GEOSCOPE: GROUNDED, STABLE AND CONTROLLABLE DRIVING WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative driving world models produce realistic video but lack grounding in a real location. Conditioned on their own predictions, they modify road layout and surrounding structures, causing long-horizon rollouts to diverge from the recorded scene. A policy evaluated in such rollouts is effectively tested on a different scene than the one recorded. We propose GeoSCope, a method that grounds driving world models in the geometry of a real route using only a single monocular recording. A dynamics-aware SLAM pipeline reconstructs a static 3D map of the route, which is rendered at the camera pose of each frame prior to its generation and used to condition a generative world model. Although the rendered signal is sparse and imperfect, the model learns to exploit it selectively. Experiments with three world models on four datasets show that this anchoring substantially reduces long-horizon drift and preserves scene structure without suppressing dynamic objects. Since the map can be rendered along arbitrary trajectories for conditioning, the same model supports counterfactual rollouts, turning existing video-based world models into grounded, stable, and controllable neural simulators. With geometric conditioning trained only on nuPlan, the approach transfers zero-shot to Waymo, MARS and nuScenes, and sets a new state of the art on the Ann-Arbor-City-Bench multi-traversal benchmark. Code and weights will be released upon acceptance.

1 INTRODUCTION

Closed-loop training and evaluation of driving policies require a simulator that reacts to the policy's actions while remaining faithful to the real world. Reconstruction-based neural simulators, including those based on 3D Gaussian splatting, provide explicit geometry and direct camera control. However, their quality deteriorates under large deviations from the observed trajectory, many require auxiliary annotations such as 3D bounding boxes or object tracks (Chen et al., 2025b; Yan et al., 2024), and dynamics must be modeled through separate components. Generative world models offer the complementary strength: they learn appearance, dynamics and interactions end-to-end from video. Yet nothing ties what they generate to a real place because they are not grounded.

Without a persistent scene reference, a generative world model is free to hallucinate the static environment, reinventing lane geometry, intersections and buildings as the rollout proceeds. Because each new frame is predicted from the model's own previous outputs, nothing in the loop can correct such deviations and they compound over long horizons. This is not merely a visual-quality issue, since a prediction can look realistic without being faithful to the real scene layout. When evaluating a driving policy, the static environment is what the policy is being tested against, so it has to remain true to the real scene. If the road bends to accommodate a lateral deviation, or an intersection disappears after a few seconds, the policy is no longer evaluated on the logged scene. A learning policy may even exploit such artifacts, and its failures can no longer be told apart from failures of the simulator. One way to ground generation is to condition on HD maps (Wang et al., 2024; Gao et al., 2024a). This fixes the road layout, but HD maps need precise annotations that are expensive to create and maintain, which is why most raw driving logs do not have them.

We introduce GeoSCope, a **G**eometry-grounded, **S**table, and **C**ontrollable method for anchoring a generative world model in the real geometry of the traversed route using only a single monocular video driving log, without LiDAR or HD maps. A dynamics-aware SLAM pipeline first reconstructs a filtered static point cloud of the route. At each rollout step, we render this reconstruction at 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.

requested pose of the frame about to be generated and use the resulting pixel-aligned image to condition a pre-trained world model. We refer to this as prospective geometric anchoring, since the reconstruction is queried at the upcoming target pose rather than aligned only with past or current viewpoints. In this setup, the reconstruction fixes the static environment wherever it was observed, while the generative prior completes sparse and uncovered regions and generates the dynamic agents that the static reconstruction does not contain. The mechanism assumes nothing about the backbone beyond an autoregressive rollout and a latent encoder, so it applies to existing world models without architectural redesign. We demonstrate this on three models that differ in scale and architecture, namely Orbis v1, the hierarchical Orbis v2, and Cosmos-Predict2.5.

Anchoring to a persistent reconstruction grounds generation in the real layout of the route, which substantially reduces structural drift and better preserves perceptual quality over long horizons. Across base models and datasets, our approach roughly doubles long-horizon keypoint matches and reduces long-horizon FVD and depth error substantially, without suppressing the dynamic content that the generative prior supplies. Beyond stabilizing the scene, the same signal provides direct control over the ego trajectory, since the reconstruction can be rendered along *any* trajectory and the model that replays a logged route can also roll out counterfactual variants of it. Our approach can be made robust to imprecise localization. This matters because many mapless, camera-only driving stacks have rough positions from satellite navigation, so the rendered map will not line up exactly with the camera. When trained with pose-augmented renders, the model tolerates position errors up to 3 m and still outperforms the base model without conditioning. This makes it possible to reuse a map from one drive for later drives in the same area, even when they are only roughly localized.

We summarize our contributions as follows:

- We introduce the *geometric anchoring* approach GeoSCope, which grounds any autoregressive driving world model in a SLAM reconstruction built from a single monocular camera log. Rendered at the pose of each upcoming frame, the map supports both faithful route replay and counterfactual trajectories, while the generative prior remains responsible for all dynamics.
- We show that these sparse and photometrically degraded renderings are sufficient for grounding and stable long-horizon prediction. The gain persists across conditioning mechanisms, at reduced coverage, and, with pose-perturbed training, under localization errors of several meters. Across three world models and four datasets, our approach roughly doubles long-horizon keypoint matches without suppressing dynamic content, transfers zero-shot to unseen datasets, and sets a new state of the art on the Ann-Arbor-City-Bench cross-traversal benchmark despite never being trained on multi-traversal data.

2 Related Work

Driving World Models. Early work established real-world generative driving simulation and action-conditioned autoregressive generation (Kim et al., 2021; Hu et al., 2023). Subsequent methods improved spatial fidelity through HD maps, 3D bounding boxes, and multi-view geometric constraints (Wang et al., 2024; Lu et al., 2024; Liao et al., 2025; Gao et al., 2024a), which ground the road layout explicitly but require annotations that most raw driving logs lack. More recent models obtain realistic, controllable rollouts through large-scale video pre-training and diffusion-based generation (Yang et al., 2024; Gao et al., 2024b; Zhang et al., 2025; NVIDIA et al., 2025). Orbis v1 (Mousakhan et al., 2025) and its hierarchical successor Orbis v2 (Mittal et al., 2026) showed that compact models trained only on monocular driving video can produce strong long-horizon rollouts. In these models, the lack of grounding leads to structural drift in rollouts longer than a few seconds, which we address by anchoring generation to a geometric reconstruction of the traversed route.

Reconstruction-based simulators. A parallel line of work reconstructs a recorded drive explicitly and renders it at new camera poses, from neural sensor simulators (Yang et al., 2023; Tonderski et al., 2024) to Gaussian-splatting methods that separate a static background from per-object dynamics (Yan et al., 2024; Chen et al., 2025b; Fischer et al., 2024). These give exact geometry and direct camera control, but the reconstruction must produce the output pixels, so unobserved viewpoints yield holes and artifacts. StreetCrafter (Yan et al., 2025) relaxes this by conditioning a video diffusion model on LiDAR renders, but it requires LiDAR and synthesizes views of a recorded clip rather than predicting a future.

Persistent Scene Context. Prior work on long-horizon video generation can be grouped by whether it maintains an explicit scene memory and by the source of that memory.

No explicit memory. Some methods reduce the training–inference gap without external memory, for example by using per-token independent noise levels or by training on the model's own degraded rollouts (Chen et al., 2024; Cui et al., 2025; Chandratreya et al., 2026). Their conditioning remains bounded by the learned prior and cannot correct hallucinated scene content as the sequence grows.

Memory from generated content. Others build an explicit memory from the model's own outputs, storing compressed frame summaries, retrieved past frames, or scene-specific weights (Henschel et al., 2025; Xiao et al., 2025; Hong et al., 2025). Others lift generated frames into 3D representations such as Gaussian splats, surfel maps or point maps, and render them at new poses to condition generation (Kong et al., 2025; Li et al., 2025; Yu et al., 2026; Wu et al., 2026). Since this memory is built from generated content, errors accumulate within it and degrade future conditioning.

Memory from real observations. The closest approaches to ours ground generation in a representation built from real observations, reconstructed offline and rendered at the target camera pose. GEN3C (Ren et al., 2025) unprojects monocular depth from seed images into a point-cloud cache, and GeoDrive (Chen et al., 2025a) extracts a static point cloud from a single seed frame, so their geometry covers only the initially visible scene and cannot anchor generation once the rollout leaves that view frustum. StreetNVS (Kuang et al., 2026) instead fuses LiDAR reprojections with surround-view imagery, Vista4D (Lin et al., 2026) lifts training frames into a persistent 4D point cloud, and VerseCrafter (Zheng et al., 2026) pairs a static background point cloud with per-object 3D Gaussian trajectories to control both camera and object motion. These approaches primarily perform geometry-conditioned novel-view synthesis that replays observed dynamics. Retrieval-based systems such as CityRAG (Chou et al., 2026) and SWM (Seo et al., 2026) query geo-registered imagery or pre-built geospatial maps, but require dense multi-pass coverage. We use a single monocular SLAM pass to anchor only the static route context, leaving dynamics to the world-model prior, and render it prospectively at future poses so that the same map also supports counterfactual trajectories.

3 METHOD

3.1 PROBLEM SETUP AND PROSPECTIVE GEOMETRIC ANCHORING

We consider an autoregressive driving world model that generates future frames one window at a time. Given the N_{ctx} most recent frames $G_{n-N_{\text{ctx}}:n-1}$, the model predicts the next F_{pred} frames and appends them to the context for the next rollout step. Since the generation is not constrained by the real environment, the model can produce any plausible future. Each window is predicted from previous predictions, so deviations accumulate and the rollout drifts away from the recorded scene.

To ground the rollout, we introduce a conditioning signal that is independent of the model's predictions (Figure 1). From a single monocular recording of the route, we reconstruct a static point cloud $\mathcal{P}$ once, offline (Section 3.2). For each frame to be generated, we render $\mathcal{P}$ at the corresponding camera pose P_n (Section 3.3), which yields a pixel-aligned geo-reference frame $R_n = \Pi(\mathcal{P}, P_n)$, where Π denotes the rendering function. The world model is then conditioned on this frame,

$$G_n \sim p_\theta\big(\cdot \mid G_{n-N_{\text{ctx}}:n-1},\, R_n\big). \tag{1}$$

We refer to this as *prospective geometric anchoring*, since the point cloud is rendered at the pose of the upcoming frame before it is generated. During training, P_n is the SLAM-estimated pose of the recorded frame, so that R_n is aligned with the target frame. At inference, the choice of poses determines the rollout. The recorded poses enable a replay in the original drive's environment, and modified poses produce counterfactual rollouts in the same scene.

3.2 STATIC SCENE RECONSTRUCTION

Dynamics-aware SLAM. We estimate the scene geometry with DROID-W (Li et al., 2026), an off-the-shelf dynamics-aware SLAM system. It extends the optical-flow bundle adjustment of DROID-SLAM (Teed & Deng, 2021) with a monocular metric-depth prior and a learned per-pixel uncertainty, which makes the reconstruction robust to moving objects and to single-traversal monocular

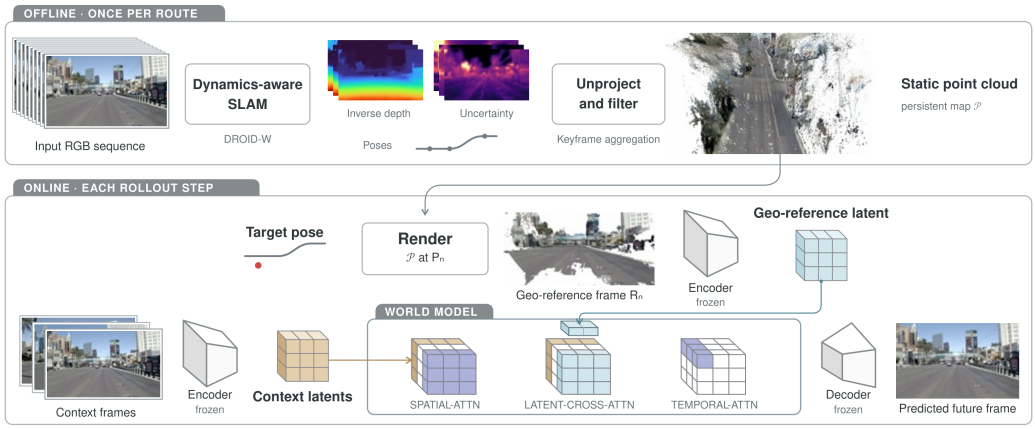

Figure 1: **GeoSCope overview**. Top: SLAM reconstruction and point cloud filtering (Section 3.2) are performed *offline*, once per route. Bottom: the resulting point cloud is reused *online* at every rollout step to render geo-reference frames (Section 3.3) that condition the world model (Section 3.4).

input. Unlike the usual online deployment of SLAM, we run DROID-W offline over the full recording, so that all frames of the route contribute to the optimization. For each selected keyframe, it returns a camera pose, an inverse-depth map and a per-pixel uncertainty map. We refer to the original work for details on tracking and optimization, and list our configuration in Table 9.

Point cloud construction and filtering. The keyframe poses, inverse-depth maps, and input RGB images are unprojected to a colored point cloud in world coordinates. Filtering is performed once, on the complete point cloud accumulated over the full traversal, rather than per rendered frame. Five sequential consistency filters (disparity, uncertainty, multi-view geometric and photometric, voxel-outlier) clean the raw cloud and suppress unreliable geometry arising from dynamic objects, occlusions, and mismatched correspondences. The result is a static point cloud $\mathcal{P}$ that is computed once per route. Filter definitions and thresholds are given in Table 10.

3.3 Rendering Geo-Reference Frames

While $\mathcal{P}$ is precomputed offline, rendering is performed online at each rollout step. Each frame is rendered from the complete point cloud rather than only from the points observed up to that frame, since the route has already been traversed and the reconstruction is available before the rollout begins. Points projecting to the same pixel are composited with a depth-gated weighted blend rather than a hard z-buffer, which fills small gaps and suppresses flicker on the sparse cloud. Details are given in Section B of the Appendix.

