GeoDYN: 4D Geometry-Grounded Video Diffusion for Dynamic View Synthesis

Anonymous authors
Paper under double-blind review

Abstract

Dynamic novel view synthesis requires both geometric consistency with the observed video and the ability to synthesize unobserved content. Existing geometry-conditioned methods render a reconstructed 4D prior into pixel-space conditions, so reconstruction errors propagate directly into the video diffusion backbone, forcing it to decide on its own what to trust. We argue that how the prior is delivered matters as much as its quality, and introduce **GeoDYN**, which uses a 4D Gaussian prior to transport source video features to target camera views and timesteps at multiple DiT blocks of the backbone, instead of passing rendered appearance as input. Each GeoDYN block further refines the transported features, preserving source context while suppressing unreliable geometry instead of propagating its errors. In a controlled comparison with matched 1.3B backbones, GeoDYN reduces LPIPS by 34% and FID by 48% relative to a pixel-rendering baseline on the Cam×Time benchmark. Despite using 9× fewer parameters, GeoDYN achieves target-view reconstruction quality comparable to a 14B pixel-rendering model while requiring roughly half the inference time per step.

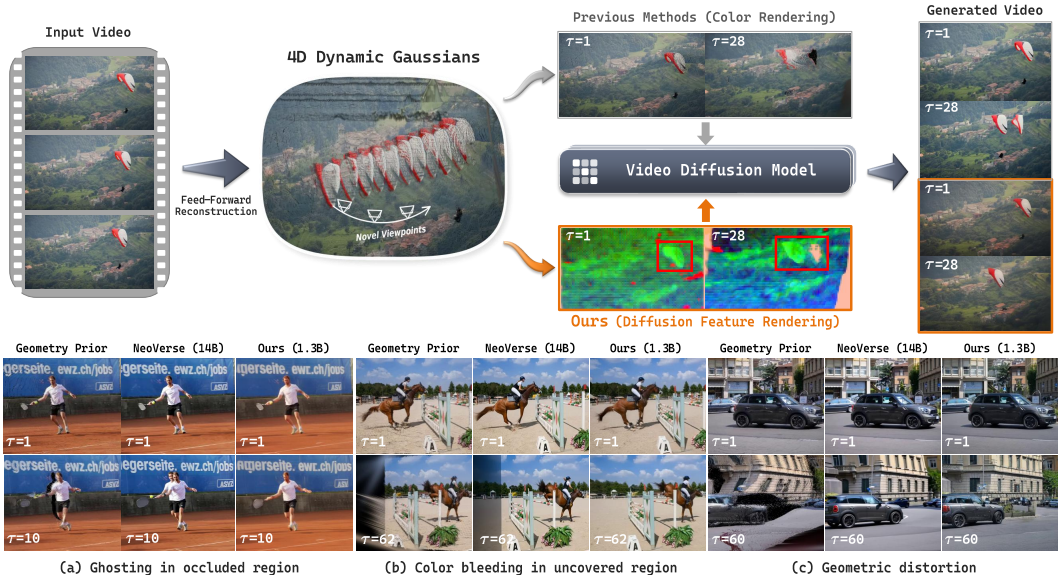


Figure 1 Pixel rendering vs. feature transport. Top: Previous methods (Yang et al., 2026; Lin et al., 2026; Ren et al., 2025) condition video diffusion on rendered appearance from the geometry prior. GeoDYN instead transports diffusion features through the 4D Gaussians and repeatedly injects them across the backbone. **Bottom:** Compared with NeoVerse, feature transport reduces **(a)** ghosting in occluded regions, **(b)** color bleeding in unseen regions, and **(c)** geometric distortion inherited from the prior across timestamp τ.

1 Introduction

Dynamic novel view synthesis (NVS) aims to render a dynamic scene from arbitrary camera viewpoints and at arbitrary scene times, given only a monocular video. Such controllable 4D generation is important for video reframing, robotics simulation, and autonomous driving, and is becoming a core capability of interactive world models such as Atlas (Team, 2026). Achieving this requires both geometric consistency with the input video and the ability to synthesize unobserved regions.

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

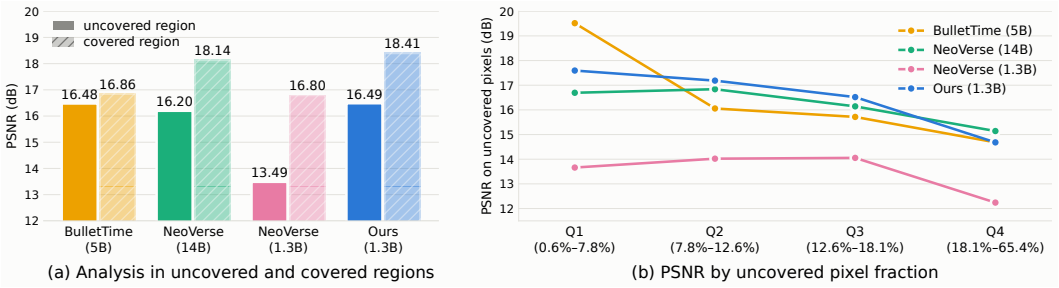


Figure 2 Robustness to incomplete geometry priors on Cam×Time (Huang et al., 2026). (Un)covered regions are defined by thresholding the rendered alpha map of the 4D Gaussian prior ($\eta = 0.125$). **(a)** Pixel rendering degrades strongly in uncovered regions, while ours (1.3B) shows smaller degradation. **(b)** As geometry coverage decreases, ours remains close to the 14B model and outperforms the backbone-matched 1.3B pixel-rendering baseline, demonstrating the robustness of feature transport to incomplete geometry.

Existing approaches follow two complementary directions. **Feed-forward reconstruction** (Liu et al., 2026; Xu et al., 2025) provides strong geometric grounding but struggles to synthesize unobserved regions. **Camera- and time-controlled video generation** (Wang et al., 2026b; Huang et al., 2026) generates unseen content, but relies on implicit geometry, often yielding inaccurate camera control under challenging trajectories (Tab. 3). **Geometry-conditioned generation** (Huang et al., 2025b; Cao et al., 2025; Wang et al., 2025b; Cao et al., 2026; Song et al., 2026; Lin et al., 2026; Yang et al., 2026; Kim et al., 2026) bridges the two by guiding video diffusion with explicit geometry. While this paradigm has largely advanced through better reconstruction, comparatively less attention has been paid to *how* the geometry prior should be delivered to the diffusion model.

Most existing methods (Yang et al., 2026; Lin et al., 2026) render the geometry prior into RGB, depth, or geometric buffers and feed them to a control branch (Jiang et al., 2025). We refer to this family as **pixel-rendering methods**. Using NeoVerse (Yang et al., 2026) as a representative method, we observe three characteristic failures (Fig. 1): ghosting from residual geometry in occluded regions, color bleeding from erroneous reconstructions in unseen regions, and geometric distortion inherited from the geometry prior. These failures stem from how the prior is injected. Pixel-rendering methods rasterize the prior only once at the input, so any rasterization error propagates directly into the conditioning. Since a rendered pixel does not reveal whether it reflects correct geometry or a reconstruction error, the diffusion backbone is left to decide what to trust. This burden has so far fallen on model capacity: when scaled down from 14B to 1.3B, NeoVerse degrades sharply, especially where the prior is incomplete (Tab. 2, Fig. 2). A natural question thus arises: *how can we exploit 4D priors to guide representations directly in feature space, thereby insulating the video diffusion backbone from errors introduced by premature rasterization?*

We address this problem by introducing **GeoDYN**, which repeatedly transports source-video features through the 4D geometry prior instead of passing its rendered appearance to the backbone. Inserted between control blocks, each GeoDYN block transports these source-aligned control features at high spatial resolution through the 4D Gaussians, preserves source space–time correspondence via Gaussian-indexed coordinates, refines Gaussian features and opacities, and adaptively fuses the resulting target-aligned features into the DiT backbone (Peebles & Xie, 2023) in Fig. 3. This directly addresses the failures in Fig. 1: 4D-GS refinement and adaptive fusion limit the influence of residual geometry in occluded regions, reducing **ghosting**; uncovered regions receive weak feature guidance rather than erroneous rendered colors, reducing **color bleeding**; and repeated refinement with selective fusion down-weights unreliable geometry, mitigating inherited **geometric distortion**.

Experiments support this design. In a controlled comparison with the same Wan2.1-1.3B backbone (Wan et al., 2025), WorldMirror prior (Liu et al., 2026), training data, and schedule, GeoDYN consistently outperforms the backbone-matched pixel-rendering baseline, reducing LPIPS by 34% and FID by 48% on Cam×Time (Huang et al., 2026). The advantage is concentrated where the geometry prior is incomplete: feature transport gains 3.0 dB on uncovered pixels compared with 1.6 dB on covered ones (Tab. 2, Fig. 2). Despite its 1.3B backbone, GeoDYN approaches the target-view reconstruction quality of NeoVerse (14B), while using $9\times$ fewer parameters and half the inference time per step. On the two monocular benchmarks, GeoDYN is the fastest geometry-conditioned method and ranks second in FID, while outperforming camera- and time-controlled generators in both perceptual fidelity and camera control (Tab. 3). Ablations identify high-resolution feature transport and source correspondence as the main contributors, while refinement and fusion analyses in Sec. 4.4

Table 1 Comparison methods. We categorize baselines by camera/time-controlled generation and geometry-conditioned generation. GeoDYN belongs to the latter, but unlike prior methods that condition on rendered geometry, our method uses feature transport via 4D Gaussians.

Method	Diffusion backbone	Camera control	Time control
Camera/Time-Controlled Generation			
ReCamMaster (Bai et al., 2025)	Wan2.1 1.3B (Wan et al., 2025)	Extrinsics	✗
BulletTime (Wang et al., 2026b)	CogVideoX 5B (Yang et al., 2025)	Extrinsics	✓
SpaceTimePilot (Huang et al., 2026)	Wan2.1 1.3B (Wan et al., 2025)	Extrinsics	✓
Geometry Conditioned Generation			
TrajectoryCrafter (Yu et al., 2025)	CogVideoX-Fun 5B (aigc apps, 2026)	Point cloud	✗
GEN3C (Ren et al., 2025)	Cosmos 7B (Agarwal et al., 2026)	Point cloud	✗
Vista4D (Lin et al., 2026)	Wan2.1 14B (Wan et al., 2025)	Point cloud	✗
NeoVerse (Yang et al., 2026)	Wan2.1 14B (Wan et al., 2025)	4D Gaussians (RGB)	✓
GeoDYN (Ours)	Wan2.1 1.3B (Wan et al., 2025)	4D Gaussians (Feature)	✓

show that GeoDYN suppresses unreliable Gaussian support and increases prior injection where geometry coverage is stronger. Together, these results show that feature transport improves robustness to incomplete geometry without increasing backbone capacity.

