GeoWAM: Visual Geometry World Action Models for Autonomous Driving

Anonymous authors
Paper under double-blind review

Abstract

World action models (WAMs) have recently gained increasing attention as a framework for jointly modeling scene evolution and ego actions in autonomous driving. Most existing WAMs learn scene dynamics in pixel space by combining a video-generation backbone for future-observation prediction with an action head for ego-trajectory prediction. Pixels, however, provide only an indirect representation of these dynamics: they entangle geometry and motion with appearance, texture, and illumination, forcing the model to infer three-dimensional transformations from two-dimensional observations. We argue that point-based geometry provides a more natural state space for driving. It explicitly captures spatial structure and both rigid and non-rigid scene dynamics while remaining aligned with the 3D space of driving actions. Building on this insight, we introduce **GeoWAM**, a visual geometry world action model for autonomous driving. Rather than predicting future images, GeoWAM is pretrained to forecast future scene geometry, yielding representations that jointly encode spatial structure and temporal evolution. A geometry-conditioned action head then leverages these learned geometric dynamics to predict future ego-trajectories. Extensive experiments show that GeoWAM outperforms image-based alternatives, achieving a combined EPDMS of 36.6 on `navhard` without PDMS supervision and strong zero-shot generalization to nuScenes, with a collision rate of 0.24%. Scaling geometry pretraining with unlabeled data further improves performance, increasing the `navhard` score by 8.2% to 39.6 and strengthening zero-shot transfer to nuScenes, where the collision rate is reduced by 50% to 0.12%. Together, these results establish geometry as an effective state representation for autonomous driving and geometry pretraining as a general, *scalable* strategy for downstream planning.

1 Introduction

Autonomous driving is fundamentally a problem of anticipation. A capable driving agent must understand not only the current scene, but also how the scene is likely to evolve and how that evolution constrains its actions. Recent end-to-end driving systems increasingly build on pretrained foundation models to acquire the broad representations needed for this task.

Vision-Language-Action (VLA) models transfer the semantic knowledge and reasoning capabilities of pretrained vision-language models to ego-trajectory prediction (Hwang et al., 2024; Zhou et al., 2026; Tian et al., 2024; Jiang et al., 2026; Wang et al., 2025d; Li et al., 2026b; Yao et al., 2026a). While effective for high-level understanding and decision making, their action-centric objectives leave future scene dynamics implicit in the learned policy representation.

World models instead learn scene dynamics by forecasting future observations, often with high-capacity video-generation backbones (Hu et al., 2023a; Gao et al., 2024; Wang et al., 2024c;b; Hassan et al., 2025). World action models (WAMs) connect such prediction to planning by coupling future-state forecasting with trajectory generation (Li et al., 2026a; Zheng et al., 2025; Bartoccioni et al., 2025; Xiong et al., 2026). However, most WAMs inherit image-based state representations from video world models and therefore learn driving dynamics through future-frame prediction.

In this paper, we argue that pixels are not an ideal state representation for modeling driving dynamics. Images encode geometry and motion *indirectly*, entangling them with appearance, texture, and

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.

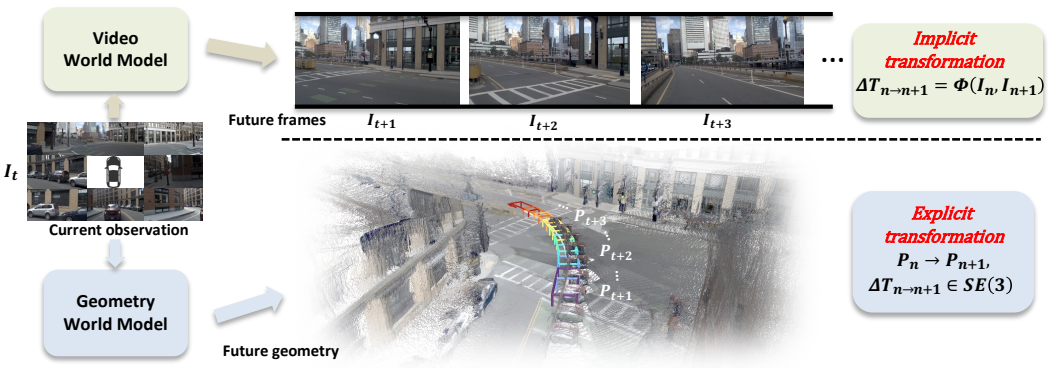


Figure 1: **Video-based versus geometry-based world modeling.** Video world models predict future pixels, from which the 3D transformations required for planning are implicitly included and difficult to recover. Geometry world models instead predict explicit future 3D structure, whose temporal evolution directly exposes object and ego motion in a planning-aligned state space.

illumination. As illustrated in fig. 1, pixel-space predictions are not naturally aligned with trajectory planning in 3D, making the underlying 3D transformations difficult to recover. Geometry, by contrast, provides a *native* state space that directly represents 3D scene structure, while its temporal evolution reveals object and ego motion in a form naturally aligned with planning.

Building on this insight, we introduce **GeoWAM**, a visual geometry world action model for autonomous driving. Instead of forecasting future images, GeoWAM is pretrained to predict future scene geometry, learning representations that jointly capture spatial structure and temporal evolution. A geometry-conditioned action head then leverages these learned geometric dynamics to predict future ego trajectories. By placing both world modeling and action prediction in a shared geometric space, GeoWAM provides the driving policy with representations explicitly organized around 3D structure and motion. Moreover, we find that geometry pretraining is a data-scalable strategy that can improve downstream planning with increasing pretraining data while transferring effectively across datasets. Extensive experiments on geometry prediction, open-loop and close-loop planning validate geometry as an effective state representation for modeling driving dynamics and supporting downstream planning.

In summary, our contributions are:

- We introduce **GeoWAM**, a world action model built on geometry as a native state representation for driving. GeoWAM couples future-geometry modeling with trajectory prediction through an inverse-dynamics-like formulation, representing scene dynamics and ego trajectories within the same 3D coordinate space.
- We develop a data-scalable geometry pretraining strategy that learns spatial structure and temporal dynamics by forecasting future scene geometry from historical multiview observations. Scaling this pretraining with additional driving data consistently strengthens downstream planning and cross-dataset generalization.
- Extensive experiments on future-geometry prediction, planning under different settings showcase that GeoWAM consistently outperforms image-based alternatives, demonstrating the effectiveness of geometry as a state representation for autonomous driving.

2 Related Work

2.1 World Models and World Action Models for Autonomous Driving

World models learn environment dynamics from past observations, in autonomous driving, they are predominantly formulated as future-video generation (Hu et al., 2023a; Wang et al., 2024b; Gao et al., 2024; Hassan et al., 2025). Occupancy-based alternatives (Zheng et al., 2024a; Yang et al., 2025) forecast 3D scene evolution but require ground-truth occupancy supervision.

