RoGSW4RLD: Feed-Forward 4D Gaussian Lifting for Robot World Model Rollouts

Anonymous authors
Paper under double-blind review

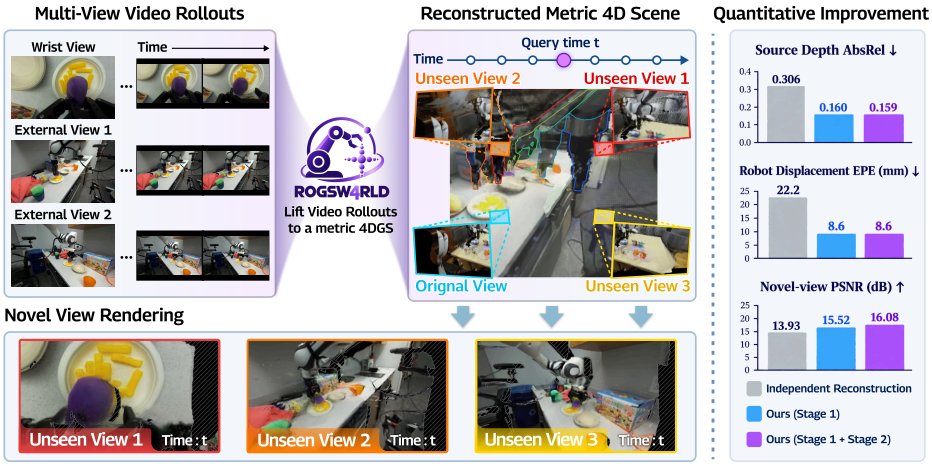


Figure 1: **Overview of RoGSW4RLD for metric 4D lifting of robot-camera rollouts.** Given synchronized multi-view videos, RoGSW4RLD reconstructs a shared, time-queryable metric 4D Gaussian scene that supports rendering from unseen viewpoints at arbitrary query times. Compared with independent camera-wise reconstruction using MoVieS (Lin et al., 2026), followed by stereo-depth metric normalization and calibrated merging in the robot frame, joint lifting substantially improves metric depth, robot-motion accuracy, and novel-view rendering, with Stage 2 further refining the reconstructed field. Bars report results on recorded observations from Table 1.

Abstract

Action-conditioned video world models predict future robot interactions from multiple cameras, yet their outputs remain disparate video collections rather than a shared metric scene queryable across viewpoints and time. While existing 4D reconstruction methods offer a path to spatialize these predictions, independently reconstructing and merging each camera stream fails to enforce cross-view consistency. This limitation is particularly detrimental when combining moving robot-mounted cameras with fixed external views. To address this, we introduce RoGSW4RLD, a feed-forward framework that lifts synchronized multi-camera rollouts into a unified, time-queryable metric 4D Gaussian field. Rather than learning a separate geometric transition model, RoGSW4RLD directly reconstructs the visual future generated by existing world models. Its core innovation is a two-stage architecture: Stage 1 jointly forms the metric 4D field by fusing cross-view evidence with robot-specific articulated geometry and kinematics, while Stage 2 refines the field's geometry and appearance while strictly preserving the initial temporal displacements. Evaluated on 256 held-out DROID episodes, RoGSW4RLD significantly outperforms camera-wise reconstruction with calibrated merging, improving novel-view PSNR by 2.15 dB, reducing depth AbsRel by 47%, and lowering robot displacement error by 61%. These robust gains extend to action-conditioned Cosmos 3 rollouts, demonstrating that predicted video futures can be successfully translated into consistent, spatially queryable 4D metric representations.

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.

1 INTRODUCTION

Video world models predict how physical interactions evolve under future actions (Ebert et al., 2018; Du et al., 2023; Yang et al., 2024a). Spatial world models forecast geometric evolution (Lu et al., 2025; Huang et al., 2026; Zhen et al., 2025), and related approaches construct dynamic Gaussian scenes from video generators (Bahmani et al., 2026; Pan et al., 2026). In contrast, multi-camera models like Cosmos 3 (NVIDIA, 2026) output synchronized interaction videos but still lack a shared, temporally queryable metric scene. To address this, we lift these multi-view rollouts into a unified metric 4D Gaussian field (Figure 1). Rather than learning a new geometric transition model, our approach reconstructs the generated visual future directly.

A naive approach would independently reconstruct each camera stream and merge the fields using camera calibration. However, post-hoc alignment fails to enforce cross-camera consistency during geometry and motion estimation, often leaving duplicated or conflicting surfaces (Figure 2). This issue is especially prominent in setups like DROID (Khazatsky et al., 2024), which combine fixed external and moving robot-mounted cameras where views entangle ego-motion with scene dynamics. While recent feed-forward methods enable spatio-temporal reconstruction (Ma et al., 2025; Balice et al., 2026), robot-camera settings specifically require joint metric estimation guided by the robot's physical geometry and kinematics.

We propose RoGSW4RLD, a feed-forward framework that jointly lifts synchronized robot-camera rollouts into a shared metric 4D Gaussian field. Instead of relying on post-hoc alignment, our key insight is to integrate robot-specific spatial and motion priors directly during field formation. Using joint states, hand–eye calibration, articulated meshes, and forward kinematics (FK), we ground the geometry and motion estimation within a common robot frame, while inferring object and scene dynamics solely from the visual rollout.

Our method operates in two feed-forward stages without per-scene optimization. Stage 1 performs robot-aware lifting: combining multi-view evidence with articulated geometry to anchor robot surfaces at source times, while using FK to guide, rather than analytically replace, the learned deformations of all trajectories. Stage 2 applies displacement-preserving refinement to correct residual placement and appearance errors via source-view rendering feedback. By sharing position corrections across query times, this stage perfectly preserves the relative temporal displacements established in Stage 1.

Experiments on held-out DROID episodes demonstrate that RoGSW4RLD significantly improves novel-view rendering, metric geometry, and robot-motion accuracy compared to calibrated unions of independent reconstructions. These gains persist for action-conditioned Cosmos 3 rollouts, yielding a more consistent spatial representation of predicted futures.

Our contributions are threefold: i) a novel framework for jointly lifting synchronized multi-view robot rollouts into a shared metric 4D Gaussian field; ii) a two-stage architecture featuring robot-aware lifting and displacement-preserving refinement via articulated geometry and kinematics; and iii) substantial empirical improvements in metric geometry, novel-view synthesis, and motion accuracy on both recorded and generated robotic rollouts.

2 RELATED WORK

2.1 VIDEO AND SPATIAL WORLD MODELS

Video world models. Video prediction has evolved from action-conditioned dynamics for control (Finn et al., 2016; Ebert et al., 2018) to comprehensive video-based planning and simulation (Du et al., 2023; Yang et al., 2024a). Recent models like Cosmos 3 (NVIDIA, 2026) and Ctrl-World (Guo et al., 2026) generate controllable robot rollouts, while TesserAct (Zhen et al., 2025) predicts RGB alongside depth and normals. Other approaches focus on improving cross-view consistency (Liu et al., 2026; Wang et al., 2026) or utilize rendered robot motion as a visual condition (Kim et al., 2026; Gu et al., 2026). Instead of modifying the video generation process itself, RoGSW4RLD focuses on lifting the synchronized rollouts of a frozen world model into a shared metric scene.

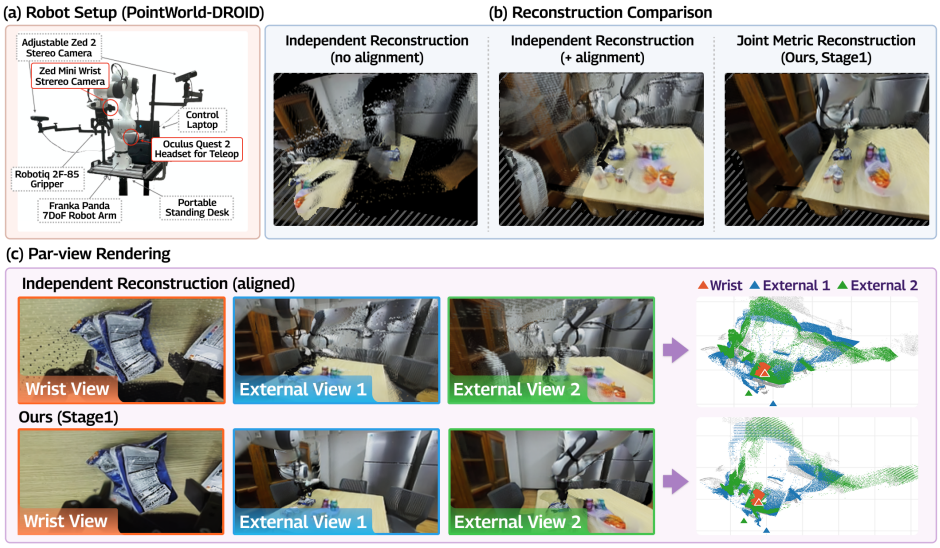


Figure 2: **Robot-camera setup and limitations of post-hoc field merging.** (a) PointWorld-DROID setup with a moving wrist camera and fixed external cameras. (b) Independent reconstruction retains inconsistent geometry after calibrated alignment, while our joint metric reconstruction (Stage 1) forms a more coherent scene. (c) Per-view renderings and merged geometry show reduced duplication and misalignment across cameras.

Spatial world models. Spatial world models directly forecast geometric evolution. For instance, GWM (Lu et al., 2025) learns action-conditioned transitions within a Gaussian latent space, and PointWorld (Huang et al., 2026) predicts 3D point flows driven by robot surface motion. Rather than learning a novel geometric transition model, RoGSW4RLD leverages the supplied visual rollout as a predicted future, reconstructing its underlying geometry and motion into a unified metric 4D field.

2.2 Dynamic Scene Reconstruction

Optimization-based reconstruction. Dynamic neural and Gaussian fields typically model scene evolution via deformation or structured space-time representations (Pumarola et al., 2021; Park et al., 2021; Wu et al., 2024; Yang et al., 2024b; Lei et al., 2025; Wang et al., 2025b). While these methods conventionally require per-scene optimization at test time, RoGSW4RLD employs a feed-forward approach, performing joint lifting and recurrent refinement using fixed network weights.

Feed-forward reconstruction. Feed-forward techniques amortize the reconstruction cost across scenes. For monocular videos, 4DGT (Xu et al., 2025) predicts temporally coherent Gaussians, and MoVieS (Lin et al., 2026) estimates depth, Gaussian attributes, and query-time deformation. 4D-LRM (Ma et al., 2025) extends reconstruction across varying views and times. Recent pose-free methods address monocular (Yang et al., 2026), driving (Chen et al., 2026), two-image (Hur et al., 2026), and multi-view (Balice et al., 2026) setups. Building upon MoVieS, our framework integrates robot geometry and kinematics into the joint metric reconstruction of calibrated moving or fixed cameras. Furthermore, we adapt ReSplat (Xu et al., 2026) by introducing a time-shared refinement mechanism that strictly preserves the per-Gaussian center displacements.