3.4 Conditioning Mechanisms

The rendered geo-reference frames have an error profile that is complementary to the generative prior. They are geometrically grounded, stable, and controllable, but sparse, incomplete, and photometrically degraded. See Section E for examples and quality assessment. The design question is thus how strongly the model should trust them.

Geo-reference encoding. Each geo-reference frame is encoded with the frozen tokenizer of the base model. We denote by $Z^{\text{geo}} \in \mathbb{R}^{F_{\text{pred}} \times H_{\text{lat}} \times W_{\text{lat}} \times C}$ the latents of the geo-reference frames of one rollout window, and by M the binary mask of latent positions covered by projected points. Further, $g_{f,i} \in \mathbb{R}^C$ denotes the geometry token of frame f at spatial position i.

Injection mechanisms. We compare mechanisms that range from hard to soft trust in the signal (Table 1). *Latent blending* is training-free and interpolates the latent toward a noised version of Z^{geo} in covered regions during the early, noise-dominated denoising steps. *Concatenation* fuses Z^{geo} with the noisy input latent along the channel axis, using a zero-initialized extension of the patch embedding. The *adaLN adapter* adds per-token offsets derived from the geometry to the global adaLN

modulation of every block, which gives spatially local control. *Gated cross-attention* lets every token attend non-locally to the geometry and can therefore discount unreliable regions. We also test cross-attention combined with adaLN. See Section D for full formulations. We used cross-attention for all ablation experiments.

Table 1: Overview of the geometric conditioning mechanisms, ranging from hard latent-space constraints to soft learned modulation. Parameter counts refer to Orbis v1.

Variant	Spatial scope	Trust	New params
Latent blending	masked tokens only	hard	0
Concatenation	local (per token)	soft	0.02 M
AdaLN	local (per token)	soft	43.2 M
Cross-attn	non-local (attention)	soft	59.1 M
Cross-attn + AdaLN	local + non-local	soft	102.3 M

Gated cross-attention. Each geometry token is lifted to the hidden dimension by a shared linear projection, and the frozen sinusoidal positional embedding p_i of the query tokens is added to establish a shared spatial coordinate system,

$$h_{f,i} = W_{\text{proj}}\, g_{f,i} + p_i, \qquad W_{\text{proj}} \in \mathbb{R}^{D\times C}. \tag{2}$$

A gated cross-attention sub-layer is inserted into every block of the backbone, with queries from the denoising tokens and keys and values from h. Queries and keys pass through a parameter-free LayerNorm on the per-head dimension, which prevents logit explosion from the heterogeneous magnitudes of the geometry features. The sub-layer is placed after the spatial self-attention and spatial MLP and before temporal attention (Zheng et al., 2024), so that queries carry local spatial context and temporal attention propagates the geometry-conditioned features across frames,

$$x \leftarrow x + \tanh(\alpha) \cdot \mathrm{CrossAttn}\big(\mathrm{LN}(x), h\big). \tag{3}$$

The scalar gate α is initialized to zero, so the block computes the pretrained function at initialization, and the gate opens during fine-tuning. Attention is block-diagonal across frames, so predicted frame j attends only to geo-reference frame j, while context frames do not participate. The final projection layer likewise applies gated cross-attention before the global adaLN.

3.5 Integration into Pre-trained World Models

Base world models. We apply our conditioning to three base world models of different scales and architectures, all initialized from their released weights. Architecture details are given in Section C. **Orbis v1** (Mousakhan et al., 2025) is a 469M-parameter STDiT that rolls out frame by frame, as described in Section 3.1. We use it for the comparison of all conditioning mechanisms. **Orbis v2** (Mittal et al., 2026) is a 1.07B-parameter hierarchical model, in which an abstract predictor generates subgoals at 2 Hz and a detail predictor generates frames conditioned on them. We apply gated cross-attention to the abstract predictor, so that the geometric signal guides the subgoals, which the detail predictor then turns into frames. **Cosmos-Predict2.5** (NVIDIA et al., 2025) is a 2B-parameter video DiT with a temporally compressing VAE, rolled out chunk by chunk. We apply gated cross-attention between the denoising target latents and the encoded geometric signal latents.

Training. Latent blending is applied directly to the released model and requires no training. For all learned variants, the tokenizer remains frozen, and the full backbone is fine-tuned jointly with the new modules using the flow-matching objective of the respective base model. For a fixed fraction of training samples, Z^{geo} is set to zero, which enables classifier-free guidance at inference.

Training with perturbed poses. To make the model tolerant to imprecise localization, we optionally train with geo-reference frames rendered from displaced viewpoints. For each frame, we sample a displacement uniformly within a circle of radius R=3 m around the true pose, holding orientation fixed. We additionally add small yaw/pitch jitter ($\pm 2°/\pm 1°$). This teaches the model to trust what the rendering shows without assuming it is exactly aligned.

4 Experiments

4.1 Experiment Details

Datasets. We train on nuPlan (Caesar et al., 2021), using only the front camera without LiDAR, maps, or object annotations. Following Orbis v1 and Orbis v2, we test on their curated nuPlan-Turns subset of 400 clips which includes challenging turning maneuvers. We also zero-shot test

on three more datasets. On Waymo (Ettinger et al., 2021), we use 182 full-length, non-overlapping validation clips, each the single longest contiguous window its segment supports (up to 197 raw frames at 10 Hz, ≈19.7 s). We apply the same recipe used to generate nuPlan-Turns to MARS's (Li et al., 2024) 66 multitraversal recordings, yielding 400 turn-event scenes: initial yaw rate exceeding 1 standard deviation of MARS's measured distribution, with the ego vehicle maintaining a mean speed of at least 1 m/s over a 20 s horizon. On nuScenes (Caesar et al., 2020), we roll out all 150 official validation scenes in full, without windowing.

To compare against the concurrent Seoul World Model (Seo et al., 2026), we follow their protocol and construct ANN-ARBOR-CITY-BENCH from 30 MARS cross-traversal pairs, where the geo-reference signal is reconstructed from one traversal of a route and rendered at the camera poses of a second, independent traversal.

Training details. For each training scene, we run DROID-W and the rendering pipeline on a contiguous 300-frame subsequence and draw all training windows from it. All models are trained on 16 NVIDIA GH200 GPUs (4 nodes × 4 GPUs) with an effective batch size of 64.

Baselines. We compare against the released state-of-the-art driving world models Orbis v1, Orbis v2, and Cosmos2.5-predict-2B. For Cosmos, we use the 480p variant for both the baseline and the anchored model. We additionally compare against the concurrent retrieval-based method Seoul World Model (Seo et al., 2026). Since no code or checkpoints are released, we follow and reproduce their evaluation protocol. We retain the other models from their table for comparison.

4.2 EVALUATION

Our method replays logged routes, so generated videos can be compared frame by frame with the recorded future. We organize the evaluation along three complementary axes, each targeting a distinct claim of the paper. Every metric is reported over a short horizon (0–10 s) and a long horizon (10–20 s), as autoregressive drift only becomes visible in long rollouts. For the average across four datasets (Table 2), we use 0–8 s and 8–16 s. All rollouts are assessed at 5 FPS.

1. Consistency (reference-based). These metrics compare each generated frame with its temporally aligned ground-truth frame, quantifying how faithfully the model reproduces the observed future and, combined with the horizon axis, how quickly it drifts. We report three complementary metrics. **LPIPS** (Zhang et al., 2018) measures perceptual similarity and degrades gracefully under the small viewpoint and timing offsets that pixel-aligned measures over-penalize. The number of **LightGlue keypoint matches** (Lindenberger et al., 2023) between generated and ground-truth frames is structure-aware and requires the local scene layout to be preserved at approximately corresponding locations. **Depth AbsRel** error between monocular depth predictions on both frames isolates geometric from photometric consistency. For the latter we use DepthAnything v3 (Lin et al., 2025), avoiding a shared backbone with the Metric3D (Yin et al., 2023) prior inside DROID-W so that the conditioning signal and its evaluator remain independent.

2. Perceptual quality is measured with FVD (Unterthiner et al., 2019), which requires no frame-level correspondence and tests whether reduced drift comes at the cost of visual realism.

3. Trajectory adherence (controllability). On the logged route, high consistency already implies trajectory adherence; this axis isolates it. We recover the camera path of the generated rollout with VGGT (Wang et al., 2025), align it to the commanded ego-trajectory with a Sim(3) Umeyama fit to resolve the gauge ambiguity of monocular reconstruction, and report the **ADE** (average displacement error) between the two paths.

5 RESULTS

5.1 GEOSCOPE ANCHORING REDUCES STRUCTURAL DRIFT

We train gated cross-attention as the ingestion mechanism across all three base models and compare each released model with its anchored variant in Table 2. Three of the four datasets are evaluated zero-shot, since training uses nuPlan only. The two information sources have complementary error profiles. The geo-reference signal alone has poor realism (FVD >1700), yet its keypoint matches do not degrade from the short to the long horizon, so it remains geometrically stable where the



Figure 2: Qualitative long rollouts at different prediction horizons. GeoSCope follows the GeoRef geometry where it is available and fills in the gaps, while the baseline slowly drifts away from the recorded layout. Moving objects are still generated by the world model.

generative prior drifts. The base models fail in the opposite way, starting from strong visual realism and losing structure as the rollout proceeds, with long-horizon FVD rising by 58 to 99% and keypoint matches dropping by more than half. Our anchoring approach combines the strengths of both sources and improves all three evaluation axes at once. It preserves structural consistency, as measured by keypoint matches and depth error, keeps perceptual quality high by flattening the FVD slope substantially, and improves trajectory adherence.

Table 2: Average across NUPLAN-TURNS, Waymo, MARS and nuScenes for short/long-horizon metrics; the full table is in the appendix. Each base world model is shown directly above its anchored variant (shaded). ADE in $\times 10^{-3}$ m. Best per column in **bold**.

Method	LPIPS ↓		FVD ↓			KP Matches ↑		Depth AbsRel ↓		ADE ↓	
	Short	Long	Short	Long	Slope	Short	Long	Short	Long	Short	Long
GeoRef signal only	0.506	0.493	2044.9	1755.3	–	188.6	200.1	0.341	0.342	3.9	7.1
Orbis v1	0.417	0.508	323.9	539.4	64.0	217.3	104.8	0.214	0.307	4.8	14.7
+ GeoSCope	0.345	0.411	**227.1**	**325.0**	**27.2**	267.8	184.5	0.157	0.217	**2.3**	**6.5**
Orbis v2	0.393	0.494	282.9	446.8	45.9	218.3	98.4	0.202	0.300	4.0	11.4
+ GeoSCope	**0.313**	**0.379**	229.8	337.0	30.4	296.1	236.0	0.126	0.172	2.6	6.7
Cosmos 2.5	0.433	0.525	325.4	506.5	52.4	292.7	124.9	0.216	0.314	6.5	10.9
+ GeoSCope	0.358	0.421	418.7	623.3	57.7	**421.6**	**328.3**	**0.124**	**0.153**	–	–

5.2 COMPARISON WITH GROUNDED WORLD MODELS

We compare against the Seoul World Model (SWM) on their multi-traversal benchmark, where the geo-reference is reconstructed from one traversal of a route and rendered at the camera poses of a second independent traversal. Our model is never trained on multi-traversal data, so the setting is entirely zero-shot for it, unlike for the retrieval-based SWM. As shown in Table 3, our approach outperforms all competing methods on FID, FVD, mPSNR and mLPIPS, with SWM (TF) leading narrowly on Img. Q. This is consistent with our single-traversal findings. The static structure of a route is the same on every pass, so it gives the model a stable reference to exploit, and it does not matter whether that reference was reconstructed from the same traversal or a different one.

5.3 CONTROLLABLE COUNTERFACTUAL GENERATION

The reconstruction can be rendered along any trajectory, so the same model that replays a logged route can also execute trajectories never observed during training. We command three perturbation types: lateral offsets, a sinusoidal wiggle, and speed changes; see Table 4. The model follows the

Table 3: **Comparison with other grounded world models.** We evaluate visual and temporal fidelity, and conditioning signal adherence. Values are reported on Ann-Arbor-City-Bench.

Method	FID↓	FVD↓	Img.Q.↑	mPSNR↑	mLPIPS↓
Yume1.5 (Mao et al., 2025)	85.62	993.62	0.61	14.15	0.623
HY-World1.5 (HunyuanWorld, 2025)	67.02	864.76	0.54	14.26	0.575
FantasyWorld (Dai et al., 2026)	67.72	917.57	0.49	11.97	0.592
Lingbot (Team et al., 2026)	57.99	1039.50	0.60	12.51	0.641
SWM (TF) (Seo et al., 2026)	56.61	640.17	**0.66**	15.18	0.481
SWM (SF) (Seo et al., 2026)	43.97	779.94	0.57	14.20	0.573
GeoRef Signal	201.06	2298.45	0.55	8.10	0.516
Ours (Orbis v1 + GeoSCope)	**39.27**	486.69	0.60	**18.61**	**0.313**
Ours (Orbis v2 + GeoSCope)	39.34	**443.16**	0.65	17.03	0.326

commanded path closely. Under lateral perturbations, trajectory adherence and realism degrade only mildly and in proportion to the perturbation magnitude (see Table 4). Compliance measures use different units: metres of lateral shift, slope of the commanded sine, and fraction of samples closer to the target speed. Full statistics are included in Table 18. These offsets only change the viewpoint, which the reconstruction supports directly. Speed and wiggle changes are the harder cases, as they push the rollout out of the ego motion distribution the base model was trained on.

Table 4: Counterfactual trajectory perturbations. Compliance measures the followed response to the command. Details in Table 18.

