VIDLIFT4D: LIFTING CAMERA-CONDITIONED VIDEOS INTO DYNAMIC GAUSSIAN SCENES

Anonymous authors
Paper under double-blind review

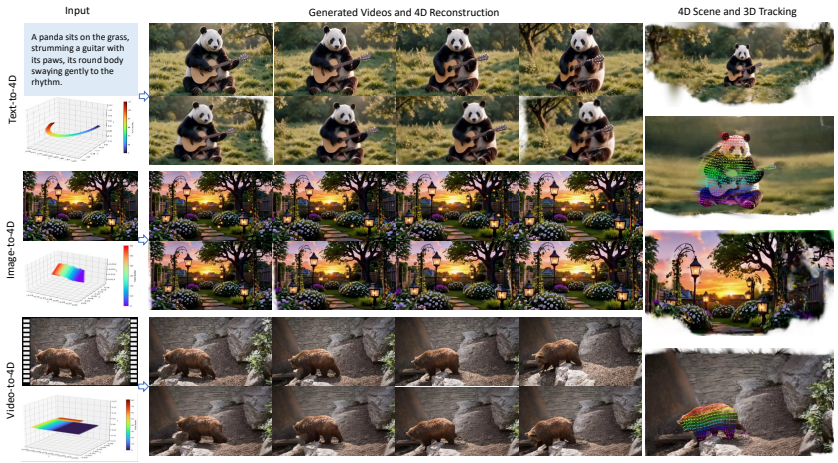


Figure 1: VidLift4D generates videos from text, image, or video conditions and a requested camera trajectory, then optimizes a 4D Gaussian scene. The example text prompt describes a panda strumming a guitar. The displayed renderings illustrate the generated appearance and novel-view capability; adherence to the requested trajectory is not established by this figure.

ABSTRACT

Generating a persistent 4D scene from text, an image, or a video requires both plausible observations and a representation that can be rendered from new views. We present VidLift4D, a video-first pipeline that generates a clip conditioned on the input and a requested camera trajectory, then fits dynamic 3D Gaussians to that clip. During video-model fine-tuning, the published Geometry Forcing objective aligns diffusion features with VGGT features. At inference, a separate VGGT pass estimates cameras, depth, and point maps for reconstruction; Gaussian motion combines instance-shared rigid transforms with per-Gaussian residuals. We evaluate monocular reconstruction, synthetic ground-truth depth and object motion, and text-, image-, and video-conditioned generation. In the reported custom Kubric-derived protocol, depth AbsRel and object-trajectory error are 0.082 and 0.041 m, compared with SoM's 0.103 and 0.058 m. On identical raw iPhone videos, PSNR is 15.91 dB versus 15.89 dB for SoM; diffusion-assisted input raises it to 16.23 dB while changing the observations. Missing per-scene predictions, strong independent-frontend controls, and requested-versus-achieved camera paths currently limit attribution of these reported differences to the shared geometry model.

1 INTRODUCTION

Building models that can generate dynamic three-dimensional worlds is a key step toward general spatial intelligence Kong et al. (2025); Cao et al. (2025). With the rapid emergence of diffusion-based video models Ho et al. (2022); He et al. (2022); Brooks et al. (2024); Yang et al. (2025b), world models that synthesize temporally evolving 3D scenes (i.e., 4D scenes) have become increasingly relevant for embodied intelligence and immersive applications.

Existing approaches have made significant progress on monocular 4D reconstruction Lei et al. (2024); Wang et al. (2025b); Wu et al. (2024) and on 4D generation from text or image Bahmani

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.

et al. (2024a;b); Jiang et al. (2023); Yu et al. (2024); Wu et al. (2025b); Sun et al. (2025); Liu et al. (2025). However, two practical gaps remain. First, video diffusion models excel at appearance and motion but largely ignore that videos are 2D projections of a dynamic 3D world; their outputs often exhibit geometric distortions Wu et al. (2025a), which limit downstream 4D reconstruction. Second, monocular reconstruction methods consume a cascade of independent estimators (SLAM, monocular depth, 2D tracker), which can disagree on each other's coordinate systems, scales, and visibility, especially on diffusion-generated videos.