Generative priors for 4D reconstruction. Video generation models offer rich observations and features for reconstruction. CAT4D (Wu et al., 2025) generates multi-view videos prior to optimizing dynamic Gaussians, whereas Diff4Splat (Pan et al., 2026) decodes video diffusion representations directly into dynamic Gaussians. Lyra (Bahmani et al., 2026) distills camera-controlled video diffusion into a feed-forward Gaussian decoder, extending this to dynamic video-to-4D generation. In contrast, we specifically target the joint metric lifting of robot-camera rollouts. We primarily rely on decoded RGB inputs, though an optional latent adapter can replace the token-input pathway for tighter integration.

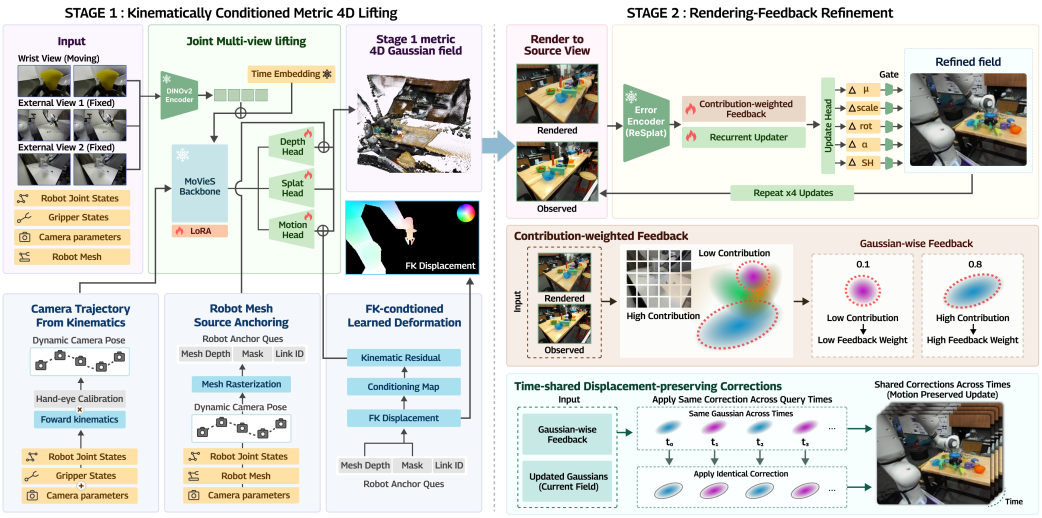


Figure 3: **Overview of RoGSW4RLD.** Stage 1 performs kinematically conditioned metric 4D lifting by jointly processing multi-view videos with calibrated camera trajectories, robot-mesh source anchoring, and FK-conditioned learned deformation. Stage 2 recurrently refines the resulting Gaussian field using contribution-weighted rendering feedback, while time-shared corrections preserve each Gaussian's temporal displacement across query times.

3 Problem Formulation

We reconstruct dynamic robotic scenes from synchronized multi-view observations, either recorded or generated by a video world model. For source cameras $\mathcal{V}$ and source times $\mathcal{S}$, inputs comprise RGB images $\mathbf{I}_{v,s}$, intrinsics $\mathbf{K}_v$, and calibrated camera-to-reconstruction transforms $\mathbf{C}_{v,s} \in SE(3)$. All 3D positions and displacements are expressed in a shared metric reconstruction frame, fixed throughout each clip and related to the robot base by a known rigid transform. Cameras may be fixed or follow supplied trajectories.

An articulated robot mesh is given, and joint configurations $\mathbf{q}_t$ and normalized gripper states $g_t \in [0, 1]$ are supplied at source and query times. Forward kinematics (FK), followed by the fixed frame conversion, yields link-to-reconstruction transforms $\mathbf{T}_\ell(t) \in SE(3)$. Robot-mounted camera poses follow from these transforms and fixed hand–eye calibration.

Our goal is to reconstruct a time-queryable Gaussian field

$$\mathcal{G}(t) = \{\boldsymbol{\mu}_i(t), \boldsymbol{\Sigma}_i(t), \alpha_i(t), \mathbf{c}_i(t, \boldsymbol{\omega})\}_{i=1}^{N}, \tag{1}$$

where the attributes denote center, covariance, opacity, and color for viewing direction $\boldsymbol{\omega}$, respectively. Gaussian identities persist across query times. The field is rendered as $\hat{\mathbf{I}}_{v,t} = \mathcal{R}(\mathcal{G}(t); \mathbf{K}_v, \mathbf{C}_{v,t})$, where $\mathcal{R}$ denotes differentiable Gaussian rasterization (Kerbl et al., 2023).

4 RoGSW4RLD: Robot-Aware Metric 4D Lifting

Stage 1 combines multi-camera observations with robot mesh source anchoring and FK-conditioned deformation to form a shared metric field. Stage 2 then refines its placement and appearance using rendering feedback and time-shared corrections (Figure 3). Both stages use fixed network weights at inference, without per-scene optimization.

4.1 Stage 1: Kinematically Conditioned Metric 4D Lifting

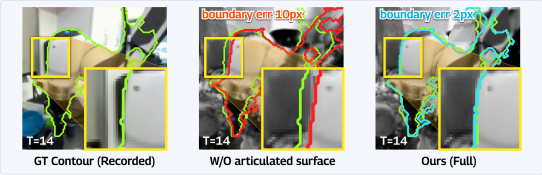
Joint multi-camera lifting. We initialize the backbone from MoVieS (Lin et al., 2026), which builds on VGGT (Wang et al., 2025a). To adapt this backbone from monocular-video reconstruction to joint multi-camera lifting, we apply LoRA (Hu et al., 2022) to the QKV and output projections

in its frame-wise and global attention blocks. Source tokens are processed jointly across views and times with camera and time encodings. The depth and splatter heads predict source depth and base Gaussian attributes, respectively. Following MoVieS, a query-time-conditioned motion head predicts 3D displacements and time-dependent Gaussian attributes. We use a fixed conversion between internal length units and metres across clips, without per-scene or camera-specific rescaling (Appendix A.1).

Robot mesh source anchoring. Each Gaussian i retains its source camera v_i, source time s_i, and source pixel $\mathbf{u}_i$. We rasterize the articulated robot mesh at each source state to obtain a depth anchor map and per-pixel link identities. During both training and inference, we select source depth D_i from mesh depth at valid robot-support pixels and from the depth-head prediction elsewhere. Back-projecting D_i, measured along the source camera's optical axis, gives the source position in the shared reconstruction frame:

$$\mathbf{p}_i = \mathbf{R}_{v_i,s_i}\left(D_i \mathbf{K}_{v_i}^{-1}\bar{\mathbf{u}}_i\right) + \mathbf{t}_{v_i,s_i}, \qquad (2)$$

where $\mathbf{R}_{v_i,s_i}$ and $\mathbf{t}_{v_i,s_i}$ are the rotation and translation of $\mathbf{C}_{v_i,s_i}$, and $\bar{\mathbf{u}}_i$ is the homogeneous source pixel. Each mesh-anchored Gaussian also retains its link identity ℓ_i for the FK-conditioned deformation described next. Support and boundary criteria are detailed in Appendix A.



(a) Robot Mesh Anchoring (Where the robot is)



(b) FK Conditioning (How the robot moves)

Figure 5: **Robot-aware geometry and motion cues.** (a) Robot mesh anchoring improves novel-view contour alignment (cyan), providing accurate robot placement. (b) FK conditioning suppresses remnants of the earlier pose (dashed white) and follows the query pose (green).

FK-conditioned learned deformation. For a mesh-anchored Gaussian i associated with link ℓ_i, the FK displacement from source time s_i to query time t is

$$\mathbf{d}_i^{\mathrm{FK}}(t) = \mathbf{T}_{\ell_i}(t)\mathbf{T}_{\ell_i}(s_i)^{-1}\mathbf{p}_i - \mathbf{p}_i, \qquad (3)$$

where rigid transforms act on 3D points through homogeneous coordinates. We construct an FK conditioning map $\boldsymbol{\kappa}_{v,s}(t)$ aligned with each source image grid. It encodes robot support, mesh depth, and per-pixel 3D FK displacements; auxiliary cues are detailed in Appendix A.

Rather than applying FK displacements directly, we use this map to augment the motion head's visual displacement prediction and obtain the query-time center:

$$\mathbf{d}_i(t) = \mathbf{d}_i^{\mathrm{vis}}(t) + \left[\mathcal{B}(\boldsymbol{\kappa}_{v_i,s_i}(t))\right]_{\mathbf{u}_i}, \qquad (4)$$

$$\boldsymbol{\mu}_i^{(0)}(t) = \mathbf{p}_i + \mathbf{d}_i(t), \qquad (5)$$

where $\mathbf{d}_i^{\mathrm{vis}}(t)$ is the visual displacement prediction and $\mathcal{B}$ is a convolutional residual branch with a zero-initialized final convolution. The same FK displacement supervises mesh-anchored Gaussians during training. Robot and non-robot Gaussians share the query-time deformation and attribute-decoding pathways. Their centers and attributes together define the shared field rendered from each query camera. Figure 5 illustrates the roles of mesh anchoring and FK conditioning.

4.2 Stage 2: Displacement-Preserving Rendering-Feedback Refinement

Stage 2 adapts the recurrent refiner of ReSplat (Xu et al., 2026) to correct placement and appearance errors revealed by source-view renderings. Stage 1 remains frozen, while source bindings, camera geometry, and the shared reconstruction frame remain fixed.

Contribution-weighted rendering feedback. At refinement update r, where $r = 0$ denotes the Stage 1 field, we render the full field $\mathcal{G}^{(r)}(s)$ at each source time s from all source cameras. A frozen error encoder converts discrepancies between rendered and observed RGB images into contextual

features $\mathbf{E}_s^{(r)} \in \mathbb{R}^{P_s \times F}$, where P_s counts pixels across cameras and F is the feature dimension. To assign feedback according to each Gaussian's current visibility and overlap with other Gaussians, we use an alpha-compositing contribution matrix $\mathbf{W}_s^{(r)} \in \mathbb{R}^{P_s \times N}$, whose rows correspond to pixels and columns to Gaussians. This matrix is recomputed after each update.

For Gaussians with $s_i = s$, we compute contribution mass $m_i^{(r)}(s)$ and pool feedback only when $m_i^{(r)}(s) > \epsilon$:

$$m_i^{(r)}(s) = \left[\left(\mathbf{W}_s^{(r)} \right)^\top \mathbf{1} \right]_i, \tag{6}$$

$$\mathbf{f}_i^{(r)}(s) = \frac{\left[\left(\mathbf{W}_s^{(r)} \right)^\top \mathbf{E}_s^{(r)} \right]_i}{m_i^{(r)}(s)}, \tag{7}$$

where $\epsilon > 0$ excludes negligible contributions. The recurrent updater combines this contribution-normalized feedback with current Gaussian attributes and its hidden state to predict corrections.

