GEOSTATE: LEARNING VIEW-DECOUPLED GEOMETRY STATES FOR 3D WORLD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent 3D world models increasingly use features from geometry foundation models (GFMs) as predictive world states, benefiting from their rich geometric priors. However, the view dependence of these features entangles scene dynamics with camera-induced variation, making the resulting state evolution harder to predict. We propose GeoState, a view-decoupled geometry representation for predictive 3D world modeling that separates scene-state evolution from camera-dependent observation. Specifically, a scene tokenizer aggregates multi-view GFM features into a view-decoupled scene state, which a geometry decoder queries using camera rays. An autoregressive world model then predicts future scene states, while camera motion is modeled separately. The predicted states can therefore be decoded under either predicted or externally specified cameras, while retaining a single context-derived scale throughout the rollout without per-frame scale alignment. Experiments show that GeoState outperforms prior 3D world models on Waymo, supports geometrically consistent cross-view querying, and generalizes zero-shot to KITTI and nuScenes.

1 INTRODUCTION

Learning to predict how the physical world evolves is a central goal of world modeling (Ha & Schmidhuber, 2018), with direct relevance to autonomous driving, robotics, and embodied intelligence. Visual world models predict the future in pixel space (Hu et al., 2023; Wang et al., 2024b; Gao et al., 2024; NVIDIA et al., 2025), which can introduce substantial representational redundancy and make geometric consistency difficult to maintain. This has motivated increasing interest in 3D world models, which represent and predict future environments in geometrically grounded state spaces. By explicitly encoding spatial structure and 3D relationships, these representations provide a structured and geometrically grounded state space that more directly reflects the real world.

Recent 3D world models have explored a range of structured geometric state spaces, including depth, occupancy, point-based representations, and 3D Gaussians (Zheng et al., 2024; Zhang et al., 2024; Yang et al., 2024; Zhu et al., 2025; Chen et al., 2025; Zuo et al., 2025). In parallel, geometry foundation models (GFMs) have substantially advanced feed-forward 3D perception by learning rich geometric representations directly from RGB observations (Wang et al., 2024a; Yang et al., 2025; Wang et al., 2025b;a; 2026). Benefiting from large-scale pretraining, these representations capture strong geometric priors that generalize across scenes and tasks, making them attractive state spaces for predictive world modeling. Building on this capability, recent approaches have begun to directly model temporal evolution in pretrained geometry feature spaces (Sun et al., 2026; Lu et al., 2026). In particular, VGGT-World (Sun et al., 2026) treats frozen VGGT features as world states and predicts their future evolution, demonstrating the potential of GFM representations for 3D forecasting.

Despite the strong performance achieved by recent GFM-based 3D world models, a strong representation for geometric perception is not necessarily an appropriate state representation for world dynamics. GFM features are typically constructed from and organized around the observing views, so changes in camera viewpoint can induce substantial changes in the representation even when the underlying physical scene remains unchanged. When such features are directly used as predictive states, a temporal model would account simultaneously for genuine scene evolution and camera-induced variation, although these arise from fundamentally different sources. In particular, even a static scene can undergo substantial changes in feature space as the camera moves, forcing the

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

temporal model to predict viewpoint-dependent transformations that do not correspond to physical scene evolution. This unnecessary variation makes the latent dynamics more complex and therefore harder to predict over long horizons. FR3D (Morbitzer et al., 2026) also shows that separating ego-motion from scene forecasting can improve temporal modeling, but its predictive state still inherits the view-dependent structure of the underlying reconstruction representation. We argue that such separation should instead be built into the representation itself: *the state should describe* ***what the world is****, while the camera should determine only* ***how that world is observed***.

To this end, we introduce GeoState, a view-decoupled geometry state learned from pretrained GFM features. Given multi-view observations, a scene tokenizer compresses their geometry features into a compact scene state represented in a shared scene coordinate system, while the query viewpoint is introduced only during decoding. Specifically, a geometry decoder takes the scene state together with rays from a query camera and predicts the geometry observed from that viewpoint. Consequently, the same scene state can be queried repeatedly from different viewpoints without re-encoding the scene. This transforms view-conditioned GFM features into a representation in which scene geometry is explicitly separated from camera-dependent observation.

Building on this representation, we formulate 3D world modeling directly in the scene-state space. The temporal model can therefore focus solely on predicting scene dynamics, while camera evolution is modeled independently. At inference time, a predicted future state can be decoded using either a separately predicted camera trajectory or a user-controlled camera. Our formulation also maintains a consistent scene gauge throughout temporal prediction and geometry decoding, avoiding the need for per-frame scale alignment during rollout.

We evaluate our approach on large-scale dynamic driving scenes, using Waymo (Sun et al., 2020) for in-domain evaluation and KITTI (Geiger et al., 2012) and nuScenes (Caesar et al., 2020) for zero-shot transfer. Our approach outperforms the compared world models in depth forecasting across all reported horizons, while maintaining a consistent scale throughout the forecasting rollout. Experiments further demonstrate geometrically consistent querying across viewpoints and generalization across driving datasets. Ablation results show that view decoupling makes the scene representation more predictable. Together, these findings support view-decoupled geometry states derived from pretrained GFMs as an effective representation for predictive 3D world modeling.

Our contributions are threefold:

- We identify viewpoint coupling in geometry foundation model features as a limitation when using them as predictive world states, and introduce **GeoState**, a view-decoupled geometry state derived from pretrained GFM representations that separates scene content from how the scene is observed.
- We develop a scene-state learning framework that realizes GeoState through a Scene Tokenizer that aggregates multi-view GFM features into a shared scene state and a query-conditioned Geometry Decoder that introduces the query viewpoint only at decoding time.
- We formulate a 3D world model that predicts scene evolution directly in GeoState space while modeling camera evolution separately, and demonstrate long-horizon geometry forecasting, geometrically consistent cross-view querying, and zero-shot generalization across driving datasets without per-frame scale alignment.

2 Related Work

2.1 3D World Models

Recent world models increasingly move beyond pixel-space prediction toward structured 3D forecasting. Copilot4D (Zhang et al., 2024) and ViDAR (Yang et al., 2024) forecast future point clouds, OccWorld (Zheng et al., 2024) models scene evolution in semantic occupancy space, and GaussianWorld (Zuo et al., 2025) models scene evolution in 3D Gaussian space. More recently, several approaches directly use learned geometry representations as predictive world states. VGGT-World (Sun et al., 2026) forecasts future VGGT features, while GeoWAM (Lu et al., 2026) learns geometry-aware predictive states for autonomous driving. FR3D (Morbitzer et al., 2026) predicts the latent state of a pretrained 3D reconstruction model while modeling ego-motion separately. In

contrast, we remove viewpoint dependence from the geometry state itself and introduce viewpoint only as a query condition during decoding.

2.2 Geometry Foundation Models

Geometry foundation models provide strong pretrained priors for multi-view 3D perception. DUSt3R (Wang et al., 2024a) formulates reconstruction through pointmap prediction, while Fast3R (Yang et al., 2025) scales joint reconstruction to larger collections of views. CUT3R (Wang et al., 2025b) introduces a persistent latent state for continuously integrating observations, and VGGT (Wang et al., 2025a) unifies camera estimation, depth, point maps, and tracking in a feed-forward architecture. VGGT-Ω (Wang et al., 2026) further strengthens this paradigm for dynamic 3D scenes, making it a natural geometry backbone for our setting. We build on its pretrained geometric features and reformulate them into a compact, view-decoupled scene-level representation better suited for temporal world modeling.

2.3 Compact 3D Representations

A related line of work compresses multi-view observations into compact scene representations. LVSM (Jin et al., 2025) encodes input views into a fixed-length set of latent tokens for novel-view synthesis, while Surflo (Guédon et al., 2026) aggregates dense VGGT features into a fixed-size global latent representation for efficient 3D surface reconstruction. 3DRAE (Wei et al., 2026) further learns a fixed-length, view-decoupled 3D latent space that can be queried from arbitrary cameras. These works demonstrate that multi-view observations can be represented without retaining a separate dense representation for each view. While they focus primarily on static scene reconstruction or novel-view synthesis, our work investigates whether such scene-level representations can serve as predictive states for temporal world modeling.

3 Method

3.1 Method Overview

Our goal is to predict future scene geometry without requiring the world model to simultaneously account for variations induced by camera motion. To this end, as illustrated in Figure 1, we first construct a view-decoupled geometry state from multi-view observations, and then model its temporal evolution while handling camera evolution separately.

GeoState learning. Given multi-view RGB observations $\mathcal{I}_t = \{I_t^v\}_{v\in\mathcal{V}_t}$, a frozen geometry foundation model extracts geometry-aware features $\mathbf{M}_t$, with the multi-view geometry expressed in a shared scene coordinate system defined by a common anchor view. A learnable Scene Tokenizer E_ϕ then aggregates $\mathbf{M}_t$ into a fixed-length scene state:

$$S_t = E_\phi(\mathbf{M}_t). \tag{1}$$

The state is not tied to any single observation viewpoint; we refer to it as a *view-decoupled geometry state*, or GeoState. Given a query camera Q, a query-conditioned geometry decoder predicts

$$\widehat{D}_Q = G_\psi(S_t, Q). \tag{2}$$

Thus, the same GeoState can be queried from different viewpoints without re-encoding the scene.

GeoState world modeling. With viewpoint variation removed from the predictive state, temporal modeling can focus on scene dynamics rather than camera-induced changes in the representation. Given four historical GeoStates $\{S_{t-3}, S_{t-2}, S_{t-1}, S_t\}$, a temporal world model autoregressively predicts future states using a sliding four-state context. Each predicted state $\widehat{S}_{t+h}$ is then queried under a future camera Q_{t+h} to obtain $\widehat{D}_{t+h} = G_\psi(\widehat{S}_{t+h}, Q_{t+h})$. The query camera can be externally specified or provided by a separately trained camera prediction model, whose architecture and training details are given in Appendix A.2.