World action models (WAMs) couple future-state prediction with action generation, so that learned environment dynamics directly inform the policy. Driving into the Future (Wang et al., 2024c) imagines multiview futures conditioned on candidate maneuvers and selects the trajectory whose predicted future receives the highest image-based reward. VaViM and VaVAM (Bartoccioni et al., 2025) augment autoregressive video modeling with an action expert, while Epona (Zhang et al., 2025) uses separate diffusion-based heads for image and trajectory prediction. DriveVLA-W0 (Li et al., 2026a) further uses future-image prediction as a dense self-supervised signal for driving vision-language-action models. PWM (Zhao et al., 2026) jointly forecasts state and action evolution, whereas DriveLaW (Xia et al., 2026) unifies planning and video generation within a shared latent driving world. UniDrive-WM (Xiong et al., 2026) unifies scene understanding, trajectory planning, and trajectory-conditioned future image generation within a shared vision-language architecture, whereas DriveWAM (Shi et al., 2026) adapts a pretrained video diffusion transformer into a unified video-action policy under a joint flow-matching objective. EponaV2 (Xu et al., 2026) additionally supervises future depth and semantic features alongside image prediction. Despite their architectural differences, most driving WAMs use image or video generation as their world-modeling interface, leaving 3D scene evolution implicit. GeoWAM makes metric geometry the primary prediction space and conditions trajectory generation on the resulting future geometric dynamics.

2.2 Geometry Foundation Models

General visual geometry models have increasingly replaced task-specific reconstruction pipelines with feed-forward prediction of dense geometric quantities. DUSt3R (Wang et al., 2024a) introduced point-map regression for uncalibrated image pairs, removing the need to explicitly estimate camera parameters before reconstruction. CUT3R (Wang et al., 2025b) extends this formulation with a persistent state for continuous 3D perception, while VGGT (Wang et al., 2025a) jointly infers camera parameters, depth, point maps, and tracks from a variable number of views. MapAnything (Keetha et al., 2026) further supports flexible geometric inputs and directly recovers metric-scale scene geometry in a unified feed-forward model. Together, these methods establish dense point maps as a scalable representation for 3D reconstruction.

Driving scenes introduce additional requirements, including long-range metric accuracy, temporal motion, surround-view observations, and substantial variation across camera configurations. DVGT (Zuo et al., 2026a) addresses these requirements with an ego-centric formulation that predicts metric point maps and ego poses from multi-frame, multi-view images using factorized spatial-temporal attention, while DVGT-2 (Zuo et al., 2026b) scales this geometry representation toward autonomous-driving action modeling. Most visual geometry models, however, primarily reconstruct the scene contained in observed images. VGGT-World (Sun et al., 2026) is a recent exception that transforms VGGT into an autoregressive geometry world model for forecasting future geometry in its feature space. GeoWAM instead forecasts dense metric point maps and connects the learned future geometric dynamics to downstream trajectory planning.

3 Methodology

We introduce **GeoWAM**, a visual geometry world action model for autonomous driving, as illustrated in fig. 2. Unlike conventional video-based world action models, GeoWAM first learns to forecast the three-dimensional evolution of a driving scene and then uses the predicted geometric dynamics to plan the future motion of the ego vehicle. The training of GeoWAM consists of two stages. In the first stage (section 3.1), we train a visual geometry world model using driving sequences to predict dense future scene geometry from historical multiview images. In the second stage (section 3.2), we extend the pretrained visual geometry world model with a geometry-conditioned action head and jointly finetune the complete model for both future-geometry prediction and trajectory planning.

3.1 Visual Geometry World Model

Multiview geometry encoding. Let $\mathbf{I}_{t-K+1:t} = \{\mathbf{I}_\tau^v \mid \tau = t-K+1, \ldots, t;\ v = 1, \ldots, V\}$ denote the K-frame multiview image history, where $\mathbf{I}_\tau^v \in \mathbb{R}^{3\times H\times W}$ is the image from camera v at time τ. We adopt DVGT-2 (Zuo et al., 2026b) as our geometry encoder $\mathcal{E}_\theta$ to encode the multiview

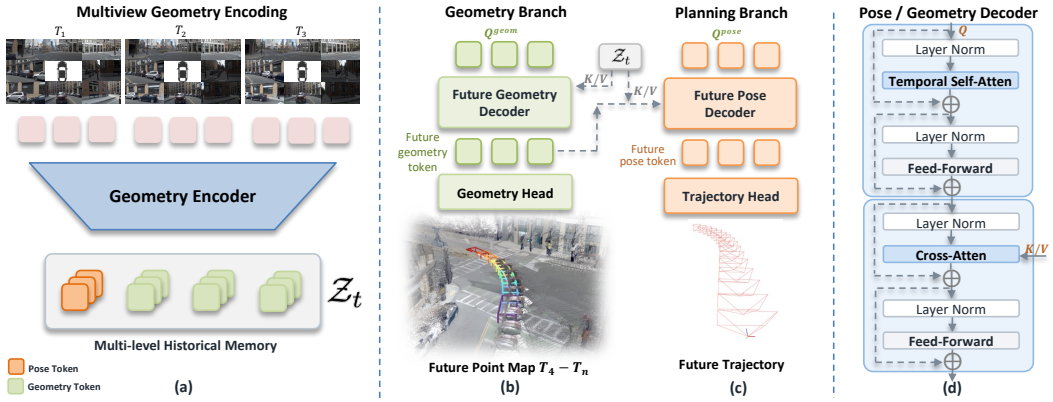


Figure 2: **GeoWAM pipeline.** (a) The geometry encoder builds a multi-level historical memory $\mathcal{Z}_t$ from multiview images. (b) Future-geometry queries attend to $\mathcal{Z}_t$ to produce future geometry tokens, which are then decoded into dense point maps. (c) Future-pose queries attend to the predicted geometry and $\mathcal{Z}_t$ to produce future pose tokens, which are then decoded into trajectories. (d) Detailed structure of pose/geometry decoder.

image sequence into multi-level historical memory, as shown in fig. 2 (a). We retain the outputs from L selected feature levels, indexed by $\ell = 1, \ldots, L$. At each time step τ and feature level ℓ, the encoder produces two types of tokens. The geometry tokens $\mathbf{X}_\tau^\ell \in \mathbb{R}^{V\times P\times D}$ represent the spatial scene structure, while the ego tokens $\mathbf{E}_\tau^\ell \in \mathbb{R}^{V\times N_e\times D}$ encode ego-motion context. Here, P and N_e are the number of geometry tokens and ego tokens per view, respectively, and D is their feature dimension. We concatenate them along the token dimension as $\mathbf{Z}_\tau^\ell = [\mathbf{X}_\tau^\ell; \mathbf{E}_\tau^\ell] \in \mathbb{R}^{V\times(P+N_e)\times D}$. The complete historical geometry memory is defined as follows:

$$\mathcal{Z}_t = \left\{\mathbf{Z}_\tau^\ell \mid \tau = t-K+1, \ldots, t;\ \ell = 1, \ldots, L\right\} = \mathcal{E}_\theta(\mathbf{I}_{t-K+1:t}). \tag{1}$$

Future geometry decoding. As shown in fig. 2 (b), the future-geometry branch predicts the scene over the next F steps from a set of learned geometry queries. Let $\mathbf{q}^{\text{geom}} \in \mathbb{R}^d$ be a learned query seed, where d is the hidden dimension of the future geometry decoder. We replicate this seed for every future step, camera view, and spatial location. The geometry query at future step k, view v, and spatial location p is

$$\mathbf{Q}_{t+k}^{\text{geom},v,p} = \mathbf{q}^{\text{geom}} + \mathbf{e}_{K+k}^{\text{time}} + \mathbf{e}_v^{\text{view}} + \mathbf{e}_p^{\text{2D}}, \tag{2}$$

where $\mathbf{e}_{K+k}^{\text{time}}$ and $\mathbf{e}_v^{\text{view}}$ are learned temporal and view embeddings, and $\mathbf{e}_p^{\text{2D}}$ is a two-dimensional positional embedding. We use $\mathcal{Q}_{t+1:t+F}^{\text{geom}} = \{\mathbf{Q}_{t+k}^{\text{geom},v,p}\}_{k,v,p}$ to denote all future geometry queries.