Our contributions are as follows.

- We introduce **GeoDYN**, replacing pixel-rendered conditioning with repeated feature transport through a 4D Gaussian prior, reducing the propagation of reconstruction artifacts into generation.
- We design high-resolution transport, Gaussian-indexed correspondence, and 4D Gaussian refinement to preserve original source and improve geometric consistency during feature transport.
- Backbone-matched experiments show that GeoDYN (1.3B) clearly outperforms the 1.3B pixel-rendering baseline. Despite using $9\times$ fewer parameters, it achieves target-view reconstruction quality comparable to the 14B pixel-rendering model.

2 Related Work

2.1 Generative Dynamic Novel View Synthesis

Reconstruction-based dynamic NVS methods explicitly model scene geometry and motion using dynamic NeRFs, 4D Gaussians, or monocular reconstruction priors (Pumarola et al., 2021; Park et al., 2021; Wu et al., 2024; Wang et al., 2025a; Lei et al., 2025; Zhang et al., 2025), but remain limited in synthesizing unobserved regions. Generative dynamic NVS addresses this through two paradigms. **Camera/time-controlled generation** directly conditions video diffusion on camera trajectories (Bai et al., 2025; Li et al., 2026), with BulletTime (Wang et al., 2026b) and SpaceTimePilot (Huang et al., 2026) additionally enabling temporal control, but relies on implicitly learned geometry. **Geometry-conditioned generation** instead combines reconstruction with video diffusion using point cloud priors (Yu et al., 2025; Ren et al., 2025; Lin et al., 2026; Wizadwongsa et al., 2026; Cao et al., 2026) or Gaussian representations (Zhu et al., 2026; Liu et al., 2026; Yang et al., 2026; Kim et al., 2026). These priors are typically rendered into RGB or geometric buffers and injected through ControlNet (Zhang et al., 2023; Jiang et al., 2025), allowing video diffusion to correct reconstruction artifacts (Yang et al., 2026; Lin et al., 2026; Zhu et al., 2026; Kim et al., 2026). **GeoDYN** follows this setting with the feed-forward 4D reconstruction model WorldMirror (Liu et al., 2026; Yang et al., 2026), using geometry to transport source features rather than rendered appearance.

2.2 Feature Lifting onto Gaussian Representations

Feature 3DGS (Zhou et al., 2024) and LangSplat (Qin et al., 2024) distill 2D foundation-model features into 3D Gaussians for open-vocabulary scene understanding, while Dr.Splat (Jun-Seong et al., 2025) and LUDVIG (Marrie et al., 2025) directly associate 2D features with Gaussians without per-scene feature-field optimization. Feature4X (Zhou et al., 2025) extends feature fields to dynamic scenes, and C4G (Kim et al., 2026) lifts foundation features onto feed-forward 4D Gaussians for tracking and scene understanding. These methods primarily use lifted features for understanding rather than generation. GeoNVS (Kang et al., 2026) extends feature lifting to generative NVS by projecting diffusion features onto static 3D Gaussians and fusing their renderings into a U-Net backbone. GeoDYN extends this paradigm to dynamic scenes and DiT-based video diffusion, transporting control features through a 4D Gaussian prior across camera and time. This requires high-resolution feature transport and explicit source space–time correspondence for retimed generation.

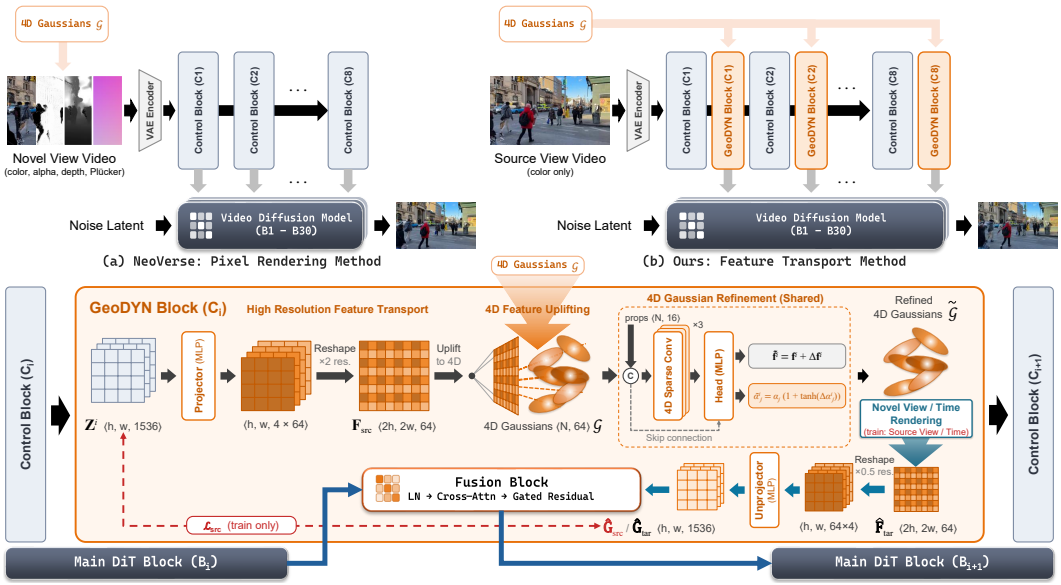


Figure 3 Overview of GeoDYN. (a) Pixel rendering (NeoVerse (Yang et al., 2026)). The 4D Gaussian prior is rendered into pixel-space conditions, allowing reconstruction artifacts to propagate into generation. **(b) Feature transport (Ours).** GeoDYN instead transports source-video features through the 4D Gaussians and repeatedly injects them into the main DiT backbone. **Bottom: GeoDYN block.** $\mathbf{Z}^i$ from control block C_i is projected to a high-resolution feature map, uplifted onto the 4D Gaussians $\mathcal{G}$, and refines feature and opacity of $\mathcal{G}$. The refined Gaussians $\mathcal{G}'$ are rendered at the requested camera and time, mapped back to token space, and fused into the DiT backbone. During training, source-view rendering is additionally supervised by $\mathcal{L}_{\mathrm{src}}$ (dashed red). Blue path in GeoDYN block denotes the flow of target-view/time tokens.

3 Method

3.1 Preliminaries and Overview

Given a monocular source video $V_{\mathrm{src}} = \{I_t^{\mathrm{src}}\}_{t=1}^{T_c}$, we synthesize a target video $\widehat{V}_{\mathrm{tar}}$ following a requested camera–time trajectory $\mathcal{Q}_{\mathrm{tar}} = \{(\pi_t, \tau_t)\}_{t=1}^{T_o}$, where π_t and τ_t specify the target camera and scene time. Following the geometry-conditioned generation pipeline, we first obtain a 4D Gaussian representation $\mathcal{G} = \{g_j\}_{j=1}^{N}$ from the source video using the feed-forward reconstruction model WorldMirror (Liu et al., 2026; Yang et al., 2026). This reconstructed geometry serves as a prior for a video diffusion model, which generates the target video at the requested camera and time.

From pixel rendering to feature transport. Conventional geometry-conditioned generation methods (Yang et al., 2026; Lin et al., 2026; Yu et al., 2025; Ren et al., 2025; Kim et al., 2026), such as NeoVerse (Yang et al., 2026), first render the reconstructed geometry $\mathcal{G}$ into the target camera and time, and feed the rendered RGB and geometric buffers to a ControlNet (Jiang et al., 2025). Thus, reconstruction artifacts in the geometry prior can be directly encoded into the pixel-space conditioning signal. **GeoDYN** instead uses $\mathcal{G}$ as a geometric carrier for transporting features extracted from the source video. The control branch consists of eight blocks $\{\mathrm{C}_i\}_{i=1}^{8}$, while the frozen DiT backbone contains 30 blocks $\{\mathrm{B}_j\}_{j=1}^{30}$. We denote the output feature of control block C_i by $\mathbf{Z}^i \in \mathbb{R}^{T_z \times h \times w \times d}$. A GeoDYN block transports $\mathbf{Z}^i$ through $\mathcal{G}$ to the camera–time trajectory $\mathcal{Q}_{\mathrm{tar}}$ and fuses the target-aligned feature into backbone block $\mathrm{B}_{\kappa(i)}$, where $\kappa(i)$ denotes the control-to-backbone mapping.

3.2 GeoDYN Block: Leveraging Geometry Priors via Feature Transport

Pipeline of GeoDYN Block. As illustrated in Fig. 3, each GeoDYN block transports control features from the source view to the requested target view and time through the 4D Gaussian prior. Given the control feature $\mathbf{Z}^i$, we first project it to a higher-resolution feature map for geometry-aware transport. We then uplift these features onto the 4D Gaussians and jointly refine their feature payloads and opacities. The refined Gaussians are rendered at the target camera and time, where the rendered features are augmented with Gaussian-indexed correspondence cues and mapped back to the control-token grid. Finally, the resulting target-aligned tokens are fused into the corresponding main DiT features through gated cross-attention.

High-resolution feature transport. Lifting high-dimensional DiT tokens into 4D Gaussians incurs substantial memory and inference costs as the per-Gaussian feature dimension grows (Zhou et al., 2024; Li et al., 2025). We therefore keep the feature dimension fixed and perform geometric transport at a higher spatial resolution. Given source video tokens $\mathbf{Z}^i$ from the stage-i control block, a projector ϕ_i predicts r^2 feature vectors per token, which are spatially rearranged as

$$\mathbf{F}^i_{\mathrm{src}} = \mathrm{Shuffle}_r\big(\phi_i(\mathbf{Z}^i)\big) \in \mathbb{R}^{T_z \times rh \times rw \times c}. \tag{1}$$

Both feature uplifting and target-view rendering are performed at this higher resolution. This transports source information into 4D Gaussians without increasing per-Gaussian feature dimensionality.