Perturbation	FVD	ADE	Compliance
Lateral +1 m	224.5	10.8	+0.66 m
Lateral +0.5 m	202.7	10.3	+0.47 m
Nominal	188.4	9.9	—
Lateral -0.5 m	202.8	10.2	-0.42 m
Lateral -1 m	224.0	11.3	-0.75 m
Wiggle 0.5 m	250.4	11.8	0.86 slope
Speed 0.85	208.7	11.9	60%

5.4 Dynamics are not suppressed

We investigate whether the static geometric reference signal suppresses the occurrence of dynamic objects such as vehicles, buses, bicycles, or pedestrians. We find that throughout the rollout their number stays within the range of the unconditioned base models rather than decreasing, so the static reference constrains the scene while also generating dynamic agents moving through it. The dynamic object count is compared with the base models in Figure 3. More details in section I.4.

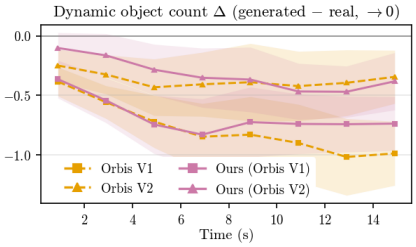

Figure 3: Dynamic object count over the rollout, measured as the difference between the generated and the recorded frames, where zero is ideal.

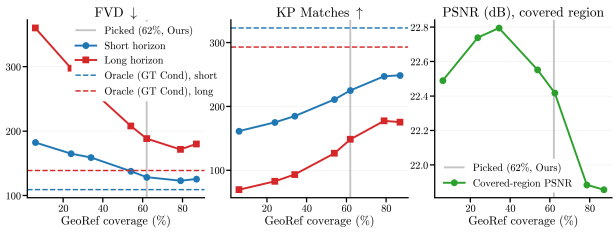

Figure 4: Effect of geo-reference coverage on generation quality. Left and center show rollout quality at the short and long horizon; right shows PSNR over the covered region. GeoSCope operates at 62%.

5.5 Robustness to Localization Error

In all preceding experiments the renderings come from the same SLAM run that produced the evaluation poses, so the map and the camera are aligned by construction. This favorable case does not hold when poses come from satellite navigation or otherwise have inaccurate localization. We test whether the anchoring mechanism tolerates a reference signal that is misaligned with the ego pose.

We compare a model trained on aligned renders against one trained with pose perturbations up to 3 m, and evaluate each under both conditions.

Trained on aligned renderings alone, the model attempts to follow the displaced geometry rather than discounting it, and degrades sharply once the signal is misaligned. Augmenting training with pose perturbations resolves this, and the perturbation-trained model performs similarly whether the evaluation signal is aligned or not, indicating that it has learned to treat the signal as a soft hint rather than a hard constraint. The cost relative to the aligned-trained model is moderate, while the gain over the unconditioned world model is largely preserved. Full results on Orbis v1 are in Table 17.

Table 5: Effect of localization errors when trained with and without localization error.

	Evaluation			
	Pose error 0m		Pose error 0–3m	
Training	FVD ↓	ADE ↓	FVD ↓	ADE ↓
Pose error 0m	188.4	9.9	728.4	28.0
Pose error 0–3m	285.3	11.5	299.3	12.1

5.6 How much should the model trust the signal?

Comparing conditioning mechanisms. Table 6 compares the four injection mechanisms on long-horizon frames, against the unconditioned backbone and the training-free variant. Latent blending improves keypoint matching substantially without any training, but its hard latent blend also inherits artifacts from the noisy geo-reference signal. All learned mechanisms improve long-horizon stability by a wide margin over the unconditioned backbone, and the differences between them are small. Cross-attention is best on all three metrics and channel concatenation recovers most of the gain with three orders of magnitude fewer added parameters. Combining adaLN and cross-attention yields no gain, and the adaLN gates open early while the cross-attention gates remain closed.

Table 6: Conditioning mechanism comparison. Best per column in **bold**, second best underlined.

Method	FVD ↓	KP Matches ↑	ADE ↓
Orbis v1	501.3	61.4	16.9
Latent blending	433.5	153.1	15.1
Concat	197.1	<u>144.5</u>	10.2
AdaLN	<u>195.5</u>	137.7	<u>10.0</u>
CrossAttn	**188.4**	**148.4**	**9.9**
CA + AdaLN	197.5	142.8	10.1

Coverage of the geometric reference We ablate the fraction of the frame covered by the projected point cloud by sweeping the multi-view depth-consistency threshold. All metrics improve monotonically as coverage increases from 6% to 79%, with the largest gains on long-horizon FVD and keypoint matches, precisely where the generative baseline drifts most. Beyond 79%, performance plateaus and slightly reverses: loosening the threshold further admits geometrically inconsistent points that introduce noise rather than useful structure. Our operating point at 62% lies on the conservative side of this trade-off between coverage and point cloud quality.

Conditioning on ground-truth geometry reveals substantial remaining headroom, nearly doubling long-horizon keypoint matches and lowering FVD further. The bottleneck is therefore the density of the reconstruction rather than the conditioning mechanism, which motivates future work on denser monocular reconstruction. Full results are in Table 16.

6 Discussion

Our experiments show that anchoring an autoregressive driving world model to a prospectively rendered SLAM reconstruction substantially reduces structural drift while turning a single monocular pass into a controllable neural simulation.

Limitations. Our conditioning signal is static and upper-bounded by the SLAM system's quality. Refining the input reconstructions via a diffusion prior, or substituting higher-fidelity 3DGS or mesh representations, remains open for future work. Finally, our evaluation measures rollout fidelity against logged or counterfactual trajectories rather than closed-loop performance of an actual driving policy, we plan on tackling this in future work.

AI USE STATEMENT

We used generative AI tools for implementing methods, for surveying and retrieving related work, and for editing the paper to improve readability. They were not used for developing the method or conceptual framework, for designing experiments, or for interpreting results. All AI-assisted code was reviewed line by line by the authors and validated by reproducing the reported experiments end-to-end, all references retrieved with AI assistance were verified against the original publications, and all AI-edited text was checked by the authors against the underlying experimental results. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only public, anonymized driving datasets (nuPlan, Waymo, MARS, nuScenes) under their research licenses, with no new human-subject data collection. We note the dual-use consideration that realistic synthetic driving video could be misused for deceptive media, within our project our intended use is closed-loop policy evaluation and simulation. More broadly, we contribute to the development and evaluation of world models for autonomous driving, technology that could improve road safety and mobility access, but that also raises concerns around job displacement, liability, and over-reliance on imperfect automation should it be deployed prematurely.

REPRODUCIBILITY STATEMENT

We will publicly release the adjusted SLAM processing pipeline, the point-cloud rendering code, and pretrained weights for all three base models upon acceptance. The SLAM configuration is given in Table 9 and the point-cloud filtering and rendering settings in Table 10. The conditioning mechanisms are specified with equations in Section 3.4 and Section D. The training and evaluation setup is described in Sections 4.1 and 4.2, with hyperparameters in Section F and metric definitions in Section G.

REFERENCES

Zechen Bai, Jianxiong Gao, Ziteng Gao, Pichao Wang, Zheng Zhang, Tong He, and Mike Zheng Shou. Factorized visual tokenization and generation. *arXiv preprint arXiv:2411.16681*, 2024.

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 (CVPR)*, 2020.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *CVPR ADP3 workshop*, 2021.

Ishaan Preetam Chandratreya, David Charatan, Basile Van Hoorick, Sergey Zakharov, Vitor Guizilini, Phillip Isola, and Vincent Sitzmann. Millivid: Hierarchical latents for long-range consistency in video generation. In *arXiv*, 2026.

Anthony Chen, W. Zheng, Y. Wang, X. Zhang, Kun Zhan, P. Jia, et al. Geodrive: 3d geometry-informed driving world model with precise action control. *arXiv preprint arXiv:2505.22421*, 2025a.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: next-token prediction meets full-sequence diffusion. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnire: Omni urban

scene reconstruction. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025b.

Gene Chou, Charles Herrmann, Kyle Genova, Boyang Deng, Songyou Peng, Bharath Hariharan, Jason Y. Zhang, Noah Snavely, and Philipp Henzler. Cityrag: Stepping into a city via spatially-grounded video generation, 2026. URL https://arxiv.org/abs/2604.19741.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation, 2025. URL https://arxiv.org/abs/2510.02283.

Yixiang Dai, Fan Jiang, Chiyu Wang, Mu Xu, and Yonggang Qi. Fantasyworld: Geometry-consistent world modeling via unified video and 3d prediction. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aur'elien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9710–9719, October 2021.

Tobias Fischer, Jonas Kulhanek, Samuel Rota Bulo, Lorenzo Porzi, Marc Pollefeys, and Peter Kontschieder. Dynamic 3d gaussian fields for urban areas. *arXiv preprint arXiv:2406.03175*, 2024.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *International Conference on Learning Representations (ICLR)*, 2024a.

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 (NeurIPS)*, 2024b.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2568–2577, June 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Yining Hong, Beide Liu, Maxine Wu, Yuanhao Zhai, Kai-Wei Chang, Linjie Li, Kevin Lin, Chung-Ching Lin, Jianfeng Wang, Zhengyuan Yang, Ying Nian Wu, and Lijuan Wang. Slowfast-vgen: Slow-fast learning for action-driven long video generation. In *International Conference on Learning Representations (ICLR)*, 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, 2023. URL https://arxiv.org/abs/2309.17080.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Team HunyuanWorld. Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. *arXiv preprint*, 2025.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. URL https://github.com/ultralytics/ultralytics.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5128–5137. IEEE, 2021.

Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5816–5825, 2021.

Hanyang Kong, Xingyi Yang, Xiaoxu Zheng, and Xinchao Wang. Worldwarp: Propagating 3d geometry with asynchronous video diffusion, 2025. URL https://arxiv.org/abs/2512.19678.

Zhengfei Kuang, Adam Sun, Liyuan Zhu, Tong Wu, Shengqu Cai, Jonathan Tremblay, Iro Armeni, Ehsan Adeli, Lior Yariv, and Gordon Wetzstein. Effective multi-sensor conditioning for street-view novel-view synthesis, 2026. URL https://arxiv.org/abs/2606.01590.

Moyang Li, Zihan Zhu, Marc Pollefeys, and Daniel Barath. Droid-slam in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25690–25699, October 2025.

Yiming Li, Zhiheng Li, Nuo Chen, Moonjun Gong, Zonglin Caps Lyu, Zehong Wang, Peili Jiang, and Chen Feng. Multiagent multitraversal multimodal self-driving: Open mars dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22041–22051, June 2024.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 12037–12047. Computer Vision Foundation / IEEE, 2025.

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

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, Paul Debevec, and Ning Yu. Vista4D: Video reshooting with 4d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32671–32682, June 2026.

Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Jiachen Lu, Ze Huang, Jiahui Zhang, Zeyu Yang, and Li Zhang. Wovogen: World volume-aware diffusion for controllable multi-camera driving scene generation. In *European Conference on Computer Vision (ECCV)*, 2024.

Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, Jiangmiao Pang, Mingmin Chi, Yu Qiao, and Kaipeng Zhang. Yume: An interactive world generation model. *arXiv preprint arXiv:2507.17744*, 2025.

Sudhanshu Mittal, Arian Mousakhan, Silvio Galesso, Karim Farid, Johannes Dienert, Rajat Sahay, and Thomas Brox. Orbis 2: A hierarchical world model for driving. *arXiv preprint arXiv:2607.15898*, 2026.

A. Mousakhan, S. Mittal, S. Galesso, K. Farid, and T. Brox. Orbis: Overcoming challenges of long-horizon prediction in driving world models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

NVIDIA, :, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai, 2025. URL `https://arxiv.org/abs/2511.00062`.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Junyoung Seo, Hyunwook Choi, Minkyung Kwon, Jinhyeok Choi, Siyoon Jin, Gayoung Lee, Junho Kim, JoungBin Lee, Geonmo Gu, Dongyoon Han, Sangdoo Yun, Seungryong Kim, and Jin-Hwa Kim. Seoul World Model: Grounding World Simulation Models in a Real-World Metropolis. In *European Conference on Computer Vision (ECCV)*, 2026.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14895–14904. IEEE, 2024.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. In *ICLR Workshop on Deep Generative Models for Higher-Order Tensors in Multi-Modal Applications*, 2019.

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 (CVPR)*, 2025.

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

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:49371–49393, 2026.

Zeqi Xiao, LAN Yushi, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024.

Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, and Sida Peng. Streetcrafter: Street view synthesis with controllable video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1389–1399. IEEE, 2023.

Wei Yin et al. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9043–9053, 2023.

Wei Yu, Runjia Qian, Yumeng Li, Liquan Wang, Songheng Yin, Sri Siddarth Chakaravarthy P, Dennis Anthony, Yang Ye, Yidi Li, Weiwei Wan, and Animesh Garg. Mosaicmem: Hybrid spatial memory for controllable video world models. *arXiv preprint arXiv:2603.17117*, 2026.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Sixiao Zheng, Minghao Yin, Wenbo Hu, Xiaoyu Li, Ying Shan, and Yanwei Fu. Versecrafter: Dynamic realistic video world model with 4d geometric control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40277–40290, June 2026.

Zangwei Zheng et al. Open-sora: Democratizing efficient video production for all, March 2024. URL `https://github.com/hpcaitech/Open-Sora`.

APPENDIX

A Overview of Persistent Scene Context Methods **16**

B Reconstruction and Rendering Details **16**

C Base World Model Details **17**

D Conditioning Mechanisms Details **18**

D.1 Latent Blending 18

D.2 Concatenation 18

D.3 AdaLN Adapter 18

E Geo-Reference Signal Quality **19**

F Hyperparameters **21**

G Evaluation Metrics **22**

G.1 Consistency metrics 22

G.2 Perceptual quality metrics 23

G.3 Trajectory adherence metrics 23

H Additional Ablations **23**

H.1 Geo-Reference Encoding 23

H.2 Multi-Camera Conditioning. 24

I Extended Results **25**

I.1 Long-Horizon Rollout Performance Per Dataset 25

I.2 Coverage vs. Quality. 25

I.3 Robustness to Localization Error 25

I.4 Scene Dynamics 26

I.5 Counterfactual Evaluation 27

J Qualitative Results **29**

J.1 Conditioning mechanism 29

J.2 Long-Horizon Rollouts 30

J.3 Coverage of the Geometric Reference 30

J.4 Localization error 31

J.5 Counterfactual Rollout 32

J.6 Multi-seed Rollout Diversity 33

J.7 Multi-view rollouts 34

A Overview of Persistent Scene Context Methods

Long-horizon world models need a representation of the scene that persists beyond the context window. Existing approaches differ in whether this memory is built from the model's own generated frames or grounded in real observations, and in whether it is aligned to past or to upcoming viewpoints. Table 7 summarizes them along these axes.