The decoder first projects the historical memory $\mathcal{Z}_t$ to its hidden dimension. It then updates the geometry queries in two steps at each decoder layer, as shown in fig. 2 (d). First, causal temporal self-attention models the evolution of each spatial location across the F future steps. Second, the updated queries cross-attend to $\mathcal{Z}_t$ to retrieve the relevant context from the memory. After that, we obtain a future geometry latent $\hat{\mathbf{U}}_{t+k} \in \mathbb{R}^{V\times P\times d}$ for each future step. For each feature level ℓ, an output projection $\mathbf{W}_\ell \in \mathbb{R}^{D\times d}$ maps this latent to the feature space required by the geometry head:

$$\begin{aligned} \hat{\mathcal{U}}_{t+1:t+F} &= \mathcal{D}_\phi(\mathcal{Q}_{t+1:t+F}^{\text{geom}}, \mathcal{Z}_t), \\ \hat{\mathbf{X}}_{t+k}^\ell &= \hat{\mathbf{U}}_{t+k}\mathbf{W}_\ell^\top, \quad k = 1, \ldots, F,\ \ell = 1, \ldots, L. \end{aligned} \tag{3}$$

where $\hat{\mathcal{U}}_{t+1:t+F} = \{\hat{\mathbf{U}}_{t+k}\}_{k=1}^F$. The level-specific outputs $\hat{\mathbf{X}}_{t+k}^\ell \in \mathbb{R}^{V\times P\times D}$ form a multi-level feature representation of the future scene. The shared geometry head $\mathcal{G}_\psi$ decodes this representation into a dense point map and a per-pixel confidence map:

$$\left(\hat{\mathbf{P}}_{t+k}, \hat{\mathbf{C}}_{t+k}\right) = \mathcal{G}_\psi\left(\left\{\hat{\mathbf{X}}_{t+k}^\ell\right\}_{\ell=1}^L\right). \tag{4}$$

Here, $\hat{\mathbf{P}}_{t+k} \in \mathbb{R}^{V\times H\times W\times 3}$ stores one 3D point per image pixel in the ego coordinate system at time $t+k$, and $\hat{\mathbf{C}}_{t+k} \in \mathbb{R}^{V\times H\times W}$ contains the corresponding confidence values.

Future geometry supervision. To supervise future-geometry prediction, we pass the ground-truth future images through the same geometry encoder in a target branch, obtaining geometry-token targets $\bar{\mathbf{X}}^{\ell}_{t+k} \in \mathbb{R}^{V\times P\times D}$. We use these tokens as supervision for the predicted future features, and the future feature prediction loss is defined as:

$$\mathcal{L}_{\text{feat}} = \frac{1}{FL}\sum_{k=1}^{F}\sum_{\ell=1}^{L}\left(1-\cos\left(\hat{\mathbf{X}}^{\ell}_{t+k}, \text{sg}\big(\bar{\mathbf{X}}^{\ell}_{t+k}\big)\right)\right), \tag{5}$$

where $\text{sg}(\cdot)$ denotes stop-gradient and $\cos(\cdot,\cdot)$ averages cosine similarity over cameras and patch locations. The predicted point maps are additionally supervised by dense future point-map targets $\mathbf{P}_{t+1:t+F}$ and their validity masks generated by MoGe-2 (Wang et al., 2025c). The point-map objective, averaged over future steps, views, and valid pixels, combines Euclidean point regression, confidence-aware regression, and multi-scale surface-normal consistency:

$$\mathcal{L}^{\text{future}}_{\text{point}} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{conf}} + \mathcal{L}_{\text{normal}}. \tag{6}$$

We apply the same point-map objective to the encoded current frame, producing $\mathcal{L}^{\text{current}}_{\text{point}}$ and anchoring the geometry encoder while it learns to forecast. The geometry pretraining objective is

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{feat}} + \mathcal{L}^{\text{future}}_{\text{point}} + \mathcal{L}^{\text{current}}_{\text{point}}. \tag{7}$$

3.2 GeoWAM: Visual Geometry World Action Model

After geometry pretraining, the model is able to predict how the scene geometry will evolve from historical observations. We extend this capability to trajectory planning through the action branch illustrated in fig. 2 (c). Our design follows an inverse-dynamics-like pathway: the future pose decoder and trajectory head conditions on predicted future geometry to infer future trajectory.

Future ego-token decoding. We introduce N_e learned ego-query seeds $\{\mathbf{q}^{\text{ego}}_n\}_{n=1}^{N_e}$, with one seed for each ego-token slot. For every future step k and camera view v, the corresponding query is constructed as

$$\mathbf{Q}^{\text{ego},v,n}_{t+k} = \mathbf{q}^{\text{ego}}_n + \mathbf{e}^{\text{time}}_{K+k} + \mathbf{e}^{\text{view}}_v, \quad n = 1,\dots,N_e. \tag{8}$$

The future pose decoder follows the same model structure as the future geometry decoder in fig. 2 (d). Unlike the geometry decoder, its queries cross-attend to both the historical memory $\mathcal{Z}_t$ and the predicted future geometry tokens $\hat{\mathcal{U}}_{t+1:t+F}$, producing future ego tokens:

$$\hat{\mathbf{E}}_{t+1:t+F} = \mathcal{D}^{\text{ego}}_{\eta}\left(\mathbf{Q}^{\text{ego}}, \mathcal{Z}_t, \text{sg}\Big(\hat{\mathcal{U}}_{t+1:t+F}\Big)\right). \tag{9}$$

Here, $\hat{\mathbf{E}}_{t+k} \in \mathbb{R}^{V\times N_e\times D}$ denotes the predicted ego tokens at future step $t+k$. The stop-gradient operation prevents the trajectory loss from propagating through the predicted future geometry, helping preserve the geometry forecasting capability acquired during pretraining. This one-way connection also reflects our inverse-dynamics-like design.

Trajectory decoding. The action head takes the deepest-level historical ego tokens $\mathbf{E}^{L}_{t-K+1:t}$ together with the predicted future ego tokens $\hat{\mathbf{E}}_{t+1:t+F}$. It appends the two sequences along the temporal dimension and refines them with a causal temporal transformer, allowing each future ego token to incorporate the preceding historical and predicted motion context. A learned trajectory query then cross-attends to the refined historical and future ego features, and a regression head maps the resulting query to a single future trajectory.