We present VidLift4D, a video-first pipeline that studies these gaps in a modular setting (Figure 1). It combines (i) camera-conditioned video diffusion taking text, image, or video input and a requested trajectory with (ii) per-scene optimization of a persistent 4D Gaussian representation. A geometric foundation model connects the stages through training-time feature supervision and a separate post-generation geometry estimate. These are two uses of the same model family, rather than literal reuse of per-scene features or ground-truth geometry.

We reuse camera-conditioned video diffusion He et al. (2024); Bahmani et al. (2025), Geometry Forcing (GF) Wu et al. (2025a), VGGT Wang et al. (2025a), dynamic Gaussian motion bases Wang et al. (2025b); Kratimenos et al. (2023), and instance masks Kirillov et al. (2023); Yang et al. (2023). Closely related video-to-4D pipelines and hierarchical motion models already exist Wang et al. (2024); Chu et al. (2024); Pan et al. (2026); Liang et al. (2025); Lee et al. (2026); our technical subject is the specific interface and its measured system behavior. The present component grid does not yet isolate teacher matching from VGGT's standalone estimation quality.

- **A geometry-teacher interface.** VGGT supervises diffusion features during training through GF and, in a separate pass on each output video, supplies camera, depth, and point-map estimates to the reconstructor. This replaces the pose/depth estimator cascade; instance masks remain external. A 2×2 experiment measures a component interaction but does not establish a teacher-matching benefit over an equally strong independent frontend.
- **Structured motion in an integrated pipeline.** Group-shared $\mathbb{SE}(3)$ motion and per-Gaussian residuals are evaluated under fixed geometry inputs. The hierarchy is related to existing grouped and multiscale motion models, rather than a new motion primitive; its parameter efficiency remains to be established with an auditable count.
- **Evaluation across complementary protocols.** iPhone tests held-out rendering with and without a video-to-video prior; Kubric Movi-F Greff et al. (2022) supplies ground-truth depth and trajectories; text/image generation and DAVIS Perazzi et al. (2016) test visual quality and internal consistency. These protocols support different claims and are interpreted separately.

The video generator and per-scene reconstructor are optimized separately. Their coupling is therefore an architectural choice whose value must be tested against stronger independent geometry estimators, not assumed from sharing a teacher.

2 Related Work

Camera-controlled video diffusion. Building on the standard video diffusion paradigm Ho et al. (2022); He et al. (2022); Kong et al. (2024); Brooks et al. (2024); Yang et al. (2025b), a series of recent methods condition video generation on user-specified camera trajectories GUO et al. (2024); He et al. (2024); Kuang et al. (2024); Zhang et al. (2024); Zhou et al. (2025); Bahmani et al. (2025). These models inject Plücker camera embeddings or similar parameterizations as auxiliary conditioning signals, but operate fundamentally in pixel space; their outputs frequently lack multi-view geometric consistency Wu et al. (2025a), which complicates downstream 4D reconstruction. VidLift4D builds on this line of work and uses Geometry Forcing Wu et al. (2025a) as a training-time auxiliary loss to address the geometric-consistency gap.

Dynamic 3D reconstruction. Dynamic-scene reconstruction ranges from synchronized multi-view methods Cao & Johnson (2023); Kratimenos et al. (2023); Luiten et al. (2024); Park et al. (2021); Wu et al. (2024) to monocular methods Lei et al. (2024); Li et al. (2021); Wang et al. (2025b). Monocular pipelines often combine external masks, depth, tracks, and camera poses Kirillov et al. (2023); Yang et al. (2024a); Doersch et al. (2023); Teed & Deng (2021). VidLift4D instead runs VGGT Wang et al. (2025a) on each generated video for pose, depth, and point maps, while retaining external instance masks. ViPE Huang et al. (2025) is an independent modern geometry alternative whose comparison

is needed to isolate teacher matching. USplat4D Guo et al. (2026) models uncertainty in monocular dynamic Gaussian reconstruction and is a relevant iPhone baseline.