Time-shared displacement-preserving corrections. Stage 1 uses visual and FK cues to learn displacements that move Gaussians from different source views and times to a common query time. Independent position updates across query times could improve individual renderings but alter these learned displacements. To preserve them while refining Gaussian centers, we apply an accumulated position correction $\boldsymbol{\delta}_i^{(r)}$ to each Gaussian after r updates, shared across query times and rendering cameras:

$$\boldsymbol{\mu}_i^{(r)}(t) = \boldsymbol{\mu}_i^{(0)}(t) + \boldsymbol{\delta}_i^{(r)}, \qquad \boldsymbol{\delta}_i^{(0)} = \mathbf{0}. \tag{8}$$

Thus, for any t and t',

$$\boldsymbol{\mu}_i^{(r)}(t) - \boldsymbol{\mu}_i^{(r)}(t') = \boldsymbol{\mu}_i^{(0)}(t) - \boldsymbol{\mu}_i^{(0)}(t'). \tag{9}$$

The updater also predicts time-shared residuals for scale, rotation, opacity, and spherical-harmonic coefficients. These residuals use the corresponding Stage 1 parameterizations and are applied to the attributes evaluated at each query time.

4.3 Training and Inference

Training. During Stage 1 training, we freeze the original backbone weights and optimize the LoRA adapters, prediction heads, and FK-conditioning branch using rendering, metric-depth, 3D-track, robot-FK, and opacity-coverage supervision. We then freeze Stage 1 and the observation and error encoders to train the Stage 2 recurrent updater and residual gates. Its objective combines rendering, metric-depth, robot-position, rendered-motion, track-preservation, and coverage losses. Paired-right views contribute training losses but are excluded from reconstruction tokens and recurrent feedback. Stereo depth and 3D track references are used for supervision and evaluation, not as inference inputs. Stage 2 is trained with one to four refinement updates. Further training details are given in Appendix A.

Inference. At inference, we use four refinement updates with fixed network weights and without per-scene optimization. An optional latent-token adapter replaces only the RGB-token pathway, leaving the remaining RGB pathways and both reconstruction stages unchanged (Appendix E).

5 Experiments

5.1 Experimental Setup

We evaluate 256 DROID episodes from ILIAD and PennPAL (Khazatsky et al., 2024), held out from training and validation of both stages and the optional latent adapter. Episodes are selected from the official PointWorld test manifest (Huang et al., 2026) before inspecting outputs. Three left-eye cameras supply reconstruction inputs; paired right-eye cameras provide held-out viewpoints.

Table 1: **Quantitative results on recorded observations and Cosmos 3 rollouts.** RoGSW4RLD improves rendering, metric depth, and robot-displacement accuracy over camera-wise baselines. Both input settings are evaluated against recorded references.

Method	Source view			Novel view			Robot	Efficiency	
	PSNR ↑	LPIPS ↓	AbsRel ↓	PSNR ↑	LPIPS ↓	AbsRel ↓	EPE (mm) ↓	Time (s) ↓	VRAM (GiB) ↓
Recorded observations									
MoVieS†	19.71	0.306	0.306	13.93	0.489	0.373	22.2	1.04	4.46
4DGT	16.54	0.456	0.600	12.90	0.584	0.681	31.0	4.80	3.89
SoM†	14.22	0.443	0.383	12.14	0.557	0.429	32.2	1713	2.35
Ours (Stage 1)	21.92	0.171	0.160	15.52	**0.363**	0.200	**8.6**	0.59	5.20
Ours (Stage 1+2)	**25.46**	**0.166**	**0.159**	**16.08**	0.371	**0.197**	**8.6**	9.08	13.11
Cosmos 3 rollouts									
MoVieS†	18.33	0.348	0.343	13.69	0.503	0.404	22.6	1.04	4.46
4DGT	15.78	0.483	0.625	12.78	0.591	0.697	32.3	4.80	3.89
SoM†	13.92	0.463	0.391	12.05	0.564	0.440	37.6	1985	2.39
Ours (Stage 1)	19.87	0.235	0.182	15.04	**0.394**	0.218	**8.7**	0.59	5.20
Ours (Stage 1+2)	**22.12**	**0.231**	**0.181**	**15.52**	0.402	**0.215**	**8.7**	9.08	13.11

$n = 256$ (246 successful clips for SoM). †: one scale factor per camera from recorded source stereo depth.
See Appendix B for evaluation details and Appendix D.4 for method-specific timing and memory measurements.

Recorded and Cosmos 3 (NVIDIA, 2026) rollout settings share episodes, a 17-frame interval, calibration, robot trajectories, query times, and recorded targets. Rollouts are conditioned on the initial three-view observation and a 16-step FK-derived end-effector action chunk; reconstruction retains the original joint and gripper trajectories without inverse kinematics. RoGSW4RLD and MoVieS use frames $\{0, 4, 8, 12, 16\}$; 4DGT and SoM use all 17. A fixed plan samples up to eight query requests per clip, evaluating duplicate timestamps once. Source views denote cameras, not input times.

We compare with MoVieS (Lin et al., 2026), 4DGT (Xu et al., 2025), and Shape of Motion (SoM) (Wang et al., 2025b). Each follows its released monocular setting, reconstructing camera streams separately and rendering the resulting fields after calibrated merging. MoVieS and SoM receive one source-stereo scale scalar per camera; RoGSW4RLD and 4DGT receive no test-time stereo depth. Held-out RGB is not directly supplied, and no field is fitted to query-view depth. SoM completes 246 clips; others complete 256.

We report source- and novel-view PSNR, LPIPS (Zhang et al., 2018), and depth AbsRel. Robot EPE measures displacement; robot depth error measures placement on fixed support. Depth references are stereo estimates, not sensor ground truth. We average image scores over cameras and distinct times within each clip, then equally across clips. Confidence intervals use 10,000 clip-bootstrap resamples, reflecting test-clip rather than training-seed variability. Appendix B details the protocol.

5.2 Reconstruction Results and Efficiency

Table 1 shows improvements over camera-wise reconstruction followed by calibrated merging. Stage 1 and the full model both outperform the baselines in rendering, metric depth, and robot displacement. On recorded inputs, our final model improves novel-view PSNR over MoVieS by 2.15 dB, reduces AbsRel from 0.373 to 0.197, and lowers robot EPE from 22.2 to 8.6 mm. Most geometric and motion gains are already present in Stage 1. These are system-level comparisons, since architectures, training, and robot inputs differ.

With generated rollouts, final novel-view PSNR decreases by 0.56 dB and AbsRel increases by 0.018 relative to recorded inputs. Nevertheless, both remain better than MoVieS: 15.52 versus 13.69 dB and 0.215 versus 0.404. The reconstruction advantage therefore persists when the visual future is generated. Evaluation against recorded futures combines rollout and reconstruction errors. Robot trajectories are unchanged, so stable robot EPE does not establish accurate visual motion generation. Figure 6 illustrates reduced ghosting in our source-view reconstructions of generated rollouts; recorded-input novel-view comparisons appear in Appendix C.1.

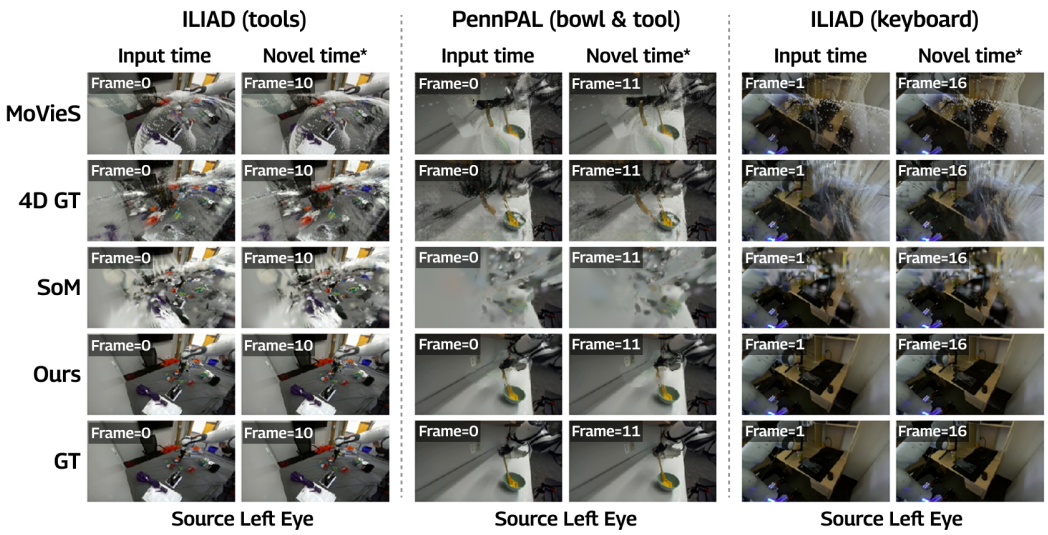


Figure 6: **Qualitative comparison on action-conditioned Cosmos 3 rollouts.** Across ILIAD and PennPAL scenes, RoGSW4RLD produces cleaner reconstructions with less ghosting and sharper robot and object boundaries than the baselines at both input and novel times. The bottom row shows the corresponding generated Cosmos 3 frames, rather than recorded observations; asterisks denote non-input query times for RoGSW4RLD and MoVieS.

Stage 2 improves source and novel-view PSNR by 3.54 and 0.56 dB on recorded inputs, respectively, and by 2.25 and 0.48 dB on rollouts. In both settings, novel-view scores show a 0.003 decrease in AbsRel but worse LPIPS, while robot EPE remains nearly unchanged. Refinement thus primarily improves photometric fitting. Time-shared corrections preserve Gaussian center displacements, but rendered motion can change with compositing weights.

Stage 1 requires 0.59 s and 5.20 GiB per clip, versus 9.08 s and 13.11 GiB with four refinement updates. Both use fixed inference weights without per-scene optimization. A 32-clip recurrence diagnostic shows diminishing PSNR gains beyond two updates (Appendix D.3); Appendix D.4 details timing and memory.

5.3 Ablation Study

We examine cross-camera interaction, rendering-feedback refinement, robot mesh source anchoring, and FK conditioning through frozen-checkpoint interventions (Table 2). These diagnose inference-time dependence rather than compare separately retrained variants.

Joint lifting and refinement. We block cross-camera interaction while retaining within-camera temporal interaction, calibration, and robot inputs, then merge the resulting camera-wise fields. In Table 2(a), camera-wise refinement raises recorded source PSNR from 19.64 to 23.71 dB but leaves novel-view AbsRel at 0.391–0.392. Joint Stage 1 instead achieves 0.200 AbsRel at 21.92 dB source PSNR. Thus, stronger source-view fitting does not recover the held-out geometry obtained through joint multi-camera lifting.