Planning objective. During planning finetuning, we retain the future and current geometry objectives and supervise the trajectory with an ℓ_1 regression loss $\mathcal{L}_{\text{traj}}$. We additionally predict the relative poses between historical frames using an auxiliary ℓ_1 loss $\mathcal{L}_{\text{pose}}$. The complete finetuning objective is

$$\mathcal{L}_{\text{plan}} = \mathcal{L}_{\text{pre}} + \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}}. \tag{10}$$

Table 1: **Future geometry prediction performance on nuScenes at different horizons.** PAI denotes the NVIDIA PhysicalAI-Autonomous-Vehicles dataset (NVIDIA, 2025). Bold and underlined values indicate the best and second-best results, respectively.

	Abs Rel ↓					$\delta < 1.25$ ↑				
Method/Horizon	**1s**	**2s**	**3s**	**4s**	**mean**	**1s**	**2s**	**3s**	**4s**	**mean**
Epona (Zhang et al., 2025) + DVGT (Zuo et al., 2026a)	0.229	0.263	0.292	0.310	0.274	0.732	0.677	0.620	0.589	0.655
Cosmos 3 (Agarwal et al., 2026) + DVGT (Zuo et al., 2026a)	0.300	0.376	0.405	0.422	0.376	0.588	0.513	0.464	0.447	0.503
VGGT-World (Sun et al., 2026)	0.272	0.329	0.342	0.357	0.325	0.612	0.553	0.513	0.497	0.544
GeoWAM (Ours) w/o PAI pretraining	0.171	0.190	0.227	**0.245**	0.208	0.849	0.813	0.777	**0.716**	0.789
GeoWAM (Ours) w/ PAI pretraining	**0.146**	**0.176**	**0.208**	0.276	**0.202**	**0.873**	**0.835**	**0.797**	0.704	**0.802**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. For geometry pretraining, we use image sequences from the NAVSIM `navtrain` split (Dauner et al., 2024) and the training set of the NVIDIA PhysicalAI-Autonomous-Vehicles dataset (PAI) (NVIDIA, 2025). We evaluate future-geometry prediction on the nuScenes validation set (Caesar et al., 2020). For planning, we finetune GeoWAM on the NAVSIM `navtrain` split and evaluate it on the open-loop `navtest` benchmark (Dauner et al., 2024) and the more challenging, two-stage pseudo-closed-loop `navhard` benchmark (Cao et al., 2025). We additionally evaluate zero-shot planning on nuScenes to showcase the generalizability of our method.

Metrics. For future geometry prediction, we convert each predicted future point map to ray depth, and report absolute relative error and threshold accuracy $\delta < 1.25$. For planning, we use the Extended Predictive Driver Model Score (EPDMS) from NAVSIM v2. For nuScenes planning, we report L2 displacement error and collision rate at 1, 2, and 3 seconds, together with their averages.

4.2 FUTURE GEOMETRY PREDICTION

Table 1 compares GeoWAM with the geometry world model VGGT-World (Sun et al., 2026) and the video world models Epona (Zhang et al., 2025) and Cosmos 3 (Agarwal et al., 2026) on nuScenes. To ensure a fair comparison, none of the evaluated models is trained on nuScenes. For Epona, we use the checkpoint trained on NAVSIM `navtrain`. For the video baselines, we reconstruct geometry from their predicted RGB frames using DVGT (Zuo et al., 2026a). Both GeoWAM variants achieve better Abs Rel and $\delta < 1.25$ than every baseline at every horizon. Geometry pretraining on PhysicalAI is most effective at short horizons, lowering Abs Rel from 0.171 to 0.146 at 1 s and from 0.227 to 0.208 at 3 s, while the variant without pretraining remains slightly better at the longest 4 s horizon. These results show that GeoWAM outperforms both geometry world model and geometry reconstructed from video world model.

4.3 PLANNING ON NAVSIM V2

`navtest`. Table 2 compares GeoWAM with perception-based planners and recent world-action models on the `navtest` split. Without geometry pretraining, GeoWAM achieves an EPDMS of **90.2**. Geometry pretraining on PhysicalAI further improves the score to **90.7**.

Two-Stage Planning on `navhard`. We further evaluate GeoWAM under the two-stage pseudo-closed-loop planning protocol on the `navhard` split (Cao et al., 2025). Importantly, GeoWAM ***uses no PDMS-derived supervision*** during training, including reinforcement learning (RL) or a learned PDMS scorer. As reported in table 3, GeoWAM without PAI pretraining achieves an EPDMS of **36.6**, outperforming all baselines trained without PDMS supervision, as well as two baselines trained with PDMS supervision (Cao et al., 2025; Xu et al., 2026). It trails only RAP (Feng et al., 2026) (36.9), which is trained with PDMS-score supervision. Geometry pretraining on PAI further raises the score by 3.0 points to **39.6**, surpassing all baselines, including RAP.

Together, these results show that future-geometry modeling provides an effective representation for downstream planning. The larger pretraining gain on `navhard` further suggests that learned geometric dynamics are especially valuable when planning errors influence subsequent observations.

Table 2: Open-loop planning results on the NAVSIM v2 `navtest` split.

Method	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
Transfuser (Chitta et al., 2022)	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	84.0
Hydra-MDP++ (Li et al., 2025a)	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
DriveSuprim (Yao et al., 2026b)	97.5	96.5	99.4	99.6	**88.4**	96.6	95.5	98.3	77.0	83.1
ARTEMIS (Feng et al., 2025)	98.3	95.1	98.6	99.8	81.5	97.4	96.5	98.3	–	83.1
DiffusionDrive (Liao et al., 2025)	98.2	96.2	99.5	99.8	87.4	97.3	96.9	**98.4**	87.7	88.2
WoTE (Li et al., 2025b)	98.5	96.8	98.8	99.8	86.1	97.9	95.5	98.3	82.9	87.7
DriveVLA-W0 (Li et al., 2026a)	98.4	95.2	99.4	**99.9**	86.6	97.9	97.8	98.3	82.7	86.9
PWM (Zhao et al., 2026)	98.8	95.9	99.4	**99.9**	86.4	98.4	97.6	98.3	85.3	88.2
DriveLaW (Xia et al., 2026)	98.7	96.9	99.6	99.8	87.5	98.3	97.6	**98.4**	77.4	88.6
DVGT-2 (Zuo et al., 2026b)	98.7	**97.9**	**99.7**	**99.9**	87.9	98.0	**98.2**	98.2	77.0	89.6
EponaV2 (Xu et al., 2026)	98.5	97.4	99.5	**99.9**	87.9	98.1	97.7	98.2	77.4	88.9
GeoWAM (Ours) w/o PAI pretraining	98.7	97.7	**99.7**	**99.9**	87.0	98.1	97.9	98.3	86.8	90.2
GeoWAM (Ours) w/ PAI pretraining	**99.0**	**97.9**	**99.7**	**99.9**	86.7	**98.5**	97.9	98.3	87.8	**90.7**

Table 3: **`navhard` leaderboard.** Methods trained with reinforcement learning or PDMS-score supervision are shown in gray and marked with †. Among the remaining methods, bold and underlined values indicate the best and second-best results, respectively.