Uplifting control features to 4D Gaussians. For each source timestamp τ, let $\mathcal{G}_\tau = \{g^\tau_j\}_{j=1}^{N_\tau}$ denote the active 3D Gaussian slice of the 4D representation $\mathcal{G}$. Let $\mathbf{F}^i_{\mathrm{src}}(\tau, p) \in \mathbb{R}^c$ denote the high-resolution feature at source pixel p. We uplift the 2D features to each Gaussian by a rendering-weighted average:

$$\mathbf{f}^{i,\tau}_j = \frac{\sum_p W^{\mathrm{src},\tau}_{pj} \mathbf{F}^i_{\mathrm{src}}(\tau, p)}{\sum_p W^{\mathrm{src},\tau}_{pj}}, \qquad g^\tau_j \in \mathcal{G}_\tau, \tag{2}$$

where $W^{\mathrm{src},\tau}_{pj}$ is the alpha-compositing contribution of Gaussian g^τ_j to source pixel p, including its projected footprint, opacity, and front-to-back transmittance. Gaussians with zero accumulated weight are assigned zero features. Applying this operation across source timestamps produces the feature-augmented 4D Gaussian representation. Because uplifting is fully feed-forward and requires no per-scene optimization, it can be applied to evolving control features throughout denoising.

4D Gaussian refinement. Gaussian opacity is originally estimated for RGB reconstruction, whereas our transport path operates on learned features. We therefore introduce a lightweight 4D Gaussian refinement network $\mathcal{A}_\theta$, based on 4D sparse convolutions (Choy et al., 2019), that jointly adapts lifted features and opacities:

$$\begin{aligned} (\Delta\mathbf{f}^i, \Delta\boldsymbol{\alpha}^i) &= \mathcal{A}_\theta(\mathbf{f}^i, \mathcal{G}), \\ \widetilde{\mathbf{f}}^i = \mathbf{f}^i + \Delta\mathbf{f}^i, &\qquad \widetilde{\alpha}^i_j = \alpha_j \left(1 + \tanh(\Delta\alpha^i_j)\right). \end{aligned} \tag{3}$$

The residual prediction head is initialized to zero so that refinement starts from the original Gaussian representation. We denote the resulting feature-augmented representation by $\mathcal{G}'^i$. The detailed architecture of $\mathcal{A}_\theta$ is provided in the Appendix. F.1.

Target rendering and source correspondence. At control block i, we render the refined 4D Gaussians $\mathcal{G}'^i$ at the requested target camera and time:

$$\widehat{\mathbf{F}}^i_{\mathrm{tar}} = \widetilde{\mathbf{W}}^{\mathrm{tar}}_i \widetilde{\mathbf{f}}^i, \tag{4}$$

where $\widetilde{\mathbf{W}}^{\mathrm{tar}}_i$ denotes alpha-compositing weights from the refined opacities. To trace each target feature to the source video, we construct a soft source-correspondence map. Let $\mathbf{q}^{\mathrm{src}}_p = (\bar\tau_p, \bar u_p, \bar v_p) \in [-1, 1]^3$ denote the normalized space–time coordinate of source location p, and W^{src}_{pj} its base-prior rendering weight for Gaussian j. Each Gaussian is assigned the weighted-average source coordinate

$$d_j = \sum_p W^{\mathrm{src}}_{pj}, \qquad \bar{\mathbf{q}}_j = \begin{cases} \frac{\sum_p W^{\mathrm{src}}_{pj} \mathbf{q}^{\mathrm{src}}_p}{d_j}, & d_j > 0, \\ \mathbf{0}, & \text{otherwise}, \end{cases} \tag{5}$$

which is rendered into the target view using the base-prior target weights:

$$s_p = \sum_j W^{\mathrm{tar}}_{pj} \mathbb{1}[d_j > 0], \qquad \mathbf{Q}_{\mathrm{tar},p} = \left[\frac{\sum_j W^{\mathrm{tar}}_{pj} \mathbb{1}[d_j > 0] \bar{\mathbf{q}}_j}{s_p} \,;\, s_p \right], \tag{6}$$

with the coordinate term set to zero when $s_p = 0$. Thus, $\mathbf{Q}_{\mathrm{tar},p}$ encodes the corresponding source space–time location and its support, and is shared across control stages. We augment $\mathbf{Q}_{\mathrm{tar},p}$ with normalized target depth $\widehat{D}_p$, binary coverage mask m_p, and the six-dimensional target Plücker ray $\boldsymbol{\rho}_p$ to form the geometric cues $\mathbf{D}_{\mathrm{tar},p} = [\widehat{D}_p; m_p; \boldsymbol{\rho}_p; \mathbf{Q}_{\mathrm{tar},p}]$. We then concatenate $\mathbf{D}_{\mathrm{tar}}$ with the rendered feature $\widehat{\mathbf{F}}^i_{\mathrm{tar}}$ on the high-resolution grid, spatially unshuffle the result, and map it with ψ_i to the $d = 1536$ diffusion dimension to obtain the target-aligned geometry feature $\widehat{\mathbf{G}}^i_{\mathrm{tar}}$:

$$\widehat{\mathbf{G}}^i_{\mathrm{tar}} = \psi_i\Big(\mathrm{Unshuffle}_r[\widehat{\mathbf{F}}^i_{\mathrm{tar}} \| \mathbf{D}_{\mathrm{tar}}]\Big). \tag{7}$$

Adaptive feature fusion. Following GeoNVS (Kang et al., 2026), we adopt its adaptive fusion module to integrate the target-aligned geometry feature $\widehat{\mathbf{G}}^i_{\mathrm{tar}}$ into the feature $\mathbf{H}^{\kappa(i)}$ of backbone block

$\mathrm{B}_{\kappa(i)}$. We first obtain a geometry-attended feature by cross-attending from the backbone feature to the transported geometry feature:

$$\mathbf{A}_i = \mathrm{CrossAttn}_i\left(\mathrm{LN}(\mathbf{H}^{\kappa(i)}), \mathrm{LN}(\widehat{\mathbf{G}}^i_{\mathrm{tar}})\right). \tag{8}$$

A gating MLP predicts fusion weight $\boldsymbol{\gamma}_i \in [-1, 1]$ from diffusion and geometry-attended features:

$$\widehat{\mathbf{H}}^{\kappa(i)} = \mathbf{H}^{\kappa(i)} + \boldsymbol{\gamma}_i \odot \mathbf{A}_i, \qquad \boldsymbol{\gamma}_i = \tanh\Big(\mathrm{MLP}_i[\mathbf{H}^{\kappa(i)} \| \mathbf{A}_i]\Big). \tag{9}$$

This allows the diffusion backbone to incorporate transported geometry based on local relevance.

3.3 Training Objective

To preserve the stage-wise control representation $\mathbf{Z}^i$, we additionally render $\mathcal{G}'^i$ back to the source views and unproject it to $\widehat{\mathbf{G}}^i_{\mathrm{src}}$ through the same path as the target branch, and enforce

$$\mathcal{L}_{\mathrm{src}} = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{P_{\mathrm{tok}}}\sum_{p}\left[1 - \cos\left(\mathcal{C}(\widehat{\mathbf{G}}^i_{\mathrm{src}})_p, \mathrm{sg}[\mathcal{C}(\mathbf{Z}^i)_p]\right)\right], \tag{10}$$

where $L = 8$ is the number of GeoDYN blocks, $P_{\mathrm{tok}} = T_z hw$ is the token count per feature map, $\mathcal{C}$ removes the token-wise mean, and sg denotes stop-gradient. The total loss is $\mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{src}}\mathcal{L}_{\mathrm{src}}$, where $\mathcal{L}_{\mathrm{FM}}$ is the standard flow-matching objective (Wan et al., 2025). The control branch, the per-block projection, unprojection, and fusion modules, and a GS refinement network shared across blocks are trained; the reconstruction model, VAE, and diffusion backbone remain frozen.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate on three benchmarks: SpatialVID (Wang et al., 2026a), DynPose (Rockwell et al., 2025), and Cam×Time (Huang et al., 2026). SpatialVID and DynPose are source-only monocular benchmarks that evaluate perceptual fidelity and camera control under novel trajectories, whereas Cam×Time provides ground-truth target videos for evaluating temporal control. All methods are evaluated at 81 frames and 480×832 using their default inference settings. Further benchmark and trajectory details are provided in the Appendix. H.

Metrics. Cam×Time (Huang et al., 2026) provides ground-truth target videos, allowing direct evaluation with PSNR, SSIM, and LPIPS. SpatialVID (Wang et al., 2026a) and DynPose (Rockwell et al., 2025) are monocular and provide no ground-truth target views, so we assess perceptual quality using FID, FVD, and VBench (Huang et al., 2024). To measure camera controllability, we estimate the generated camera trajectory with ViPE (Huang et al., 2025a), align it to the target trajectory using Sim(3) Umeyama alignment, and report translation and rotation errors. For monocular benchmarks, the target trajectory is defined by composing the source pose with the instructed camera motion.

Compared methods. As summarized in Tab. 1, we compare camera/time-controlled generation (Wang et al., 2026b; Huang et al., 2026; Bai et al., 2025) and geometry-conditioned generation (Yang et al., 2026; Ren et al., 2025; Lin et al., 2026; Yu et al., 2025), including NeoVerse as the closest pixel-rendering baseline. All methods use officially released checkpoints.

Implementation. GeoDYN uses frozen Wan2.1-1.3B (Wan et al., 2025) and WorldMirror (Liu et al., 2026) as the diffusion backbone and 4D Gaussian prior estimator, respectively; DL3DV (Ling et al., 2024) uses precomputed 3D Gaussians. We train the control, projection/unprojection, fusion, and Gaussian refinement modules with $c = 64$, $r = 2$, and $\lambda_{\mathrm{src}} = 0.03$. Training runs for 15K steps on a mixture of Cam×Time and SpatialVID, followed by 17K additional steps on a mixture of six datasets (Grauman et al., 2024; Ling et al., 2024; Wang et al., 2026a; Huang et al., 2026; Jin et al., 2025; Denninger et al., 2023), for 32K steps in total. We use four H200 GPUs with an effective batch size of 16. The full model has 1.92B parameters, of which ∼0.50B are trainable. Further training and inference details are provided in the Appendix. G.

A matched pixel-rendering baseline. We re-implement NeoVerse on the same frozen Wan2.1-1.3B backbone and WorldMirror prior as GeoDYN, trained on the same data for 15K steps. NeoVerse (1.3B) uses rendered pixel-space geometry conditions, whereas GeoDYN uses feature transport through the Gaussian representation. The original model is denoted NeoVerse (14B). In Tab. 2 and Fig. 2, GeoDYN is trained for the same 15K steps as NeoVerse (1.3B) for a matched comparison; all other GeoDYN results use the full 32K-step schedule.