Structured Gaussian motion. SoM Wang et al. (2025b) already uses low-dimensional motion bases. HiMoR Liang et al. (2025) organizes coarse-to-fine motion in a tree, MoGaF Lee et al. (2026) optimizes motion-aware Gaussian groups for forecasting, and continuous $\mathbb{SE}(3)$ B-spline bases have been used for monocular reconstruction Zhang et al. (2026). VidLift4D uses fixed semantic groups and per-Gaussian residual transforms within a video-first generation pipeline. Grouping and hierarchical motion by themselves are not new.

4D generation. Score-distillation 4D approaches Poole et al. (2023); Singer et al. (2023); Bahmani et al. (2024a;b); Jiang et al. (2023); Ling et al. (2024); Ren et al. (2023); Yin et al. (2023); Yu et al. (2024); Zhao et al. (2023); Zheng et al. (2024) and feed-forward generators Li et al. (2024); Pan et al. (2024); Yang et al. (2025a); Liang et al. (2024); Ren et al. (2024); Chen et al. (2025) have made progress. Free4D Liu et al. (2025), CAT4D Wu et al. (2025b), GenXD Zhao et al. (2025), and DimensionX Sun et al. (2025) provide relevant text/image-conditioned comparisons. Vidu4D Wang et al. (2024) reconstructs dynamic Gaussian surfels from generated video; DreamScene4D Chu et al. (2024) factorizes object deformation, object-to-world motion, and camera motion. Diff4Splat Pan et al. (2026) predicts dynamic Gaussians directly from video latents in one forward pass. These are close pipeline-level precedents.

VidLift4D differs by per-scene optimization from generated video and by using VGGT both as a training-time feature teacher and a post-generation geometry estimator. It does not reuse the same per-scene features across stages. The reported ablations test parts of this combination, but a stronger independent estimator control remains necessary.

3 METHOD

Given a multimodal prompt $\boldsymbol{c} = (\mathbf{c}_{\text{text}}, \mathbf{c}_{\text{image}}, \mathbf{c}_{\text{video}})$ and a camera trajectory encoded as Plücker embeddings Jia (2020) $\mathcal{P} \in \mathbb{R}^{T\times H\times W\times 6}$, VidLift4D produces a 4D scene that supports novel-view rendering. The pipeline runs in two stages, illustrated in Figure 2: (i) camera-conditioned video diffusion trained with the published Geometry Forcing (GF) loss, and (ii) per-scene reconstruction with dynamic Gaussians. VGGT acts as a feature teacher during Stage-I training; a separate VGGT inference pass on each output video supplies Stage-II geometry. We summarize latent video diffusion Rombach et al. (2022), GF Wu et al. (2025a), and 3D Gaussian Splatting Kerbl et al. (2023) in the supplement (Section C); below we describe only what is novel or specific to VidLift4D.

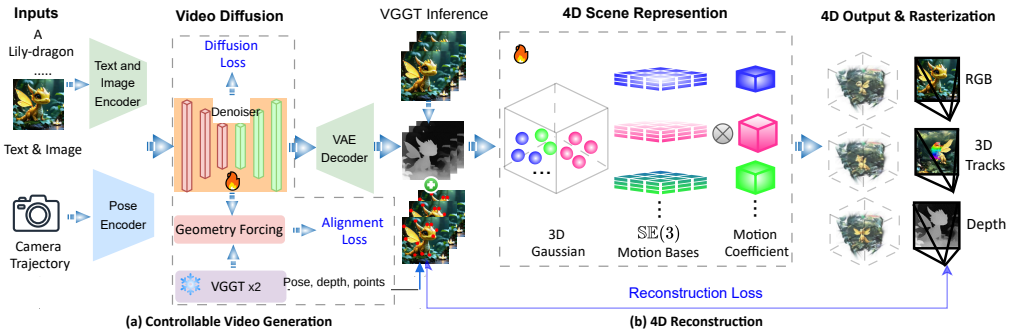


Figure 2: VidLift4D pipeline. The shown text/image branch is illustrative; prior-assisted video inputs use SDEdit. The frozen VGGT model is invoked separately for Stage-I training feature supervision and post-generation Stage-II pose, depth, and point-map estimation ("VGGT x2"); the latter does not provide dynamic point tracks. Instance masks are external. Stage II uses estimated poses, which may differ from the requested path.