Rollouts show the same separation. Stage 2 raises source PSNR of the camera-wise reconstruction from 18.38 to 21.37 dB, while novel-view AbsRel remains at 0.390–0.391. Joint Stage 1 instead reaches 0.218 AbsRel. Because this intervention blocks cross-camera interaction only at inference, it should not be interpreted as training three independent monocular models from scratch.

Robot geometry and kinematics. We disable mesh anchoring, FK conditioning, or both from the same frozen checkpoint. When mesh anchoring is disabled, supported robot geometry falls back to the learned visual geometry path while FK conditioning remains available. Conversely, removing FK conditioning retains mesh anchoring. Camera calibration is unchanged. FK-conditioning re-

Table 2: **Frozen-checkpoint component interventions.** (a) Joint lifting improves novel-view geometry; refinement mainly improves RGB fitting. (b) Mesh anchoring improves robot placement, while FK conditioning improves displacement accuracy.

(a) Joint multi-camera lifting and refinement

Cross camera	Stage2	Source PSNR ↑	Novel PSNR ↑	Novel AbsRel ↓	Robot EPE (mm) ↓	Robot depth err. (mm) ↓
Recorded Observations						
✓	✓	**25.46**	**16.08**	**0.197**	**8.6**	13.7
✓	–	21.92	15.52	0.200	**8.6**	**13.4**
–	✓	23.71	12.93	0.392	8.8	101.6
–	–	19.64	12.64	0.391	8.8	100.6
Cosmos 3 Rollouts						
✓	✓	**22.12**	**15.52**	**0.215**	**8.72**	15.5
✓	–	19.87	15.04	0.218	**8.72**	**15.2**
–	✓	21.37	12.75	0.391	8.90	97.9
–	–	18.38	12.50	0.390	8.92	97.2

(b) Robot geometry and motion

Mesh anchoring	FK cond.	Source PSNR ↑	Novel PSNR ↑	Novel AbsRel ↓	Robot EPE (mm) ↓	Robot depth err. (mm) ↓
Recorded Observations						
✓	✓	**25.46**	**16.08**	0.197	**8.64**	**13.7**
✓	–	23.26	15.77	0.206	30.15	22.1
–	✓	25.29	16.07	**0.196**	8.81	37.6
–	–	23.29	15.79	0.207	30.12	42.9
Cosmos 3 Rollouts						
✓	✓	**22.12**	15.52	**0.215**	**8.72**	**15.5**
✓	–	21.20	15.34	0.223	30.16	24.5
–	✓	22.02	**15.56**	0.218	8.90	56.6
–	–	21.21	15.39	0.227	30.14	62.7

All interventions are applied at inference without retraining ($n = 256$); (b) uses Stage 1+2.
Robot depth error is source-view depth MAE on fixed robot support.
See Appendix B for evaluation details and Appendix D.1 for intervention definitions and Stage 1 results.

moval is supported by conditioning dropout during training, whereas mesh anchoring removal and removal of both are diagnostic out-of-distribution interventions.

Table 2(b) distinguishes placement from displacement. On recorded inputs, removing FK conditioning raises EPE from 8.64 to 30.15 mm. Removing mesh anchoring instead raises robot depth error from 13.7 to 37.6 mm, while EPE remains at 8.81 mm. Removing both degrades both metrics.

Rollouts show the same pattern. FK removal raises EPE from 8.72 to 30.16 mm. Removing mesh anchoring raises robot depth error from 15.5 to 56.6 mm while EPE remains at 8.90 mm. The larger rollout placement penalty is observed, but its cause is not isolated by these interventions. Scene-wide AbsRel does not uniformly improve with mesh anchoring because the intervention targets the robot region.

Figure 5 visualizes the same distinction at non-input query times. Without mesh anchoring, the metric robot surface shows a larger disparity shift. Without FK conditioning, stale robot content remains near an earlier pose. Appendix D.1 reports the stage-wise intervention details. Analytic FK transport slightly improves robot EPE, so learned deformation is not claimed to be more accurate than analytic kinematics (Appendix D.2). Its role is instead to keep robot and non-robot dynamics within a shared learned deformation representation.

6 Conclusion

We presented RoGSW4RLD, a feed-forward framework that converts synchronized robot-camera rollouts into a shared, time-queryable metric 4D Gaussian field. Rather than learning a separate geometric transition model, it directly reconstructs visual futures from existing video world models. Our core novelty lies in jointly forming this metric field across moving wrist and fixed external cameras by integrating articulated robot geometry and kinematics. This approach circumvents the cross-view inconsistencies of post-hoc alignment, substantially enhancing geometric consistency and motion accuracy. Additionally, our displacement-preserving refinement boosts visual quality without altering the estimated temporal dynamics.

Evaluations on recorded DROID observations and action-conditioned Cosmos 3 rollouts validate these advantages across novel-view synthesis, metric geometry, and displacement accuracy. These results prove that predicted multi-camera videos can seamlessly become coherent, queryable metric scenes without per-scene optimization or retraining the underlying world model. Ultimately, RoGSW4RLD bridges video predictions and explicit spatial representations to facilitate downstream robotic reasoning. Future work will extend this framework beyond calibrated setups to unconstrained sensing, longer horizons, and diverse robotic configurations.

AI USE STATEMENT

In this work, we used generative AI tools to support research ideation and discussions of experimental design, identify relevant literature, and draft and polish portions of the manuscript.

We have not used generative AI tools for synthetic dataset generation, mathematical proofs, or any purposes beyond those described above.

At least two authors reviewed all AI-assisted content and revised it where necessary. This review included checking references against the original sources and verifying that technical descriptions and interpretations were consistent with our methods and experimental results. The authors retain full responsibility for the accuracy and integrity of the paper and its associated artifacts.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix A describe the architecture, losses, and training schedules of both stages, and Appendix A.1 describes the coordinate and unit conventions. Section 5.1 and Appendix B describe the evaluation cohort, input protocol, baseline adaptations, and metrics, and Table 7 reports confidence intervals. The supplementary material contains anonymized code for training, evaluation, and result aggregation, including the configuration files with all hyperparameters. Its README links to anonymous downloads of the trained weights and of the complete evaluation data of the main tables, together with our per-clip reference results. All data are derived from public releases (DROID, PointWorld, Cosmos 3, Wan2.2).

REFERENCES

Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David B. Lindell, Zan Gojcic, Sanja Fidler, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra: Generative 3D scene reconstruction via self-distillation with video diffusion models. In *International Conference on Learning Representations*, 2026. URL https://research.nvidia.com/labs/toronto-ai/lyra/.

Matteo Balice, Yanik Kunzi, Chenyangguang Zhang, Matteo Matteucci, Marc Pollefeys, and Sunghwan Hong. No pose, no problem in 4D: Feed-forward dynamic gaussians from unposed multi-view videos, 2026. URL https://arxiv.org/abs/2605.22190. Accepted at NeurIPS 2026.

Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, Hangjun Ye, Hongyang Li, Ya-Qin Zhang, and Hao Zhao. DGGT: Feedforward 4D reconstruction of dynamic driving scenes using unposed images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Chen_DGGT_Feedforward_4D_Reconstruction_of_Dynamic_Driving_Scenes_using_Unposed_CVPR_2026_paper.html.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0403. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1d5b9233ad716a43be5c0d3023cb82d0-Abstract-Conference.html.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control, 2018. URL https://arxiv.org/abs/1812.00568.

Chelsea Finn, Ian J. Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, volume 29, pp. 64–72, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/d9d4f495e875a2e075a1a4a6e1b9770f-Abstract.html.

Chenghao Gu, Hanyang Yu, Jingbo Zhang, Haitao Lin, Wenyao Zhang, Jinghe Wang, Hanglei Jin, Shuzhao Xie, Jingyan Jiang, and Zhi Wang. GeniWorld: A generalizable interactive world model

for robotic manipulation via visual actions, 2026. URL https://arxiv.org/abs/2608.06332.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, 2026. URL https://ctrl-world.github.io/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. PointWorld: Scaling 3D world models for in-the-wild robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://point-world.github.io/.

Junhwa Hur, Charles Herrmann, Songyou Peng, Philipp Henzler, Zeyu Ma, Todd Zickler, and Deqing Sun. UFO-4D: Unposed feedforward 4D reconstruction from two images. In *International Conference on Learning Representations*, 2026. URL https://ufo-4d.github.io/.

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

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R. Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J. Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. doi: 10.15607/RSS.2024.XX.120. URL https://www.roboticsproceedings.org/rss20/p120.html.

Byungjun Kim, Taeksoo Kim, Hyunsoo Cha, and Hanbyul Joo. Robot-factored world models via robot rendering, 2026. URL https://arxiv.org/abs/2607.22535.

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*, pp. 6165–6177, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Lei_MoSca_Dynamic_Gaussian_Fusion_from_Casual_Videos_via_4D_Motion_CVPR_2025_paper.html.

Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Tao Hu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu. MoVieS: Motion-aware 4D dynamic view synthesis in one second. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–306, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lin_MoVieS_Motion-Aware_4D_Dynamic_View_Synthesis_in_One_Second_CVPR_2026_paper.html.

Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4D video generation for robot manipulation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=18gC6pZVVc.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. GWM: Towards scalable gaussian world models for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9263–9274, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Lu_GWM_Towards_Scalable_Gaussian_World_Models_for_Robotic_Manipulation_ICCV_2025_paper.html.

Ziqiao Ma, Xuweiyi Chen, Shoubin Yu, Sai Bi, Kai Zhang, Ziwen Chen, Sihan Xu, Jianing Yang, Zexiang Xu, Kalyan Sunkavalli, Mohit Bansal, Joyce Chai, and Hao Tan. 4D-LRM: Large space-time reconstruction model from and to any view at any time. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2730. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/753d9584b57ba01a10482f1ea7734a89-Abstract-Conference.html.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI, 2026. URL https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Panwang Pan, Chenguo Lin, Chenxin Li, Jingjing Zhao, Yuchen Lin, Haopeng Li, Yunlong Lin, Kairun Wen, Yixuan Yuan, and Yadong Mu. Diff4Splat: Repurposing video diffusion models for dynamic scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4232–4244, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Pan_Diff4Splat_Repurposing_Video_Diffusion_Models_for_Dynamic_Scene_Generation_CVPR_2026_paper.html.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5865–5874, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Park_Nerfies_Deformable_Neural_Radiance_Fields_ICCV_2021_paper.html.

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*, pp. 10318–10327, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Pumarola_D-NeRF_Neural_Radiance_Fields_for_Dynamic_Scenes_CVPR_2021_paper.html.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.

Jiaxu Wang, Yicheng Jiang, Tianlun He, Jingkai Sun, Qiang Zhang, Junhao He, Jiahang Cao, Zesen Gan, Mingyuan Sun, Qiming Shao, and Xiangyu Yue. MVISTA-4D: View-consistent 4D world model with test-time action inference for robotic manipulation. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.09878.

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*, pp. 9660–9672, 2025b. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wang_Shape_of_Motion_4D_Reconstruction_from_a_Single_Video_ICCV_2025_paper.html.