Method	Stage	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
DriveVLA-W0 (Li et al., 2026a)	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)	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	
DVGT-2 (Zuo et al., 2026b)	S1	97.2	91.3	98.4	99.8	**84.8**	95.5	95.5	97.5	71.4	31.7
	S2	77.8	73.8	81.3	98.3	**91.5**	73.2	48.0	83.9	45.1	
LTFv6 (Nguyen et al., 2026)	S1	96.5	86.6	99.2	99.5	84.4	95.1	94.4	97.7	76.4	31.9
	S2	79.8	75.5	86.2	97.8	89.5	76.0	50.0	95.2	66.7	
NavFormer† (Cao et al., 2025)	S1	96.2	92.4	95.7	99.6	83.8	96.0	94.7	96.4	60.9	34.1
	S2	85.7	81.0	83.5	97.6	90.1	82.4	48.2	94.9	48.4	
EponaV2† (Xu et al., 2026)	S1	97.3	90.7	99.4	100.0	83.3	97.3	97.3	97.6	60.9	36.1
	S2	83.6	78.0	88.0	98.9	86.0	80.3	50.1	96.1	52.0	
RAP-DINO† (Feng et al., 2026)	S1	97.1	94.4	98.8	99.8	83.9	96.9	94.7	96.4	66.2	36.9
	S2	83.2	83.9	87.4	98.0	86.9	80.4	52.3	95.2	52.4	
GeoWAM (Ours) w/o PAI pretraining	S1	97.7	91.5	99.1	99.8	83.8	95.8	96.0	**97.8**	79.0	36.6
	S2	80.4	**76.3**	87.3	**98.7**	88.9	76.2	49.9	94.0	56.0	
GeoWAM (Ours) w/ PAI pretraining	S1	**98.9**	**92.9**	**99.6**	**100.0**	82.0	**97.3**	**96.9**	**97.8**	**79.5**	**39.6**
	S2	**84.1**	76.0	**87.4**	98.6	84.0	80.2	**50.5**	96.9	63.7	

4.4 Zero-shot Planning on NuScenes

To assess cross-dataset generalization and the effect of geometry pretraining, we evaluate two GeoWAM variants on nuScenes without nuScenes-specific trajectory supervision. Both planners are trained only on NAVSIM `navtrain`. The pretrained variant additionally uses image sequences from `navtrain` and PhysicalAI for geometry pretraining. As shown in Table 4, even without geometry pretraining on PAI, GeoWAM achieves an average L2 error of 1.28 m and collision rate of 0.24%, outperforming both zero-shot video-based world models (*i.e.* PWM and DriveVLA-W0).

Table 4: **Zero-shot motion planning performance on nuScenes.**

Method	nuScenes Finetune	Auxiliary Supervision	L2 (m) ↓				Collision Rate (%) ↓			
			1s	2s	3s	Avg.	1s	2s	3s	Avg.
ST-P3 (Hu et al., 2022)	✓	Map&Box&Depth	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD (Hu et al., 2023b)	✓	Map&Box&Motion	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
OccNet (Sima et al., 2023)	✓	3D-Occ&Map&Box	1.29	2.13	2.99	2.14	0.21	0.59	1.37	0.72
OccWorld (Zheng et al., 2024a)	✓	3D-Occ	0.52	1.27	2.41	1.40	0.12	0.40	2.08	0.87
VAD-Tiny (Jiang et al., 2023)	✓	Map&Box&Motion	0.60	1.23	2.06	1.30	0.31	0.53	1.33	0.72
VAD-Base (Jiang et al., 2023)	✓	Map&Box&Motion	0.54	1.15	1.98	1.22	0.04	0.39	1.17	0.53
GenAD (Zheng et al., 2024b)	✓	Map&Box&Motion	0.36	0.83	**1.55**	0.91	0.06	0.23	1.00	0.43
Doe-1 (Zheng et al., 2024c)	✓	QA	0.50	1.18	2.11	1.26	0.04	0.37	1.19	0.53
Epona (Zhang et al., 2025)	✓	Future RGB	0.61	1.17	1.98	1.25	0.01	0.22	0.85	0.36
PWM (Zhao et al., 2026)	✗	Future RGB	2.06	3.91	6.00	3.99	0.12	0.15	0.86	0.36
DriveVLA-W0 (Li et al., 2026a)	✗	Future RGB	0.43	1.26	2.60	1.43	0.22	0.66	1.42	0.77
GeoWAM (Ours) w/o PAI pretraining	✗	Future Geometry	0.42	1.19	2.24	1.28	**0.00**	**0.10**	0.60	0.24
GeoWAM (Ours) w/ PAI pretraining	✗	Future Geometry	**0.29**	**0.79**	1.58	**0.89**	0.02	0.12	**0.23**	**0.12**

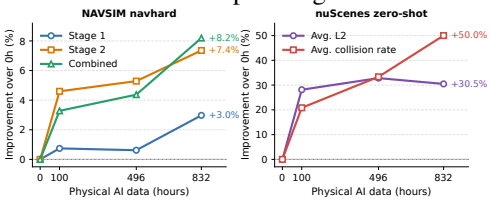
Notably, this variant already attains a lower collision rate than every baseline in the table, including planners finetuned on nuScenes, even though its L2 error still trails the strongest finetuned planners. Geometry pretraining further reduces the averages to 0.89 m and 0.12%, respectively, without additional trajectory supervision, surpassing all the nuScenes-finetuned baselines on both average metrics, suggesting that future geometry supports cross-dataset planning and that geometry pretraining strengthens this transfer.

4.5 Scaling Geometry Pretraining

Our first-stage geometry pretraining (section 3.1) learns future scene geometry directly from image sequences, allowing it to leverage additional driving data without ground-truth trajectory annotations. To evaluate this scalability, we vary the amount of PhysicalAI data used for geometry pretraining from 0 to 832 hours, with 832 hours covering the entire PhysicalAI training set. The left panel of table 5 shows that scaling the pretraining data improves `navhard` EPDMS from 36.6 to 39.6 and reduces the nuScenes zero-shot collision rate from 0.24% to 0.12%. The right panel shows the corresponding relative improvements over the model without geometry pretraining. The nuScenes L2 error is not strictly monotonic, since it measures deviation from the recorded human trajectory and may penalize other valid driving behaviors. In contrast, the collision rate decreases consistently as the pretraining data scales, indicating improved planning safety. As shown by the parameter comparison in section A.2, GeoWAM uses 1.9B parameters, substantially fewer than DriveVLA-W0 and EponaV2. Its planning performance nevertheless continues to improve with additional geometry-pretraining data, without increasing model capacity or trajectory supervision.

Table 5: **Effect of geometry-pretraining data scale on downstream planning.** Left: absolute planning performance at different pretraining-data scales. Right: relative improvements over the model without geometry pretraining on `navhard` and nuScenes zero-shot planning.