3.1 CAMERA-CONDITIONED VIDEO GENERATION WITH GEOMETRY FORCING

Stage I is a latent video diffusion model fine-tuned for camera-conditioned video generation. We follow CameraCtrl He et al. (2024) and AC3D Bahmani et al. (2025): a camera encoder $\mathcal{E}_{\text{cam}}$ converts the per-frame Plücker map $\mathcal{P}_t$ into multi-scale features that are injected into the temporal-attention modules of the diffusion U-Net at the matching spatial resolutions.

Plücker camera representation. Under a world-to-camera convention with extrinsics $(\mathbf{R}, \mathbf{t})$, the camera center in world coordinates is $\boldsymbol{o} = -\mathbf{R}^\top \mathbf{t}$. For each pixel (u, v), the Plücker embedding is

$$\boldsymbol{p}_{u,v} = \langle \boldsymbol{o} \times \hat{\boldsymbol{d}}_{u,v},\, \hat{\boldsymbol{d}}_{u,v} \rangle \in \mathbb{R}^6, \tag{1}$$

where $\hat{\boldsymbol{d}}_{u,v}$ is the unit ray direction in world coordinates obtained by normalizing $\mathbf{R}^\top \mathbf{K}^{-1}[u, v, 1]^\top$. The full trajectory is $\mathcal{P} \in \mathbb{R}^{T \times H \times W \times 6}$. Its translation component inherits the scale convention of the supplied camera poses.

Multimodal conditioning. We unify the three modalities through the conditioning channel $\boldsymbol{c} = (\mathbf{c}_{\text{text}}, \mathbf{c}_{\text{image}}, \mathbf{c}_{\text{video}})$. For *text-only* inputs, $\mathbf{c}_{\text{image}}$ is sampled by an off-the-shelf text-to-image model conditioned on $\mathbf{c}_{\text{text}}$. For *image* inputs, the provided image is used directly as $\mathbf{c}_{\text{image}}$. When Stage I is invoked for a *video* input, we treat the video as a noisy reference and run SDEdit-style partial-noise inversion He et al. (2024) at 60% of the maximum diffusion timestep. This retains some global visual content while allowing the GF-trained backbone to change the observations; the content change is quantified in the supplement. The same-input iPhone protocol skips Stage I.

Geometry Forcing. Camera conditioning alone does not enforce 3D geometric consistency. We therefore incorporate Geometry Forcing (GF) Wu et al. (2025a) as a training-time auxiliary loss only. GF aligns the diffusion latent features $\boldsymbol{h} \in \mathbb{R}^{N \times P' \times D'}$ with intermediate features $\boldsymbol{y} \in \mathbb{R}^{L \times N \times P \times D}$ produced by a pretrained geometric foundation model VGGT Wang et al. (2025a), where L, N, P, D are the number of VGGT layers, views, patches, and feature dimensions. A lightweight projector f_ϕ maps $\boldsymbol{h}$ to $\boldsymbol{m}_{n,p} = f_\phi(\boldsymbol{h}_{n,p})$, and a prediction head g_ϕ recovers feature scale from the ℓ_2-normalized $\boldsymbol{m}$. The Angular Alignment loss enforces directional consistency

$$\mathcal{L}_{\text{Ang}} = -\tfrac{1}{LNP} \sum_{\ell,n,p} \cos(\boldsymbol{y}_{\ell,n,p},\, \boldsymbol{m}_{n,p}), \tag{2}$$

and the Scale Alignment loss recovers magnitude

$$\mathcal{L}_{\text{Sca}} = \tfrac{1}{LNP} \sum_{\ell,n,p} \left\| g_\phi\big(\boldsymbol{m}_{n,p}/\|\boldsymbol{m}_{n,p}\|_2\big) - \boldsymbol{y}_{\ell,n,p} \right\|_2^2. \tag{3}$$

Note that $\boldsymbol{m}_{n,p}$ does not depend on the layer index ℓ: one projected feature is aligned to all selected VGGT layers. The total Stage-I objective is

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{VDM}} + \lambda_{\text{GF}} \left(\lambda_{\text{Ang}} \mathcal{L}_{\text{Ang}} + \lambda_{\text{Sca}} \mathcal{L}_{\text{Sca}}\right), \tag{4}$$

where $\mathcal{L}_{\text{VDM}}$ is the standard latent-video noise-prediction loss. GF is used during fine-tuning only; sampling is unchanged.