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. doi: 10.1109/TIP.2003.819861.

Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. FoundationStereo: Zero-shot stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5249–5260, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wen_FoundationStereo_Zero-Shot_Stereo_Matching_CVPR_2025_paper.html.

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*, pp. 20310–20320, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wu_4D_Gaussian_Splatting_for_Real-Time_Dynamic_Scene_Rendering_CVPR_2024_paper.html.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T. Barron, and Aleksander Holynski. CAT4D: Create anything in 4D with multi-view video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26057–26068, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wu_CAT4D_Create_Anything_in_4D_with_Multi-View_Video_Diffusion_Models_CVPR_2025_paper.html.

Haofei Xu, Daniel Barath, Andreas Geiger, and Marc Pollefeys. ReSplat: Learning recurrent gaussian splatting. In *European Conference on Computer Vision*, 2026. URL https://github.com/cvg/resplat.

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*, volume 38, pp. 47576–47602, 2025. doi: 10.52202/085713-1419. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3cbf33008024aa1612ce853ef78e0e53-Abstract-Conference.html.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c4d66eae503694424123b93ac0fbaf17-Abstract-Conference.html.

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. URL https://arxiv.org/abs/2601.00393.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20331–20341, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Deformable_3D_Gaussians_for_High-Fidelity_Monocular_Dynamic_Scene_Reconstruction_CVPR_2024_paper.html.

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*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Learning 4D embodied world models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhen_Learning_4D_Embodied_World_Models_ICCV_2025_paper.html.

APPENDIX CONTENTS

A Implementation Details . . . 16
A.1 Coordinate system and robot geometry . . . 16
A.2 Stage 1 architecture and training . . . 16
A.3 Stage 2 architecture and training . . . 17
B Evaluation Protocol . . . 17
B.1 Evaluation cohort . . . 17
B.2 Cameras and temporal sampling . . . 18
B.3 Inference inputs and evaluation references . . . 19
B.4 Baseline adaptations and coverage . . . 19
B.5 Metric definitions . . . 20
C Additional Baseline Comparisons and Uncertainty . . . 21
C.1 Qualitative reconstruction from recorded observations . . . 21
C.2 Metrics omitted from the main table . . . 21
C.3 Confidence intervals . . . 21
C.4 Number of input frames . . . 21
C.5 Scale-adjusted depth diagnostic . . . 22
D Additional Design Analyses . . . 23
D.1 Intervention definitions and Stage 1 effects . . . 23
D.2 Learned motion and analytic FK transport . . . 24
D.3 Number of rendering-feedback refinement updates . . . 24
D.4 Reconstruction and rendering cost . . . 25
E Optional Latent-Token Interface . . . 25
E.1 Adapter and training . . . 25
E.2 Effect of replacing RGB tokens . . . 26
E.3 Spatial-resolution diagnostic . . . 26
F Scope and Limitations . . . 26

A Implementation Details

A.1 Coordinate system and robot geometry

Coordinate convention. Supplied camera calibrations and FK define camera and link poses in the robot base frame. For each clip, we set the reconstruction frame's origin and orientation to those of the wrist camera at the first source time $s_0 = \min \mathcal{S}$. This frame remains fixed throughout the clip, even as the wrist camera moves. Let $\mathbf{C}_{v,t}^{\text{base}}$ and $\mathbf{T}_{\ell}^{\text{base}}(t)$ denote camera-to-base and link-to-base transforms. The fixed base-to-reconstruction transform is $\mathbf{H} = (\mathbf{C}_{\text{wrist},s_0}^{\text{base}})^{-1}$, giving

$$\mathbf{C}_{v,t} = \mathbf{H}\mathbf{C}_{v,t}^{\text{base}}, \qquad \mathbf{T}_{\ell}(t) = \mathbf{H}\mathbf{T}_{\ell}^{\text{base}}(t). \tag{10}$$

The same frame is used for all source views, query times, Gaussian centers, and 3D displacements. Wrist-camera poses follow the supplied joint states and fixed hand–eye calibration. Joint configurations $\mathbf{q}_t$ and normalized gripper states $g_t \in [0, 1]$ are supplied at source and query times, not predicted by the lifter.

Internal units. The coordinate definitions above use metres. Network and rasterizer computations use a fixed conversion of 3.5 internal units per metre across clips. Geometric positions, displacements, Gaussian scales, and camera/link translations use this conversion consistently; rotation matrices are not rescaled. This changes numerical units, not the physical scale of a scene, and no clip-specific scale normalization is applied. Metric quantities can be expressed in the robot base frame using the inverse fixed transform after undoing the unit conversion.

Robot mesh source anchoring. We rasterize the Panda collision mesh and Robotiq visual mesh at each source state to obtain source-camera depth anchor maps and link identities. For Gaussian i, source depth D_i is selected from mesh depth at valid robot-support pixels and from the depth-head prediction elsewhere. This deterministic selection is used during training and inference, rather than learned as a fusion rule. Depth denotes source-camera optical-axis depth, not Euclidean distance to the camera center. We retain the resulting source position $\mathbf{p}_i$, mesh material-point correspondence, link identity ℓ_i, and fixed source binding $(v_i, s_i, \mathbf{u}_i)$. This *robot mesh source anchoring* constrains supported robot geometry; all remaining pixels use learned depth. Cached stereo surfaces are supervision or evaluation references, not reconstruction inputs.

A.2 Stage 1 architecture and training

Stage 1 is initialized from released MoVieS (Lin et al., 2026) weights, whose backbone builds on VGGT (Wang et al., 2025a). We freeze the original backbone parameters and train LoRA (Hu et al., 2022) adapters, the depth, motion, and Gaussian-attribute heads, and the FK-conditioning branch. LoRA uses rank 32 and $\alpha_{\text{LoRA}} = 16$ and is applied to the QKV and output projections in the frame-wise and global attention blocks. Tokens from source views and times are processed jointly before geometry and motion decoding.

The conditioning map $\boldsymbol{\kappa}_{v,s}(t)$ is aligned with the source image grid and contains robot support, source mesh depth, source-to-query 3D FK displacement $\mathbf{d}_i^{\text{FK}}(t)$, an availability flag, and a coarse gripper-neighborhood prior. The three displacement channels store vectors in the shared reconstruction frame and internal length units, not image-plane flow; the source mesh-depth channel remains in metres. The final convolution of $\mathcal{B}$ is zero-initialized, initially preserving the visual displacement prediction. Training includes FK-conditioning dropout. Mesh anchoring constrains source placement and Gaussian footprint, whereas query-time centers $\boldsymbol{\mu}_i^{(0)}(t) = \mathbf{p}_i + \mathbf{d}_i(t)$ remain learned predictions rather than analytic FK replacements.

The Gaussian-attribute head receives backbone features and a resized RGB shortcut; the depth head uses tokens only. Following MoVieS, the main text uses *motion head* collectively for query-time displacement and attribute prediction. The implementation separates these into `motion_head`, which predicts 3D center displacement, and `motion_splat_head`, which predicts query-time color, scale, rotation, and opacity from backbone features, base Gaussian attributes, and the query-time encoding. FK conditioning directly augments only the displacement prediction through $\mathcal{B}$. For both robot and non-robot Gaussians, query-time attributes replace static values under the released

MoVieS motion gates. Color, scale, and rotation use a center-motion threshold of 0.005 internal units (approximately 1.4 mm); opacity retains its separate gate. This permits time-dependent appearance.

Training uses AdamW with $\beta = (0.9, 0.95)$ and weight decay 0.05. Peak learning rates are 4×10^{-5} for backbone adapters and standard heads, and 4×10^{-4} for Gaussian-attribute and kinematic branches. A 150-update linear warmup precedes cosine decay; gradients are clipped at 1.0. We use seed 73 and the final checkpoint at update 3,462, after approximately ten data passes on four B200 GPUs. Each clip samples two to five source times and eight query requests. Batch size adapts to source-frame count within a budget of 80 source images per GPU. Neural modules use bfloat16 mixed precision; geometric operations use float32.

Supervision combines source and paired-right RGB rendering, metric depth, 3D tracks, robot-FK consistency, and opacity coverage. Paired-right images contribute training losses, not reconstruction tokens. Table 3 lists the separately weighted loss terms.

Table 3: Stage 1 loss weights. Keys identify separately weighted terms in the training configuration. Values are reported to at most six significant figures.

Loss key	Weight	Loss key	Weight
`movies_mse`	1.0	`movies_lpips`	0.5
`depth_head`	1.0	`track_head`	10.0
`stereo`	1.04051	`track_final`	0.105593
`fk_head`	0.464340	`fk_final`	0.157137
`metric_final`	0.00182939	`metric_right`	0.00484463
`coverage`	0.0589678	`coverage_right`	0.0648524

A.3 Stage 2 architecture and training

Stage 2 freezes Stage 1 and initializes the refiner from released ReSplat (Xu et al., 2026) weights. The observation and error encoders remain frozen. We train the recurrent updater and residual gates for 1,928 optimization updates over two exposure epochs, sampling one to four refinement updates during training and using four at inference. AdamW learning rates are 10^{-4} for the updater and 1.521×10^{-4} for the gates, with weight decay 0.01 and the released 1% warmup schedule.

At source time s, the full current field is rendered across source cameras. Feedback is assigned only to Gaussians with $s_i = s$ and contribution mass $m_i^{(r)}(s) > \epsilon$. Contextual errors are pooled using current alpha-compositing contributions and normalized by contribution mass; weights are recomputed after each update.

For each eligible Gaussian, the updater predicts a 59-channel, time-shared residual for position, scale, rotation, opacity, and spherical-harmonic coefficients. Shared corrections use the corresponding attribute parameterizations and apply to Stage 1's time-specific values. The accumulated position correction $\boldsymbol{\delta}_i^{(r)}$ is shared across query times and cameras and bounded by 5 cm. This preserves Gaussian center displacements exactly, but not rendered motion, which can change with compositing weights. No separate trajectory-correction branch is used.

The objective combines source and paired-right rendering with metric-depth and absolute FK-referenced robot-position losses, plus rendered-motion, track-preservation, and coverage terms. These losses constrain placement and rendering, while time sharing preserves Stage 1 center displacements. Only source RGB enters recurrent feedback; paired-right images serve as training loss targets, not feedback inputs. Inference uses fixed network weights without per-scene optimization.

B Evaluation Protocol

B.1 Evaluation cohort

Lab holdout and test membership. Our internal split reserves all 1,026 ILIAD and PennPAL episodes for evaluation. Neither lab contributes episodes to our training or validation of Stage 1, Stage 2, or the optional latent adapter. The prepared cache contains one 33-frame clip with stereo

references for each of 804 episodes. Episodes appearing in the preparation script's training or validation cohorts are also excluded.

We use the released DROID test assignments from PointWorld (Huang et al., 2026), rather than regenerate the split. The manifest `droid_paper_split_manifest.json` specifies the DROID domain, seed 42, and test fraction 0.1. Removing the final temporal-clip suffix from each manifest key gives its episode ID. Membership is therefore checked at the episode level, without requiring our evaluation interval to match the manifest's temporal clip.

