y_i \, \mathbf{1}[\hat{Y}_i \geq \tau_j]}{\sum_i \mathbf{1}[\hat{Y}_i \geq \tau_j]}, \qquad R_j = \frac{\sum_i y_i \, \mathbf{1}[\hat{Y}_i \geq \tau_j]}{\sum_i y_i}, \quad R_0 = 0. \tag{18}$$