3.2 Post-generation Geometry Frontend

Standard SoM-style monocular reconstruction Wang et al. (2025b); Lei et al. (2024) runs a cascade of independent estimators on the input video: per-frame masks $\{\mathbf{M}_t\}$ from Track-Anything Kirillov et al. (2023); Yang et al. (2023), depth $\{\mathbf{D}_t\}$ from Depth Anything Yang et al. (2024a), 2D tracks $\{\mathbf{U}_t\}$ from TAPIR Doersch et al. (2023), and camera poses $\{\mathbf{P}_t\}$ from DROID-SLAM Teed & Deng (2021). These independently estimated quantities can disagree on coordinate system, scale, and visibility. Whether a VGGT frontend offers a benefit specifically because it shares a training-time teacher with Stage I, beyond its standalone estimation quality, requires a controlled comparison with an independent frontend.

We run VGGT Wang et al. (2025a) on the generated video in overlapping 24-frame windows (stride 12), then align overlapping camera predictions with a similarity transform. This yields $\hat{\mathcal{A}}_t = (\hat{\mathbf{P}}_t, \hat{\mathbf{D}}_t, \hat{\mathbf{X}}_t)$: estimated poses, dense depth, and point maps in a common coordinate system. These are outputs of a new inference pass, not geometry exported by Stage I. A static 3D point reprojected across frames is not a dynamic point correspondence, so we do not use such reprojections as long-range foreground tracks or as a track-supervision loss. Foreground masks still come from Track-Anything. Stage II uses $\hat{\mathcal{A}}_t$ for initialization and camera/depth supervision. The requested trajectory and $\hat{\mathbf{P}}_t$ may differ; the current evaluation does not quantify that difference.

3.3 Hierarchical $\mathbb{SE}(3)$ Motion for Dynamic 3D Gaussians

We represent the scene as N canonical 3D Gaussians $\{\boldsymbol{g}_0^j\}_{j=1}^N$, each with mean $\boldsymbol{\mu}_0^j \in \mathbb{R}^3$, orientation $\boldsymbol{R}_0^j \in \mathbb{SO}(3)$, and time-shared scale, opacity, and color. Following DynMF Kratimenos et al. (2023)

and SoM Wang et al. (2025b), Gaussian j moves by $\boldsymbol{T}_{0:t}^{j} \in \mathbb{SE}(3)$ applied to $(\boldsymbol{\mu}_0^j, \boldsymbol{R}_0^j)$. Our implementation combines a fixed semantic-group transform with a per-Gaussian residual.

Hierarchical decomposition. We partition the canonical Gaussians into B disjoint semantic groups $\{\mathcal{G}^{(b)}\}_{b=1}^{B}$ and define

$$\boldsymbol{T}_{0:t}^{j} = \boldsymbol{T}_{0:t}^{(\pi(j))} \circ \boldsymbol{T}_{0:t}^{(j)}, \tag{5}$$

where $\pi(j) \in \{1, \dots, B\}$ assigns Gaussian j to its group. The two factors are

$$\boldsymbol{T}_{0:t}^{(b)} = \exp\Big(\sum_{k=1}^{K_g} c_{0:t}^{(b,k)} \boldsymbol{e}_g^{(k)}\Big), \quad \boldsymbol{T}_{0:t}^{(j)} = \exp\Big(\sum_{k=1}^{K_p} c_{0:t}^{(j,k)} \boldsymbol{e}_p^{(j,k)}\Big), \tag{6}$$

where $\boldsymbol{T}_{0:t}^{(b)}$ uses a global pool of K_g bases $\boldsymbol{e}_g^{(k)} \in \mathfrak{se}(3)$ with group-specific coefficients, and $\boldsymbol{T}_{0:t}^{(j)}$ uses K_p point-specific bases. The composition in Equation (5) is the implemented transform. In general, its logarithm is not the sum of the two Lie-algebra terms because $\mathbb{SE}(3)$ is noncommutative.