3.2 View-Decoupled Geometry State

Given synchronized multi-view images $\mathcal{I}_t = \{I_t^v\}_{v\in\mathcal{V}_t}$, let $\mathcal{I}^{\text{in}}$ denote the observations jointly processed by the frozen VGGT-Ω (Wang et al., 2026). To maintain a common coordinate anchor across

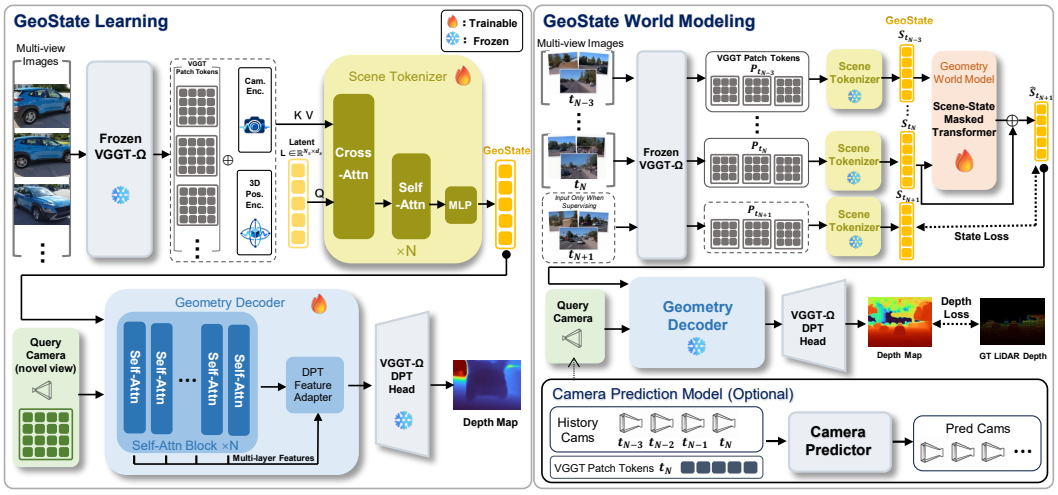

Figure 1: **Overview of GeoState.** GeoState Learning constructs view-decoupled scene states with a scene tokenizer and query-conditioned geometry decoder. GeoState World Modeling predicts future states directly in this representation space.

timestamps, $\mathcal{I}^{\text{in}}$ may include observations from multiple timestamps that share the same first input view. After the joint VGGT-Ω forward pass, however, the Scene Tokenizer uses only the outputs associated with the current synchronized observations $\mathcal{I}_t$:

$$\left\{\{F_t^{v,(\ell)}\}_{\ell\in\{4,11,17,23\}}, X_t^v, C_t^v, \widetilde{Q}_t^v\right\}_{v\in\mathcal{V}_t} = \mathcal{G}_\Omega(\mathcal{I}^{\text{in}})\big|_t\,. \tag{3}$$

Here, $F_t^{v,(\ell)}$ denotes intermediate patch features from the four levels used by the VGGT-Ω dense prediction head, X_t^v and C_t^v denote the pixel-aligned 3D point map and its confidence, respectively, and $\widetilde{Q}_t^v = (\widetilde{K}_t^v, \widetilde{P}_t^v)$ is the GFM-estimated camera. Thus, cross-time observations are used by VGGT-Ω to establish a shared coordinate frame, while each GeoState is constructed only from the outputs corresponding to its own timestamp. For the observed context states, all context observations are processed in a single forward pass; future target states use growing observation prefixes as described in Sec. 3.5. We omit the time index t below for notational simplicity.

Geometry-conditioned GFM tokens. For each patch p in source view v, we project and fuse the GFM features $\{F_{p,v}^{(\ell)}\}_{\ell\in\{4,11,17,23\}}$ with per-level linear projections and an MLP to obtain $e_{p,v}^{\text{feat}} \in \mathbb{R}^{512}$. To facilitate cross-view feature aggregation and reduce scale inconsistency across GFM forwards, following Surflo (Guédon et al., 2026), we augment the GFM patch features with explicit geometric cues derived from X^v, C^v, and $\widetilde{Q}^v$. For each patch, we obtain its 3D location and confidence, source-camera pose, and viewing ray, and encode them as $e_{p,v}^{\text{point}}$, e_v^{pose}, and $e_{p,v}^{\text{ray}}$, respectively, after median-normalizing their explicit spatial scale. These embeddings form the explicit geometric encoding

$$g_{p,v} = [e_{p,v}^{\text{point}}, e_v^{\text{pose}}, e_{p,v}^{\text{ray}}], \tag{4}$$

which is fused with the GFM feature through

$$m_{p,v} = \text{LN}\big(e_{p,v}^{\text{feat}} + \text{MLP}\big([e_{p,v}^{\text{feat}}, g_{p,v}]\big)\big)\,. \tag{5}$$

Scene-state aggregation. To obtain a scene representation independent of the number of views, we collect the geometry-conditioned tokens from all source views into a variable-length memory

$$\mathbf{M}_t = \text{Concat}_{v,p}\, m_{p,v} \in \mathbb{R}^{N_t\times 512}. \tag{6}$$

A Perceiver-style Tokenizer (Jaegle et al., 2021) E_ϕ then aggregates this memory using learned scene queries $\mathbf{L} \in \mathbb{R}^{N_s\times d_s}$:

$$S_t = E_\phi(\mathbf{M}_t; \mathbf{L}) \in \mathbb{R}^{N_s\times d_s}. \tag{7}$$

We use $N_s = d_s = 512$ and share the same learned scene queries across timestamps. The resulting GeoState has a fixed size independent of the number of source views. Detailed encoder architecture is provided in Appendix A.1.1.

3.3 QUERY-CONDITIONED GEOMETRY DECODER

The query viewpoint is introduced only during geometry decoding. Given a query camera $Q = (K, P)$, we construct a Plücker ray field (Sitzmann et al., 2021) from its intrinsics and camera-to-world pose. The rays are expressed in the same coordinate frame and scale as the encoded scene geometry, then Fourier-encoded (Tancik et al., 2020) and patchified into query tokens R_Q.

We first process the scene state with a scene-only self-attention conditioner, then concatenate it with R_Q. A joint Transformer exchanges information among scene and ray tokens to produce view-conditioned representations:

$$Z_Q = \Psi_{\text{query}}(S_t, R_Q), \tag{8}$$

where Ψ_{query} denotes the query Transformer. A feature decoder Φ_{dec} maps Z_Q to the four feature levels expected by the frozen VGGT-Ω dense prediction head H_{GFM}, which produces the depth:

$$\widehat{D}_Q^{\text{norm}} = H_{\text{GFM}}(\Phi_{\text{dec}}(Z_Q)) \,. \tag{9}$$

The normalized prediction is converted back to metric depth using the same context-derived scale convention used by the scene representation. Reusing the frozen GFM head constrains the decoder to recover geometry-compatible features while avoiding relearning dense geometry prediction. Detailed model architectures are provided in Appendix A.1.2.

3.4 SCENE-STATE WORLD MODEL

The world model predicts future evolution directly in GeoState space. Given four consecutive historical states, it predicts a residual update to the latest state:

$$\widehat{S}_{t+1} = S_t + \Delta S_{t+1}, \qquad \Delta S_{t+1} = F_\theta(S_{t-3:t}). \tag{10}$$

We adopt the masked factorized Transformer architecture of FR3D (Morbitzer et al., 2026) for scene-state prediction. A masked prediction state is appended to the four historical states, followed by factorized temporal and scene attention. The predicted state is produced through a residual head, which is zero-initialized so that the model is exactly equivalent to latent Copy-Last at initialization, and outputs ΔS_{t+1} in Eq. 10.

Future GeoStates are generated autoregressively using a sliding four-state context:

$$\widehat{S}_{t+h} = \widetilde{S}_{t+h-1} + F_\theta\Big(\widetilde{S}_{t+h-4:t+h-1}\Big), \qquad h \geq 1, \tag{11}$$

where $\widetilde{S}_\tau = S_\tau$ for observed timestamps $\tau \leq t$ and $\widetilde{S}_\tau = \widehat{S}_\tau$ otherwise. Detailed world-model architecture is provided in Appendix A.1.3.

3.5 LEARNING OBJECTIVES

Geometry-state learning. We train the scene tokenizer and decoder for geometry reconstruction and novel-view querying, keeping the GFM and its dense head frozen. The objective is

$$\mathcal{L}_{\text{rep}} = \mathcal{L}_{\log D} + \lambda_{\text{grad}}\mathcal{L}_{\text{grad}}, \tag{12}$$

where $\mathcal{L}_{\log D}$ measures mean absolute log-depth error over valid pixels, and $\mathcal{L}_{\text{grad}}$ supervises local depth gradients. The learned tokenizer and decoder are frozen during world-model training.

World-model training. We train the world model in two stages: single-step initialization followed by autoregressive rollout training.

Target-state construction. To construct future supervision in a consistent scene coordinate frame, target GeoStates are constructed by feeding growing observation prefixes into the frozen VGGT-Ω. For the target at $t + h$, the prefix terminates at $t + h$, while all prefixes share the same first input view to maintain a common coordinate anchor across time. Only the outputs corresponding to the synchronized observations at $t+h$ are retained and passed to the Scene Tokenizer to construct S^*_{t+h}.

Since VGGT-Ω predictions may vary in scale across forward passes, we normalize the metric spatial quantities in $g_{p,v}$ (Eq. 4) using a calibration derived from the observed context before fusing them

with the GFM features. This gives target GeoStates a consistent context-derived scale without target-frame alignment. Further details are provided in Appendix A.3.