Table 7: Comparison of scene memory approaches. *Anchoring*: *retrospective* methods condition on geometry aligned to past or current viewpoints, while *prospective* methods render geometry to the future camera pose before generation. *Memory input*: the data used to construct the persistent scene representation, not the full input to the generative model. *Dyn. objects*: whether the conditioning signal itself represents dynamic objects, or instead delegates dynamics entirely to the generative prior. *Dyn. scene*: whether the method's intended output contains a dynamic scene with moving objects, or is restricted to static environments.

Method	Memory input	Anchoring	Memory representation	Dyn. objects	Dyn. scene
Generated content memory					
StreamingT2V (Henschel et al., 2025)	Previously generated frames	Retrospective	Frame buffer + global token	✓	✓
WorldMem (Xiao et al., 2025)	Previously generated frames	Retrospective	Retrieved frame index	✓	✓
SlowFastGen (Hong et al., 2025)	Previously generated frames	Retrospective	LoRA weights	✓	✓
WorldWarp (Kong et al., 2025)	Previously generated frames	Prospective	Online 3DGS	✗	✗
VMem (Li et al., 2025)	Previously generated frames	Retrospective	Surfel map	✗	✗
MosaicMem (Yu et al., 2026)	Previously generated frames	Retrospective	3D memory patches	✗	✓
SpMem (Wu et al., 2026)	Previously generated frames	Prospective	Static point maps	✗	✓
Real-world observations					
CityRAG (Chou et al., 2026)	Route imagery (multi-pass)	Retrospective	Image database	✓	✓
Seoul WM (Seo et al., 2026)	Pre-built geospatial map	Retrospective	Geospatial image map	✓	✓
GEN3C (Ren et al., 2025)	Monocular seed image	Prospective	Point cloud cache	✗	✗
GeoDrive (Chen et al., 2025a)	Monocular seed frame	Prospective	Decomposed point cloud	✓	✓
StreetNVS (Kuang et al., 2026)	LiDAR + multi-camera	Prospective	LiDAR point cloud	✗	✓
Vista4D (Lin et al., 2026)	Training frames	Prospective	4D point cloud	✓	✓
VerseCrafter (Zheng et al., 2026)	Input image + obj. trajs.	Prospective	4D control maps	✓	✓
Ours	Monocular driving log	Prospective	Static point cloud (SLAM)	✗	✓

B Reconstruction and Rendering Details

Point-cloud filtering. Filtering is performed once, *offline*, on the complete unprojected point cloud accumulated over the full traversal, rather than per rendered frame. Five sequential filters clean the raw point cloud, ordered from cheap local per-pixel tests to more expensive cross-view and voxel-density-based checks. The uncertainty and multi-view geometric filters reapply the consistency criteria that DROID-W already computes internally during optimization, now as explicit, tunable filters on the point cloud. We also add a disparity/depth threshold, a photometric color-agreement test, and a final 3D outlier removal step. Together, these filters suppress unreliable geometry arising from dynamic objects, occlusions, and mismatched correspondences, reducing erroneous reprojections and improving the stability of the subsequent rendering stage (thresholds in Table 10):

1. **Disparity filter.** Points whose source disparity is below a threshold, or whose depth exceeds d_{max} are discarded.
2. **Uncertainty filter.** Points whose source pixel has tracking uncertainty exceeding τ_{unc} are removed.
3. **Multi-view geometric filter.** A point survives only if it reprojects into at least N_{mv} other keyframes with relative depth error below τ_{mv}.
4. **Multi-view photometric filter.** A point is kept only if its reprojected color agrees (mean L1 $< \tau_{\text{photo}}$) with at least N_{photo} other frames.
5. **Voxel outlier filter.** Points whose voxel of side length s_{vox} contains fewer than k_{vox} points (including the point itself) are discarded.

Per-frame projection. In contrast to filtering, projection is performed *online*, once per target frame. Each world-space point is transformed into the camera frame of the target pose P_n, yielding camera coordinates (X, Y, Z), and projected with the pinhole model

$$u = f_x \frac{X}{Z} + c_x, \quad v = f_y \frac{Y}{Z} + c_y, \tag{4}$$

where f_x, f_y, c_x, c_y are the intrinsics of the target camera at the model's input resolution $H \times W$. During training and replay, P_n is the DROID-W pose of the recorded frame. For counterfactual and perturbed rollouts, it is the modified pose.

Each point is projected to the nearest pixel. Points with $Z \leq 0.01$ or outside the image are discarded. We use a single-pixel footprint and apply no hole filling. Instead of a hard z-buffer, all points projecting to the same pixel are composited with a depth-gated weighted blend. Let Z_{front} be the smallest depth among them. Each point receives the weight

$$w(Z) = \frac{1}{Z^2} \mathbb{1}\big[Z \leq (1+\delta)\, Z_{\text{front}}\big], \tag{5}$$

and the pixel color is the weighted average $\bar{\mathbf{c}} = \sum_i w(Z_i)\, \mathbf{c}_i / \sum_i w(Z_i)$. We use a permissive gate (δ=2), so occlusion is resolved mainly by the inverse-square weighting: a point at three times the front depth receives $1/9$ of the front point's weight. Compared to a nearest-point z-buffer, this averaging suppresses flickering caused by near-coincident points on the sparse cloud. The accompanying depth map stores Z_{front} per pixel. Rendering from the complete scene point cloud, rather than only the points observed up to the current frame, matches the simulation setting we target: the route has already been traversed, and the reconstructed map is available before rollout begins, so the model replays the logged route, or counterfactual variants of it, such as altered speed, lane offsets, or other traffic participants.

C Base World Model Details

We evaluate our conditioning on three base world models spanning different scales and architectures: Orbis v1 (Mousakhan et al., 2025), Orbis v2 (Mittal et al., 2026), and Cosmos2.5-Predict-2B (NVIDIA et al., 2025). We summarize only the components relevant to our conditioning and refer to the original works for details. For all learned variants, we initialize from the released weights and fine-tune the full backbone jointly with the conditioning modules; latent blending uses the released weights unchanged. Geo-reference frames are encoded with the frozen tokenizer of the respective model.

Orbis v1. A 469M-parameter continuous-latent autoregressive world model trained from scratch on 280h of front-camera driving video. Frames are encoded by a hybrid VQGAN-style tokenizer (86M encoder / 91M decoder) into semantic (DINOv2-distilled) and detail token streams. The generator is a causal STDiT of L=24 blocks (Swin-windowed spatial attention, causal temporal attention, adaLN-Zero (Peebles & Xie, 2023) timestep conditioning), trained with a rectified-flow objective (Lipman et al., 2023; Liu et al., 2023; Esser et al., 2024): it learns a velocity field $v_\theta(\mathbf{z}_t; \mathbf{z}^{1:N_{\text{ctx}}}, t)$ that transports noise to a clean latent via Euler integration. At inference, the model rolls out frame-by-frame: given N_{ctx} context latents it denoises one new frame, appends it to the context, drops the oldest frame, and repeats, generating long-horizon videos one step at a time.

Orbis v2. A hierarchical extension of Orbis v1 with a 512M high-level abstract predictor and a 555M low-level detail predictor, 1067M params total, both flow-matching-based. The abstract predictor operates on a compact 16-dim DINOv2-aligned latent at 2Hz, predicting subgoals over a 3s strided context. The detail predictor generates pixel-aligned frames at 10Hz conditioned on these subgoals, reusing the Orbis v1 tokenizer. Training uses diffusion-forcing pretraining followed by teacher-forcing fine-tuning, on 5,890h of video (OpenDV, BDD100K, PhysicalAI-AV, NATIX) at 512×288. Steering control is added via adaLN on the abstract branch, fine-tuned on 500h with IMU-derived trajectories. Rollout is likewise autoregressive but at two nested rates: the abstract predictor advances its sliding window one frame at a time to emit a new subgoal, and the detail predictor advances its own window 5 frames at a time, generating the intervening fine-grained frames conditioned on the latest predicted subgoal.

Cosmos2.5-Predict-2B. NVIDIA's 2B-parameter flow-matching video DiT unifying Text2World, Image2World, and Video2World generation. The backbone has 32 blocks, model dimension 2048, 16 attention heads, AdaLN-LoRA conditioning, and 3D RoPE. Video is compressed via a causal WAN2.1 VAE ($4 \times 8 \times 8$) plus $1 \times 2 \times 2$ patchification, generating $\sim$5.8s clips (93 frames, 16fps). Pretrained on 200M curated clips spanning driving, robotics, and human-dynamics domains, then refined with supervised fine-tuning, RL-based post-training, and timestep distillation. Since the fixed context window only covers $\sim$5.8s, longer rollouts are produced chunk-autoregressively whereby

the last frames of the previously generated chunk are fed back via a frame-replacement strategy to condition the denoising of the next chunk.

D Conditioning Mechanisms Details

D.1 Latent Blending

Latent blending injects the geo-reference signal without training or architectural changes by correcting the sampler state in latent space between ODE steps. The rectified-flow forward process noises a clean latent Z_0 as $z_t = (1-t)\,Z_0 + t\,\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, so that $t{=}0$ is clean and $t{=}1$ is pure noise. After each Euler step that brings the state to timestep t', we construct an anchor by applying this forward process to the geo-reference latent,

$$Z^{\text{anc}}(t') = (1-t')\,Z^{\text{geo}} + t'\,\epsilon, \tag{6}$$

i.e. the state that a denoising trajectory ending at Z^{geo} would visit at t'. During the first $\lfloor \rho_{\text{geo}} S \rfloor$ of the S ODE steps, the state is interpolated toward the anchor at covered positions,

$$z_{t'}[M] \leftarrow w(t')\,Z^{\text{anc}}(t')[M] + \big(1 - w(t')\big)\,z_{t'}[M], \qquad w(t') = \min(s_{\text{geo}}\,t',\,1), \tag{7}$$

where $[M]$ selects the tokens of the F_{pred} predicted frames that are covered by the geo-reference, broadcast over channels. Uncovered tokens and context frames are left unchanged. Blending is therefore restricted to the noise-dominated regime, where it fixes the coarse layout at little cost to diversity, while the remaining steps refine appearance freely. With $S{=}30$, $\rho_{\text{geo}}{=}0.2$ and $s_{\text{geo}}{=}1$, blending is active for the first 6 steps with $w \in [0.8, 1]$.

D.2 Concatenation

The most direct injection point is the patch embedding layer itself: rather than modulating features deeper in the network, the geo-reference signal is fused with the noisy input latent along the channel axis before the first projection. Every block downstream is an unmodified STDiT block, which makes this the cheapest learned variant.

The patch embedding is a $1{\times}1$ convolution $W_{\text{embed}} \in \mathbb{R}^{D \times C}$ mapping tokenizer-latent channels to the backbone width D. To admit the geo-reference latent, W_{embed} is widened along the input axis to $W'_{\text{embed}} \in \mathbb{R}^{D \times 2C}$,

$$W'_{\text{embed}} = \big[\, W_{\text{embed}} \;\big|\; \mathbf{0} \,\big], \tag{8}$$

copying the pretrained weights into the first C columns and zero-initializing the new ones (bias unchanged), so the model computes the pretrained function at initialization. Only the input side changes. The output head is untouched.

D.3 AdaLN Adapter

Each STDiT block of Orbis v1 modulates three sub-layers $\ell \in \{\text{sa}, \text{smlp}, \text{tmlp}\}$ (spatial self-attention, spatial MLP, temporal MLP) via adaLN-Zero (Peebles & Xie, 2023). For each sub-layer, it produces modulation vectors $\boldsymbol{\gamma}^{k,\ell} \in \mathbb{R}^D$, $k \in \{\text{shift}, \text{scale}, \text{gate}\}$, which are shared by all N tokens. Our adapter augments this global signal with a per-token additive offset derived from the geo-reference, injected at every layer. The idea is related to SPADE (Park et al., 2019), but the offset is added to the adaLN signal rather than replacing it, so timestep information is preserved at every token.

Each block contains a lightweight MLP applied independently to every geometry token $\mathbf{g}_{f,i}$,

$$\Delta(\mathbf{g}_{f,i}) = W_2\,\text{SiLU}(W_1\mathbf{g}_{f,i} + \mathbf{b}_1) + \mathbf{b}_2, \quad W_1 \in \mathbb{R}^{B_{\text{neck}} \times C},\; W_2 \in \mathbb{R}^{9D \times B_{\text{neck}}}, \tag{9}$$

where B_{neck} is the bottleneck width. The output is split into nine offsets $\boldsymbol{\delta}^{k,\ell}_{f,i} \in \mathbb{R}^D$, one for each modulation vector k and sub-layer ℓ, which are added to the base modulation:

$$\boldsymbol{\Gamma}^{k,\ell}_{f,i} = \boldsymbol{\gamma}^{k,\ell} + \boldsymbol{\delta}^{k,\ell}_{f,i}, \qquad \boldsymbol{\delta}^{k,\ell}_{f,i} = \mathbf{0} \;\text{ for context frames}, \tag{10}$$

i.e. only the F_{pred} predicted frames receive a geometric offset, while context frames pass through the base modulation unchanged. Sub-layer ℓ, with function $\mathcal{F}_\ell$, then updates each token as

$$\mathbf{x}_{f,i} \leftarrow \mathbf{x}_{f,i} + \mathbf{\Gamma}^{\text{gate},\ell}_{f,i} \odot \mathcal{F}_\ell\Big(\text{LN}(\mathbf{x})_{f,i} \odot \big(1 + \mathbf{\Gamma}^{\text{scale},\ell}_{f,i}\big) + \mathbf{\Gamma}^{\text{shift},\ell}_{f,i}\Big). \tag{11}$$

Causal temporal self-attention is unmodulated in the base block and remains so. Each of the L blocks contains one adapter, and the final layer, which uses only a shift and scale receives an adapter with a $2D$-dimensional output. The last layer of each adapter MLP is initialized with $\mathcal{N}(0, 0.02^2)$ weights and zero bias, so the offsets are small but non-zero at initialization.