Group assignment. We assign each Gaussian to the persistent instance ID of the mask covering its canonical-frame projection and hold the assignment fixed. Background Gaussians have an identity group transform; whether their per-Gaussian residuals are also frozen is not documented, so this does not by itself guarantee static background geometry. Track-Anything supplies mask identities in later frames for mask supervision, not per-point correspondences. The equal-sequence mean of the seven listed group-purity values on iPhone is 93.2% (lowest 82.1% on `teddy`); see supplement.

Optimization. At each iteration we render the RGB image $\hat{\mathbf{I}}_t$, depth $\hat{\mathbf{D}}_t$, and foreground mask $\hat{\mathbf{M}}_t$ from the post-generation VGGT camera estimate $\hat{\mathbf{P}}_t$ via 3DGS rasterization Kerbl et al. (2023), and then supervise them with a per-frame loss $\mathcal{L}_{\text{render}} = \|\hat{\mathbf{I}}_t - \mathbf{I}_t\|_1 + \lambda_{\text{D}} \|\hat{\mathbf{D}}_t - \hat{\mathbf{D}}_t^{\text{vggt}}\|_1 + \lambda_{\text{M}} \|\hat{\mathbf{M}}_t - \mathbf{M}_t\|_1$. Three regularizers act on the motion: a group temporal-smoothness $\mathcal{L}_{\text{group}} = \sum_{b,k,t} \|c_{0:t}^{(b,k)} - c_{0:t-1}^{(b,k)}\|_2^2$, a point-coefficient penalty $\mathcal{L}_{\text{point}} = \sum_{j,k,t} \|c_{0:t}^{(j,k)}\|_2^2$, and a residual-transform penalty $\mathcal{L}_{\text{hier}} = \sum_{j,t} \|\log((\boldsymbol{T}_{0:t}^{(\pi(j))})^{-1} \boldsymbol{T}_{0:t}^{j})\|_2^2$ where $\log : \mathbb{SE}(3) \to \mathfrak{se}(3)$ is the matrix logarithm. Since $(\boldsymbol{T}_{0:t}^{(\pi(j))})^{-1} \boldsymbol{T}_{0:t}^{j} = \boldsymbol{T}_{0:t}^{(j)}$, $\mathcal{L}_{\text{hier}}$ penalizes residual motion in twist space; it is related to $\mathcal{L}_{\text{point}}$ and is not an independent group-consistency constraint. The total Stage-II objective is

$$\mathcal{L} = \mathcal{L}_{\text{render}} + \lambda_{\text{g}} \mathcal{L}_{\text{group}} + \lambda_{\text{p}} \mathcal{L}_{\text{point}} + \lambda_{\text{h}} \mathcal{L}_{\text{hier}}. \tag{7}$$

The learned bases and coefficients have a rescaling ambiguity, and an unweighted twist norm depends on the translation scale of the reconstructed scene. Basis constraints, a canonical-frame anchor, and rotation/translation units are not documented in the available implementation record; Section E states the resulting interpretation limits. The hierarchy is therefore a motion parameterization used in this pipeline, not an established scale-invariant or parameter-efficient regularizer.

3.4 Pipeline Summary and Implementation

The inference algorithm is provided in the supplementary material (Section F).

Implementation. Stage I is built on a camera-controllable I2V backbone He et al. (2024); Bahmani et al. (2025), with the text/image encoders and VAE frozen. We fine-tune its temporal-attention adapters, camera encoder, and GF projectors on a curated subset of SEVA Zhou et al. (2025) ($\sim$320K clips, 49 frames each, 480×720); filtering is described in the supplement. Stage II optimizes each scene on a single NVIDIA RTX 3090: $1{,}000$ canonical-frame iterations followed by 600 joint epochs. The default has $B = 15$ semantic-group slots, $K_g = 15$ shared group bases, and $K_p = 2$ bases per Gaussian; B and K_g are distinct parameters that happen to share a default value. An 80-frame clip at 960×720 takes about 0.5 h to optimize and renders at 20 fps.