Single-step initialization. We first train F_θ for one-step $4 \to 1$ prediction with joint state and geometric supervision:

$$\mathcal{L}_{\text{init}} = \lambda_{\text{state}}\mathcal{L}_{\text{state}} + \lambda_{\text{depth}}\mathcal{L}_{\log D}(\widehat{D}_{t+1}, D_{t+1}), \tag{13}$$

where $\mathcal{L}_{\text{state}}$ combines Smooth L1 and cosine losses between the predicted state $\widehat{S}_{t+1}$ and the encoded target S^*_{t+1}. The predicted state is also decoded as $\widehat{D}_{t+1} = G_\psi(\widehat{S}_{t+1}, Q_{t+1})$ using the ground-truth future FRONT camera Q_{t+1} and supervised against LiDAR depth D_{t+1}. Thus, state supervision directly guides prediction toward the encoded target, while depth supervision constrains the geometry decoded from the predicted state.

Autoregressive rollout training. We then unroll the world model for $H = 7$ future steps and supervise each predicted state through its decoded geometry. At each step, $\widehat{D}_{t+h} = G_\psi(\widehat{S}_{t+h}, Q_{t+h})$ is decoded using the ground-truth future FRONT camera Q_{t+h}:

$$\mathcal{L}_{\text{roll}} = \frac{1}{H}\sum_{h=1}^{H} \mathcal{L}_{\log D}(\widehat{D}_{t+h}, D_{t+h}), \tag{14}$$

Unlike single-step initialization, we omit direct state matching during rollout, allowing autoregressive states to be optimized for their decoded geometry rather than exact agreement with GFM features. We examine the role of state supervision in both training stages in Section 4.5.

4 Experiments

4.1 Experimental Setup

Datasets. We pretrain the Scene Tokenizer and ray-conditioned decoder on approximately 10k multi-view scenes from DL3DV (Ling et al., 2024), TartanAir (Wang et al., 2020), RealEstate10K (Zhou et al., 2018), CO3D (Reizenstein et al., 2021), and WildRGB-D (Xia et al., 2024), using ground-truth depth where available and VGGT-Ω (Wang et al., 2026) pseudo-depth otherwise. We then adapt both modules on a subset of the Waymo Open Dataset (Sun et al., 2020) selected for static scene content. World Model training uses approximately 140k windows from 800 segments, each containing four context and seven future states at $0.2\,\text{s}$ intervals, with FRONT, FRONT_LEFT, and FRONT_RIGHT context views. We evaluate on 100 Waymo windows held out from model selection, 113 eligible KITTI (Geiger et al., 2012) drives, and all 150 scenes in the nuScenes (Caesar et al., 2020) validation split. Further dataset selection and sampling details are provided in Appendix A.5.

Evaluation Metrics. We report frame-wise mean AbsRel (↓) and δ_1 (↑) (Eigen et al., 2014) over valid pixels. Our method flexibly adapts VGGT-Ω's default balanced resolution preprocessing to both training and evaluation, while baselines follow their original resolution settings. For scale handling, we follow the original evaluation procedure of each baseline: VGGT-World performs per-target-frame scale alignment, while FR3D performs per-target-frame scale-and-shift alignment. For DINO-Foresight, we fit its relative inverse-depth predictions to the LiDAR measurements of each target frame. In contrast, we calibrate metric scale once using LiDAR at the last observed FRONT frame and retain this scale throughout the entire forecasting rollout without target-frame alignment.

We additionally measure cross-query geometric consistency using symmetric reprojection AbsRel. Given decoded depths D_0, D_i and z-buffered reprojection $\mathcal{R}_{a\to b}$, we compute

$$E_{\text{sym}}(Q_0, Q_i) = \frac{1}{2}\big[\,\text{AbsRel}(\mathcal{R}_{0\to i}(D_0), D_i) + \text{AbsRel}(\mathcal{R}_{i\to 0}(D_i), D_0)\,\big]. \tag{15}$$

Lower values indicate better geometric agreement.

Baselines. We compare with DINO-Foresight (Karypidis et al., 2025), which forecasts semantic image features, and geometry-based world models FR3D (Morbitzer et al., 2026) and VGGT-World (Sun et al., 2026). We train DINO-Foresight and VGGT-World on the same Waymo training

Table 1: **Long-horizon depth forecasting on Waymo.** Bold and underlined values indicate the best and second-best results among methods using predicted future cameras.

Method	Camera	Per-frame alignment	t+0.2 s		t+0.6 s		t+1.0 s		t+2.0 s	
			AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
Copy-Last	Fixed View	No	0.094	0.906	0.147	0.840	0.184	0.797	0.248	0.734
DINO-Foresight	Pred.	Yes	0.119	0.841	0.138	0.825	0.151	0.798	0.243	0.663
VGGT-World	Pred.	Yes	<u>0.073</u>	<u>0.931</u>	0.112	0.892	0.143	0.826	0.213	0.731
FR3D	Pred.	Yes	0.110	0.879	0.121	0.858	0.131	0.840	0.152	0.800
Copy-Last	Pred.	No	0.074	0.928	<u>0.088</u>	<u>0.905</u>	<u>0.100</u>	<u>0.882</u>	<u>0.132</u>	<u>0.827</u>
GeoState	Pred.	No	**0.065**	**0.939**	**0.079**	**0.921**	**0.092**	**0.902**	**0.127**	**0.848**
Copy-Last	GT	No	0.071	0.929	0.082	0.913	0.092	0.894	0.108	0.865
GeoState	GT	No	0.062	0.940	0.071	0.929	0.081	0.915	0.099	0.888

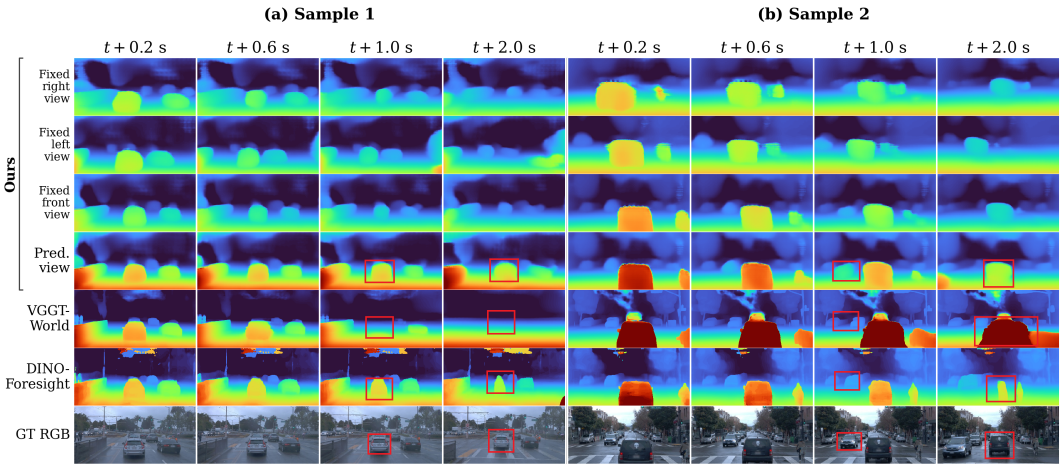

Figure 2: **Qualitative multi-view depth forecasting.** Red boxes highlight prediction differences.

data as our method, but following their respective training procedures. All FR3D results are taken directly from the original paper.

We also include three *Copy-Last* baselines. *Fixed View* repeats the last observed VGGT-Ω depth at every future timestamp, while *GT Camera* and *Pred. Camera* decode the last observed scene state from ground-truth or predicted future cameras. These baselines allow us to assess the effects of future camera motion and scene-state prediction.

Implementation Details. We pretrain the Scene Tokenizer and ray-conditioned decoder on the multi-view pretraining data for 200k steps with an initial learning rate of 10^{-5}, and then finetune on the static Waymo subset for 50k steps with an initial learning rate of 5×10^{-6}. Both stages use a global effective batch size of 36. We sample one to five source views per scene and supervise depth reconstruction at held-out query views. For static Waymo sequences, source and query views may span both cameras and timestamps. We then train the World Model in two stages: 20k steps of one-step prediction with an initial learning rate of 10^{-5}, followed by 35k steps of seven-step autoregressive rollout training with an initial learning rate of 10^{-6}. VGGT-Ω, Tokenizer, and the decoder remain frozen throughout World Model training. Additional details are provided in Appendix A.4.

4.2 Long-Horizon Depth Forecasting on Waymo

Table 1 compares long-horizon depth forecasting performance on Waymo. With predicted cameras, our model consistently outperforms all compared methods on both depth metrics at every reported horizon. The advantage over VGGT-World increases with the forecasting horizon, reaching a 40.4% reduction in AbsRel at $2.0\,$s. These improvements are achieved while maintaining a single context-derived scale without per-frame alignment and persist beyond the $1.4\,$s rollout horizon used

	Small		Medium		Large	
Δt (s)	World	Enc.	World	Enc.	World	Enc.
0.2	0.023	0.024	0.054	0.055	0.089	0.091
0.6	0.022	0.023	0.053	0.055	0.088	0.091
1.0	0.022	0.024	0.054	0.055	0.089	0.090
2.0	0.024	0.024	0.058	0.055	0.096	0.091

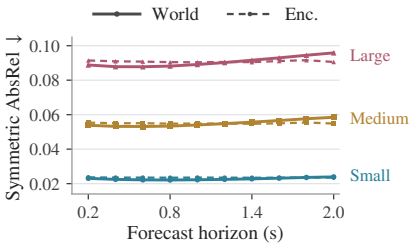

Figure 3: **Cross-query geometric consistency on Waymo.** We evaluate symmetric reprojection AbsRel (↓) across perturbation levels.