Table 2 Pixel rendering vs. feature transport. We compare GeoDYN with backbone-matched NeoVerse (1.3B) under the same training and evaluation protocol, isolating feature transport from pixel-space geometry conditioning. Cam×Time (Huang et al., 2026) has held-out target views, whereas SpatialVID (Wang et al., 2026a) is source-only. WorldMirror (Liu et al., 2026) represents the prior alone; NeoVerse (14B)† is an unranked upper-capacity reference. Camera errors are averaged over the 111 (Cam×Time) and 262 (SpatialVID) cases where every method admits a positive similarity fit. **Bold** marks the best backbone-matched result.

Method	Cam×Time (Huang et al., 2026)							SpatialVID (Wang et al., 2026a)			
	PSNR ⇑	SSIM ⇑	LPIPS ⇓	FID ⇓	FVD ⇓	Trans. err. ⇓	Rot. err. ⇓	FID ⇓	FVD ⇓	Trans. err. ⇓	Rot. err. ⇓
Geometry Prior											
WorldMirror (Liu et al., 2026)	12.87	0.463	0.613	128.96	860.37	2.74	3.11	34.03	456.54	7.16	0.83
Pixel Rendering Method											
NeoVerse (14B)†	17.35	0.644	0.345	35.35	349.52	1.91	1.53	23.98	361.05	7.21	0.73
NeoVerse (1.3B)	16.15	0.565	0.529	100.63	715.91	3.24	4.06	42.20	484.64	9.55	1.72
Feature Transport Method											
GeoDYN (Ours, 1.3B)	**17.45**	**0.683**	**0.347**	**52.42**	**379.61**	**2.77**	**2.72**	**28.27**	**393.22**	**8.90**	**1.37**

† NeoVerse (14B) (Yang et al., 2026) uses a 10× larger backbone and is therefore not ranked. Here, NeoVerse (1.3B) is our backbone-matched re-implementation, trained with the same training data and 15K-step schedule as GeoDYN.

4.2 Pixel Rendering versus Feature Transport

Tab. 2, Fig. 2 isolate the effect of how the geometry prior is delivered to the diffusion model. NeoVerse (1.3B) and GeoDYN share the same backbone, prior, data, and schedule, differing only in pixel rendering versus feature transport; WorldMirror provides the reconstruction reference.

Matched backbone. With backbone, prior, data, and training schedule matched, GeoDYN consistently outperforms pixel rendering in visual fidelity, distributional quality, and camera control on both benchmarks. This suggests that feature transport provides a more effective interface to the geometry prior than directly conditioning the diffusion backbone on its rasterized appearance.

Robustness to incomplete geometry. Fig. 2 further localizes this advantage. The improvement over pixel rendering is substantially larger in regions where the 4D Gaussian prior has low coverage than in covered regions. Moreover, as the fraction of uncovered pixels increases, GeoDYN remains close to the much larger NeoVerse model while the backbone-matched pixel-rendering baseline degrades more rapidly. This suggests that feature transport is especially beneficial when the geometry prior is incomplete, where rasterized conditioning provides weak or unreliable appearance evidence.

Comparison with a larger backbone. Despite using a substantially smaller backbone, GeoDYN approaches NeoVerse (14B) in direct target-view reconstruction quality and in the coverage-based analysis. The larger model retains an advantage in distributional fidelity and camera control, showing that additional generative capacity still benefits pixel-rendering approaches. Overall, these results indicate that feature transport recovers much of the performance otherwise obtained by scaling the diffusion backbone, while remaining markedly more robust than pixel rendering at matched capacity.

4.3 Comparison with Existing Methods

Tab. 3 evaluates GeoDYN against existing methods in Tab. 1 on the two source-only monocular benchmarks (Wang et al., 2026a; Rockwell et al., 2025), while Tab. 5 compares temporal control on Cam×Time (Huang et al., 2026) with ground-truth target videos.

Overall comparison. Compared with camera/time-controlled generation, GeoDYN achieves better perceptual fidelity and camera control than the strongest comparable baseline, SpaceTimePilot, on both benchmarks, while requiring fewer parameters and the lowest inference time per step. Among geometry-conditioned methods, GeoDYN achieves the second-best FID on both datasets despite using a substantially smaller backbone, while maintaining competitive FVD and camera accuracy. Its VBench scores remain comparable to the larger geometry-conditioned models, showing that the efficiency gain does not come at a clear loss in perceptual video quality.

Temporal control. Tab. 5 further evaluates methods by temporal-control type. GeoDYN improves over NeoVerse (14B) in reconstruction quality for direction changes and bullet-time generation, despite its much smaller backbone, whereas NeoVerse retains an advantage for slow-motion generation and camera accuracy. SpaceTimePilot performs strongly across these categories, indicating that explicit camera–time conditioning remains effective for pure retiming tasks. Overall, GeoDYN provides a favorable trade-off between generative quality, geometry control, and computational cost across both novel-view and temporal control settings.

Table 3 Evaluation on monocular video datasets. We compare seven baselines on DynPose (Rockwell et al., 2025) and SpatialVID (Wang et al., 2026a), evaluating efficiency, fidelity, camera control, and VBench quality. As these monocular datasets provide no held-out target views, evaluation focuses on camera controllability and preservation of the source video's appearance and dynamics. **Bold** indicates the best result per dataset.

Method	Efficiency		Video fidelity		Camera control		VBench 1.0 Huang et al. (2024)				
	Time (s/step) ⇓	Params (B) ⇓	FID ⇓	FVD ⇓	Trans. err. ⇓	Rot. err. ⇓	Subject cons. ⇑	Backgr. cons. ⇑	Aesthetic quality ⇑	Imaging quality ⇑	Temporal flicker. ⇑
DynPose Dataset [(Rockwell et al., 2025)]											
Camera/Time-Controlled Generation											
ReCamMaster	21.33	**1.49**	46.001	626.087	22.356	9.750	0.830	0.872	0.448	0.550	**0.948**
BulletTime	11.41	5.67	39.262	499.337	21.517	9.336	0.841	0.896	0.440	0.532	0.944
SpaceTimePilot	23.02	2.00	22.517	388.860	18.522	8.666	0.837	0.881	**0.461**	0.577	0.933
Geometry Conditioned Generation											
TrajectoryCrafter	5.44	5.65	21.501	342.554	14.276	5.660	0.837	0.904	0.437	0.582	0.942
GEN3C	46.90	7.24	25.804	339.139	14.643	6.100	0.824	0.897	0.441	0.576	0.938
Vista4D	45.16	19.56	**13.155**	**293.725**	13.895	**5.290**	0.840	**0.906**	0.439	0.605	0.941
NeoVerse	10.44	17.65	22.450	303.456	**13.660**	5.750	**0.844**	0.893	0.451	**0.625**	0.922
GeoDYN (Ours)	**4.97**	1.92	18.377	337.495	15.411	6.643	0.834	0.894	0.442	0.596	0.923
SpatialVID Dataset [(Wang et al., 2026a)]											
Camera/Time-Controlled Generation											
ReCamMaster	21.33	**1.49**	36.916	584.276	27.001	4.469	0.885	0.890	0.500	0.632	0.948
BulletTime	11.41	5.67	38.884	698.522	27.263	5.634	**0.892**	0.903	0.484	0.582	**0.954**
SpaceTimePilot	23.02	2.00	19.375	382.947	13.615	3.440	0.868	0.892	**0.501**	0.651	0.932
Geometry Conditioned Generation											
TrajectoryCrafter	5.44	5.65	26.433	430.180	10.743	1.347	0.850	0.911	0.476	0.605	0.937
GEN3C	46.90	7.24	22.649	350.642	10.407	1.351	0.854	0.908	0.482	0.591	0.940
Vista4D	45.16	19.56	**13.743**	**322.695**	10.585	1.262	0.868	**0.920**	0.476	0.672	0.941
NeoVerse	10.44	17.65	23.977	361.054	**9.833**	**0.994**	0.865	0.900	0.488	**0.683**	0.922
GeoDYN (Ours)	**4.97**	1.92	18.474	361.087	11.520	1.351	0.857	0.907	0.486	0.651	0.924

Table 4 Component ablation on Cam×Time (Huang et al., 2026) dataset. GeoDYN variants are trained for 8K steps on Cam×Time only; the NeoVerse row is the 15K-step model from Tab. 2. High-resolution feature transport mainly improves perceptual fidelity, while GS index projection further improves reconstruction quality and camera control. Ablating GS refinement degrades performance, supporting the complementary roles of opacity and feature refinement. Camera errors are averaged over all 186 cases. Best per column in **bold**.

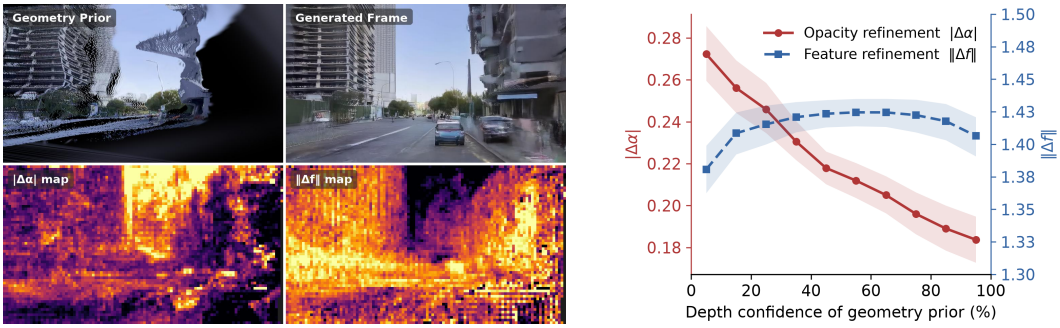
Variants	Model components				Visual fidelity					Camera control	
	α refine	Feature refine	HR	Index Proj.	PSNR↑	SSIM↑	LPIPS↓	FID↓	FVD↓	Trans.↓	Rot.↓
Pixel Rendering Method											
NeoVerse (1.3B, 15K steps)	–	–	–	–	16.15	0.565	0.529	100.6	715.9	6.45	5.98
GeoDYN Baseline (8K steps)	✓	✓	✗	✗	16.66	0.634	0.430	72.4	544.7	6.19	6.47
+ high-res feature transport	✓	✓	✓	✗	16.46	0.646	**0.387**	54.6	**444.0**	5.82	5.59
+ GS index projection (**final**)	✓	✓	✓	✓	**16.76**	**0.663**	0.389	**53.8**	467.3	5.56	**4.93**
Ablations of GS Refinement											
w/o refinement	✗	✗	✓	✓	16.63	0.657	0.400	58.8	501.8	5.70	5.34
w/o learned α	✗	✓	✓	✓	16.75	0.663	0.390	54.0	467.4	**5.44**	5.24
w/o learned feature	✓	✗	✓	✓	16.63	0.657	0.400	58.9	499.3	5.58	5.24

4.4 Ablation Study

Component ablation. Tab. 4 evaluates the GeoDYN components introduced in Sec. 3.2 on Cam×Time. The base feature-transport model already improves most metrics over the backbone-matched pixel-rendering baseline. High-resolution transport mainly improves perceptual fidelity, while Gaussian-index projection further benefits reconstruction and camera control. Removing Gaussian refinement degrades performance, showing the value of adapting the Gaussians to uplifted control features. All ablation models are trained for 8K steps using only the Cam×Time dataset.