PAI Data (hours)	navhard EPDMS ↑			nuScenes Zero-shot L2 (m) ↓				nuScenes Zero-shot Collision Rate (%) ↓			
	S1	S2	Comb.	1s	2s	3s	Avg.	1s	2s	3s	Avg.
0	80.7	43.5	36.6	0.42	1.19	2.24	1.28	**0.00**	**0.10**	0.60	0.24
100	81.3	45.5	37.8	0.31	0.81	1.64	0.92	0.02	0.12	0.44	0.19
496	81.2	45.8	38.2	**0.29**	**0.76**	**1.55**	**0.86**	0.04	0.12	0.33	0.16
832	**83.1**	**46.7**	**39.6**	**0.29**	0.79	1.58	0.89	0.02	0.12	**0.23**	**0.12**



4.6 Ablation Study

We ablate three factors in two groups (table 6): future-geometry modeling and PAI pretraining in (a), and the prediction horizon in (b). (b) is conducted without PAI pretraining.

Future geometry and PAI pretraining. Removing future-geometry prediction while retaining current-frame geometry supervision lowers EPDMS from 90.2 to 89.2 on `navtest` and from 36.6 to 31.7 on `navhard`. These gaps isolate the benefit of forecasting future geometric evolution. PAI pretraining further improves EPDMS by 0.5 points on `navtest` and 3.0 points on `navhard`. The larger gain on `navhard` suggests that pretraining on more diverse road layouts and traffic scenes strengthens the learned geometric dynamics, particularly when planning errors affect subsequent observations under pseudo-closed-loop evaluation.

Prediction horizon. With F future frames predicted at 2 Hz, the forecast horizon ranges from 1 s (F=2) to the full 4 s planning horizon (F=8). EPDMS increases monotonically with F, by 1.0 on `navtest` and 5.9 on `navhard` in total, and the per-step gain on `navhard` grows rather than saturates (+1.5, +2.1, and +2.3). In contrast, forecasting only 1 s ahead brings no benefit: F=2 matches the variant without future geometry on `navtest` (89.2) and falls below it on `navhard` (30.7 vs. 31.7). This suggests that the improvement does not stem from the additional decoder capacity alone, but from exposing scene dynamics over a horizon aligned with the planned trajectory.

4.7 Ego-Motion Recovery from Predicted States

To examine how directly different predicted states expose future ego motion, we compare trajectories recovered from PWM's future video (Zhao et al., 2026) and GeoWAM's future geometry. For PWM,

Table 6: **Ablation studies on downstream planning.**

Variant	Current Geometry	Future Geometry	PAI Pretraining	navtest EPDMS ↑	navhard EPDMS ↑
w/o Future Geometry	✓			89.2	31.7
w/o PAI Pretraining	✓	✓		90.2	36.6
Full	✓	✓	✓	**90.7**	**39.6**

(a) Future geometry and PAI pretraining

# Frames	navtest EPDMS ↑	navhard EPDMS ↑
2	89.2	30.7
4	89.5	32.2
6	89.9	34.3
8	**90.2**	**36.6**

(b) Prediction horizon

we recover ego motion from generated RGB frames using two separate methods: classical epipolar geometry (Hartley & Zisserman, 2004) and the learning-based VGGT (Wang et al., 2025a). For GeoWAM, we match static 3D keypoints across predicted point maps and recover the relative rigid transforms with Kabsch alignment (Kabsch, 1976).

We show two examples in fig. 3, including each model's predicted future state alongside the trajectories recovered by all three methods and the ground truth trajectory. In these examples, the geometry-derived trajectories follow the ground truth more closely, whereas the trajectories recovered from future video show larger deviations. This qualitative comparison illustrates how future geometry makes ego motion more directly accessible than pixel-space predictions. More visualizations are provided in section A.1.

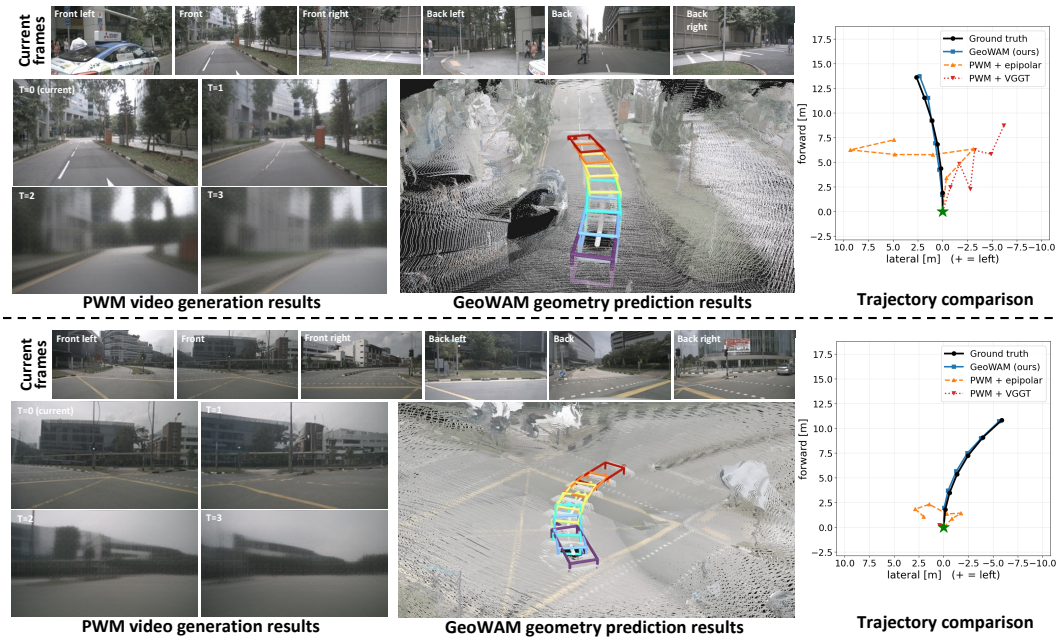


Figure 3: **Ego-motion recovery from predicted future states.** Each row shows current observations, PWM-generated future video, GeoWAM-predicted future geometry, and trajectories recovered from different future representation, compared with the logged ground truth.

5 Conclusion

We introduced GeoWAM, a visual geometry world action model that treats geometry as the state space for autonomous driving. Rather than forecasting future pixels, GeoWAM predicts future scene geometry from historical multiview observations and uses the learned geometric dynamics to condition trajectory prediction through an inverse-dynamics-like action branch. Experiments on future-geometry prediction and planning under various settings consistently demonstrate the advantage of this geometry-based formulation over image-based alternatives. Scaling geometry pretraining with additional driving data further improves planning performance, without increasing model capacity or trajectory supervision. These results establish geometry as an effective, planning-aligned state representation and geometry pretraining as a general, scalable strategy for downstream planning.

AI USE STATEMENT

In this work, we did not use generative AI tools for research tasks requiring disclosure, including developing or refining research hypotheses, designing the methodology or experiments, implementing the proposed method, analyzing data, or interpreting experimental results. Other required-disclosure tasks, such as generating synthetic datasets, developing theoretical models, formulating or proving mathematical claims, translation, and qualitative or thematic data analysis, were not applicable to this work.