Table 2: **Zero-shot depth forecasting on KITTI and nuScenes.** Bold and underlined values indicate the best and second-best results among methods using predicted cameras.

Method	Camera	KITTI						nuScenes					
		t+0.6 s		t+1.0 s		t+2.0 s		t+0.75 s		t+1.25 s		t+2.5 s	
		AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
DINO-Foresight	Pred.	0.131	0.849	0.161	0.793	0.195	0.732	0.231	0.691	0.253	0.665	0.291	0.602
VGGT-World	Pred.	0.109	0.886	0.127	0.862	0.198	0.732	0.228	0.715	0.241	0.693	0.282	0.637
FR3D	Pred.	0.116	0.868	0.132	0.835	0.178	0.758	0.182	0.740	0.197	0.710	0.229	<u>0.660</u>
Copy-Last	Pred.	<u>0.097</u>	<u>0.913</u>	<u>0.113</u>	<u>0.878</u>	<u>0.160</u>	<u>0.802</u>	<u>0.142</u>	<u>0.804</u>	<u>0.177</u>	<u>0.751</u>	<u>0.221</u>	0.646
GeoState	Pred.	**0.095**	**0.920**	**0.112**	**0.883**	**0.155**	**0.821**	**0.136**	**0.820**	**0.172**	**0.764**	**0.213**	**0.666**
Copy-Last	GT	0.088	0.932	0.095	0.915	0.114	0.882	0.131	0.825	0.164	0.786	0.195	0.697
GeoState	GT	0.086	0.939	0.093	0.918	0.108	0.895	0.125	0.840	0.159	0.797	0.190	0.707

during training. Together, these results demonstrate that GeoState supports effective long-horizon autoregressive geometry forecasting, including at horizons beyond those seen during training. Appendix A.8.2 further reports results on the Dynamic and Static subsets. Figure 2 visualizes depth forecasts across future horizons and query views, alongside comparisons with prior methods.

The Copy-Last variants further disentangle the effects of camera conditioning and scene-state evolution. Re-querying the last observed state with future cameras substantially outperforms simply repeating the last observed depth map, demonstrating that viewpoint-induced changes can be accounted for by querying a fixed scene state from future viewpoints without evolving the state itself. Our predicted states consistently outperform their camera-matched Copy-Last counterparts at every horizon under both GT and predicted cameras, showing that the World Model captures future scene evolution beyond what can be explained by camera re-querying alone.

4.3 Cross-Query Geometric Consistency

We evaluate whether a predicted scene state remains geometrically consistent when queried from different viewpoints. At each future timestamp, we decode the same state from the ground-truth FRONT camera Q_0 and perturbed virtual cameras Q_i, jointly varying translation and rotation. We consider Small (0–0.5 m, 0–2°), Medium (0.5–1.0 m, 2–5°), and Large (1.0–1.5 m, 5–10°) perturbations. Consistency is measured by symmetric reprojection AbsRel E_{sym} in Eq. 15 over valid overlap. We compare World Model-predicted states (World) with target-time states encoded from RGB observations (Enc.). Further protocol details are provided in Appendix A.6.

Figure 3 shows that world-model-predicted states closely match observation-encoded states across all perturbation levels at every forecast horizon. This indicates that although geometric supervision is provided only through depth, the model learns a scene state that captures the underlying geometric structure of the world and supports consistent querying from unseen cameras.

4.4 Zero-Shot Generalization

Table 2 evaluates zero-shot depth forecasting on KITTI and nuScenes. Without target-dataset adaptation, our model with predicted cameras outperforms DINO-Foresight, VGGT-World, and the reported FR3D results on both metrics across all reported horizons on both datasets. These results

Table 3: **Geometric reconstruction on Waymo.**

Method	t+0.2 s		t+0.6 s		t+1.0 s		t+2.0 s	
	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
VGGT-Ω	0.074	0.934	0.079	0.932	0.076	0.932	0.078	0.930
GeoState Reconstruction	0.068	0.934	0.073	0.932	0.072	0.932	0.072	0.929

Table 4: **Representation ablation.**

Mode	**State**	t+0.6 s		t+1.0 s		t+2.0 s	
		AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
Rec.	Compressor	0.075	0.930	0.073	0.930	0.074	0.928
	Ours	0.073	0.932	0.072	0.932	0.072	0.929
Pred.	Compressor	0.092	0.896	0.111	0.863	0.144	0.804
	Ours	**0.081**	**0.919**	**0.095**	**0.896**	**0.133**	**0.834**

Table 5: **State supervision ablation.**

One-step	**Rollout**	t+0.6 s		t+1.0 s		t+2.0 s	
		AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
×	✓	0.077	0.919	0.086	0.902	0.108	0.878
×	×	0.075	0.922	0.084	0.909	0.104	0.879
✓	✓	0.071	0.928	0.082	0.913	0.107	0.879
✓	×	**0.071**	**0.929**	**0.081**	**0.915**	**0.099**	**0.888**

support that GeoState captures transferable geometric structure. Moreover, our predicted states consistently outperform camera-matched Copy-Last under both ground-truth and predicted cameras, showing that the learned scene dynamics also generalize beyond the training domain.

4.5 Ablation Studies

Geometric Reconstruction Quality. We first examine whether compressing GFM features into GeoState preserves their geometric information. Table 3 compares direct VGGT-Ω depth predictions with depth reconstructed through our Scene Tokenizer and Geometry Decoder using ground-truth query cameras. Both use target-time RGB observations and are evaluated against ground-truth LiDAR depth. Their comparable accuracy shows that the fixed-length GeoState preserves the geometric information required for accurate reconstruction, providing a compact and geometrically faithful representation for subsequent world modeling.

Effect of the View-Decoupled Scene State. We compare GeoState with a feature-compression baseline that directly compresses VGGT-Ω features into a 512×512 latent state. For a fair comparison with the feature-compression baseline, only the target FRONT GFM output block is passed to the Scene Tokenizer. Both representations use the same world model, training data, forecasting objectives, optimization, and evaluation protocol; implementation details are provided in Appendix A.5.4. As shown in Table 4, the two representations achieve comparable reconstruction quality, but GeoState is substantially easier to predict, improving AbsRel and δ_1 at every horizon.

Because the two representations achieve comparable reconstruction quality, the forecasting gap is unlikely to arise from reconstruction accuracy. Instead, the results suggest that GeoState provides a more predictable state space for temporal modeling. By separating scene content from viewpoint-dependent observation, GeoState reduces camera-induced variation in the state dynamics, allowing the World Model to focus more directly on the underlying evolution of scene geometry.

Effect of State Supervision. We ablate state supervision during one-step initialization and autoregressive rollout while retaining decoded-depth supervision in all settings. Table 5 shows that one-step state supervision consistently improves forecasting, providing a useful target for initializing local state transitions. In contrast, extending latent matching throughout the rollout degrades long-horizon performance. This supports supervising autoregressive states through their decoded geometry rather than enforcing a unique latent trajectory; further analysis is provided in Appendix A.7.

5 Conclusion

We presented GeoState, a view-decoupled geometry state learned from GFM features for predictive 3D world modeling. By separating scene evolution from camera-dependent observation, GeoState provides a scene representation that can be predicted more effectively over time and queried from different viewpoints. Future work may extend this framework to embodied settings, where predictive scene states could support perception, planning, and interaction in dynamic environments.

AI USE STATEMENT

Generative AI tools were used in a supporting role during the preparation and execution of this work. Specifically, they were used to improve the clarity and language of the manuscript, assist with literature retrieval, and provide assistance with code implementation and debugging. The core research idea, methodological design, experimental setup, analysis, and scientific conclusions were developed by the authors rather than generated by AI. All AI-assisted text, code, and retrieved information were reviewed and verified by the authors. The authors take full responsibility for the content and results presented in this paper.

REPRODUCIBILITY STATEMENT

We have made extensive efforts to ensure the reproducibility of our work. Detailed architectural specifications of GeoState are provided in Appendix A.1, while the implementation and training details of the camera predictor are described in Appendix A.2. The coordinate and scale conventions used throughout the method and evaluation are documented in Appendix A.3. Complete training configurations and implementation details are provided in Appendix A.4, while the evaluation-set construction, random seed, and evaluation protocols are documented in Appendix A.5. Upon acceptance, we will release the code together with the trained models to facilitate reproduction and further research.

REFERENCES

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.html.

Junyi Chen, Haoyi Zhu, Xianglong He, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Zhoujie Fu, Jiangmiao Pang, and Tong He. DeepVerse: 4D Autoregressive Video Generation as a World Model. *arXiv preprint arXiv:2506.01103*, 2025. URL https://arxiv.org/abs/2506.01103.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://papers.nips.cc/paper_files/paper/2014/hash/91c56ce4a249fae5419b90cba831e303-Abstract.html.

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. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.17398.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. URL https://www.cvlibs.net/datasets/kitti/.

Antoine Guédon, Shu Nakamura, Nicolas Dufour, Jiahui Lei, Ko Nishino, and Angjoo Kanazawa. Surflo: Consistent 3D Surface Flow Model with Global State. *arXiv preprint arXiv:2606.13644*, 2026. URL https://arxiv.org/abs/2606.13644.

David Ha and Jürgen Schmidhuber. World Models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

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*, 2023. URL https://arxiv.org/abs/2309.17080.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General Perception with Iterative Attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664. PMLR, 2021. URL https://proceedings.mlr.press/v139/jaegle21a.html.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A Large View Synthesis Model with Minimal 3D Inductive Bias. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QQBPWtvtcn.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. DINO-Foresight: Looking into the Future with DINO. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=gimtybo07H.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A Large-Scale Scene Dataset for Deep Learning-based 3D Vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Ling_DL3DV-10K_A_Large-Scale_Scene_Dataset_for_Deep_Learning-based_3D_Vision_CVPR_2024_paper.html.