E GEO-REFERENCE SIGNAL QUALITY

Coverage of the projected geo-reference signal ranges from roughly 50% to 78% of each frame across the single-camera splits, with reprojection quality tracking SLAM trajectory accuracy far less than expected: nuScenes attains the best trajectory estimate (0.74 m ATE) with a masked PSNR of 22.96 dB, while MARS, despite the lowest trajectory error (0.42 m) and highest coverage (68.5%), yields by far the weakest masked PSNR (12.83 dB), degrading further under cross-traversal reconstruction (MARS (swm): 77.5% coverage, 12.36 dB). In the multi-view setting, the forward camera clearly outperforms the oblique left/right views (e.g., 21.83 dB vs. 13.25/12.56 dB masked PSNR on the nuPlan train split), pulling the multi-view mean well below the single-view front-camera numbers. Full results are in Table 8.

Table 8: Coverage and reprojection quality of the geo-reference signal.

							PSNR	
View	Split	Dataset	Traj. RMSE	Scenes	Frames	Coverage	Full	Masked
Single	Train	nuPlan	3.25 m	1007	2,882,118	50.4%	8.88 dB	21.72 dB
	Val	nuPlan turns	3.34 m	128	270,263	62.6%	9.88 dB	22.42 dB
		Waymo	2.70 m	182	35,854	52.9%	9.07 dB	24.43 dB
		nuScenes	0.74 m	150	34,948	55.5%	10.27 dB	22.96 dB
		MARS	0.42 m	401	88,621	68.5%	7.28 dB	12.83 dB
	Cross-trav.	MARS	—	30	10,950	77.5%	7.81 dB	12.36 dB
Multi	Train	nuPlan (left)	3.25 m	986	2,844,008	26.6%	7.60 dB	13.25 dB
		nuPlan (front)	3.25 m	986	2,844,008	50.5%	8.89 dB	21.83 dB
		nuPlan (right)	3.25 m	986	2,844,008	31.2%	7.77 dB	12.56 dB
		nuPlan (mean)	3.25 m	986	2,844,008	36.1%	8.09 dB	15.88 dB
	Val	nuPlan (left)	3.24 m	225	505,139	35.7%	7.98 dB	13.20 dB
		nuPlan (front)	3.24 m	225	504,569	59.4%	9.56 dB	22.45 dB
		nuPlan (right)	3.24 m	225	505,139	37.3%	8.03 dB	12.82 dB
		nuPlan (mean)	3.24 m	225	504,949	44.1%	8.52 dB	16.16 dB

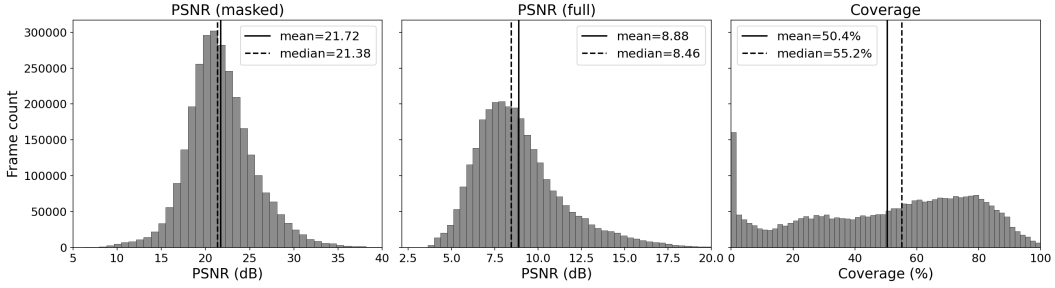

Figure 5: Distribution of geo-reference frame statistics over the training set, computed per frame: pixel coverage of the validity mask $\mathbf{M}$, the PSNR within masked region, and across the full geo-reference frame. Coverage varies substantially across frames, motivating conditioning mechanisms that tolerate partial and unreliable geometry rather than enforcing it uniformly.



Figure 6: The geo-reference frames differ in coverage and quality. While some frames have very low coverage (e.g., due to being at the beginning of the collected trajectory), other frames provide a dense signal. The quality, measured via PSNR in the covered region, differs due to floaters or dynamic objects missing from the image. While dynamic objects are removed from the point cloud before rendering, parked cars remain in the scene.



Figure 7: Example geo-reference validation frames from nuPlan turns, Waymo, nuScenes, our generated MARS validation set, and ANN-ARBOR-CITY-BENCH from 30 MARS cross-traversal pairs.

F HYPERPARAMETERS

DROID-W SLAM

Table 9: DROID-W SLAM hyperparameters.

Parameter	Symbol	Value
Tracker		
Metric3D input resolution	$H \times W$	616×1064
Metric depth scale (raw prediction $\hat{d}$)	—	$\hat{d} \cdot f_x/1000$
Metric depth clamp range	—	$[0,\ 300]$ m
DINOv2-small patch stride	s	14
Motion filter		
Minimum flow to admit keyframe	τ_{motion}	2.8 px
Frontend (local BA)		
Local BA iterations before keyframe decision	k_1	6
Local BA iterations after keyframe decision	k_2	4
Temporal edge window	T_w	25 frames
Keyframe pruning flow threshold	τ_{kf}	3.5 px
Backend (global BA)		
Global BA trigger interval	N_{ba}	20 keyframes
Final BA coarse steps	N_c	12
Final BA fine steps	N_f	20

POINT CLOUD RENDERING

Table 10: Point-cloud filtering and rendering hyperparameters.

Parameter	Symbol	Value
Point-cloud filters		
Disparity gate (fraction of mean disparity)	τ_{disp}	10^{-7}
Maximum projection depth	d_{max}	200 m
Max. tracking uncertainty	τ_{unc}	2.0
Min. keyframes for multi-view survival	N_{mv}	5
Max. relative depth error (multi-view)	τ_{mv}	0.03
Max. mean L1 reprojection color error	τ_{photo}	0.20
Min. frames for photometric survival	N_{photo}	2
Voxel outlier min. neighbors	k	2
Voxel outlier side length	ℓ	0.08 m
Weighted depth blending		
Splat radius	r	1 px
Hard depth gate threshold	δ	2.0
Soft depth decay rate (λ=0: hard gate)	λ	0

LATENT BLENDING

Table 11: Latent blending hyperparameters.

Parameter	Symbol	Value
Flow-matching schedule minimum noise	$\sigma_{\min}$	10^{-5}
Number of ODE integration steps	S	30
Sampler stochasticity (η=0: deterministic ODE)	η	0
Fraction of steps with blending correction	ρ_{geo}	0.2
Blend scale	s_{geo}	1.0

CROSS-ATTENTION AND ADALN TRAINING

Table 12: Training hyperparameters shared by the adaLN adapter and the cross-attention adapter, and architecture constants specific to each.

Parameter	Symbol	AdaLN	CrossAttn
Architecture			
STDiT blocks	L	24	24
Token grid	$H_{\text{lat}} \times W_{\text{lat}}$		18×32
Tokenizer latent channels	C		32
Adapter MLP bottleneck width	B_{neck}	256	—
Adapter output layer init	—	$\mathcal{N}(0, 0.02^2)$, $\mathbf{b}_2$=$\mathbf{0}$	—
Cross-attention gate init	α	—	0
Shared geo-reference projection	W_{proj}	—	$\mathbb{R}^{D\times C}$, shared across blocks
Training			
Optimizer	—		AdamW
Learning rate	lr		5×10^{-5}
Weight decay	—		0.01
LR warmup	—		5,000 steps (linear)
Geo-ref dropout rate	—		10%
Pose perturbation radius (optional)	R		3 m
Yaw / pitch jitter (optional)	—		$\pm 2^\circ$ / $\pm 1^\circ$

G EVALUATION METRICS

G.1 CONSISTENCY METRICS

All consistency metrics compare a generated frame G_t to its temporally aligned ground-truth frame I_t and are averaged over all frames within the respective horizon.

PSNR. The peak signal-to-noise ratio measures pixel-wise reconstruction fidelity,

$$\text{PSNR}(G_t, I_t) = 10 \log_{10} \frac{\text{MAX}^2}{\text{MSE}(G_t, I_t)},$$

where MAX is the maximum pixel intensity and MSE the mean squared error over all pixels and channels. PSNR rewards exact pixel alignment and therefore penalizes even sub-pixel spatial or sub-second temporal offsets between the generated and observed future.

LPIPS. The learned perceptual image patch similarity (Zhang et al., 2018) measures the distance between deep features of a pre-trained network ϕ,

$$\text{LPIPS}(G_t, I_t) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot \left(\phi_l(G_t)_{hw} - \phi_l(I_t)_{hw} \right) \right\|_2^2,$$

where ϕ_l denotes unit-normalized activations of layer l and w_l learned per-channel weights. We use the VGG variant. LPIPS correlates better with human judgments than pixel-wise metrics and degrades more gracefully under small misalignments.

mPSNR and **mLPIPS**, PSNR and LPIPS (Zhang et al., 2018) restricted to the static scene: dynamic agents (e.g. pedestrians and vehicle classes) are segmented with YOLOv8-seg (Jocher et al., 2023) in both frames and their union excluded, so the score reflects geometric fidelity rather than incidental disagreement in agent placement. We report this score to compare with the Seoul World Model paper (Seo et al., 2026).

LightGlue keypoint matches. We extract keypoints with SuperPoint and match them between the generated and ground-truth frames with LightGlue (Lindenberger et al., 2023), reporting the number of matches with confidence above 0.5. A high match count requires that the local scene structure

is preserved at approximately corresponding locations, making this a structure-aware consistency measure that tolerates moderate appearance differences.

Depth error. To isolate geometric from photometric consistency, we predict monocular depth for both frames with DepthAnything V3 (Lin et al., 2025) and report the absolute relative error after per-frame median-scale alignment,

$$\text{AbsRel}(G_t, I_t) = \frac{1}{|\Omega|} \sum_{p \in \Omega} \frac{\left| s\,\hat{d}(p) - d(p) \right|}{d(p)}, \qquad s = \frac{\text{median}(d)}{\text{median}(\hat{d})},$$

where $\hat{d}$ and d are the depth maps predicted from the generated and ground-truth frame, and Ω is the set of valid pixels. The scale factor s resolves the scale ambiguity of monocular depth. We deliberately use an estimator that does not share a backbone with the Metric3D (Yin et al., 2023) prior used in DROID-W, so that the conditioning signal and its evaluator remain independent.

G.2 Perceptual quality metrics

These metrics compare feature *distributions* of generated and real frames and are computed over the full evaluation set within each horizon.

Fréchet Inception Distance (FID). FID (Heusel et al., 2017) fits Gaussians $\mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ and $\mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ to Inception-v3 features of generated and real frames and reports their Fréchet distance,

$$\text{FID} = \|\boldsymbol{\mu}_g - \boldsymbol{\mu}_r\|_2^2 + \text{Tr}\big(\boldsymbol{\Sigma}_g + \boldsymbol{\Sigma}_r - 2(\boldsymbol{\Sigma}_g \boldsymbol{\Sigma}_r)^{1/2}\big).$$

It captures per-frame image realism but is blind to temporal artifacts such as flicker.

Img.Q. (Huang et al., 2024) is VBench's no-reference Imaging Quality score (MUSIQ-SPAQ (Ke et al., 2021), averaged per frame then per video). We report this score to compare with the Seoul World Model paper (Seo et al., 2026).

Fréchet Video Distance (FVD). FVD (Unterthiner et al., 2019) applies the same Fréchet formulation to spatio-temporal features of an I3D network computed on video clips rather than individual frames, and is therefore sensitive to temporal artifacts and motion realism in addition to per-frame quality.

FVD slope. Following Orbis v2 (Mittal et al., 2026), we split each rollout into T consecutive 2 s chunks (10 frames at 5 Hz), compute FVD per chunk, and report $(\text{FVD}_T - \text{FVD}_1)/(T-1)$. Lower values indicate slower degradation over the rollout.

G.3 Trajectory adherence metrics

We recover the camera path of each generated rollout with VGGT (Wang et al., 2025). As reference, we run VGGT on the corresponding ground-truth frames, so that both trajectories share VGGT's reconstruction errors, and align the generated path to the reference with a Sim(3) Umeyama fit. Since VGGT reconstructions are defined only up to scale, ADE is expressed in the units of the reference reconstruction rather than in meters.

Average Displacement Error (ADE). Given the aligned generated positions $\hat{\mathbf{p}}_1, \ldots, \hat{\mathbf{p}}_T$ and the reference positions $\mathbf{p}_1, \ldots, \mathbf{p}_T$,

$$\text{ADE} = \frac{1}{T} \sum_{t=1}^{T} \|\hat{\mathbf{p}}_t - \mathbf{p}_t\|_2.$$

This metric quantifies pointwise deviations and is sensitive to minor spatial misalignments.

H Additional Ablations

H.1 Geo-Reference Encoding

We compare three encodings of the geo-reference frames R_n, all with gated cross-attention on Orbis v1 (Table 13).

(i) Orbis tokenizer (default). RGB renders are encoded with the frozen hybrid tokenizer of Orbis v1 (Bai et al., 2024) in its continuous mode. It concatenates a DINOv2-distilled (Oquab et al., 2024) semantic branch and a detail branch, so the condition lies in the same latent space as the generated frames.