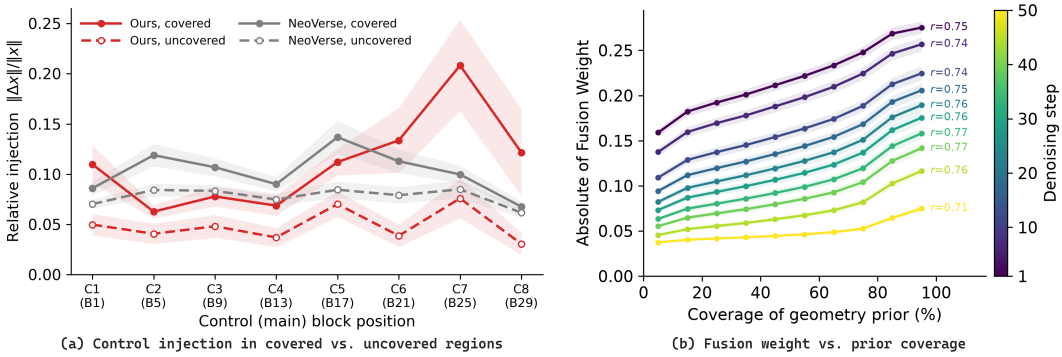
Role of 4D-GS refinement. We further analyze how feature and opacity refinement modify the geometry prior in Fig. 4. Opacity updates concentrate in geometrically unreliable regions, such as floaters and poorly reconstructed backgrounds, and decrease as the prior's depth confidence increases. In contrast, feature refinement occurs broadly across the scene and is largely independent of geometry confidence, with stronger responses around fine-detail regions. Together with the ablation results, this suggests complementary roles: opacity refinement suppresses unreliable geometry, while feature refinement adapts the transported representation more globally.

Adaptive use of the geometry prior. Fig. 5 analyzes how geometry guidance is injected across network depth and denoising time. At each fusion point, we measure the relative injection $\|\Delta x\|/\|x\|$,



(a) Opacity and feature updates in 4D Gaussian refinement (b) 4D Gaussian refinement vs. geometry confidence

Figure 4 Analysis of 4D-GS refinement. (a) Opacity edits $|\Delta\alpha|$ concentrate in geometrically uncertain regions such as sky and floaters, while feature edits $\|\Delta f\|$ are global and stronger on fine details. **(b) Refinement vs. depth confidence:** Opacity refinement decreases with depth confidence, showing that α is corrected where geometry is unreliable, while feature refinement stays nearly constant, regardless of geometry confidence.



(a) Control injection in covered vs. uncovered regions (b) Fusion weight vs. prior coverage

Figure 5 Where and when the model trusts the geometry prior. (a) Relative control injection across network depth in covered and uncovered regions. **(b)** Fusion weight versus prior coverage across denoising steps. GeoDYN increasingly favors well-supported geometry, whereas NeoVerse exhibits a flatter injection profile.

where x is the DiT hidden feature before fusion and Δx the feature added by the control block, separately for regions covered and uncovered by the prior; the same probe is applied to NeoVerse. The analysis uses 100 novel-trajectory samples, with 50 from Cam×Time (Huang et al., 2026) and 50 from SpatialVID (Wang et al., 2026a). GeoDYN keeps injection weak in uncovered regions, while increasing it with network depth where the prior is well supported. NeoVerse instead shows a flatter profile across depth and coverage. Across denoising, the magnitude of fusion weight $\mathbb{E}[|\gamma_i|]$ increases with prior coverage, concentrating guidance on reliable regions as generation proceeds.

Why is GeoDYN robust to incomplete geometry? Together, these analyses explain GeoDYN's robustness to incomplete geometry: the refinement module suppresses unreliable Gaussian support, while adaptive fusion limits its influence where coverage is weak. This reduces the direct propagation of unreliable reconstruction into the diffusion backbone, consistent with the smaller degradation observed in uncovered regions compared with pixel-rendering conditioning.

5 Conclusion

We presented **GeoDYN**, which replaces pixel-rendered geometry conditioning with repeated feature transport through a 4D Gaussian prior for dynamic novel view synthesis. By transporting source-video features rather than rendered appearance, GeoDYN reduces the propagation of geometry-induced artifacts such as ghosting, color bleeding, and inherited geometric distortion, particularly where the prior is incomplete. High-resolution transport preserves spatial detail, Gaussian-indexed correspondence keeps track of where source-video content comes from, and 4D Gaussian refinement and adaptive fusion suppress unreliable geometry and modulate its influence according to prior coverage. At matched 1.3B capacity, GeoDYN clearly outperforms pixel rendering and achieves target-view reconstruction quality comparable to a 14B pixel-rendering model with $9\times$ fewer parameters. These results show that improving *how* geometry is delivered to video diffusion can matter as much as improving the reconstruction or scaling the backbone. GeoDYN still inherits errors in the geometry prior, so advances in 4D reconstruction and in feature transport remain complementary directions for controllable dynamic generation.

AI USE STATEMENT

We used generative AI tools only to aid and polish the writing of this paper; they were not used for method development, implementation, experiments, or analysis. All AI-assisted text was reviewed by the authors, who take full responsibility for the final content.

REFERENCES

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

aigc apps. Videox-fun: A video generation pipeline for diffusion transformer, 2026. URL https://github.com/aigc-apps/VideoX-Fun.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14834–14844, 2025.

Chenjie Cao, Jingkai Zhou, Shikai Li, Jingyun Liang, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Uni3c: Unifying precisely 3d-enhanced camera and human motion controls for video generation. In *Proceedings of the SIGGRAPH Asia Conference Papers*, pp. 1–12, 2025.

Wei Cao, Hao Zhang, Fengrui Tian, Yulun Wu, Yingying Li, Shenlong Wang, Ning Yu, and Yaoyao Liu. Freeorbit4d: Training-free arbitrary camera redirection for monocular videos via foreground-complete 4d reconstruction. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2026.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3070–3079, 2019.

Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. doi: 10.21105/joss.04901. URL https\protect\protect\leavevmode@ifvmode\kern+.2222em\relax//doi.org/10.21105/joss.04901.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the International Conference on Machine Learning*, 2024.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19383–19400, 2024.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

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*, 2017.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025a.

Jiaxin Huang, Sheng Miao, Bangbang Yang, Yuewen Ma, and Yiyi Liao. Vivid4d: Improving 4d reconstruction from monocular video by video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1–13, 2025b.

Zhening Huang, Hyeonho Jeong, Xuelin Chen, Yulia Gryaditskaya, Tuanfeng Y Wang, Joan Lasenby, and Chun-Hao Huang. Spacetimepilot: Generative rendering of dynamic scenes across space and time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4D: Learning How Things Move in 3D from Internet Stereo Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Kim Jun-Seong, GeonU Kim, Kim Yu-Ji, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. splat: Directly referring 3d gaussian splatting via direct language embedding registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14137–14146, 2025.

Minjun Kang, Inkyu Shin, Taeyeop Lee, Myungchul Kim, In So Kweon, and Kuk-Jin Yoon. Geonvs: Geometry grounded video diffusion for novel view synthesis. In *European Conference on Computer Vision*, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Mungyeom Kim, Minkyeong Jeon, Honggyu An, Jaewoo Jung, Hyunah Ko, Jisang Han, Hyeonseo Yu, Donghwan Shin, Sunghwan Hong, Takuya Narihira, et al. Learning global motion with compact gaussians for feed-forward 4d reconstruction. *arXiv preprint arXiv:2605.31595*, 2026.

Jiahui Lei, Yijia Weng, Adam W Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Wanhua Li, Yujie Zhao, Minghan Qin, Yang Liu, Yuanhao Cai, Chuang Gan, and Hanspeter Pfister. Langsplatv2: High-dimensional 3d language gaussian splatting with 450+ fps. *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2507.07136.

Yixuan Li, Yanhong Zeng, Ka Leong Cheng, Jiayi Zhu, Hanlin Wang, Wen Wang, Yihao Meng, Hao Ouyang, Qiuyu Wang, Yue Yu, et al. Cameraanything: Refilming videos with arbitrary camera control. In *European Conference on Computer Vision*, 2026.

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, et al. Vista4d: Video reshooting with 4d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32671–32682, 2026.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

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

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting. In *Proceedings of the International Conference on Machine Learning*, 2026.

Juliette Marrie, Romain Menegaux, Michael Arbel, Diane Larlus, and Julien Mairal. Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7440–7450, 2025.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4172–4182, 2023.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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*, pp. 6121–6132, 2025.

Chris Rockwell, Joseph Tung, Tsung-Yi Lin, Ming-Yu Liu, David F. Fouhey, and Chen-Hsuan Lin. Dynamic camera poses and where to find them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Chenxi Song, Yanming Yang, Tong Zhao, Ruibo Li, and Chi Zhang. Taming video models for 3d and 4d generation via zero-shot camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40352–40363, 2026.

World Labs Team. Atlas: A world model for spatial intelligence. *World Labs Blog*, 2026. https //wwẇorldlabs.ai/blog/atlas.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

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

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Chang Zeng, Yanxi Zhou, Xiao-Xiao Long, et al. Spatialvid: A large-scale video dataset with spatial annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42592–42603, 2026a.