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

Nils Morbitzer, Jonathan Evers, Artem Savkin, Thomas Stauner, Nassir Navab, Federico Tombari, and Stefano Gasperini. Future dynamic 3d reconstruction: Toward 3d world modeling with disentangled ego-motion. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2026. URL https://arxiv.org/abs/2606.18250.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai, 2025. URL https://arxiv.org/abs/2501.03575.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common Objects in 3D: Large-Scale Learning and Evaluation of Real-Life 3D Category Reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10901–10911, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Reizenstein_Common_Objects_in_3D_Large-Scale_Learning_and_Evaluation_of_Real-Life_ICCV_2021_paper.html.

Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light Field Networks: Neural Scene Representations with Single-Evaluation Ren-

dering. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2106.02634.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Sun_Scalability_in_Perception_for_Autonomous_Driving_Waymo_Open_Dataset_CVPR_2020_paper.html.

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. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari (eds.), *Computer Vision – ECCV 2026*, pp. 382–400, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-37271-0. URL https://link.springer.com/chapter/10.1007/978-3-032-37271-0_21.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.nips.cc/paper_files/paper/2020/hash/55053683268957697aa39fba6f231c68-Abstract.html.

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*, pp. 5294–5306, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2605.15195.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D Perception Model with Persistent State. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10510–10522, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_Continuous_3D_Perception_Model_with_Persistent_State_CVPR_2025_paper.html.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wang_DUSt3R_Geometric_3D_Vision_Made_Easy_CVPR_2024_paper.html.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916, 2020. doi: 10.1109/IROS45743.2020.9341801. URL https://ieeexplore.ieee.org/document/9341801/.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards Real-World-Drive World Models for Autonomous Driving. In *Computer Vision – ECCV 2024*, volume 15106 of *Lecture Notes in Computer Science*, pp. 55–72. Springer, 2024b. doi: 10.1007/978-3-031-73195-2_4. URL https://link.springer.com/chapter/10.1007/978-3-031-73195-2_4.

Dongxu Wei, Qi Xu, Zhiqi Li, Hangning Zhou, Cong Qiu, Hailong Qin, Mu Yang, Zhaopeng Cui, and Peidong Liu. Any 3D Scene is Worth 1K Tokens: 3D-Grounded Representation for Scene Generation at Scale. *arXiv preprint arXiv:2604.11331*, 2026. URL https://arxiv.org/abs/2604.11331.

Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. RGBD Objects in the Wild: Scaling Real-World 3D Object Learning from RGB-D Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22378–22389, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Xia_RGBD_Objects_in_the_Wild_Scaling_Real-World_3D_Object_Learning_CVPR_2024_paper.html.

Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D Reconstruction of 1000+ Images in One Forward Pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21924–21935, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_Fast3R_Towards_3D_Reconstruction_of_1000_Images_in_One_Forward_CVPR_2025_paper.html.

Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual Point Cloud Forecasting enables Scalable Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14673–14684, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Visual_Point_Cloud_Forecasting_enables_Scalable_Autonomous_Driving_CVPR_2024_paper.html.

Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. Copilot4D: Learning Unsupervised World Models for Autonomous Driving via Discrete Diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Psl75UCoZM.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 55–72, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-72624-8. URL https://link.springer.com/chapter/10.1007/978-3-031-72624-8_4.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo Magnification: Learning View Synthesis using Multiplane Images. In *SIGGRAPH*, 2018. URL https://tinghuiz.github.io/projects/mpi/.

Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-Aware Unified World Modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8535–8546, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhu_Aether_Geometric-Aware_Unified_World_Modeling_ICCV_2025_paper.html.

Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. GaussianWorld: Gaussian World Model for Streaming 3D Occupancy Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6772–6781, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zuo_GaussianWorld_Gaussian_World_Model_for_Streaming_3D_Occupancy_Prediction_CVPR_2025_paper.html.

A Appendix

A.1 Architecture Details

This section provides the implementation details of the Scene Tokenizer, geometry decoder, and world model used in the main experiments. Table 6 summarizes their principal architectural configurations and parameter counts.

A.1.1 Geometry-Conditioned Memory and Scene Tokenizer

Each selected GFM feature level is independently normalized and projected from 2,048 to 512 channels. The four projected features are concatenated and fused by an MLP with dimensions $2048 \to 2048 \to 512$. For each spatial patch, confidence-weighted pooling of the dense point map provides a representative 3D point. Its coordinates, normalized by the scene scale, are Fourier-encoded with ten bands and concatenated with $\log(1 + c)$, where c is the pooled confidence, before projection to a 512-dimensional geometry embedding.

Source-camera conditioning uses a six-dimensional rotation representation formed from the first two rotation-matrix columns and a four-band Fourier encoding of normalized translation. A camera MLP produces a 512-dimensional embedding shared by all patches in that view. The source-ray branch encodes six-dimensional Plücker coordinates with four Fourier bands and applies a convolution with kernel size and stride 16, followed by LayerNorm, to produce one 512-dimensional embedding per patch.

The feature, point, camera, and ray embeddings are concatenated. A $2048 \to 1024 \to 512$ fusion MLP predicts a correction to the feature embedding; their sum is normalized to form the patch memory. A bank of 512 learned queries of width 512 is updated by eight Perceiver layers, each comprising latent-to-memory cross-attention and latent self-attention. Each attention block uses eight heads, pre-normalization, residual connections, and a feed-forward network with expansion ratio four. A final LayerNorm produces $S \in \mathbb{R}^{512\times512}$.

A.1.2 Query-Conditioned Geometry Decoder

For a camera-to-world pose $P = (R, \mathbf{o})$ and intrinsic matrix K, a pixel $\mathbf{u} = (u, v, 1)^\top$ defines the unit direction

$$\mathbf{d}(\mathbf{u}) = \frac{RK^{-1}\mathbf{u}}{\|RK^{-1}\mathbf{u}\|_2}. \tag{16}$$

In normalized scene coordinates, the ray representation is $[\mathbf{d}, \bar{\mathbf{o}} \times \mathbf{d}]$, where $\bar{\mathbf{o}}$ is the normalized camera center. Query rays are Fourier-encoded with six bands and patchified by a stride-16 convolution. At 416×624 resolution, this produces $26 \times 39 = 1{,}014$ query tokens, which are concatenated with the 512 scene tokens to form a sequence of 1,526 tokens.

The scene tokens first pass through LayerNorm, a linear projection, and two scene-only self-attention blocks. The resulting scene tokens and query-ray tokens are then processed jointly by sixteen self-attention Transformer blocks, allowing each query to retrieve view-dependent geometry from the shared scene state. Query-token representations are retained after blocks 4, 8, 12, and 16, providing multi-layer features at four stages of the decoder.

These multi-layer features are subsequently processed by a *DPT Feature Adapter*, whose role is to transform the decoder representations into the feature interface expected by the frozen VGGT-Ω DPT head. For each level, the final query representation is fused with the corresponding intermediate representation through a learned sigmoid gate and a level embedding. Two additional Perceiver layers allow the query tokens to attend to the final scene tokens, and an independent MLP maps each level as $512 \to 1024 \to 1024 \to 2048$.

The resulting four feature maps are passed directly to the frozen VGGT-Ω DPT head to predict camera z-depth. The decoder receives only the scene state, query camera, and output dimensions, such that all scene information must pass through the fixed-size state bottleneck. During world-model training, both the Geometry Decoder and DPT head remain fixed, while gradients are propagated through them with respect to the input scene state.

Table 6: Detailed architecture of the model.

Component	Configuration
Scene Tokenizer	512 queries of width 512; eight Perceiver layers; eight attention heads; MLP expansion ratio 4; 70.46M parameters
Geometry decoder	Width 512; two scene-conditioning blocks; 16 joint scene–ray blocks; four output branches with two Perceiver layers each; 135.55M parameters
World Model	Four-state context; 12 factorized blocks; width 1,152; eight heads; MLP expansion ratio 4; dropout 0.1; 256.55M parameters

A.1.3 Residual World Model

The four observed scene states form an input tensor of shape $4 \times 512 \times 512$, corresponding to four time steps, 512 spatial slots per state, and 512 feature channels per slot. Each token is first linearly projected from 512 to a model width of 1,152. A learned mask token is then replicated over 512 slots to construct a fifth state, which serves as the prediction position. To encode token identity, we use learned temporal and slot embeddings of 576 dimensions each. Their concatenation forms a 1,152-dimensional positional embedding, which is added to the projected tokens.

Each Transformer block factorizes attention over time and space. Temporal attention is first applied across the five time positions independently for each of the 512 slots. Spatial attention then operates across the 512 slots independently within each time position. The two attention operations are followed by a $1152 \to 4608 \to 1152$ feed-forward network with GELU. All sublayers use pre-normalization, residual connections, and dropout. After the final block, the tokens at the fifth, masked prediction position are mapped back to 512 channels by a bias-free linear residual head. The head is zero-initialized, such that the model initially predicts a zero residual and therefore exactly recovers the Copy-Last prediction.

A.2 Camera Predictor

The camera predictor forecasts future FRONT-camera poses from the observed ego-motion and the current FRONT image. The predicted trajectory is used as the future query-camera sequence for both GeoState (Pred. Camera) and Copy-Last (Pred. Camera).