(ii) DINOv2. RGB renders are encoded with a frozen DINOv2 encoder (Oquab et al., 2024). Its features serve directly as condition tokens, providing a purely semantic representation without a reconstruction objective. Frames encoded this way are never compressed unlike the ones encoded via the Orbis tokenizer.

(iii) Depth + Canny. Depth maps rendered from the filtered point cloud are replicated to three channels and encoded with the Orbis tokenizer, together with Canny edge maps of the RGB renders. This signal carries geometry and edges but no color.

Table 13: Geo-reference encoding ablation on NUPLAN-TURNS (Orbis v1, cross-attention): RGB renders encoded with the Orbis tokenizer or a frozen DINOv2 encoder, and depth and Canny-edge renders encoded with the Orbis tokenizer. Metrics over frames 0–49 (short) and 50–99 (long) at 5 Hz. Best per column in **bold**, second best underlined.

	LPIPS ↓		FVD ↓		KP Matches ↑		Depth AbsRel ↓		ADE ($\times 10^{-3}$) ↓	
Encoder	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long
Orbis tokenizer	**0.393**	**0.437**	**128.5**	188.4	**225.1**	**148.4**	**0.167**	0.206	**3.6**	**9.9**
DINOv2	0.405	0.443	139.3	**184.8**	193.0	121.0	0.170	**0.200**	3.8	10.4
Depth+Canny	0.429	0.503	166.0	297.6	172.4	81.0	0.189	0.254	3.9	10.9

H.2 MULTI-CAMERA CONDITIONING.

A natural extension is synchronized multi-camera generation with the front, left, and right cameras. We add another cross-attention block performing epipolar band cross-attention between nearby frames. We train this model with and without geo-reference conditioning. The shared point cloud is rendered separately for each camera, using the rig extrinsics for the side views, and each camera has its own cross-attention conditioning module. Geo-reference conditioning improves every metric in every view (Table 14). The front view benefits most: long-horizon KP matches triple ($60.1 \rightarrow 180.0$) and long-horizon FVD drops by 42% ($360.7 \rightarrow 209.6$), consistent with the front camera covering the most geometry-rich part of the reconstruction. For the side cameras, long-horizon KP matches roughly double, with smaller absolute gains, as the side views cover sparser regions of the reconstruction. Averaged over views, cross-attention lowers long-horizon FVD from 272.1 to 191.2 and raises long-horizon KP matches from 49.4 to 117.2, confirming that the shared geometric anchor enforces structural consistency across the full rig.

Table 14: Multi-view rollouts on NUPLAN-TURNS for the left, front, and right cameras, without (Base) and with cross-attention conditioning. *Average* is the mean over the three views. Metrics over frames 0–49 (short) and 50–99 (long) at 5 Hz. Cross-attention improves every metric in every view, best average values in **bold**.

		LPIPS ↓		FVD ↓		KP Matches ↑		Depth AbsRel ↓		ADE ($\times 10^{-3}$) ↓	
Variant	View	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long
Base	Left	0.478	0.556	145.1	210.7	130.8	47.2	0.231	0.303	13.3	28.5
	Front	0.448	0.541	185.3	360.7	171.1	60.1	0.214	0.314	4.9	14.7
	Right	0.487	0.564	156.9	245.0	123.3	40.9	0.231	0.297	10.3	26.1
	Average	0.471	0.554	162.4	272.1	141.7	49.4	0.225	0.305	9.5	23.1
Cross-attn	Left	0.432	0.499	126.2	175.9	165.8	87.1	0.194	0.245	12.3	27.2
	Front	0.337	0.416	135.4	209.6	265.0	180.0	0.144	0.208	4.2	12.7
	Right	0.436	0.506	130.6	188.0	165.2	84.5	0.186	0.244	9.8	24.8
	Average	**0.402**	**0.474**	**130.7**	**191.2**	**198.7**	**117.2**	**0.175**	**0.233**	**8.7**	**21.6**

I Extended Results

I.1 Long-Horizon Rollout Performance Per Dataset

In Table 15 we report the values obtained on nuPlan turns, the longest sequences from the Waymo validation set, the MARS dataset, and on nuScenes validation.

Table 15: Comparison against baselines on nuPlan-Turns, Waymo, MARS, and nuScenes. Short (s) / long (l) horizons: frames 0–39 / 40–79 (8 s each, 16 s total) for every dataset (both at 5 Hz). ADE in $\times 10^{-3}$ m. Best per column and dataset in **bold**, second best underlined. FVD slope over $T{=}8$ chunks (frames 0–79), see Section G.

	Method	LPIPS ↓		FVD ↓			KP Matches ↑		Depth AbsRel ↓		ADE ↓	
		Short	Long	Short	Long	Slope	Short	Long	Short	Long	Short	Long
nuPlan-T.	Orbis v1	0.451	0.545	249.9	440.4	56.3	181.1	68.4	0.217	0.305	5.0	13.9
	Orbis v2	0.440	0.536	222.3	333.8	33.9	178.7	63.6	0.218	0.310	4.7	12.1
	Cosmos 2.5	0.447	0.543	214.8	426.8	54.4	**304.2**	92.6	0.214	0.308	3.5	**6.7**
	GeoRef Signal	0.451	0.438	1325.1	1291.3	–	212.9	226.3	0.352	0.318	4.8	8.4
	Ours (Orbis v1)	0.387	0.428	**120.0**	**174.1**	**13.7**	237.7	159.7	0.162	0.197	**3.0**	8.2
	Ours (Orbis v2)	**0.304**	**0.360**	131.9	183.3	16.2	298.1	**249.4**	**0.104**	**0.132**	3.4	9.1
Waymo	Orbis v1	0.371	0.469	353.5	579.7	67.2	237.6	122.3	0.184	0.280	3.9	13.7
	Orbis v2	0.336	0.446	293.4	440.4	49.2	243.9	122.8	0.167	0.267	3.4	12.1
	GeoRef Signal	0.541	0.533	2118.0	1686.8	–	164.0	171.7	0.311	0.350	3.0	5.2
	Ours (Orbis v1)	0.313	0.388	276.2	359.9	**25.6**	286.6	208.6	0.149	0.216	**1.6**	**4.4**
	Ours (Orbis v2)	**0.267**	**0.341**	**249.8**	**346.2**	30.7	**312.8**	**251.2**	**0.116**	**0.169**	1.9	4.5
MARS	Orbis v1	0.453	0.531	311.1	528.5	64.3	216.5	116.6	0.247	0.336	4.8	11.0
	Orbis v2	0.412	0.512	285.3	536.6	57.2	218.5	104.5	0.223	0.333	3.3	7.3
	GeoRef Signal	0.497	0.477	2878.3	2480.9	–	212.9	223.6	0.328	0.334	3.5	4.1
	Ours (Orbis v1)	**0.315**	**0.394**	**207.4**	**365.1**	39.5	**287.5**	216.7	**0.145**	0.213	**1.9**	**3.4**
	Ours (Orbis v2)	0.366	0.430	247.4	404.1	**38.1**	272.6	**224.3**	**0.145**	**0.200**	2.5	3.9
nuScenes	Orbis v1	0.392	0.486	380.9	609.1	68.2	233.9	112.0	0.207	0.306	5.4	20.1
	Orbis v2	0.384	0.480	330.8	476.4	43.1	232.1	102.7	0.200	0.289	4.8	14.0
	GeoRef Signal	0.537	0.524	1858.1	1562.4	–	164.5	178.9	0.372	0.367	4.3	10.8
	Ours (Orbis v1)	0.364	0.434	304.9	**400.9**	**30.1**	259.3	152.9	0.173	0.241	**2.7**	10.0
	Ours (Orbis v2)	**0.314**	**0.387**	**290.3**	414.3	36.7	**300.9**	**219.1**	**0.137**	**0.185**	**2.7**	**9.1**

I.2 Coverage vs. Quality.

Table 16 varies the coverage of the geo-reference signal via the multi-view depth threshold τ_{mv}. Denser signals improve the rollout, but the additional points are less reliable: PSNR over covered pixels drops from 22.8 dB at 34% coverage to 21.9 dB at 79–87%, and the looser filtering admits floaters and artifacts (Figure 13). Our conservative coverage score of 62% is a trade-off between signal quality and coverage.

I.3 Robustness to Localization Error

Table 17 reports the full set of metrics for the localization-error experiment of Section 5.5 on Orbis v1. Each model is evaluated on both aligned and misaligned geo-reference renders, so that the cost of training with pose perturbations and the cost of encountering a displaced signal at inference can be read separately. The unconditioned backbone is included as the score a model would reach by ignoring the signal entirely.

Table 16: Geo-reference coverage ablation on NUPLAN-TURNS (Orbis v1, cross-attention). Coverage is varied via the multi-view depth threshold τ_{mv}. *Oracle*: shows the model being conditioned on ground truth images, also containing dynamic objects. Metrics over frames 0–49 (short) and 50–99 (long) at 5 Hz. Best per column in **bold**, second best underlined; Oracle is excluded from the ranking.

Coverage	PSNR (covered) (dB) ↑	LPIPS ↓		FVD ↓		KP Matches ↑		Depth AbsRel ↓		ADE ($\times 10^{-3}$) ↓	
		Short	Long	Short	Long	Short	Long	Short	Long	Short	Long
6%	22.49	0.446	0.523	182.3	360.0	161.3	69.4	0.216	0.300	4.2	11.8
24%	22.74	0.427	0.498	165.0	297.5	175.0	82.6	0.192	0.267	3.7	10.6
34%	**22.79**	0.418	0.484	159.1	264.4	184.8	93.3	0.185	0.252	3.7	10.4
54%	22.55	0.401	0.451	137.8	208.1	210.9	126.5	0.172	0.221	3.6	10.1
62% (Ours)	22.42	0.393	0.437	128.5	188.4	225.1	148.4	0.167	0.206	3.6	9.9
79%	21.88	0.383	**0.419**	**123.0**	**171.5**	247.3	**177.3**	0.157	**0.193**	3.6	10.1
87%	21.86	**0.383**	0.421	125.6	180.1	**248.8**	175.4	**0.157**	0.193	**3.4**	**9.5**
Oracle	–	0.353	0.361	109.0	138.8	322.8	292.9	0.138	0.149	3.0	8.0

Table 17: The nominal reprojection (*aligned eval*) and one rendered from a virtual viewpoint displaced onto a 3 m circle (*misaligned eval*). *Orbis v1* is the unconditioned backbone, which takes no geo-ref signal at all — it is what a run would score if it had learned to ignore a signal it cannot trust. Metrics averaged over frames 0–49 (s) and 50–99 (l) at 5 Hz. ADE in $\times 10^{-3}$ m.

Training	Evaluation	LPIPS ↓		FVD ↓		KP Matches ↑		Depth AbsRel ↓		ADE ($\times 10^{-3}$) ↓	
		Short	Long	Short	Long	Short	Long	Short	Long	Short	Long
	Orbis v1	0.468	0.554	276.1	501.3	160.7	61.4	0.232	0.316	6.0	16.9
Pose error 0m	Pose error 0m	0.393	0.437	128.5	188.4	225.1	148.4	0.167	0.206	3.6	9.9
	Pose error 0–3m	0.477	0.500	502.5	728.4	126.6	84.3	0.234	0.254	19.1	28.0
Pose error 0–3m	Pose error 0m	0.456	0.522	175.4	285.3	158.2	65.6	0.221	0.290	4.3	11.5
	Pose error 0–3m	0.458	0.525	177.4	299.3	156.7	65.9	0.223	0.297	4.2	12.1

I.4 SCENE DYNAMICS

Figure 8 tracks the number of dynamic objects over the rollout, measured as the difference between the generated and the recorded frames. The anchored models stay within the range of their unconditioned counterparts throughout, so conditioning on static geometry does not remove moving agents as the rollout proceeds.

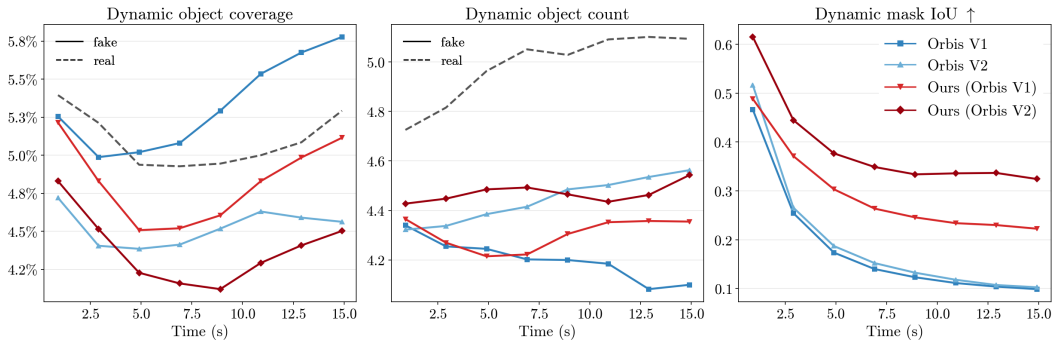

Figure 8: Dynamic object count over the rollout, comparing the base world models with their anchored variants. Values are the difference between the generated and the recorded frames, where zero is ideal.

I.5 COUNTERFACTUAL EVALUATION

Throughout, *GT* denotes the real recorded driving log (real camera frames, real ego odometry), the same log droidw reconstructed the persistent geometry from. *Nominal rollout* and *counterfactual rollout* both denote videos *generated* by our model – one conditioned on the geo-ref render along the unperturbed (replay) trajectory, the other along the perturbed (lateral/wiggle/speed) trajectory. Neither is ground truth; GT appears only as the fixed reference both are measured against.

Measuring compliance as the raw displacement between the counterfactual and nominal rollout is not enough: VGGT's own pose-estimation noise can push the reconstructed trajectory in an arbitrary direction, and this reconstruction error is easily mistaken for a genuine lateral shift.