Eligibility and sampling. Of the 804 cached episodes, 306 are absent from the manifest-derived test episode set, leaving 498 eligible episodes. We additionally require at least 24 cached frames, available camera-calibration metadata, and finite normalized gripper values in $[0, 1]$, allowing a boundary tolerance of 10^{-6}. These checks exclude no further episodes.

Using `numpy.random.default_rng(42)`, we permute eligible records sorted by cached clip name. We select the first 256 indices without replacement, then sort those indices only to fix export order. Sampling is uniform over eligible episodes, not stratified by lab or scene. The cohort contains 129 PennPAL and 127 ILIAD episodes across 69 scenes, with one fixed interval per episode shared by recorded and rollout evaluations. Table 4 summarizes the selection.

Table 4: Evaluation-cohort construction. Counts refer to episodes, not camera streams or temporal queries.

Selection stage	Episodes
ILIAD and PennPAL episodes reserved in the internal split	1,026
Episodes with a prepared clip and stereo references	804
Eligible episodes represented in the PointWorld test manifest	498
Fixed-seed evaluation sample	256

Selection record. The cohort is fixed before inspecting model outputs. Before constructing rollout inputs, the preparation script writes `selection.json`, recording the seed, candidate and eligible counts, exclusion reasons, and selected episode IDs. Subsequent method-specific execution failures affect evaluation coverage, not the initial 256-episode cohort.

B.2 Cameras and temporal sampling

Each episode provides a 33-frame clip at 15 Hz. Stored wrist and external images have resolutions 352×640 and 176×320, respectively; all six cameras are evaluated at 154×280. The wrist and two external left-eye streams provide reconstruction observations. Their paired right-eye cameras are the held-out novel views.

Both settings use cached frames 7–23, a 17-frame interval matching one Cosmos 3–Nano (NVIDIA, 2026) rollout. Frame 7 supplies the initial three-view observation, followed by 16 frames conditioned on a 16-step FK-derived end-effector action chunk. Reconstruction retains the supplied joint and gripper trajectories without inverse kinematics. Our five designated source times $\mathcal{S}$ have relative indices $\{0, 4, 8, 12, 16\}$, corresponding to cached frames $\{7, 11, 15, 19, 23\}$. RoGSW4RLD and MoVieS use these times; the main 4DGT and SoM baselines use all 17 frames per camera. Appendix C.4 also evaluates five-frame 4DGT.

A fixed per-clip random plan samples up to eight query requests within this interval. With probability 0.25, it includes all five times in $\mathcal{S}$ and samples the remaining requests from the interval; otherwise, all requests are sampled directly. Duplicate timestamps are evaluated once. Recorded and rollout settings share query times, evaluation cameras, robot trajectories, and recorded RGB/depth/track references. Only the reconstruction observations change. An input time denotes temporal inclusion in a method's input sequence; a source view denotes camera identity, irrespective of whether the queried time is an input time.

Table 5: Signals used in the main evaluation. "Scale only" denotes one scalar per camera computed at the five designated source times, not dense stereo supervision during test-time reconstruction. Right-eye maps and reference tracks are not supplied directly to reconstruction.

Signal	Ours	MoVieS	4DGT	SoM
Left-eye visual input	5 times $\times$ 3 views	5 / camera	17 / camera	17 / camera
Calibrated cameras	Yes	Yes	Yes	Yes
Joint/gripper states[‡]	Anchoring/FK	No direct input	No direct input	FK masks only
Robot mesh	Geometry prior	No	No	Foreground mask
Source-time stereo depth	No	Scale only[†]	No	Scale only[†]
Right-eye RGB/depth maps	Evaluation only	Evaluation only	Evaluation only	Evaluation only
3D track references	Evaluation only	Evaluation only	Evaluation only	Evaluation only
Post-hoc target-depth fitting	No	No	No	No

[†] Scalars use recorded left-eye stereo depth before inference or optimization, including for rollout inputs. They carry indirect information from the recorded stereo pairs, but no right-eye image or depth map is supplied directly.
[‡] All methods receive calibrated camera poses, including the wrist trajectory. SoM uses robot states to construct foreground masks, not as explicit motion-conditioning inputs.

B.3 Inference inputs and evaluation references

Recorded reconstruction uses real observations; rollout reconstruction uses generated observations with the same calibration, trajectories, queries, and recorded targets. Rollout scores therefore combine generation and reconstruction errors rather than measuring reproduction of generated inputs alone. The supplied robot trajectory is unchanged, so similar robot EPE across settings does not establish accurate visual motion generation. For our model, paired-right observations provide training supervision on training episodes, but never reconstruction tokens or recurrent feedback on evaluation clips. These are system-level comparisons: frame counts, auxiliary inputs, and training differ. Appendix E describes the optional latent-token interface.

B.4 Baseline adaptations and coverage

MoVieS, 4DGT, and Shape of Motion follow their released monocular settings: each independently reconstructs the wrist and two external-camera streams. We express their three fields in the common metric robot base frame, concatenate their Gaussians, and render all six evaluation cameras. This *camera-wise reconstruction followed by calibrated merging* differs from our joint field formation. Metric-scale conventions are method-specific; no main-comparison field is fitted to target-view depth after reconstruction.

MoVieS. Released MoVieS (Lin et al., 2026) uses the five designated source times per camera. To stabilize scale for fixed external cameras and short wrist trajectories, we compute one normalization scalar per camera from recorded left-eye stereo depth at these times, using the mean distance of valid back-projected source points. Camera translations are normalized before inference, and the same factor restores the predicted field to metres afterwards. The three fields are rigidly transformed into the robot base frame and concatenated for source- and novel-view rendering. Scale estimation uses neither depth outside the five designated times nor right-eye depth maps; a query coinciding with an input time does not introduce additional depth access.

4DGT. We use the public first-stage 4DGT (Xu et al., 2025) checkpoint with all 17 frames per camera. Inference images are resized to 182×322 with corresponding intrinsics rescaling. Following the released pipeline, camera poses are rigidly recentered around the mean input camera pose. This changes the coordinate frame without rescaling the supplied metric translations. After inference, we undo recentering and concatenate the three fields in the robot base frame for rendering. Expected camera-z depth is scored in metres, without stereo-depth input or output-scale fitting to evaluation targets. Appendix C.4 uses the same protocol with five input frames.

Shape of Motion. Released Shape of Motion (Wang et al., 2025b) uses all 17 frames independently per camera. Preprocessing uses UniDepth-v1 metric depth, Depth-Anything relative depth,

and BootsTAPIR tracks. To correct camera-specific depth-scale biases before optimization, we compute

$$\gamma_v = \mathrm{median}_{s\in\mathcal{S},\,\mathbf{u}\in\mathcal{P}^{\mathrm{st}}_{v,s}} \frac{D^{\mathrm{stereo}}_{v,s}(\mathbf{u})}{D^{\mathrm{UniDepth}}_{v,s}(\mathbf{u})}, \tag{11}$$

where $\mathcal{S}$ contains the five designated source times and $\mathcal{P}^{\mathrm{st}}_{v,s}$ is the valid cached stereo-pixel support. UniDepth estimates are multiplied by γ_v before subsequent preprocessing and optimization; recorded stereo depth is not used thereafter. Each camera uses a single γ_v across all 17 frames, rather than separate scales for individual times or evaluation views.

Calibrated cameras are used directly, scene renormalization is disabled, and projected FK robot masks replace interactive foreground selection. All runs share the resolution-adjusted settings of 3-pixel erosion and 6-pixel dilation. Each camera-specific scene follows the released 500-epoch schedule, giving 1,500 updates for 17 frames with batch size eight. Learned camera corrections are not used for evaluation. The three dynamic fields are combined in the calibrated robot base frame and rendered from all six evaluation cameras.

Dynamic displacement output. Persistent Gaussian identities define anchor-to-query displacement for RoGSW4RLD and MoVieS. The evaluation wrapper obtains 4DGT positions from its released per-Gaussian position polynomial and SoM positions from its optimized motion bases. Each method's displacements are rendered through the anchor cameras and passed to the same track-EPE scorer.

Coverage and exclusions. RoGSW4RLD, MoVieS, and 4DGT cover all 256 selected clips. The MoVieS world-coordinate render-parity check is an audit, not a cohort-selection criterion. Near-threshold cases remain included after camera-point consistency checks; their effect on reported means is below the displayed precision.

SoM is evaluated on 246 clips per setting. A clip is excluded only when at least one of its three camera optimizations fails under the common foreground-mask settings, preventing merged-field construction. Its reported means therefore condition on successful runs rather than the full cohort.

Undefined 4DGT depth pixels are assigned zero and remain penalized inside the fixed reference mask. Robot EPE uses each method row's evaluated clips, with the required track sidecars available; empty track sets are handled as specified below.

B.5 Metric definitions

Image metrics. PSNR uses full-image RGB MSE in $[0, 1]$. SSIM (Wang et al., 2004) uses an 11×11 Gaussian window with $\sigma = 1.5$; LPIPS (Zhang et al., 2018) uses VGG with RGB normalized to $[-1, 1]$. Source- and novel-view scores use their respective three-camera groups. PSNR is computed per image before averaging, not from pooled multi-camera MSE. Scoring covers the full target image, including background and regions occluded in reconstruction inputs.

Metric depth. For each camera and query time, let $\hat{z}_{\mathbf{u}}$ denote rendered expected camera-z depth and $z_{\mathbf{u}}$ the cached FoundationStereo (Wen et al., 2025) reference, both in metres. On fixed valid pixel support $\mathcal{P}$, we compute

$$\mathrm{AbsRel} = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{u}\in\mathcal{P}} \frac{|\hat{z}_{\mathbf{u}} - z_{\mathbf{u}}|}{z_{\mathbf{u}}}, \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathcal{P}|}\sum_{\mathbf{u}\in\mathcal{P}} (\hat{z}_{\mathbf{u}} - z_{\mathbf{u}})^2}, \tag{13}$$

$$\delta_1 = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{u}\in\mathcal{P}} \mathbf{1}\Big[\frac{z_{\mathbf{u}}}{1.25} < \hat{z}_{\mathbf{u}} < 1.25 z_{\mathbf{u}}\Big]. \tag{14}$$

The interval form of δ_1 implements the usual ratio threshold for positive reference depths while rejecting non-positive predictions.

Source-view support requires finite reference depth with $0.1 < z < 5$ m. Right-eye support follows the cached validity mask and its configured metric range. Neither depends on predicted opacity, depth, or error. Non-positive predictions remain in AbsRel and RMSE. Predictions are neither clipped to the reference range nor rescaled to target depth in the main evaluation. These metrics measure agreement with stereo estimates, not sensor-measured ground truth.