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025a.

Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025b.

Yiming Wang, Qihang Zhang, Shengqu Cai, Tong Wu, Jan Ackermann, Zhengfei Kuang, Yang Zheng, Frano Rajič, Siyu Tang, and Gordon Wetzstein. Bullettime: Decoupled control of time and camera pose for video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18319–18330, 2026b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Suttisak Wizadwongsa, Hyelin Nam, Supasorn Suwajanakorn, and Jeong Joon Park. Sierpinskicam: Camera-controlled video retaking with sierpinski triangle pattern cues. *arXiv preprint arXiv:2606.17310*, 2026.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4dgt: Learning a 4d gaussian transformer using real-world monocular videos. In *Advances in Neural Information Processing Systems*, 2025.

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 100–111, 2025.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Forrester Cole, Deqing Sun, Ming-Hsuan Yang, et al. Monst3r: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations*, volume 2025, pp. 82863–82886, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

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/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Shijie Zhou, Hui Ren, Yijia Weng, Shuwang Zhang, Zhen Wang, Dejia Xu, Zhiwen Fan, Suya You, Zhangyang Wang, Leonidas Guibas, et al. Feature4x: Bridging any monocular video to 4d agentic ai with versatile gaussian feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14179–14190, 2025.

Liyuan Zhu, Shengyu Huang, Amrita Mazumdar, Tianye Li, Zan Gojcic, Gordon Wetzstein, Iro Armeni, Shalini De Mello, and Alex Trevithick. World from motion: Generative dynamic gaussian reconstruction from monocular video. *arXiv preprint arXiv:2607.01202*, 2026.

APPENDIX

A COVERED AND UNCOVERED PIXEL ANALYSIS

Coverage mask. We define coverage from the Gaussian prior (Kerbl et al., 2023) reconstructed by WorldMirror (Liu et al., 2026) from the context video. The prior's alpha is rendered at each target view on the 480×832 evaluation plane, and a pixel is uncovered when its alpha falls below $\eta = 0.125$. Because the mask is computed from the base prior alone, the same mask is used for every method.

Masked PSNR. For each clip and region, we average the squared error over the selected pixels, frames, and RGB channels and convert it to PSNR; clip-level values are then averaged over the 186 Cam×Time clips.

Models. Following the matched protocol of Tab. 2 and Fig. 2, GeoDYN and the NeoVerse (Yang et al., 2026) (1.3B) reimplementation are both trained for 15K steps; NeoVerse (14B) and BulletTime use their released checkpoints.

Uncertainty. We report paired 95% confidence intervals from a scene-level bootstrap with 20,000 replicates, resampling the 31 scenes together with all of their queries. Against NeoVerse (1.3B), GeoDYN improves uncovered-region PSNR by $[2.39, 3.60]$ dB, with 163 of 186 clips improving, and the gain in uncovered regions exceeds that in covered regions by $[0.70, 2.08]$ dB. Against NeoVerse (14B), the intervals are $[-0.19, 0.76]$ dB (uncovered) and $[-0.15, 0.60]$ dB (covered), consistent with GeoDYN remaining close to the 14B model despite its smaller backbone.

Coverage bins. Clips are sorted by their uncovered fraction and split into four equal-size bins (Table 7); each point in Fig. 2b is the mean uncovered-region PSNR within a bin. Since scene content also changes across bins, these curves should be read as descriptive trends.

B RESULTS BY CONTROL CATEGORY

Table 5 breaks down the Cam×Time results by temporal control. The direction category contains backward and zigzag queries (62 clips), bullet time contains early, middle, and late frozen-time queries (93 clips), and motion speed contains slow-motion queries (31 clips); the queries are described in Appendix H.2. All GeoDYN results use the final 32K-step model, and camera errors follow the protocol of Tab. 2.

Table 5 Cam×Time results by control category. Controls retime the target view: *direction* (reverse, zigzag), *bullet time* (freeze at early/middle/late instant), and *motion speed* (slow motion). Params count the backbone and control module only. FID and FVD are computed per category and are not comparable across categories. Camera error follows Table 2. **Bold**: best per category.

Method	Params (B) ⇓	PSNR ⇑	SSIM ⇑	LPIPS ⇓	FID ⇓	FVD ⇓	Trans. err. ⇓	Rot. err. ⇓
Direction (reverse, zigzag)								
Camera/Time-Controlled Generation								
BulletTime	5.67	17.08	0.674	0.355	42.92	558.93	5.32	3.42
SpaceTimePilot	2.00	**19.79**	**0.732**	**0.231**	**24.74**	**356.81**	**2.78**	**2.51**
Geometry Conditioned Generation								
NeoVerse (14B)	17.65	17.77	0.661	0.316	39.29	470.59	3.45	3.52
GeoDYN (Ours, 1.3B)	**1.92**	18.02	0.718	0.298	47.60	546.07	4.52	4.71
Bullet time								
Camera/Time-Controlled Generation								
BulletTime	5.67	16.33	0.666	0.417	62.35	654.48	4.76	4.03
SpaceTimePilot	2.00	**20.49**	**0.748**	**0.218**	**29.01**	**304.99**	**3.06**	**2.24**
Geometry Conditioned Generation								
NeoVerse (14B)	17.65	16.64	0.618	0.395	60.47	597.17	3.89	3.18
GeoDYN (Ours, 1.3B)	**1.92**	17.78	0.686	0.306	48.33	459.12	4.17	3.56
Motion speed (slow)								
Camera/Time-Controlled Generation								
BulletTime	5.67	17.01	0.667	0.347	46.42	565.27	4.11	3.09
SpaceTimePilot	2.00	**19.51**	**0.730**	**0.235**	**32.09**	447.12	**3.33**	**2.12**
Geometry Conditioned Generation								
NeoVerse (14B)	17.65	18.64	0.688	0.252	40.46	**411.09**	3.34	2.57
GeoDYN (Ours, 1.3B)	**1.92**	17.67	0.677	0.316	59.86	578.43	3.54	2.97

Table 6 Variants in the component ablation (Tab. 4).

Variant	Implementation
GeoDYN baseline	Feature transport at the token resolution (30×52), without correspondence channels.
+ high-res feature transport	Transport at 60×104 ($r = 2$); the projector outputs $r^2c = 256$ channels per token.
+ GS index projection (final)	Adds the four source-correspondence channels, giving the full model.
w/o refinement	Removes the refiner; lifted features are rendered with the prior's opacity.
w/o learned α	Keeps feature refinement and uses the prior's opacity.
w/o learned feature	Keeps opacity refinement; Gaussian features are not updated.

C Reimplementation of the Matched Baseline

Architecture. The official NeoVerse release provides only the 14B model, so we reimplement NeoVerse on the same Wan2.1-1.3B (Wan et al., 2025) backbone and WorldMirror prior used by GeoDYN. Following the released architecture, the controller VAE-encodes the rendered RGB and depth of the prior (16 latent channels each), where depth is normalized by its 98th percentile and replicated to three channels. A binary coverage mask and six Plücker-ray channels are embedded by a Conv3D (kernel and stride $(4, 8, 8)$), GroupNorm, and SiLU into 64 channels. The resulting 96 channels are patch-embedded to width 1536 and processed by eight DiT (Peebles & Xie, 2023) control blocks attached to backbone layers $0, 4, \ldots, 28$, which add their outputs to the backbone through zero-initialized projections as in ControlNet (Zhang et al., 2023). The control branch has 393.5M trainable parameters.

Training. The baseline is trained on the same Cam×Time/SpatialVID (Wang et al., 2026a) mixture (1:2) and the same 15K-step schedule as GeoDYN, with effective batch size 16, AdamW, a learning rate of 10^{-5}, and BF16 precision. Following NeoVerse, training applies an opacity mask, masks non-context frames for non-cross-view pairs, and applies a color-consistency mask when ground-truth frames are available. Text, mask, and full-condition dropout probabilities are 0.1, 0.2, and 0.05, and reconstruction degradation is applied to SpatialVID. We evaluate the final checkpoint.

D Ablation and Analysis Details

D.1 Component Ablation

Table 6 describes each variant of Tab. 4. All variants are trained for 8K steps on Cam×Time, with learning rates of 10^{-4} for new modules and 10^{-5} for warm-started modules. The NeoVerse row of Tab. 4 is the 15K-step model of Tab. 2.

Table 7 Coverage bins used in Fig. 2b.

Bin	Uncovered fraction (%)	Clips	Scenes
Q1	0.6–7.8	46	17
Q2	7.8–12.6	47	13
Q3	12.6–18.1	46	14
Q4	18.1–65.4	47	13

D.2 Refinement Analysis

For Fig. 4, we record the opacity update $\tanh(\Delta\alpha_j)$ and feature update $\Delta\mathbf{f}_j$ of every Gaussian at the last control stage and the final denoising step. Both magnitudes are normalized by their per-sample mean and plotted against deciles of WorldMirror's depth confidence, computed within each sample. We also report the fraction of strongly suppressed Gaussians ($\tanh(\Delta\alpha_j) < -0.9$). The analysis uses 100 novel-trajectory generations (50 from Cam×Time and 50 from SpatialVID), with 95% bootstrap intervals. From the lowest to the highest confidence decile, the opacity update decreases monotonically from 1.25 to 0.83 of its mean, and the suppressed fraction falls from 7.0% to 1.5%. The feature update stays within 3% of its mean. Opacity refinement therefore concentrates where the



Figure 6 Qualitative results on Cam×Time (Huang et al., 2026). Each example shows the source video, the results of the compared methods and GeoDYN, and the ground-truth target under the requested camera and scene time. Higher PSNR indicates better performance.

prior is uncertain, while feature refinement is applied nearly uniformly. Since depth confidence is itself an estimate of the reconstructor, this analysis relates refinement to the prior's own uncertainty rather than to measured geometric error.

D.3 Fusion Analysis

For Fig. 5, we measure the relative injection $\|\Delta\mathbf{x}\|/\|\mathbf{x}\|$ at each fusion point, where $\mathbf{x}$ is the backbone feature before fusion and $\Delta\mathbf{x}$ the added feature, together with the gate magnitude $|\gamma_i|$. For NeoVerse (14B), $\Delta\mathbf{x}$ is the additive control signal. Because the two models inject control through different modules, we compare the injection relative to the backbone feature norm and focus on how it varies with depth and coverage rather than on its absolute value. Fig. 5a averages the injection over all denoising steps, separately for covered and uncovered tokens, and Fig. 5b reports $\mathbb{E}[|\gamma_i|]$ per denoising step. Token coverage is obtained by rendering the prior's alpha at the target cameras and downsampling it to the token grid; a token is uncovered when its alpha is below 0.05, matching the coverage descriptor of Appendix F.4. We use the same 100 generations as above and report 95% bootstrap intervals over samples.