A.2.1 Architecture

The camera predictor contains 8.99M parameters and follows a motion-prior-plus-visual-refinement design. Given four historical metric FRONT poses at $t-0.6$, $t-0.4$, $t-0.2$, and t seconds, we first compute the three consecutive relative transformations and map them to six-dimensional motion vectors using the $\mathrm{SE}(3)$ logarithm. The motion vectors and their first differences are standardized, projected to width 256, and encoded by a four-layer Transformer with eight attention heads, an MLP expansion ratio of four, and dropout 0.1. Future poses are initialized from a constant-body-twist extrapolation of the latest observed motion. Learned horizon queries attend to the encoded motion features through three decoding layers and predict residual corrections to this extrapolated trajectory. Future time is represented using linear and quadratic terms together with eight Fourier bands.

To incorporate scene-dependent cues, the predictor additionally conditions on the current FRONT image. Frozen VGGT-Ω spatial features are extracted from a separate forward pass and pooled to a 6×8 grid, yielding 48 tokens of width 2,048. After projection to width 256 and the addition of learned spatial embeddings, two cross-attention layers compress these features into eight visual tokens. The visual tokens are appended to the encoded motion memory, and two additional horizon-query decoding layers, with eight attention heads and an MLP expansion ratio of two, predict a six-dimensional visual correction $\Delta\xi_h$ for each requested horizon.

For each requested horizon h, the final relative pose is

$$\widehat{P}_{t+h} = \widehat{P}^{\mathrm{motion}}_{t+h} \, \mathrm{Exp}_{\mathrm{SE}(3)}(\Delta\xi_h). \tag{17}$$

Table 7: Camera prediction results on Waymo.

Horizon (s)	ATE (m) ↓	RTE (m) ↓	RRE (°) ↓
0.2	0.0125	0.0125	0.1894
0.6	0.0379	0.0246	0.2462
1.0	0.0840	0.0361	0.2783
2.0	0.3117	0.0785	0.3755

The visual correction is bounded by a scaled $\tanh$ output, with a limit of 2.0 in normalized residual coordinates and horizon-dependent motion scales. Its final projection is initialized to zero, so the complete model initially reduces to the motion-only prediction and learns visual corrections during training. At inference, the four observed poses and the current FRONT image jointly condition predictions at all ten future horizons in a single forward pass.

A.2.2 Training

We train the camera predictor in three stages. First, we optimize the motion backbone using metric FRONT-camera trajectories from Waymo. Each training sample contains four observed poses at $t-0.6$, $t-0.4$, $t-0.2$, and t, together with ten future poses at $0.2\,\mathrm{s}$ intervals. The objective combines absolute pose error, adjacent relative-pose error, and ray-endpoint error with weights 1.0, 0.5, and 0.2, respectively. Rotation errors are converted to metric displacement using a $10\,\mathrm{m}$ lever arm. We first train for 12,000 steps with direct ten-horizon supervision using AdamW, batch size 1,024, weight decay 0.05, and a cosine learning-rate schedule from 10^{-4} to 10^{-6} after 300 warm-up steps. We then continue for another 12,000 steps with closed-loop training, recursively applying the predictor over four consecutive rollout chunks. The closed-loop loss is weighted by 0.5. This phase uses batch size 384, a 100-step warm-up, and a cosine learning-rate schedule from 5×10^{-5} to 5×10^{-7}. Gradients are clipped to norm 1.0 throughout motion training.

Second, we train the visual adapter while keeping the motion backbone frozen. We use the same pose-based supervision, with the relative-pose and ray-endpoint terms both weighted by 0.5 and a $20\,\mathrm{m}$ rotation lever arm. Turning and speed-changing windows are assigned sample weights of 1.5 and 0.5, respectively. Training uses 123,789 windows for 20 epochs with AdamW, batch size 256, weight decay 0.02, and a cosine learning-rate schedule from 2×10^{-4} to 2×10^{-6}. Gradients are clipped to norm 1.0.

Finally, we adapt the visual adapter to the downstream depth-query task through the frozen geometry decoder. The last observed GeoState S_t is held fixed across future horizons and decoded using the predicted future camera poses. The decoded depth is supervised against future Waymo LiDAR. To account for the horizon-dependent difficulty of querying a fixed scene state, we normalize the AbsRel at each horizon by the corresponding GT-camera Copy-Last AbsRel estimated over the training set. This depth objective is combined with the trajectory loss and a parameter-anchor regularizer, weighted by 0.05 and 0.02, respectively. Only the visual adapter is updated in this stage. We train for 1,200 steps using AdamW with learning rate 5×10^{-6}, weight decay 0.01, and gradient clipping at 1.0.

A.2.3 Evaluation Results

We evaluate on the same Waymo validation windows as depth forecasting. Predicted and GT camera-to-world poses share the metric frame of the last observed FRONT camera, with no post-hoc trajectory alignment. At each reported horizon, errors are pooled over all 100 sequences and all future steps up to that horizon before computing RMSE. ATE measures camera-center error. RTE and RRE measure the translation magnitude and rotation angle of $(T_{k-1}^{-1} T_k)^{-1} (\widehat{T}_{k-1}^{-1} \widehat{T}_k)$ over adjacent $0.2\,\mathrm{s}$ transitions, including the first transition from the observed anchor. Table 7 reports the results.

A.3 Coordinate and Scale Conventions

GFM reconstruction is defined up to an unknown scale. For each temporal window, we therefore calibrate its scale using the latest observed FRONT view, which serves as the anchor. Let D_0^{Ω} and

D_0^{LiDAR} denote the raw GFM and LiDAR depths at the anchor, respectively, and let Ω_0 denote the set of valid pixels with matched GFM and LiDAR depths. We estimate

$$a_0 = \exp\left[\mathrm{median}_{p\in\Omega_0}\left(\log D_0^{\Omega}(p) - \log D_0^{\mathrm{LiDAR}}(p)\right)\right], \tag{18}$$

where a_0 converts metric distances to GFM units. Matched pixels are required to have finite and positive GFM depth, LiDAR depth, and GFM confidence, and LiDAR depth is restricted to 80 m. Calibration is performed only when at least 128 valid matches are available.

We additionally define a scene-normalization depth s_0 from the observed views as a confidence-weighted trimmed median of valid GFM depths. The lowest and highest 2% of cumulative confidence weight are discarded before computing the weighted median. The physical length represented by one normalized scene unit is therefore

$$L_0 = \frac{s_0}{a_0} \quad \text{meters.} \tag{19}$$

Point coordinates, camera translations, and ray origins are divided by s_0 in the reference GFM frame, while rotations and unit ray directions remain unchanged. Plücker moments are constructed using the normalized ray origins.

All four observed states within a temporal window are extracted from the same 12-image GFM forward pass and therefore share the same a_0, s_0, and L_0. A normalized decoder prediction is converted to metric depth as

$$\widehat{D}_Q^{\mathrm{metric}} = L_0 \widehat{D}_Q^{\mathrm{norm}}. \tag{20}$$

This conversion is fixed by the observed context and retained throughout the entire forecasting rollout; no future target frame is used to re-estimate the scale.

When geometry is obtained from a separate GFM forward pass, its raw scale may differ from that of the observed context. Let a_k denote the metric calibration factor of such a pass. Setting

$$s_k = a_k L_0 \tag{21}$$

expresses its geometry in the same normalized physical unit, since

$$\frac{\mathbf{x}_{\mathrm{GFM},k}}{s_k} \approx \frac{\mathbf{x}_{\mathrm{metric}}}{L_0}. \tag{22}$$

This normalization applies only to explicit geometric quantities (Eq. 4); the intermediate GFM feature vectors are not rescaled.

A.4 Training Details

In this section, we provide the sampling procedures, loss implementations, and optimization settings that supplement Sections 3.5 and 4.1.

A.4.1 Waymo Data Construction

We use a binary motion partition for Waymo. Motion labels are derived from the annotated 3D tracks of vehicles, pedestrians, and cyclists over each eleven-state window (four observed and seven future states at 0.2 s intervals). A box observation must contain at least three LiDAR points and a track must be observed in at least two states. We mark a track as moving when its median speed exceeds 0.75 m/s for a vehicle, 0.25 m/s for a pedestrian, or 0.50 m/s for a cyclist. A window is Dynamic if it contains at least one such visible moving track and Static otherwise. This partition yields 114,735 Dynamic and 24,632 Static training windows.

A.4.2 Scene Tokenizer and Decoder Training

Multi-view pretraining uses 9,496 scenes, comprising 2,834 from DL3DV, 2,818 from TartanAir, 2,142 from RealEstate10K, 941 from CO3D, and 761 from WildRGB-D. To train the tokenizer with different visible-view configurations, anchor-view and random-view dropout each use probability 0.4 at the memory level. For each training sample with at least two source views, we apply one of three mutually exclusive memory-level operations: mask the anchor-view patch block with probability 0.4, mask one uniformly sampled non-anchor view block with probability 0.4, or retain all view blocks with probability 0.2. Single-view samples retain their only view.

During Waymo adaptation, we use the Static subset described in Section A.4.1. Since these windows contain no visible moving tracks under our motion criterion, we allow temporal source–query separations up to 2.0 s while the observation and query cameras move. This sampling covers both nearby and more distant query viewpoints, so that viewpoints far from the anchor remain within the training distribution. The adaptation schedule uses a 2k-step warmup followed by cosine learning-rate decay; training budgets, initial learning rates, and effective batch sizes follow Section 4.1. For the geometry objective defined in Eq. 12, we set the gradient-loss weight to 0.1, yielding

$$\mathcal{L}_{\text{rep}} = \mathcal{L}_{\log D} + 0.1\mathcal{L}_{\text{grad}}. \tag{23}$$

A.4.3 World-Model Training

World Model training uses 139,367 windows from 798 usable Waymo training segments covering static and dynamic content. Each window contains four context states and seven future states at 0.2 s intervals, corresponding to a forecast horizon of 1.4 s from the last context state. For one-step initialization, each window supplies seven $4 \to 1$ transitions with observation-encoded histories.