Robot displacement and placement. For evaluation notation, let $\boldsymbol{\mu}_i^{\mathrm{base}}(t)$ denote Gaussian centers expressed in metres in the robot base frame, related to the reconstruction frame by the inverse fixed transform and unit conversion in Appendix A.1. Anchor-to-query displacement is $\boldsymbol{\mu}_i^{\mathrm{base}}(t) - \boldsymbol{\mu}_i^{\mathrm{base}}(a)$. Rendering these displacements into anchor camera v and normalizing by accumulated opacity gives $\hat{\mathbf{d}}_{v,a\to t}$. For valid robot tracks $\mathcal{T}_{\mathrm{robot}}^{a,t}$, with anchor camera v_j and pixel $\mathbf{u}_j^a$, robot EPE is

$$e_{\mathrm{track}}(a,t) = \frac{1000}{|\mathcal{T}_{\mathrm{robot}}^{a,t}|} \sum_{j \in \mathcal{T}_{\mathrm{robot}}^{a,t}} \left\| \hat{\mathbf{d}}_{v_j,a\to t}(\mathbf{u}_j^a) - (\mathbf{X}_j^t - \mathbf{X}_j^a) \right\|_2 \quad \text{(mm)}. \tag{15}$$

Here $\mathbf{X}_j^a$ and $\mathbf{X}_j^t$ are cached track positions in the same metric frame. Robot membership uses fixed FK-mesh support; anchor-time comparisons and empty track sets are omitted. EPE measures displacement, not absolute endpoint location or agreement with analytic FK. Stage 2 preserves individual Gaussian displacements, but rendered EPE can change through compositing weights.

Robot depth error is $1000\ \mathrm{mean}\,|\hat{z} - z|$ (mm) on the reference robot core, pooling valid source-camera pixels. The core requires a common link throughout a 3×3 neighborhood, valid stereo depth, and stereo–mesh depth consistency under the fixed threshold. These stereo checks define evaluation support only. The metric measures supported robot-surface placement, not joint-center or end-effector localization. Non-robot mover EPE is omitted because manual auditing identified false static-object drift in cached labels.

Aggregation and uncertainty. Image metrics are averaged over cameras and distinct query times within each view group, then equally over evaluated clips. Each clip corresponds to one episode, so episodes rather than scenes receive equal weight. Robot metrics pool valid support at each query time before averaging over times and clips. We report 95% confidence intervals from 10,000 clip-level bootstrap resamples per method. They describe test-clip sampling variability, not variability across training seeds. Table 7 reports the central comparison.

C Additional Baseline Comparisons and Uncertainty

C.1 Qualitative reconstruction from recorded observations

Figure 7 compares reconstructions from recorded inputs at held-out right-eye cameras. These examples show fewer ghosting artifacts and clearer object boundaries with RoGSW4RLD, complementing the generated-rollout source-view comparison in Figure 6.

C.2 Metrics omitted from the main table

Table 6 complements Table 1 using identical predictions, evaluated clips, reference supports, and metric conventions. Most depth gains are already present in Stage 1; Stage 2 further improves source-view SSIM, with smaller changes in depth RMSE and δ_1.

C.3 Confidence intervals

Table 7 reports uncertainty across evaluated clips, not training seeds. SoM intervals condition on successful runs. These are intervals for individual method means, not paired differences; they do not directly quantify uncertainty in the incremental effect of Stage 2.

C.4 Number of input frames

We compare 4DGT using all 17 frames per camera with the same five source times supplied to our model. Both variants retain camera-wise reconstruction followed by calibrated merging, using

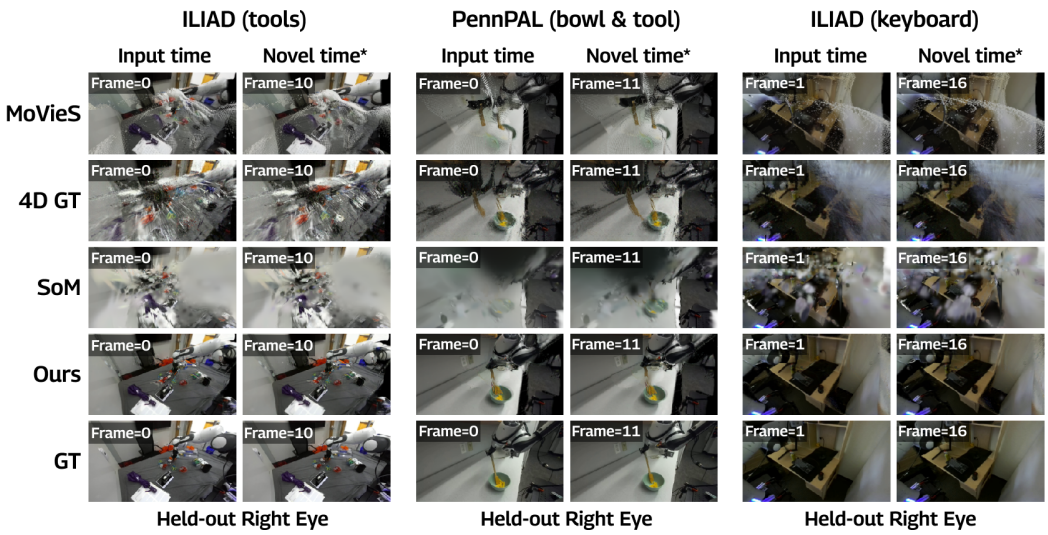


Figure 7: **Novel-view reconstruction from recorded observations.** RoGSW4RLD shows less ghosting and clearer scene details at held-out right-eye cameras. The bottom row shows recorded references; asterisks mark non-input times for RoGSW4RLD and MoVieS only.

Table 6: Complementary reconstruction metrics. Means use 256 clips for MoVieS, 4DGT, and our model, and 246 successful clips for SoM. †: one source-stereo scale scalar per camera before reconstruction.

Method	Source SSIM ↑	Source RMSE (m) ↓	Novel δ_1 (%) ↑
Recorded observations			
MoVieS†	0.713	0.333	55.1
4DGT (17f)	0.585	0.414	31.1
SoM†	0.573	0.473	38.1
Ours (Stage 1)	0.817	0.257	77.2
Ours (Stage 1+2)	0.866	0.255	77.5
Cosmos 3 rollouts			
MoVieS†	0.639	0.354	52.0
4DGT (17f)	0.533	0.427	30.6
SoM†	0.531	0.473	37.8
Ours (Stage 1)	0.736	0.286	73.0
Ours (Stage 1+2)	0.775	0.284	73.2

the same checkpoint, input resolution, query plan, evaluation cameras, and recorded references. This measures input-density sensitivity within 4DGT rather than isolating architectural differences between methods.

Seventeen frames improve every reported metric in both settings, but do not close the gap to our model. The main comparison therefore retains this stronger 4DGT configuration rather than restricting its temporal input to match ours.

C.5 Scale-adjusted depth diagnostic

The main evaluation scores metric reconstructions without target-view scale fitting. Here, we rescore 4DGT and SoM after multiplying each predicted depth image by its median reference-to-prediction depth ratio. A separate factor is fitted for each camera and query time, rather than one global factor for the field. This target-assisted operation changes only scored depth images; it neither rescales nor rerenders the Gaussian field. Unadjusted scores appear in Table 1.

Table 7: Main-comparison means and 95% clip-bootstrap confidence intervals from 10,000 resamples. MoVieS, 4DGT, and our model use 256 clips; SoM uses 246. †: one source-stereo scale scalar per camera.

Method	Novel PSNR (dB) ↑	Novel AbsRel ↓	Robot EPE (mm) ↓
Recorded observations			
MoVieS†	13.93 [13.77, 14.09]	0.373 [0.356, 0.390]	22.2 [20.5, 24.2]
4DGT (17f)	12.90 [12.75, 13.05]	0.681 [0.659, 0.704]	31.0 [28.8, 33.2]
SoM†	12.14 [12.00, 12.28]	0.429 [0.413, 0.445]	32.2 [30.1, 34.4]
Ours (Stage 1)	15.52 [15.35, 15.70]	0.200 [0.193, 0.208]	8.6 [7.8, 9.6]
Ours (Stage 1+2)	16.08 [15.90, 16.27]	0.197 [0.190, 0.205]	8.6 [7.7, 9.6]
Cosmos 3 rollouts			
MoVieS†	13.69 [13.53, 13.85]	0.404 [0.385, 0.424]	22.6 [20.9, 24.5]
4DGT (17f)	12.78 [12.64, 12.93]	0.697 [0.674, 0.720]	32.3 [30.0, 34.7]
SoM†	12.05 [11.91, 12.19]	0.440 [0.423, 0.458]	37.6 [35.0, 40.4]
Ours (Stage 1)	15.04 [14.87, 15.22]	0.218 [0.209, 0.227]	8.7 [7.8, 9.7]
Ours (Stage 1+2)	15.52 [15.34, 15.71]	0.215 [0.206, 0.224]	8.7 [7.8, 9.7]

Table 8: 4DGT with five or 17 input frames per camera under the same camera-wise reconstruction and calibrated-merging protocol on 256 clips.

		Source	Novel view		
Setting	Frames	PSNR ↑	PSNR ↑	LPIPS ↓	AbsRel ↓
Recorded	5	15.63	12.78	0.603	0.803
Recorded	17	16.54	12.90	0.584	0.681
Cosmos 3	5	15.20	12.69	0.608	0.815
Cosmos 3	17	15.78	12.78	0.591	0.697

Table 9: Target-assisted median-scale depth diagnostic (AbsRel ↓). †: source-stereo scaling before reconstruction. ‡: per-image target-assisted scaling used only here, not in the main comparison.

	Recorded		Cosmos 3 rollouts	
Method	Source	Novel	Source	Novel
4DGT‡	0.320	0.411	0.338	0.417
SoM†‡	0.402	0.489	0.411	0.498

For 4DGT, novel-view AbsRel decreases from 0.681 to 0.411 on recorded inputs and from 0.697 to 0.417 on rollouts. SoM instead worsens from 0.429 to 0.489 and from 0.440 to 0.498. Thus, this correction substantially reduces 4DGT's error but leaves a sizable residual; it does not improve SoM. The diagnostic does not establish a single field-level scale correction.

D Additional Design Analyses

D.1 Intervention definitions and Stage 1 effects

Table 2 examines joint lifting and refinement in panel (a), and robot geometry and motion in panel (b). We define these frozen-checkpoint interventions and test whether the robot-specific effects precede Stage 2.

The cross-camera intervention blocks attention to other cameras in Stage 1 while retaining within-camera temporal interaction, calibration, and robot inputs. Camera-specific fields are then combined by calibrated merging. This probes the jointly trained checkpoint's dependence on inference-time cross-camera exchange, not the performance of separately trained monocular models.

Disabling FK conditioning suppresses the kinematic input to learned deformation while retaining mesh anchoring. Disabling mesh anchoring restores learned source geometry for robot-supported pixels and removes the associated mesh anchoring pathway while retaining FK conditioning. This is a pathway-level intervention, not an isolated depth replacement. Calibration, including FK-derived

wrist poses, remains unchanged. Conditioning dropout supports FK removal; mesh anchoring removal and removal of both are out-of-distribution diagnostics. No variant is retrained.