E Additional Qualitative Results

We show additional qualitative results of the final GeoDYN model on Cam×Time (Fig. 6), DynPose (Fig. 7), and SpatialVID (Fig. 8). Cam×Time evaluates temporal control against ground-truth target videos, whereas DynPose and SpatialVID evaluate novel camera trajectories on monocular videos without target views.

F Additional Architecture and Implementation Details

F.1 4D Gaussian Refinement

Figure 9 gives an overview of the refinement module; its components are detailed below.

Gaussian input attributes. The per-Gaussian input concatenates the lifted 64-channel feature with the 16 properties in Table 8. Let $\boldsymbol{\sigma}$ be the per-axis standard deviation of the Gaussian centers in a scene and $\bar{\sigma}$ its mean. Position and timestamp determine the sparse coordinates, and diffusion-time information enters through the lifted features.

Sparse coordinates. Gaussian centers are normalized as $\mathbf{z} = (\mathbf{x} - \boldsymbol{\mu})/\boldsymbol{\sigma}$ and voxelized on a 256^3 grid with voxel width $\sigma_k/32$, covering $\pm 4\sigma$ around the scene center. The temporal coordinate is the index of the Gaussian's timestamp group. Gaussians that fall into the same 4D site are averaged, and only occupied sites are stored.

Sparse backbone and head. Three submanifold 4D convolutions (Table 9) give a receptive field of seven sites along each axis. The output is gathered back to each Gaussian and concatenated

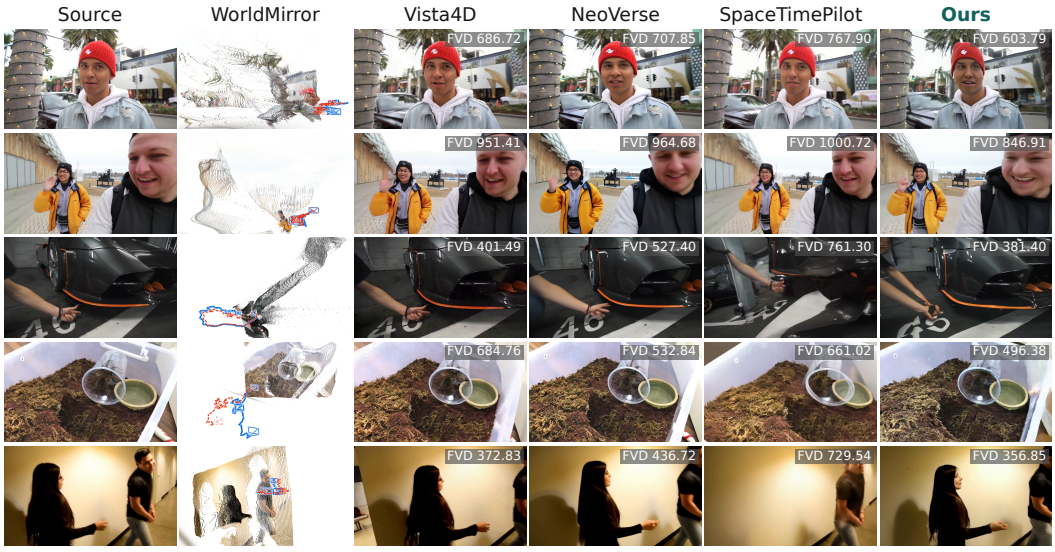


Figure 7 Qualitative results on DynPose (Rockwell et al., 2025) Each example shows the source video and the results of the compared methods and GeoDYN along a novel camera trajectory. Lower FVD indicates better performance.



Figure 8 Qualitative results on SpatialVID (Wang et al., 2026a). Each example shows the source video and the results of the compared methods and GeoDYN along a novel camera trajectory. Lower FVD indicates better performance.

with its 80-channel input ($M \times 144$), then passed through a pointwise MLP ($144 \to 128$, SiLU, $128 \to 65$) whose first 64 outputs are feature residuals and whose last output controls opacity. The skip connection (Figure 9) preserves differences between Gaussians in the same voxel. The last layer is zero-initialized, so refinement starts as the identity.

Sharing and opacity range. One refiner (1.1M parameters) is shared across the eight control stages. At each stage it receives the features lifted at that stage, and its opacity update is applied to the prior's opacity rather than accumulated across stages. Gaussian positions, shapes, and motion are left unchanged, and the sparse coordinates are computed once per prior and reused across stages. The refined opacity lies in $(0, 2\alpha_j)$ and is clipped at 0.99 per Gaussian–pixel contribution during compositing.

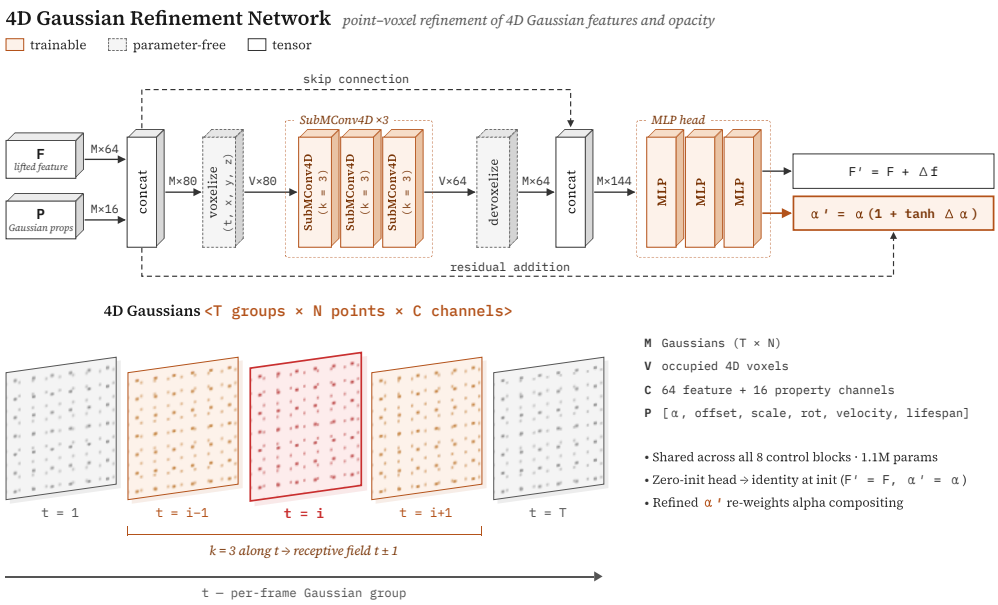


Figure 9 4D Gaussian refinement. Each of the M Gaussians carries its lifted feature $\mathbf{F}$ (64 channels) and 16 prior properties $\mathbf{P}$ (Table 8). The concatenated inputs are averaged into the V occupied sites of a sparse 4D grid indexed by timestamp group and voxelized position, processed by three submanifold 4D convolutions (Choy et al., 2019) (3^4 kernels), and gathered back to each Gaussian. The gathered context ($M \times 64$) is concatenated with each Gaussian's own input through the skip connection ($M \times 144$), and a pointwise MLP head predicts a feature residual $\Delta\mathbf{f}$ and an opacity update $\Delta\alpha$. These are added to the inputs by residual addition, $\mathbf{F}' = \mathbf{F} + \Delta\mathbf{f}$ and $\alpha' = \alpha(1 + \tanh \Delta\alpha)$. Each convolution spans adjacent timestamp groups ($t \pm 1$); the three layers together span $t \pm 3$. The head is zero-initialized, so refinement starts as the identity. Prior geometry and motion are not modified, and one refiner (1.1M parameters) is shared across all eight control blocks.

Table 8 Gaussian properties supplied to refinement.

Property	Width	Parameterization
Opacity	1	Opacity of the prior.
Subvoxel offset	3	Offset from the voxel center, scaled to $[-1, 1]$.
Scale	3	$\log s - \log \bar{\sigma}$ per axis.
Rotation	4	Quaternion with non-negative real part.
Motion	3	Forward-motion vector of the prior divided by $\bar{\sigma}$.
Temporal support	2	$(t - t_{\text{back}})/n_{\text{frames}}$ and $(t_{\text{forward}} - t)/n_{\text{frames}}$.

F.2 Control Routing, Projectors, and Fusion

Control branch. The control branch receives the clean source-video latents, the text context, and the backbone's timestep embedding. Each control block updates the source-aligned stream, whose output is lifted, refined, and rendered into a target-aligned conditioning stream; the next control block continues from the source-aligned output. Both streams have 21 latent frames and a 30×52 token grid, and transport operates at 60×104.

Projection and unprojection. Each stage has its own projector, LayerNorm–Linear$(1536, 256)$–SiLU–Linear$(256, 256)$, where $256 = r^2 c$ with $r = 2$ and $c = 64$, and its own unprojector, LayerNorm–Linear$(304, 512)$–SiLU–Linear$(512, 1536)$. At each transport location, the rendered feature (64 channels) is concatenated with normalized depth (1), binary coverage (1), the Plücker ray (6), and source correspondence (4); folding 2×2 locations yields 304 channels per token. The same unprojector is used for source reconstruction and target conditioning.

Gated fusion. Each fusion module applies eight-head cross-attention with backbone tokens as queries and target-aligned control tokens as keys and values, both normalized by LayerNorm. Attention is computed within each latent frame, with 2D rotary position embeddings. The gate takes the

Table 9 Refinement backbone. Kernel sizes are in (t, x, y, z) order.

Layer	Width	Operator
Sparse conv 1	$80 \to 64$	SubMConv4d, $(3, 3, 3, 3)$, SiLU
Sparse conv 2	$64 \to 64$	SubMConv4d, $(3, 3, 3, 3)$, SiLU
Sparse conv 3	$64 \to 64$	SubMConv4d, $(3, 3, 3, 3)$

backbone feature and the attention output and applies Linear$(3072, 1536)$, SiLU, Linear$(1536, 1)$, and tanh, producing one scalar per token. The final gate layer is zero-initialized so that fusion starts inactive. Fusion is inserted after each selected backbone block.