The concrete loss corresponding to Eq. 13 is

$$\mathcal{L}_{\text{init}} = \text{SmoothL1}_{0.1}(\widehat{S}, S^*) + 0.05\left[1 - \cos\Big(\text{vec}\,\widehat{S}, \text{vec}\,S^*\Big)\right] + 0.24\mathcal{L}_{\log D}. \tag{24}$$

The SmoothL1 transition parameter is 0.1, and cosine similarity is computed between flattened states. During autoregressive training, Eq. 14 weights the FRONT log-depth losses equally across horizons. Each window requires one GFM forward pass on its twelve context images; future LiDAR depth supplies the reconstruction targets.

We use AdamW with $\beta = (0.9, 0.95)$, weight decay 0.05, gradient clipping at 1.0, and learning-rate warmup followed by cosine decay. All experiments are conducted on four NVIDIA RTX 5090 GPUs.

A.5 Data and Evaluation Protocols

A.5.1 Datasets and Temporal Sampling

We construct one candidate window per sequence. Each window contains four historical observations ending at anchor t and ten future targets. Eligibility is determined solely by the availability of the required RGB images, camera poses, and LiDAR measurements, together with timestamp matching. Using a fixed temporal position and at most one window per sequence prevents longer sequences from receiving greater weight.

For Waymo, the candidate pool contains one window from each of the 202 segments in the official validation split. We define logical states at 5 Hz and place the anchor at the tenth logical state, 1.8 s after the first complete timestamp. A complete timestamp contains FRONT, FRONT_LEFT, and FRONT_RIGHT images together with the vehicle pose. The four observations occur at $t - 0.6$, $t - 0.4$, $t - 0.2$, and t, and the ten targets span $t + 0.2$ through $t + 2.0$ s. For each nominal time, we select the nearest complete timestamp within 20 ms. We partition the 202 segments into 102 segments used for checkpoint selection and 100 held-out segments reserved for final evaluation. The evaluation subset is sampled uniformly without replacement using seed 20260804, after which its segment identities and anchors are fixed for all paired comparisons. No segment in the held-out evaluation subset is used for checkpoint selection. This fixed sample contains 79 Dynamic and 21 Static windows under the partition defined in Appendix A.4.1.

For KITTI, we consider the 151 sequences covered by the KITTI Depth Completion annotations and construct at most one fixed window per sequence. Logical states follow the VGGT-World two-raw-frame stride, with the anchor placed at the tenth logical state, corresponding to raw-frame offset 18 from the first image. We retain a drive when all four historical and ten future RGB, pose, and LiDAR records are available on this stride, yielding 113 eligible drives. Ineligible drives are recorded as skipped and are not replaced. This common anchor placement keeps the temporal position comparable to Waymo while respecting KITTI's native sampling cadence.

For nuScenes, we construct one window for each of the 150 scenes in the official validation split. Because CAM_FRONT is sampled asynchronously, the anchor is defined as the image nearest to

1.8 s after the scene start, subject to a 120 ms tolerance, rather than as the tenth stored image. Historical observations occur at $t - 0.75, t - 0.50, t - 0.25, t$, and ten future targets follow at 0.25 s intervals. Each target is paired with the nearest LIDAR_TOP sweep within 50 ms. This construction preserves a comparable fixed temporal anchor across datasets while accommodating the native nuScenes sensor timing.

A.5.2 Input and Query Construction

Our method follows the balanced-resolution preprocessing rule of VGGT-Ω. It preserves the nominal patch-token budget while adapting the height and width to the source aspect ratio and rounding both dimensions to multiples of the 16-pixel patch size; a centered crop is applied only when needed to limit extreme aspect ratios. The resulting evaluation rasters (height×width) are 416×624 for Waymo, 368×720 for KITTI, and 384×688 for nuScenes. Camera intrinsics are updated using the same resize factors and crop offsets. External baselines retain their respective preprocessing conventions.

At each Waymo observation time, we use synchronized FRONT, FRONT_LEFT, and FRONT_RIGHT images. VGGT-Ω jointly processes all twelve historical images, placing the latest FRONT image first to define a shared reference frame. A timestamp-specific memory mask exposes only the three view blocks belonging to each state to the Scene Tokenizer. The four states are then supplied to the World Model in chronological order. KITTI and nuScenes use one camera view per timestamp.

The World Model predicts ten successive states with a four-state sliding window. GT-camera and predicted-camera evaluations share this state trajectory and differ in the decoder query poses. The camera predictor receives four observed metric poses and the current forward-facing image, as detailed in Appendix A.2. Within each camera setting, Copy-Last and forecasting share query cameras and LiDAR evaluation masks.

Camera-matched Copy-Last holds S_t fixed and evaluates $G_\psi(S_t, Q_{t+h})$, retaining the representation, decoder, and query trajectory used by forecasting. Fixed-View Copy-Last repeats the raw VGGT-Ω FRONT depth map from the observed anchor at every future timestamp. The map remains fixed while the future LiDAR reference changes.

A.5.3 Depth, Scale, and Metric Computation

Depth is evaluated against projected LiDAR measurements in the query-camera frame. For a transformed point (x_c, y_c, z_c), the reference depth is z_c. The valid set Ω contains projected measurements with finite depth satisfying $0 < D \leq 80\,\text{m}$. Predictions and references are compared at corresponding pixels of the evaluation raster:

$$\text{AbsRel}(\widehat{D}, D) = \frac{1}{|\Omega|} \sum_{p \in \Omega} \frac{|\widehat{D}(p) - D(p)|}{D(p)}, \tag{25}$$

$$\delta_1(\widehat{D}, D) = \frac{1}{|\Omega|} \sum_{p \in \Omega} \mathbf{1}\left[\max\left(\frac{\widehat{D}(p)}{D(p)}, \frac{D(p)}{\widehat{D}(p)} \right) < 1.25 \right]. \tag{26}$$

For all depth-forecasting results, we compute each metric independently on every target frame and then average over frames, giving every frame equal weight regardless of its number of valid LiDAR pixels.

Our forecasts calibrate metric scale once from same-pixel VGGT-Ω/LiDAR depth pairs at the observed anchor. The conversion and scene-normalization unit defined in Appendix A.3 are shared by the input states, query-camera translations, and decoded depths throughout the rollout. Future LiDAR provides the scoring reference, while decoded predictions retain the context-derived scale. Fixed-View Copy-Last uses the same-pixel anchor calibration for its raw GFM depth raster.

For VGGT-World, we follow its original per-target-frame scale-alignment protocol. FR3D numbers are quoted from its original evaluation, which uses per-target-frame scale-and-shift alignment. DINO-Foresight relative inverse-depth predictions are fitted to target-frame LiDAR for depth scoring.

A.5.4 COMPRESSED VGGT STATE BASELINE

Architecture. The representation ablation uses an autoencoder of VGGT-Ω patch features. For each training example, the frozen GFM jointly processes four historical timestamps and one target timestamp, with FRONT, FRONT_LEFT, and FRONT_RIGHT at each timestamp. We retain the target FRONT features from GFM layers $\{4, 11, 17, 23\}$; each layer provides a 26×39 grid of 2,048-dimensional patch features. Four independent linear projections reduce each feature vector from 2,048 to 128 dimensions. The projected features are concatenated channel-wise and spatially average-pooled from 26×39 to 16×32, producing a compressed state $Z \in \mathbb{R}^{512 \times 512}$. To reconstruct the GFM features, the state is bilinearly upsampled to 26×39, split into four 128-dimensional groups, and mapped back to four 2,048-dimensional feature grids by independent linear layers. A frozen VGGT-Ω DPT head decodes the reconstructed features into FRONT depth. The compressor has 2.11M trainable parameters.

Training. We follow the same two-stage training paradigm as the Scene Tokenizer. The compressor is first pretrained for 200k steps on the same 9,496-scene multi-view dataset, after which the best checkpoint is selected and further fine-tuned on Waymo for 50k steps.

The training procedure is identical in both stages. For each training sample, we extract the VGGT-Ω features of the target image and compress them into the latent state Z. The compressor then reconstructs the original multi-layer VGGT-Ω features, which are passed through the frozen VGGT-Ω DPT head to obtain the predicted depth. The prediction is supervised directly against the corresponding ground-truth depth using

$$\mathcal{L}_{\text{rep}} = \mathcal{L}_{\log D} + 0.1\mathcal{L}_{\text{grad}}. \tag{27}$$

The VGGT-Ω backbone and DPT head remain frozen throughout both stages, and gradients are propagated through the DPT head to optimize only the compressor.

A.6 CROSS-QUERY GEOMETRIC CONSISTENCY

A.6.1 FORECAST AND OBSERVATION-ENCODED STATES

We use the same 100 Waymo validation windows and all ten future timestamps. World denotes the world model autoregressive prediction. Enc. denotes an independent target-time reconstruction from exactly three images. For Enc., a separate historical pass provides the normalization length L_0 defined in Eq. 19.

For Enc., raw target FRONT GFM depth is calibrated using target-time LiDAR, and the normalization scale is set to its paired scale times L_0. World retains its historical-anchor calibration. Both decode under the same GT FRONT reference query and the same camera perturbations, without independently aligning decoded depths. Enc. is therefore an observation-based reference with privileged target-time scale information.

A.6.2 CAMERA PERTURBATIONS AND COVERAGE

Each perturbation combines a 3D translation with an axis–angle rotation. Translation directions and rotation axes are independently sampled uniformly on the unit sphere, while their magnitudes are sampled uniformly within one of three perturbation levels. Small perturbations use translation magnitudes in $[0, 0.5]$ m and rotation magnitudes in $[0, 2]^\circ$; Medium uses $[0.5, 1.0]$ m and $[2, 5]^\circ$; and Large uses $[1.0, 1.5]$ m and $[5, 10]^\circ$. Rotation magnitudes refer to the total axis–angle rotation, and translation magnitudes to Euclidean displacement. For a reference camera-to-world pose P_0, the perturbed pose is $P_i = P_0\Delta_i$. Perturbations are expressed in the reference camera's local coordinates, with $+x$ right, $+y$ down, and $+z$ forward.