Additionally, we used generative AI tools to improve the readability and presentation of the manuscript and to organize author-provided experimental results into tables. The AI tools were used only to assist with the presentation of existing results and were not used to generate, compute, modify, or interpret the reported experimental results. We reviewed all AI-assisted work: all AI-assisted text was proofread and revised by the authors, and all values in AI-assisted tables were manually verified against the original experimental results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the implementation details, model configurations, training settings, and hyperparameters necessary to reproduce our experiments in the main paper and the appendix. To further facilitate reproducibility, we plan to publicly release the source code after the review process, including the training and evaluation scripts as well as model checkpoints corresponding to the results reported in the paper. Together, these materials are intended to enable reproduction of both the proposed method and the reported experimental results.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Florent Bartoccioni, Elias Ramzi, Victor Besnier, Shashanka Venkataramanan, Tuan-Hung Vu, Yihong Xu, Loick Chambon, Spyros Gidaris, Serkan Odabas, David Hurych, et al. Vavim and vavam: Autonomous driving through video generative modeling. *arXiv preprint arXiv:2502.15672*, 2025.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11618–11628. IEEE, 2020.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *9th Annual Conference on Robot Learning*, 2025. URL https://openreview.net/forum?id=9uKL9FJBiz.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

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.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. In *International Conference on Learning Representations*, volume 2026, pp. 37722–37737, 2026.

Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *IEEE Robotics and Automation Letters*, 11(1):226–233, 2025.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2024.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

Mariam Hassan, Sebastian Stapf, Ahmad Rahimi, Pedro Rezende, Yasaman Haghighi, David Brüggemann, Isinsu Katircioglu, Lin Zhang, Xiaoran Chen, Suman Saha, et al. Gem: A generalizable ego-vision multimodal world model for fine-grained ego-motion, object dynamics, and scene composition control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22404–22415, 2025.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023a.

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 *European Conference on Computer Vision*, pp. 533–549. 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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023b.

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. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8306–8316. IEEE, 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. *International Journal of Computer Vision*, 134(9):407, 2026.

W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, Sep 1976. doi: 10.1107/S0567739476001873. URL https://doi.org/10.1107/S0567739476001873.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manue Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. Mapanything: Universal feed-forward metric 3d reconstruction; mapanything.github.io. In *2026 International Conference on 3D Vision (3DV)*, pp. 499–509, 2026. doi: 10.1109/3DV69130.2026.00054.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 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, 2026a.

Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, et al. Unidrivevla: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026b.

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.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. Lead: Minimizing learner-expert asymmetry in end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39775–39785, 2026.

NVIDIA. PhysicalAI-Autonomous-Vehicles. `https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles`, 2025.

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, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as occupancy. 2023.

Xiangyu Sun, Shijie Wang, Fengyi Zhang, Lin Liu, Caiyan Jia, Ziying Song, Zi Huang, and Yadan Luo. Vggt-world: Transforming vggt into an autoregressive geometry world model, 2026. URL `https://arxiv.org/abs/2603.12655`.

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`.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522. IEEE, 2025b.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 35928–35959. Curran Associates, Inc., 2025c. doi: 10.52202/085713-1207. URL `https://proceedings.neurips.cc/paper_files/paper/2025/file/336572db3e99930814d6b328d4220cb6-Paper-Conference.pdf`.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024a.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024b.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025d.

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, 2024c.

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.

Zhexiao Xiong, Xin Ye, Burhan Yaman, Sheng Cheng, Yiren Lu, Jingru Luo, Nathan Jacobs, and Liu Ren. Unidrive-wm: Unified understanding, planning and generation world model for autonomous driving. *arXiv preprint arXiv:2601.04453*, 2026.

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.

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.

Jin Yao, Dhruva Dixith Kurra, Tom Lampo, Zezhou Cheng, Danhua Guo, and Burhan Yaman. Vlga: Vision-language-geometry-action models for autonomous driving. 2026a. URL `https://arxiv.org/abs/2606.12396`.

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, 2026b.

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.

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, 2026.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024a.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 87–104. Springer, 2024b.

Wenzhao Zheng, Zetian Xia, Yuanhui Huang, Sicheng Zuo, Jie Zhou, and Jiwen Lu. Doe-1: Closed-loop autonomous driving with large world model. *arXiv preprint arXiv:2412.09627*, 2024c.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, et al. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28632–28642. IEEE, 2025.

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.

Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Shengyin Jiang, Long Chen, Zhi-Xin Yang, and Jiwen Lu. Dvgt: Driving visual geometry transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14658–14668, June 2026a.

Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Hanbing Li, Long Chen, Zhi-Xin Yang, and Jiwen Lu. Dvgt-2: Vision-geometry-action model for autonomous driving at scale. *arXiv preprint arXiv:2604.00813*, 2026b.

A Appendix

A.1 Additional Qualitative Results

We provide additional qualitative comparisons of PWM and GeoWAM in fig. 4, complementing the ego-motion recovery analysis in section 4.7. Across these examples, GeoWAM predicts coherent future 3D structure, and the trajectories recovered from its geometry follow the logged ground truth more closely than those recovered from PWM-generated video.