F.3 4D Slicing and Feature Transport

Each Gaussian is indexed by its timestamp group and its index within the group. At a query time, groups are activated by their temporal support, and positions, opacity, scale, and rotation are evaluated with the prior's motion model; the geometry itself is not learned. Because features are stored per Gaussian, features lifted from the source frames are shared across all target times. Latent frame k corresponds to video frame $4k$. For DL3DV, precomputed static Gaussians are assigned to the nearest source keyframe with zero motion.

Rasterization returns sparse pixel–Gaussian weights (up to 256 per pixel) rather than a dense transport matrix, and features are accumulated in FP32. The base geometry, correspondence, and rays are computed without gradients, while the compositing weights are recomputed from the refined opacity so that gradients flow to both the features and the opacity updates.

F.4 Camera Conventions and Geometric Descriptors

Intrinsics. Intrinsics are mapped from the WorldMirror image plane to the feature plane as

$$\mathbf{K}_{\mathrm{f}} = \mathbf{S}_{\mathrm{f}} \mathbf{A}_{\mathrm{vae}} \mathbf{A}_{\mathrm{wm}}^{-1} \mathbf{K}_{\mathrm{wm}}, \tag{11}$$

where $\mathbf{A}_{\mathrm{wm}}$ and $\mathbf{A}_{\mathrm{vae}}$ are the image transforms of the two inputs and $\mathbf{S}_{\mathrm{f}}$ is the feature-grid scaling. For 832×480 input, WorldMirror resizes to 582×336 and center-crops to 560×336. Rays pass through pixel centers.

Descriptors. Cameras follow the OpenCV convention, and world units follow the scale of the reconstructed scene. Source correspondence coordinates are the source frame index and pixel location normalized to $[-1, 1]$, so a target view can refer back to any source time, including in reversed or repeated order. The fourth correspondence channel is the accumulated prior weight of source-observed Gaussians. Depth is the composited camera depth, normalized by its 98th percentile and clipped to $[0, 1]$. The coverage descriptor thresholds the rendered alpha at 0.05. The Plücker channels contain the ray direction and moment.

F.5 Source Objective

Source reconstruction shares the descriptor channels and stage-specific unprojector with the target path. The loss in Eq. (10) of the main paper is computed in FP32 on mean-centered tokens, averaged over tokens and control stages, with $\lambda_{\mathrm{src}} = 0.03$. It is used only during training. The diffusion objective is flow matching (Lipman et al., 2023; Esser et al., 2024) with a timestep shift of 5, and text dropout is 0.1.

G Training, Inference, and Parameters

G.1 Data and Training Stages

GeoDYN is trained for 32K steps in two stages. Stage 1 (steps 0–15K) uses Cam×Time and SpatialVID; its final checkpoint is the matched model of Tab. 2 and Fig. 2. Stage 2 (steps 15K–32K) continues on six datasets, with the mixture adjusted during training (Table 10). Cam×Time provides 417 training scenes with both same-trajectory and cross-trajectory pairs. For SpatialVID and Stereo4D, where source and target coincide, we degrade the reconstructed prior (random culling and occlusion) so that the model learns to handle incomplete geometry; the degradation is strengthened in Stage 2c. Ego-Exo4D pairs are used in both directions, and DL3DV uses precomputed 3D Gaussians. Training priors use 11–21 context views, and evaluation uses 21. WorldMirror Gaussians are subsampled with stride four.

Table 10 Training datasets and sampling weights (%).

Dataset	Units	Stage 1	Stage 2a	Stage 2b	Stage 2c
Cam×Time (Huang et al., 2026)	51,708 pairs	33	15	15	15
SpatialVID (Wang et al., 2026a)	99,799 clips	67	22	15	35
Ego-Exo4D (Grauman et al., 2024)	8,614 pairs	–	12	20	–
DL3DV (Ling et al., 2024)	20,712 pairs	–	18	15	10
Stereo4D (Jin et al., 2025)	70,515 clips	–	18	10	40
BlenderProc (Denninger et al., 2023) (synthetic)	23,880 pairs	–	15	25	–

Stage 2a: steps 15K–19K; 2b: 19K–26K; 2c: 26K–32K.

Table 11 GeoDYN optimization settings.

Setting	Value
Optimizer	AdamW, $\beta = (0.9, 0.999)$, weight decay 10^{-2}.
Learning rate	10^{-5} for modules initialized from VACE and 10^{-4} for new modules, with 300 warmup steps and cosine decay, restarted at 26K and ending at $0.1\times$ at 32K.
Batch	Effective batch size 16 on four H200 GPUs.
Precision	BF16 mixed precision; refinement and feature transport in FP32; gradient clipping at 1.0.
Initialization	Control stage i is initialized from VACE (Jiang et al., 2025) block $2i$; the gate and refinement head are zero-initialized.
Frozen modules	Diffusion backbone, VAE, text encoder, and WorldMirror.

Data separation. All 50 SpatialVID evaluation videos come from a 200-video holdout that is excluded from training, and the 31 Cam×Time evaluation scenes are disjoint from its 417 training scenes. DynPose is not used for training.

G.2 Optimization

G.3 Models Used for Each Result

Table 12 lists the model used for each result; in all cases we evaluate the final checkpoint of the corresponding training run. For the baselines, we use the officially released checkpoints: BulletTime with CogVideoX1.5-5B, SpaceTimePilot, ReCamMaster, GEN3C, TrajectoryCrafter with CogVideoX-Fun-5B, Vista4D with Wan2.1-14B, and NeoVerse (14B). Code and checkpoints will be released upon publication.

G.4 Inference

GeoDYN samples with 50 flow-matching steps, 21 context views, and no classifier-free guidance, generating 81 frames at 480×832. The NeoVerse (1.3B) reimplementation uses the same number of steps and resolution. Released baselines use their default inference settings, and all outputs are evaluated at 81 frames and 480×832. Time in Tab. 3 is the mean wall-clock time per denoising step.

G.5 Parameters

Table 13 lists the trainable parameters of GeoDYN. Together with the frozen backbone, VAE, and text encoder, the full model has 1.92B parameters. The Gaussian refiner accounts for only 0.2% of the trainable parameters. For reference, the control branch of the NeoVerse (1.3B) reimplementation has 393.5M trainable parameters. Our implementation builds on PyTorch, spconv (Contributors, 2022), FlashAttention (Dao et al., 2022; Dao, 2024), and Accelerate (Gugger et al., 2022).

H Benchmark and Evaluation Details

H.1 Benchmarks

Cam×Time (Huang et al., 2026) contains 31 evaluation scenes with six temporal-control queries each (186 clips) and provides ground-truth target videos. For SpatialVID and DynPose (Rockwell et al., 2025), we select 50 source videos each and apply six camera trajectories per video (300 clips

Table 12 Models behind each reported result.

Result	Model
Matched comparison (Tab. 2, Fig. 2)	GeoDYN at 15K steps and the NeoVerse (1.3B) reimplementation at 15K steps (Appendix C); NeoVerse (14B) and BulletTime use released checkpoints.
Monocular comparison (Tab. 3)	GeoDYN at 32K steps; baselines use released checkpoints.
Component ablation (Tab. 4)	Variants trained for 8K steps on Cam×Time (Appendix D.1); the NeoVerse row is the 15K model of Tab. 2.
Category results (Tab. 5)	GeoDYN at 32K steps.
Analysis and examples (Figs. 4 and 5)	GeoDYN at 32K steps; Fig. 5 also analyzes NeoVerse (14B).

Table 13 Trainable parameters of GeoDYN.

Component	Parameters
Input patch embedding	0.10M
Input linear projection	2.36M
Eight control blocks	371.53M
Eight projectors	3.70M
Eight unprojectors	7.56M
Eight gated fusion modules	113.28M
Shared Gaussian refiner	1.11M
Total	499.63M

per dataset). Videos are selected from the upper half of the candidates ranked by the fraction of dynamic pixels, at evenly spaced ranks, so that the benchmarks emphasize dynamic content.

H.2 Camera and Time Queries

Camera trajectories. For SpatialVID and DynPose, each trajectory moves from the source camera to an endpoint over 80 frames, interpolating rotation with SLERP and translation linearly. We use six trajectory families: pan and tilt (15°), lateral, vertical, and forward translation (0.1 scene units), and orbit (15° with unit radius). Each video uses one direction per family, alternating directions across videos. The target camera is $\mathbf{P}_t^{\text{desired}} = \mathbf{P}_t^{\text{src}}\mathbf{C}_t$, where $\mathbf{C}_t$ is the relative motion, and the source intrinsics are kept.

Temporal control. Cam×Time queries specify a requested scene time for each of the 81 output frames, covering direction (backward, zigzag), bullet time (early, middle, late), and motion speed (slow motion). Requested times are expressed on the source frame axis and may be fractional; camera paths and scene times are given as separate inputs.

H.3 Metrics

Reconstruction metrics. PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) (AlexNet) are computed per frame against the ground-truth target at 480×832 and averaged over clips.

FID and FVD. FID (Heusel et al., 2017) uses Inception features of all frames, and FVD (Unterthiner et al., 2018) uses I3D features of 16 uniformly sampled frames per video. On Cam×Time, the ground-truth targets serve as references; on SpatialVID and DynPose, which have no target views, the source videos serve as references, so FID and FVD there measure fidelity to the source distribution.

VBench. We report five VBench 1.0 (Huang et al., 2024) dimensions: subject consistency, background consistency, aesthetic quality, imaging quality, and temporal flickering.

H.4 Camera Evaluation

We estimate the camera trajectories of the source and generated videos with ViPE (Huang et al., 2025a), using DA3 depth for Cam×Time and UniDepth for SpatialVID and DynPose. For the

monocular benchmarks, the target trajectory is obtained by composing the estimated source trajectory with the instructed relative motion. The generated trajectory is aligned to the target with Sim(3) Umeyama alignment, and we report the camera-center error relative to the median scene depth (translation, %) and the geodesic rotation error (degrees). In Tab. 2 and Tab. 5, camera errors are averaged over the cases for which every compared method admits a valid similarity alignment (111 Cam$\times$Time and 262 SpatialVID cases in Tab. 2), and Tab. 4 averages over all 186 Cam$\times$Time cases. Because the two tables average over different case sets, the same model, such as NeoVerse (1.3B), has different camera errors in Tab. 2 and Tab. 4; comparisons are made within each table.