Table 10: Robot-specific interventions in Stage 1 on 256 recorded clips. Table 2 reports the corresponding Stage 1+2 interventions for recorded observations and Cosmos 3 rollouts.

Configuration	Robot EPE (mm) ↓	Robot depth error (mm) ↓
Full Stage 1	8.64	13.4
w/o FK conditioning	30.14	21.5
w/o mesh anchoring	8.80	36.5
w/o both	30.11	41.6

The complementary pattern already appears before refinement: FK removal raises displacement EPE from 8.64 to 30.14 mm, whereas mesh anchoring removal raises robot depth error from 13.4 to 36.5 mm with EPE at 8.80 mm. Stage 2 therefore does not create this separation between placement and displacement.

D.2 Learned motion and analytic FK transport

We replace learned robot-motion prediction with analytic FK transport of mesh-anchored source points and their covariances along the supplied link trajectories. Inputs and learned weights remain unchanged. This geometry-transport diagnostic evaluates displacement and placement against the recorded references, not prediction of unknown robot actions. Because both centers and covariances change, it is not an isolated center-motion ablation.

Table 11: Score changes from learned motion to analytic FK transport. Entries are analytic-FK minus learned-motion scores; positive PSNR and negative errors favor analytic transport.

Input	Stage	Δ Novel PSNR (dB)	Δ AbsRel	Δ Robot EPE (mm)	Δ Robot depth error (mm)
Recorded	Stage 1	0.00	-0.004	-0.16	+1.7
Recorded	Stage 1+2	+0.01	-0.004	-0.18	+0.9
Cosmos 3	Stage 1+2	+0.01	-0.004	-0.20	+0.9

Analytic transport lowers robot EPE by 0.16–0.20 mm but increases robot depth error by 0.9–1.7 mm; novel-view PSNR changes by at most 0.01 dB. These results do not support superior robot-displacement accuracy from learned deformation. Its role is instead to maintain one deformation parameterization for robot and non-robot dynamics. The small score differences describe this diagnostic, not a demonstrated statistical advantage.

D.3 Number of rendering-feedback refinement updates

Table 12 varies the number of updates r using the same Stage 2 checkpoint and initial Stage 1 fields. The fixed 32-clip subset evaluates the quality–computation trade-off; its refinement-only timings are distinct from the complete-pipeline measurements in Table 1.

Table 12: Refinement sweep on the first 32 evaluation clips under the fixed-window protocol. Quality metrics use all 32 clips. Refinement time uses CUDA-synchronized measurements on 27 clips after excluding each process's warm-up clip; lifting and final query rendering are excluded.

Updates r	Source PSNR ↑	Novel PSNR ↑	Source LPIPS ↓	Source AbsRel ↓	Robot EPE (mm) ↓	Time (s) ↓
0	22.59	16.02	0.182	0.178	6.21	–
1	24.94	16.39	0.169	0.177	6.22	2.61
2	25.59	16.44	0.168	0.177	6.22	4.78
3	25.80	16.47	0.171	0.176	6.22	6.95
4	25.87	16.48	0.173	0.176	6.22	9.12

Most PSNR gains occur within two updates, with diminishing gains thereafter. Source LPIPS is lowest at two updates and then worsens slightly, so additional updates do not uniformly improve image quality. Source AbsRel and robot EPE change little on this subset.

The time-shared correction $\boldsymbol{\mu}_i^{(r)}(t) = \boldsymbol{\mu}_i^{(0)}(t) + \boldsymbol{\delta}_i^{(r)}$ translates each Gaussian's entire Stage 1 center trajectory, preserving its displacement between query times exactly. This guarantee does not extend to rendered depth or EPE: changes in placement, covariance, opacity, and visibility alter compositing weights. Stage 2 therefore refines rendering without independently re-estimating center trajectories.

D.4 Reconstruction and rendering cost

Table 1 reports reconstruction and query-rendering time and peak allocated GPU memory. Below we specify their method-dependent measurement scopes. Model loading and world-model generation are excluded.

Our model. Stage 1 requires 0.436 s for joint lifting and 0.153 s for query rendering, totaling 0.59 s after rounding. The full model uses 0.436 s for lifting, 8.417 s for four refinement updates, and 0.228 s for final rendering, totaling 9.08 s. Both stages use fixed inference weights without per-scene optimization.

MoVieS. Timing includes three camera-wise inferences, transformation into the common metric robot frame, Gaussian merging, and rendering of the common query set. Diagnostic per-view renders are excluded.

4DGT. The 17-frame workload comprises 0.09 s of preprocessing, 1.93 s of camera-wise inference, 2.59 s of query rendering, and 0.19 s of displacement rendering, totaling 4.80 s. The five-frame variant is timed separately.

Shape of Motion. Timing includes preprocessing and optimization for all three camera streams plus merged-field rendering. It sums the camera-run workloads rather than reporting rendering latency from an already optimized scene.

Measurement conditions. RoGSW4RLD, MoVieS, and 4DGT use otherwise idle B200 workloads with explicit CUDA synchronization. SoM is measured under concurrent workloads, so its aggregate run time is not an isolated single-GPU end-to-end latency.

Peak allocation is measured using `torch.cuda.max_memory_allocated` after resetting the peak statistic for each clip. Reported memory includes model weights but excludes the CUDA context and unused allocator cache.

RoGSW4RLD and MoVieS share a process holding multiple models. Their method-specific estimates add the invoked modules' parameter and buffer memory to the peak allocation increase above the pre-invocation baseline, averaged over 16 clips after warm-up. These are not unadjusted shared-process peaks. 4DGT and SoM use method-specific processes and aggregate per-clip peaks over evaluated clips. SoM uses the largest peak among its three camera optimizations and excludes merged-field rendering memory. These differing scopes should be retained when comparing memory requirements. Network-training memory is not reported.

E Optional Latent-Token Interface

E.1 Adapter and training

Following the latent embedding design of Diff4Splat (Pan et al., 2026), the adapter maps each camera region of the Cosmos 3 (NVIDIA, 2026) latent to the Stage 1 view-time token grid. Latents have 48 channels, spatial stride eight, and temporal stride four. Cross-camera interaction remains in the joint Stage 1 backbone.

Adapter training first aligns outputs with DINOv2 (Oquab et al., 2024) visual tokens, then distills intermediate Stage 1 features and log-depth predictions. Both reconstruction stages remain frozen.

Starting from the token-aligned checkpoint, the second phase uses 300 AdamW updates with learning rate 5×10^{-5} and weight decay 0.01. Feature and log-depth losses each have weight 1.0; the input-token loss has weight 0.1. Decoded RGB remains in the Gaussian-attribute shortcut and the refinement observation and error encoders, so this interface is not RGB-free.

E.2 Effect of replacing RGB tokens

Both interfaces share downstream reconstruction weights, calibrated cameras, robot inputs, and the evaluation protocol. Table 13 reports the latent-token path; RGB references appear in Table 1, with robot depth error in Table 2(b).

Table 13: Optional latent-token interface with Stage 1+2 and four refinement updates on 256 clips. Robot EPE measures track displacement; robot depth error measures source-view placement on the reference robot core. Cosmos 3 rollouts supply generated latents directly.

Input	Novel PSNR (dB) ↑	Novel AbsRel ↓	Robot EPE (mm) ↓	Robot depth error (mm) ↓
Recorded	15.32	0.293	9.0	20.7
Cosmos 3	14.98	0.298	9.0	20.7

Relative to RGB, the latent interface lowers novel-view PSNR by 0.76 dB and raises AbsRel by 0.096 on recorded inputs; the rollout gaps are 0.54 dB and 0.083. Robot EPE increases by approximately 0.4 and 0.3 mm at the reported precision, while robot depth error increases by 7.0 and 5.2 mm, respectively. These relatively small EPE changes occur with mesh anchoring and FK conditioning retained; they do not establish motion preservation by latent tokens alone.

E.3 Spatial-resolution diagnostic

We compare the latent interface with RGB downsampled by a factor of eight per dimension and restored before reconstruction. This probes sensitivity to spatial detail, not equivalence between latent compression and image downsampling.

Table 14: Stage 1-only spatial-resolution diagnostic on the first 16 evaluation clips under the fixed-window protocol. Metrics use source views; "Far" restricts AbsRel to reference camera-z depths between 1.5 and 5 m. Every input frame and camera is area-downsampled and bilinearly restored in the $8\times$ RGB variant.

Input	PSNR (dB) ↑	AbsRel ↓	δ_1 (%) ↑	Far AbsRel ↓
Original RGB	23.35	0.160	80.3	0.191
$8\times$ down/up RGB	20.76	0.227	63.3	0.482
Latent adapter	22.37	0.228	62.5	0.411

Both alternatives increase overall depth AbsRel from 0.160 to 0.227–0.228. However, latent tokens yield higher PSNR and lower far-depth error than degraded RGB. This is consistent with sensitivity to spatial detail, but does not isolate spatial resolution from representation or adapter effects as the cause of the latent gap.

F Scope and Limitations

The lab holdout evaluates transfer to ILIAD and PennPAL, whose episodes are excluded from our training and validation of Stage 1, Stage 2, and the optional adapter. Evaluation uses calibrated wrist and external cameras, known robot geometry, supplied trajectories, and 17-frame intervals. Novel-view evaluation uses paired right-eye cameras; wider baselines, new robot morphologies, and longer rollouts remain untested. Errors in stereo and track references affect measured accuracy, and robot EPE does not establish non-robot motion accuracy.

Baseline comparisons are system-level because training, input frame counts, and auxiliary inputs differ. MoVieS and SoM receive one scale scalar per camera estimated from recorded source stereo

depth; RoGSW4RLD and 4DGT receive no test-time stereo depth. SoM uses projected FK foreground masks and completes 246 of 256 clips, so its reported means are conditional on successful runs.

Cosmos 3 scores compare reconstructions of generated observations against recorded futures, combining generation and reconstruction errors. Supplied robot trajectories remain unchanged between recorded and rollout settings, so stable robot EPE does not demonstrate accurate visual motion generation.

Reconstruction is also sensitive to the spatial detail available in its visual inputs. The 16-clip Stage 1 spatial-resolution diagnostic in Appendix E.3 shows that downsampling RGB inputs and resizing them back to the original resolution degrades rendering and depth accuracy. This supports sensitivity to input detail, but does not isolate rollout resolution as the cause of the performance gap between recorded and generated inputs. Because RoGSW4RLD reconstructs pretrained world-model outputs rather than learning a new transition model, it may benefit from future improvements in the resolution and consistency of action-conditioned rollouts.

Stage 2 preserves each Gaussian's center displacement between query times and therefore cannot correct displacement errors inherited from Stage 1. These experiments evaluate reconstruction, not physical reasoning, task success, or closed-loop robot control.