Using seed 20260804, we initially sample 250 perturbations at each level. For each perturbation, we compute its mean bidirectional coverage across all evaluation windows, timestamps, and both state variants, and retain the 200 perturbations with the highest coverage within each level. This produces a fixed set of 600 perturbations from 750 initial samples. The target-three-view evaluation reuses this fixed subset.

A.6.3 Reprojection, Masks, and Aggregation

We back-project D_a using K_a^{-1}, transform points into camera b, and project them using K_b. Destination pixels are rounded to their nearest integer location. A z-buffer retains the smallest positive destination depth when points collide. Out-of-frame projections and points with destination $z \leq 10^{-4}$ m are excluded. Let $\Omega_{a\to b}$ contain pixels with finite, positive reprojected and decoded destination depths. The directional error is

$$e_{a\to b} = \frac{1}{|\Omega_{a\to b}|} \sum_{p\in\Omega_{a\to b}} \frac{|\mathcal{R}_{a\to b}(D_a)(p) - D_b(p)|}{\max(D_b(p), 10^{-8})}. \tag{28}$$

We report $(e_{0\to i} + e_{i\to 0})/2$. This dense consistency comparison uses neither the sparse LiDAR evaluation mask nor the 80 m depth cutoff. No additional depth-agreement threshold filters the overlapping pixels.

Coverage is $(|\Omega_{0\to i}| + |\Omega_{i\to 0}|)/(2HW)$. Each horizon equally averages 100 windows and 200 perturbations per level; the overall mean averages all ten horizons. There are $100 \times 10 \times 600 = 600{,}000$ query pairs per method. This metric measures geometric agreement across queries and complements the evaluation of accuracy against physical ground truth.

A.7 Feature Analysis of VGGT-Ω

VGGT-Ω jointly processes multiple images, so the representation of an image may depend on the other images included in the same forward pass. This property is relevant to state supervision: an observation-derived target can encode both scene geometry and context-specific variation introduced by the GFM. We therefore examine whether the features of an unchanged image remain stable when only its surrounding context changes.

We conduct a controlled diagnostic on all 100 Waymo validation windows. For each window, the same FRONT image at the first future timestamp (t+0.2 s) is retained at the same input index, with its image content, camera, and reference frame fixed. The reference frame is defined by the FRONT image at the observed anchor t. Each timestamp contributes three views. The shortest prefix containing the probed image comprises four observed timestamps and the first future timestamp, giving $5 \times 3 = 15$ images. The complete diagnostic window contains the same four observed timestamps and seven future timestamps, giving $11 \times 3 = 33$ images. We compare the probed-image features produced under these two contexts. For feature layer ℓ, we measure Relative L2

$$E_F^{(\ell)} = \frac{\|F_{33}^{(\ell)} - F_{15}^{(\ell)}\|_2}{\|F_{15}^{(\ell)}\|_2}. \tag{29}$$

We additionally compare the dense DPT depth maps predicted for the same t+0.2 s FRONT image under both unaligned and calibrated settings. The unaligned comparison is performed directly on the raw DPT outputs. For the calibrated comparison, both forwards estimate their VGGT-Ω-to-metric factors from the same matched VGGT-Ω/LiDAR depths at the observed anchor t, following the physical-unit convention in Eq. 19.

As shown in Table 8, context dependence is small in the early representation but becomes pronounced in deeper layers. Relative feature differences increase from 1.38% at layer 4 to 33.55% and 31.15% at layers 17 and 23, respectively, with a four-layer mean of 19.32%. The raw DPT outputs differ by 12.08% AbsRel. After expressing both outputs in the anchor-calibrated metric unit, the difference decreases to 3.82%. Although feature relative L2 and depth AbsRel are not directly commensurate, this controlled contrast shows that substantial variation in the VGGT-Ω feature coordinates can coexist with considerably more stable metric geometry. The features therefore carry context-conditioned information that is not uniquely determined by scene geometry, in addition to the geometric information used by the DPT head.

This property inevitably affects a state representation built on VGGT-Ω features. The Scene Tokenizer can suppress part of the context-specific variation, but its observation-derived targets are still constructed from GFM features that are not purely geometric. This observation helps explain the state-supervision ablation in Table 5. During one-step initialization, latent supervision provides a useful local target for learning the state transition. During autoregressive rollout, however, exact

Table 8: **Context dependence of frozen VGGT-Ω.** All metrics come from the same image while varying only its jointly processed context. Values are sample mean ± standard deviation.

Quantity	Measure	15 vs. 33 images
GFM patch features		
Layer-4 patch features	Relative L2	$1.38 \pm 0.59\%$
Layer-11 patch features	Relative L2	$11.21 \pm 1.96\%$
Layer-17 patch features	Relative L2	$33.55 \pm 8.71\%$
Layer-23 patch features	Relative L2	$31.15 \pm 5.97\%$
Four-layer mean	Relative L2	$\mathbf{19.32 \pm 4.16}\%$
DPT depth of the same image		
Raw DPT depth	AbsRel	$12.08 \pm 7.59\%$
Anchor-calibrated DPT depth	AbsRel	$\mathbf{3.82 \pm 1.58}\%$

Table 9: **Long-horizon depth forecasting on Waymo with FRONT-only history.** GeoState receives only the FRONT image at each observed timestamp; Bold and underlined values indicate the best and second-best results among methods using predicted future cameras.

Method	Camera	Per-frame	t+0.2 s		t+0.6 s		t+1.0 s		t+2.0 s	
		alignment	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
DINO-Foresight	Pred.	Yes	0.119	0.841	0.138	0.825	0.151	0.798	0.243	0.663
VGGT-World	Pred.	Yes	<u>0.073</u>	<u>0.931</u>	<u>0.112</u>	<u>0.892</u>	0.143	0.826	0.213	0.731
FR3D	Pred.	Yes	0.110	0.879	0.121	0.858	<u>0.131</u>	<u>0.840</u>	<u>0.152</u>	<u>0.800</u>
GeoState	Pred.	No	**0.070**	**0.936**	**0.089**	**0.914**	**0.105**	**0.881**	**0.143**	**0.813**

latent matching then asks the World Model to reproduce both scene geometry and context-specific feature variation, including directions with limited influence on scene geometry. Supervising rollout states through decoded depth instead constrains the scene geometry required at inference while allowing flexibility in these context-sensitive latent directions. This analysis supports our use of latent supervision for one-step initialization and decoded-geometry supervision for closed-loop rollout.

A.8 Additional Quantitative Results

A.8.1 Single-View-Controlled Baseline Comparison

The main GeoState evaluation uses the synchronized FRONT, FRONT_LEFT, and FRONT_RIGHT images at each observed timestamp, whereas our compared methods use a single image. To control for this difference in input-view count, we re-evaluate GeoState on the same 100 Waymo evaluation windows using only FRONT. Both the frozen GFM and the Scene Tokenizer therefore receive four historical images in total, one from each observed timestamp. The model checkpoints, timestamps, query cameras, metric-scale calibration, and LiDAR masks remain unchanged. Table 9 compares this FRONT-only variant with the external forecasting baselines from Table 1.

Under predicted future cameras, FRONT-only GeoState outperforms FR3D, VGGT-World, and DINO-Foresight in both AbsRel and δ_1 at every reported horizon. At t+2.0 s, it obtains an AbsRel of 0.143 and a δ_1 of 0.813, compared with 0.152/0.800 for FR3D, 0.213/0.731 for VGGT-World, and 0.243/0.663 for DINO-Foresight. This comparison removes the additional side-view observations available to the main GeoState configuration, while GeoState continues to use a single context-derived scale rather than per-target-frame alignment. The performance advantage over the external forecasting baselines therefore cannot be attributed solely to using three camera views per observed timestamp.

A.8.2 Motion-Stratified Forecasting Results

We further report results under the Dynamic/Static partition defined in Appendix A.4.1. Dynamic combines all windows containing at least one qualifying visible moving track, while Static con-

Table 10: **Motion-stratified forecasting on Waymo.** Results are frame-wise means over all ten horizons on the Waymo validation set. "Improve" is the relative AbsRel reduction of GeoState over the camera-matched Copy-Last, and $\Delta\delta_1$ is the absolute accuracy gain.

		AbsRel↓			**δ_1 ↑**		
Subset	**Camera**	Copy-Last	GeoState	Improve	Copy-Last	GeoState	$\Delta\delta_1$
Dynamic	GT	0.0980	0.0852	13.10%	0.8842	0.9077	+0.0236
Dynamic	Pred.	0.1070	0.0967	9.63%	0.8717	0.8946	+0.0229
Static	GT	0.0710	0.0675	4.92%	0.9245	0.9337	+0.0092
Static	Pred.	0.0915	0.0895	2.17%	0.8917	0.8995	+0.0078

tains the remaining windows. Table 10 averages each subset over all ten forecasting horizons and compares GeoState with the camera-matched Copy-Last reference.

GeoState improves both subsets under GT and predicted query cameras. On the Dynamic subset, it reduces AbsRel by 13.10% with GT cameras and 9.63% with predicted cameras, with corresponding δ_1 gains of 0.0236 and 0.0229. The gains on Static windows are smaller but remain positive: 4.92% and 2.17% relative AbsRel reductions, respectively. The substantially larger gains on Dynamic windows indicate that the World Model effectively captures temporal changes in scene geometry. The same pattern under GT query cameras further shows that this dynamic-scene advantage is present independently of future-camera prediction error.