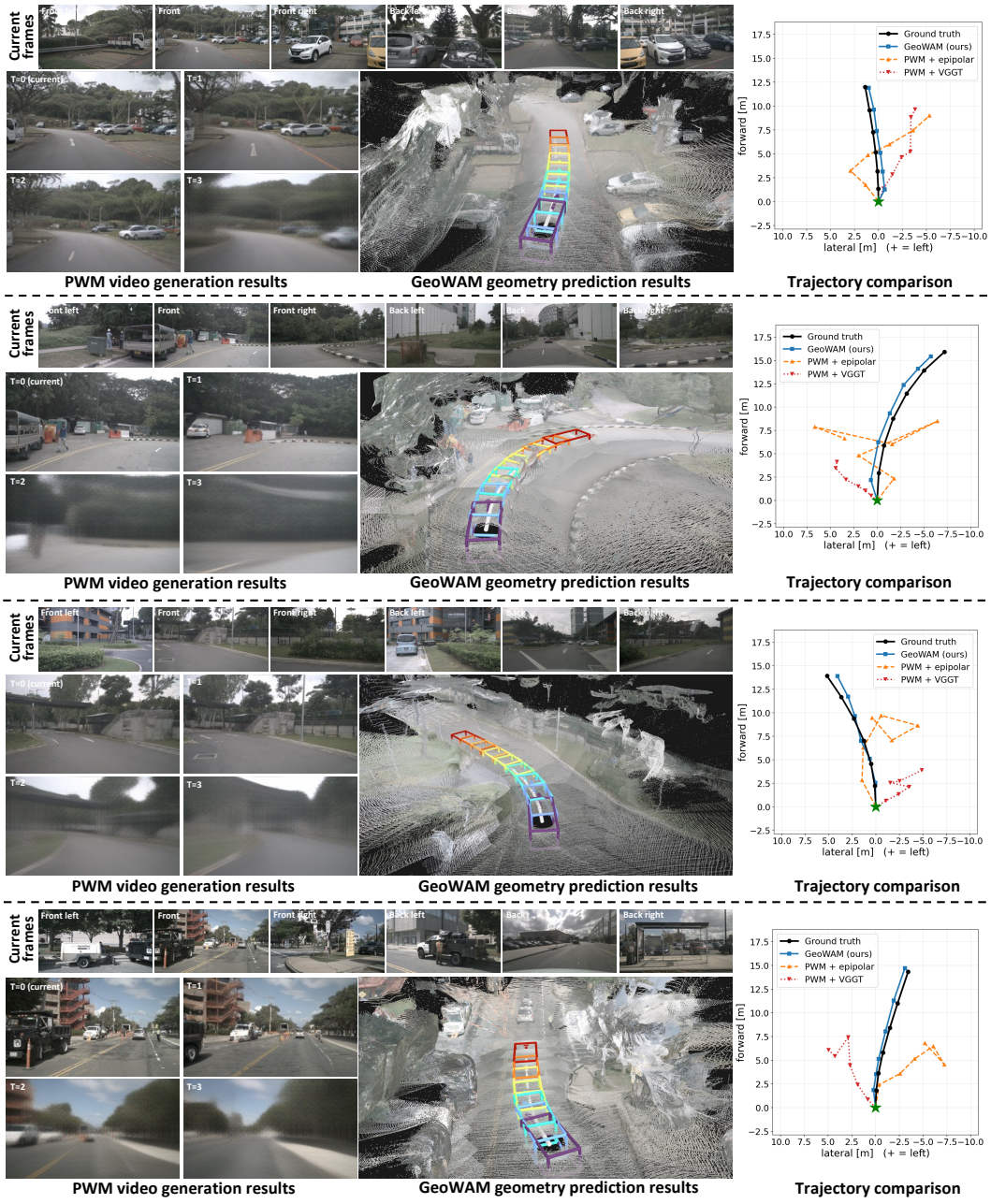


Figure 4: **Additional qualitative comparisons of predicted future states and recovered ego motion.** Each row shows current multiview observations, PWM-generated future frames, GeoWAM-predicted future geometry, and trajectories recovered from both representations alongside the logged ground truth.

Table 7: **Model size and planning performance.** We report EPDMS on `navtest` and `navhard`, and average L2 error and collision rate for zero-shot planning on nuScenes. † denotes training with PDMS-based reinforcement learning. "–" denotes an unavailable result. Bold and underlined values indicate the best and second-best results, respectively.

Method	Parameters	NAVSIM		nuScenes Zero-shot	
		`navtest` EPDMS ↑	`navhard` EPDMS ↑	Avg. L2 (m) ↓	Avg. Coll. (%) ↓
DriveLaW (Xia et al., 2026)	2.1B	88.6	30.6	–	–
DriveVLA-W0 (Li et al., 2026a)	7.5B	86.9	24.4	1.43	0.77
PWM (Zhao et al., 2026)	1.7B	88.2	–	3.99	0.36
EponaV2† (Xu et al., 2026)	6.7B	88.9	36.1	–	–
GeoWAM (Ours) w/o PAI pretraining	1.9B	<u>90.2</u>	<u>36.6</u>	<u>1.28</u>	<u>0.24</u>
GeoWAM (Ours) w/ PAI pretraining	1.9B	**90.7**	**39.6**	**0.89**	**0.12**

A.2 Model Size and Planning Performance

Table 7 compares GeoWAM with recent world-action models in terms of parameter count and planning performance across NAVSIM and nuScenes.

A.3 Implementation details.

The future geometry decoder contains six transformer layers with a hidden dimension of 1024 and 16 attention heads. Following VGGT Wang et al. (2025a), DVGT Zuo et al. (2026a), and DVGT-2 Zuo et al. (2026b), we retain the encoder outputs from layers 4, 11, 17, and 23, yielding $L = 4$ feature levels. Geometry pretraining uses three historical frames to predict $F = 8$ future frames at 2 Hz, with six to eight camera views dynamically sampled from each sequence. For the current- and future-frame point-map objectives, we generate dense point-map pseudo-targets from the corresponding RGB images using MoGe-2 (Wang et al., 2025c). We initialize the geometry encoder and point head from DVGT-2 (Zuo et al., 2026b). All geometry-pretraining variants use AdamW with a weight decay of 0.05 and a batch size of 64. The variant without PAI data is trained for 40k steps, while variants using 100, 496, and 832 hours of PAI data are trained for 40k, 70k, and 80k steps, respectively. The future decoder uses a peak learning rate of 10^{-4}, while the pretrained components use 2×10^{-5}. Both learning rates use a 5% linear warmup followed by cosine decay, and training is performed in bfloat16 precision. For downstream planning, all variants are initialized from their corresponding geometry-pretrained checkpoints and finetuned on `navtrain` for 40k steps using eight camera views and three historical frames. The future decoder and newly introduced action head use a learning rate of 10^{-4}, while the remaining pretrained parameters use 2×10^{-5}. We set both loss weights to $\lambda_{\text{traj}} = \lambda_{\text{pose}} = 5$.

A.4 Planning Evaluation Protocols

`navtest`. NAVSIM evaluates each predicted 4-second ego trajectory with a non-reactive PDM simulator, in which the ego vehicle follows the plan while surrounding agents replay their logged trajectories (Dauner et al., 2024). The resulting rollout is scored by EPDMS: NC, DAC, DDC, and TLC act as multiplicative penalties, while EP, TTC, LK, HC, and EC are combined through a weighted average. Because `navtest` applies this procedure independently to each logged scene, it remains a single-stage open-loop evaluation.

`navhard`. The more challenging `navhard` split extends the `navtest` evaluation with a two-stage pseudo-closed-loop protocol (Cao et al., 2025). After evaluating the planner on a real Stage 1 scene, the benchmark considers multiple Stage 2 follow-up scenes rendered from perturbed ego states using 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023). Their relevance is determined by the distance between each follow-up state and the endpoint of the planner's Stage 1 rollout, with closer states receiving larger Gaussian weights. The weighted Stage 2 EPDMS is multiplied by the Stage 1 score to obtain the final result. This construction approximates observation feedback and tests if a planner remains effective when earlier errors shift the ego state away from the logged trajectory.

nuScenes. We follow the standard open-loop planning protocol on the nuScenes validation set (Caesar et al., 2020; Hu et al., 2023b). The predicted ego trajectory is compared with the logged human trajectory at 1, 2, and 3 seconds using L2 displacement error, while collision rate measures how often the planned ego trajectory intersects annotated future obstacles. Because predictions are not fed back into subsequent observations, L2 primarily measures imitation of one logged behavior rather than closed-loop driving quality; collision rate provides a more direct indication of safety. For our zero-shot evaluation, GeoWAM is trained with trajectory supervision only on NAVSIM `navtrain` and receives no nuScenes-specific trajectory supervision.

A.5 Limitations and Future Work

GeoWAM predicts dense future geometry from multiple camera views, which introduces additional computation and memory overhead. Future work could explore sparse or adaptive geometric representations to support longer prediction horizons and more camera views efficiently. Moreover, the current action head produces a single deterministic trajectory and does not explicitly represent uncertainty over multiple plausible driving futures. Extending GeoWAM with probabilistic geometry prediction and multi-hypothesis planning is a promising direction.