A more fundamental problem is that ADE cannot see lateral compliance at all. Umeyama's similarity alignment jointly fits scale, rotation, and translation, which removes a pure translation exactly; since each rollout is additionally re-origined at its own first generated frame, any lateral offset – which after droidw's ramp is a fixed offset from the nominal path – is fit away entirely.

To recover a genuine compliance signal we anchor every rollout to real, shared geometry instead of aligning it to itself:

1. Run VGGT jointly over the 3 real conditioning frames and the 100 generated frames, for the nominal and counterfactual rollout separately. Since the conditioning frames are real and identical across both, VGGT's own canonicalization to its first input frame fixes a shared origin and heading for both reconstructions.
2. Fit a single scalar scale s^* per rollout on the forward (along-track) component only, by least squares against GT odometry:
$$s^* = \frac{\sum_t f(t)\, f^{\mathrm{gt}}(t)}{\sum_t f(t)^2}, \tag{12}$$
where $f(t)$ is VGGT's forward coordinate for that rollout at frame t and $f^{\mathrm{gt}}(t)$ is the corresponding real GT forward distance. Because only the along-track component enters this fit, s^* cannot absorb a sideways shift.
3. Gate out sequences where VGGT disagrees with GT motion, using the *nominal* rollout's own fit quality only (outcome-independent, since it does not depend on whether the counterfactual complied):
$$s^* > 0 \quad \text{and} \quad R^2 = 1 - \frac{\sum_t \left(s^* f(t) - f^{\mathrm{gt}}(t)\right)^2}{\sum_t \left(f^{\mathrm{gt}}(t) - \overline{f^{\mathrm{gt}}}\right)^2} \;\geq\; 0.9. \tag{13}$$
4. Measure the signed cross-track distance of each rollout's scaled VGGT trajectory from the real GT path, separately for the counterfactual and the nominal rollout:
$$d_{\mathrm{cf}}(t) = s^*_{\mathrm{cf}}\, g_{\mathrm{cf}}(t), \qquad d_{\mathrm{nom}}(t) = s^*_{\mathrm{nom}}\, g_{\mathrm{nom}}(t), \tag{14}$$
where $g(t)$ is VGGT's lateral coordinate at frame t for that rollout. Both are measured against the same real GT path, not against each other, so any lateral bias intrinsic to VGGT's reconstruction of that scene – present regardless of which trajectory was commanded – appears in both.
5. Take the paired difference
$$\Delta(t) = d_{\mathrm{cf}}(t) - d_{\mathrm{nom}}(t), \tag{15}$$
which cancels shared VGGT reconstruction bias between the two generated rollouts and isolates the shift induced specifically by the counterfactual command.

Wiggle. The wiggle perturbation is a sine-wave lateral offset (period 10 m with a maximal offset of 0.5 meters), so its commanded shift alternates sign along the rollout. Reporting a single scalar Δ (as for lateral) is meaningless here: the positive and negative half-cycles average toward zero by construction, even under full compliance. Instead, for each sample i we use droidw's own per-frame commanded offset trace $e_i(t)$ – the actual lateral component of the perturbed trajectory at frame t, which already accounts for how the ramp and the vehicle's heading through turns modulate the raw

sine amplitude – and fit a single regression slope through the origin between the measured paired shift and the commanded trace:

$$\beta_i = \frac{\sum_t \Delta_i(t)\, e_i(t)}{\sum_t e_i(t)^2}. \tag{16}$$

$\beta_i = 1$ means the measured shift tracks the commanded sine curve exactly (full compliance); $\beta_i = 0$ means no relationship. We report the median β_i across samples, together with the fraction of samples with $\beta_i > 0$ and its binomial p-value.

Speed. A speed change is not a translation: it changes how far along the route – through the turn – the rollout has progressed by a given frame, which is a change in the trajectory's own shape that Umeyama's similarity alignment cannot fit away. Plain ADE is therefore usable for speed, provided it is measured against the correctly speed-scaled target rather than the nominal-speed one. For each commanded multiplier $\mu \in \{0.85, 1.15\}$ we build a target $\text{gtcf}_\mu(t)$ by resampling real GT odometry along its own arclength at rate μ (the same route, traversed $\mu\times$ faster or slower), and compute per-sample ADE (VGGT trajectory re-origined at frame 0, aligned with $\mathrm{umeyama_2d}$) of both the counterfactual and the nominal rollout against this *same* target:

$$a_{\text{own}} = \mathrm{ADE}\big(V_{\text{cf}},\, \text{gtcf}_\mu\big)\,, \qquad a_{\text{nom}} = \mathrm{ADE}\big(V_{\text{nom}},\, \text{gtcf}_\mu\big)\,. \tag{17}$$

Compliance is the paired comparison $a_{\text{own}} < a_{\text{nom}}$ (fraction of samples, binomial p-value): a rollout that actually changed speed should track the μ-scaled target more closely than one that did not. We additionally check specificity against the *opposite* multiplier's target $\text{gtcf}_{\mu'}$, confirming $a_{\text{own}} < a_{\text{opp}} = \mathrm{ADE}(V_{\text{cf}}, \text{gtcf}_{\mu'})$.

Table 18: Counterfactual trajectory perturbations of OURS on NUPLAN-TURNS. FID/FVD/ADE are computed against the *nominal* ground-truth rollout and quantify how far the generation drifts from the unperturbed future (lower is not better here – nominal is the reference, not a target). ADE cannot see lateral compliance at all (see text), so Compliance is a separate measurement, in different units per row: for Lateral, the median counterfactual-minus-nominal sideways shift in metres, sign matching the commanded direction (samples passing a nominal-only VGGT quality gate; see fraction of samples in that direction and its p-value); for Wiggle, whose shift alternates sign and so cannot be summarized as a single offset, the per-sample regression slope of measured shift on the commanded sine curve (1 = fully follows, 0 = ignores); for Speed, the fraction of samples closer to their GT-odometry speed target than the nominal rollout is. ADE in $\times 10^{-3}$ m.

Perturbation	FID ↓ Short	FID ↓ Long	FVD ↓ Short	FVD ↓ Long	ADE ($\times 10^{-3}$) ↓ Short	ADE ($\times 10^{-3}$) ↓ Long	Compliance (p)
CFG = 1 (default guidance)							
Lateral $+1$ m	18.4	26.0	155.0	224.5	4.4 ± 13.9	10.8 ± 13.7	+0.66 m (181/237, $p = 2e{-}16$)
Lateral $+0.5$ m	17.0	24.5	139.3	202.7	4.0 ± 12.8	10.3 ± 14.1	+0.47 m (185/237, $p = 1e{-}18$)
Nominal	16.2	23.5	128.5	188.4	3.6 ± 13.8	9.9 ± 13.4	–
Lateral -0.5 m	16.9	24.5	140.9	202.8	4.0 ± 14.2	10.2 ± 13.4	-0.42 m (176/237, $p = 4e{-}14$)
Lateral -1 m	18.3	26.1	156.8	224.0	4.4 ± 14.4	11.3 ± 14.8	-0.75 m (176/237, $p = 4e{-}14$)
Wiggle 0.5 m	17.2	24.0	171.9	250.4	4.8 ± 14.1	11.8 ± 14.6	slope 0.86 (210/237, $p = 3e{-}36$)
Speed $0.85\times$	19.5	27.5	168.8	208.7	6.3 ± 15.9	11.9 ± 13.9	60% (239/400, $p = 1e{-}4$)
CFG = 2							
Lateral $+1$ m	17.2	23.5	146.3	200.7	4.4 ± 16.0	10.1 ± 14.2	+0.71 m (178/237, $p = 5e{-}15$)
Lateral $+0.5$ m	15.7	21.2	126.8	165.8	3.8 ± 14.7	9.0 ± 13.1	+0.52 m (179/237, $p = 2e{-}15$)
Nominal	14.8	19.3	115.2	151.2	3.6 ± 15.9	9.5 ± 14.0	–
Lateral -0.5 m	15.7	21.6	125.8	169.3	3.9 ± 13.3	9.5 ± 14.3	-0.40 m (176/237, $p = 4e{-}14$)
Lateral -1 m	17.5	23.9	147.7	200.7	4.5 ± 15.2	10.4 ± 14.8	-0.89 m (183/237, $p = 1e{-}17$)
Wiggle 0.5 m	15.6	21.3	155.0	213.2	4.7 ± 13.1	10.5 ± 14.4	slope 1.18 (218/237, $p = 5e{-}44$)
Speed $0.85\times$	18.1	23.0	157.6	181.9	7.1 ± 14.7	10.7 ± 13.3	62% (250/400, $p = 7e{-}7$)

J Qualitative Results

J.1 Conditioning mechanism

Figures 9 and 10 show representative rollouts for each injection mechanism. Cross-attention recovers fine scene structure that latent blending and concatenation are unable to retain over time.



Figure 9: Qualitative comparison of conditioning mechanisms on nuPlan-Turns (sequence 123).



Figure 10: Qualitative comparison of conditioning mechanisms on nuPlan-Turns (sequence 131).

J.2 Long-Horizon Rollouts

Figures 11 and 12 highlight how the geo-reference information anchors the rollouts in the actual surroundings of the original drive.

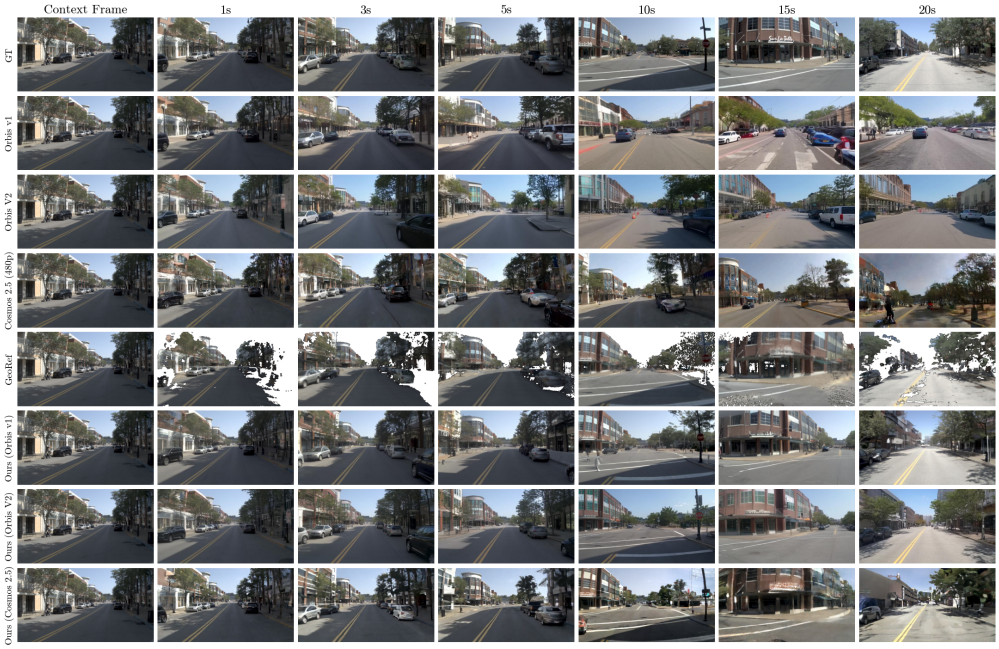

Figure 11: Qualitative comparison of the different methods on nuPlan-Turns (sequence 001).

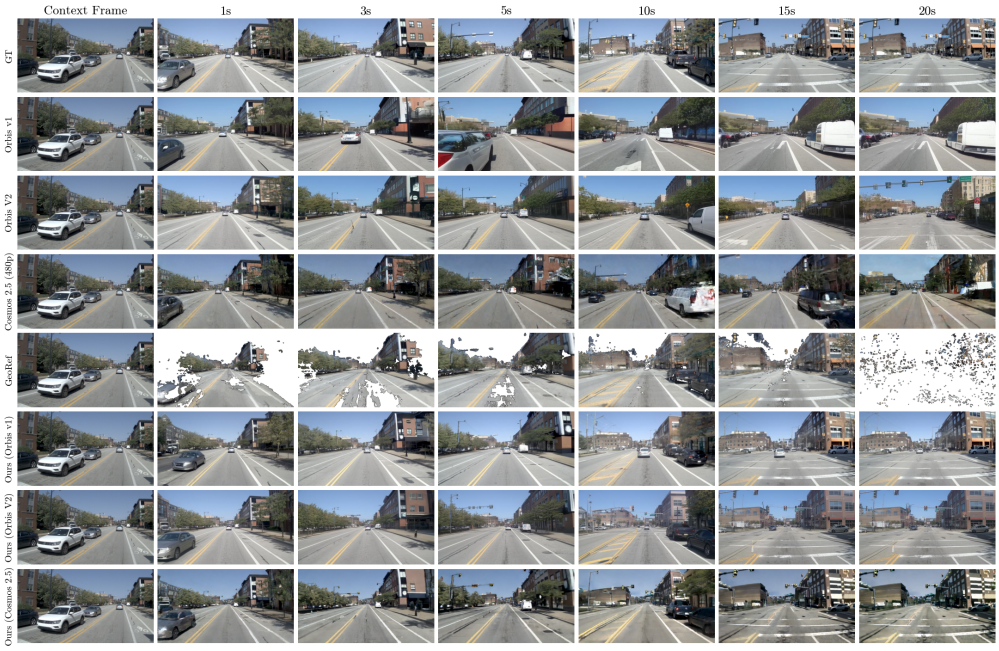

Figure 12: Qualitative comparison of the different methods on nuPlan-Turns (sequence 357).

J.3 Coverage of the Geometric Reference

Figures 13 to 15 illustrate the coverage sweep of Table 16: as point-cloud coverage increases, background structure sharpens and long-horizon drift visibly decreases.

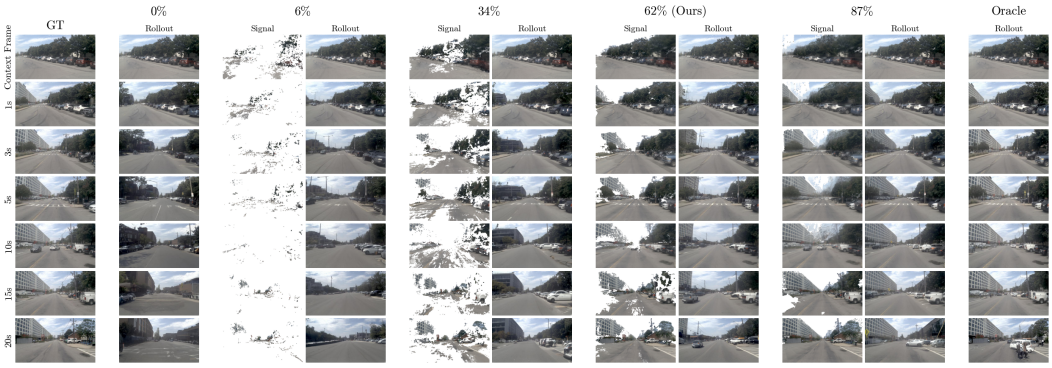

Figure 13: Qualitative effect of geo-reference coverage on nuPlan-Turns (sequence 116).

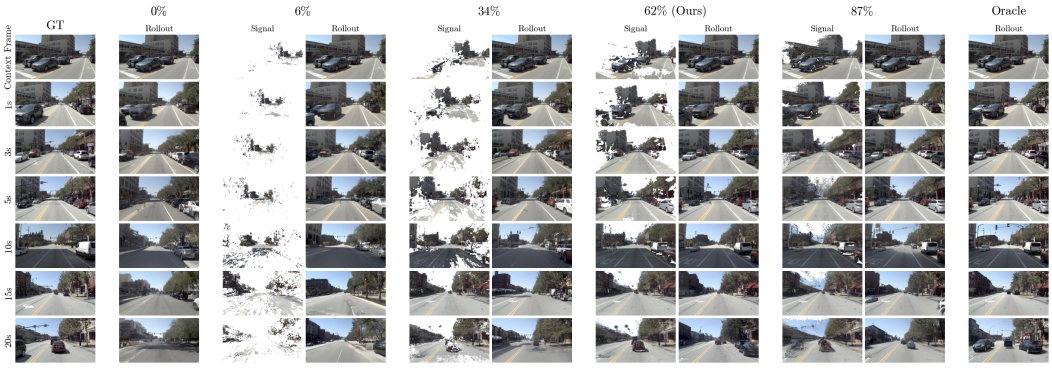

Figure 14: Qualitative effect of geo-reference coverage on nuPlan-Turns (sequence 123).

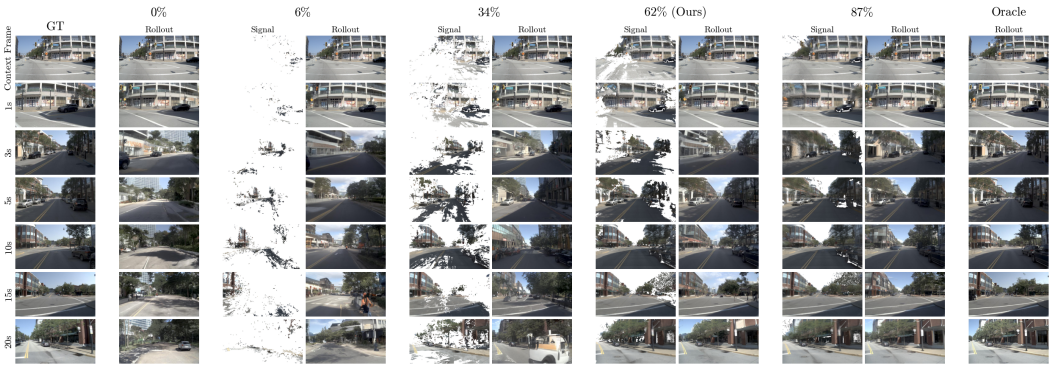

Figure 15: Qualitative effect of geo-reference coverage on nuPlan-Turns (sequence 000).

J.4 Localization error

Figures 16 and 17 show rollouts under a 3m pose perturbation (Section 5.5, Table 17): the aligned-trained model visibly chases the displaced geometry, while the perturbation-trained model ignores the misalignment and stays on the recorded route.

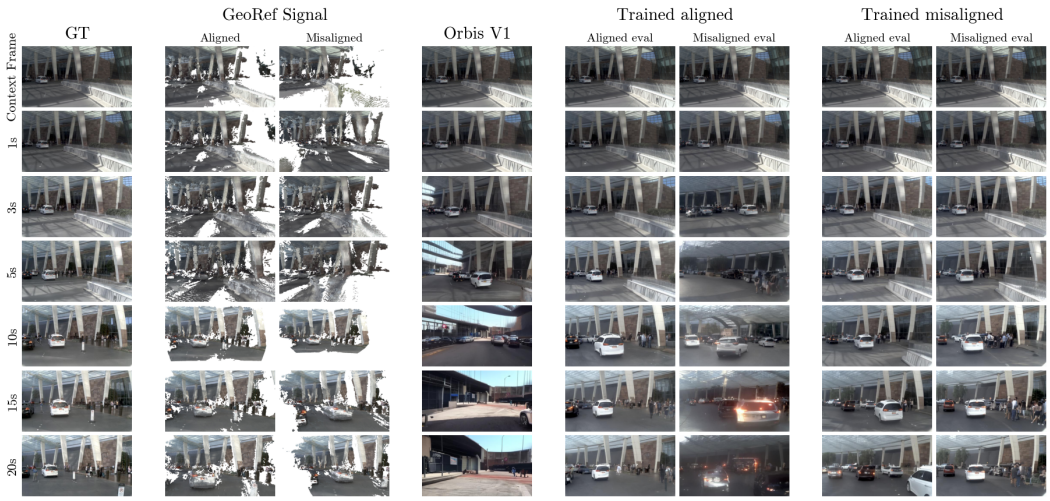

Figure 16: Rollout under a 3 m pose perturbation on nuPlan-Turns. The aligned-trained model follows the displaced geometry, while the perturbation-trained model stays on the recorded route.

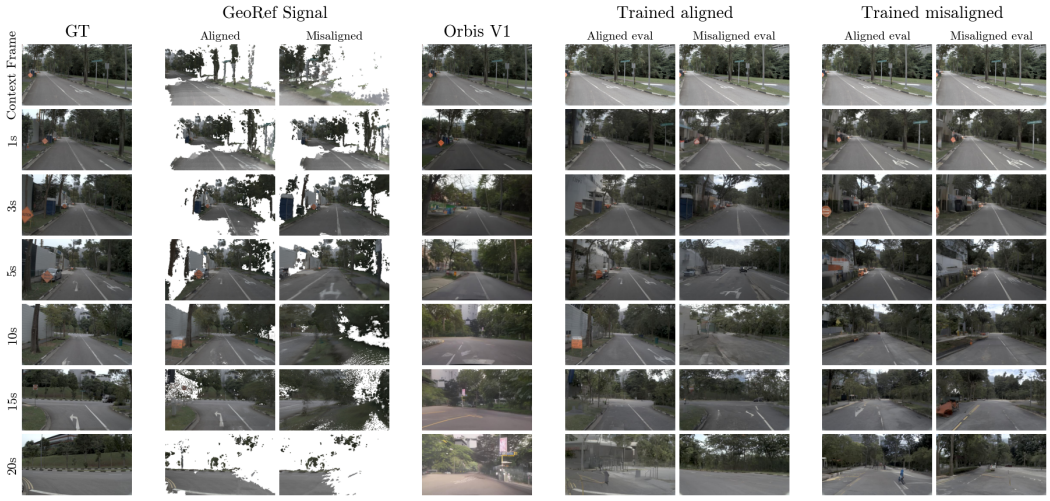

Figure 17: Rollout under a 3 m pose perturbation on nuPlan-Turns.

J.5 Counterfactual Rollout

Figures 18 and 19 illustrate counterfactual rollouts with lateral offsets ranging from $+1$ to -1 meters. The bottom two rows highlight the resulting sinusoidal lateral trajectory ("wiggle") and the corresponding reduction in speed. We refer the reader to the supplementary videos for a more detailed qualitative assessment.



Figure 18: Counterfactual rollouts on nuPlan-Turns. Rows show lateral offsets from $+1$ to -1 m, followed by the sinusoidal wiggle and the reduced-speed command.



Figure 19: Counterfactual rollouts on nuPlan-Turns

J.6 Multi-seed Rollout Diversity

Figures 20 and 21 vary only the sampling seed under identical geo-reference conditioning on nuPlan and nuScenes, showing that the geo-reference signal constrains scene structure without collapsing the diversity of dynamic content across samples.

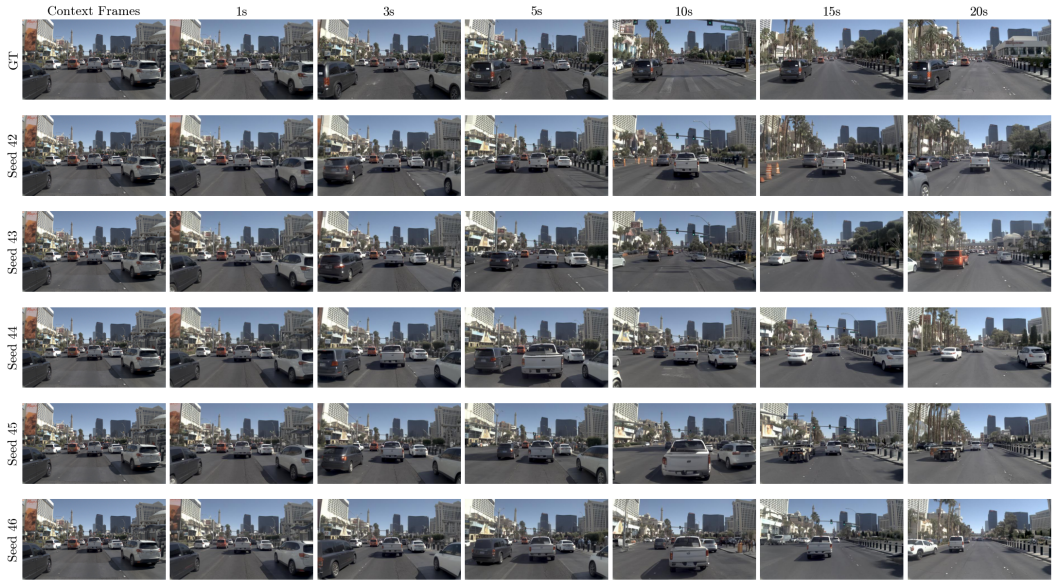

Figure 20: Rollouts from different sampling seeds under identical geo-reference conditioning on nuPlan-Turns. Scene structure is shared across seeds, while the dynamic content differs.



Figure 21: Rollouts from different sampling seeds under identical geo-reference conditioning on nuScenes

J.7 MULTI-VIEW ROLLOUTS

Figures 22 to 24 show synchronized front/left/right rollouts corresponding to Table 14: cross-attention conditioning keeps geometry consistent across all three views, most visibly on the forward-facing camera.

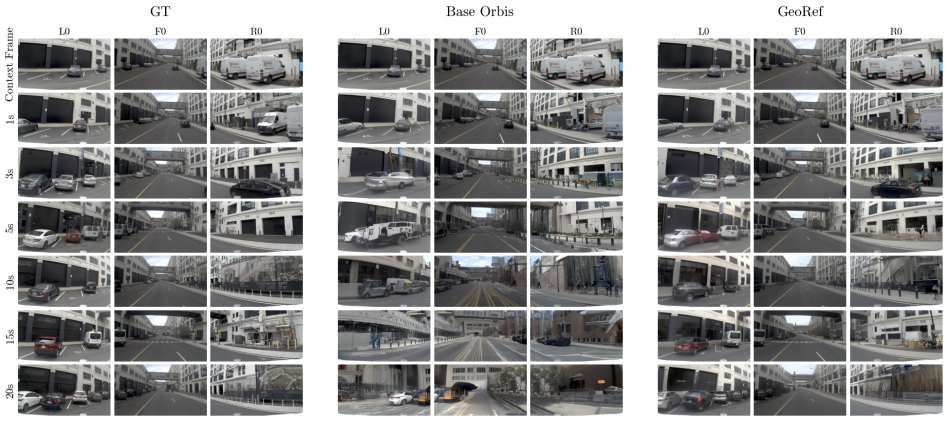

Figure 22: Synchronized left, front and right rollouts on nuPlan-Turns, without and with geo-reference conditioning.

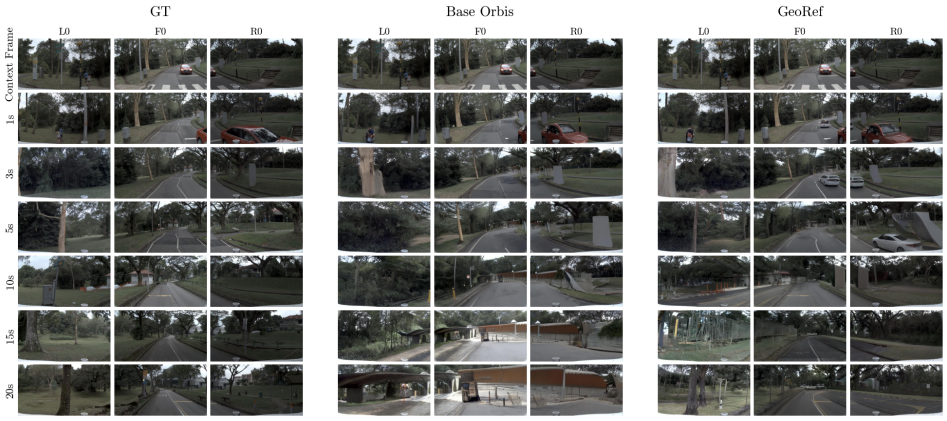

Figure 23: Synchronized multi-view rollouts on nuPlan-Turns.



Figure 24: Synchronized multi-view rollouts on nuPlan-Turns.