4 Experiments

We evaluate VidLift4D on real monocular reconstruction (iPhone Gao et al. (2022)), text/image-conditioned 4D generation (VBench Huang et al. (2024)), synthetic 4D scenes with ground-truth depth and trajectories (Kubric Movi-F Greff et al. (2022)), and video-to-4D re-rendering (DAVIS Perazzi et al. (2016)). The protocols answer different questions: iPhone raw videos support a same-input

Method	PSNR ↑	SSIM ↑	LPIPS ↓
4D Gaussian Wu et al. (2024)	15.42	0.58	**0.45**
Deformable 3DGS Yang et al. (2024b)	15.55	0.58	0.47
Dynamic 3DGS (mono.) Luiten et al. (2024)	15.30	0.57	0.48
MoSca Lei et al. (2024)	15.71	0.59	0.46
CAT4D Wu et al. (2025b)	15.18	0.57	0.49
SoM Wang et al. (2025b)	**15.89**	**0.60**	0.46
Ours, raw input (0% SDEdit)	15.91	0.60	–

Table 1: Reported iPhone validation-view aggregate over 7 sequences under the same original-video input. VidLift4D's raw-input LPIPS and per-sequence variation were not reported. Best baseline values are bold; exact aggregation weights require verification. Prior-assisted results are kept separate in Tables 5 and 7.

reconstruction comparison; prior-assisted iPhone changes VidLift4D's input; Kubric has external depth and trajectory targets; text/image VBench and DAVIS assess rendered appearance under the stated input differences. Full input and target details are in Table 8. Internal static-background diagnostics use each method's own cameras, depths, and validity set, so they cannot establish geometric accuracy or requested-camera compliance (Section B). Valid-pixel coverage, requested-camera error, and full baseline uncertainty remain unavailable.

4.1 Real Monocular Dynamic Reconstruction (iPhone)

We distinguish two iPhone protocols. In the same-input protocol, Stage II directly reconstructs the real video (0% SDEdit) and obtains 15.91 dB PSNR versus SoM's 15.89 dB; the 0.02 dB difference is too small to support a reconstruction win without uncertainty estimates. In the prior-assisted protocol, VidLift4D first changes the video with 60% SDEdit, then reconstructs the altered observations. This obtains 16.23 dB but reduces input-view CLIP-I to 0.91 (Table 7); it is a distinct video-to-video prior setting. We compare with 4D Gaussian Wu et al. (2024), Deformable 3DGS Yang et al. (2024b), Dynamic 3D Gaussians (mono.) Luiten et al. (2024), MoSca Lei et al. (2024), CAT4D Wu et al. (2025b), and SoM Wang et al. (2025b). USplat4D Guo et al. (2026) is an important missing baseline for a revised study.

Table 1 puts the same-input comparison first. The 0.02 dB PSNR difference from SoM has no raw-input per-sequence uncertainty or VidLift4D LPIPS, so it is a practical tie at the reported precision. The prior-assisted 16.23 dB result is analyzed separately in Tables 5 and 7; comparing it with SoM's raw-input 15.89 dB gives a 0.34 dB numerical difference across unequal observations, not a reconstruction gain. In the prior-assisted sequence table, all seven rounded PSNR differences from SoM are positive (two-sided exact sign-test $p = 0.0156$), but this test does not apply to the same-input protocol. The prior-assisted LPIPS ties SoM and trails 4D Gaussian.

The examples in Figure 3 illustrate render differences. A still frame cannot establish temporal stability or isolate the frontend effect; the factorial PSNR grid provides the relevant quantitative comparison.

4.2 4D Generation (text-/image-to-4D)

We use 30 text prompts and 30 single-image conditions sourced from public datasets Zhou et al. (2025), with five requested trajectories per scene (still, dolly, orbit, crane, free). These are identical requested paths where the baseline supports camera conditioning; actual output viewpoints are not verified. The original experiment description identifies the VBench inputs as *rendered videos*, which indicates Stage-II Gaussian renderings for VidLift4D; raw clip manifests are unavailable to verify this interpretation. Exact clip sampling, render resolution, baseline seed counts, and achieved camera paths are not preserved; we report three-seed scores for VidLift4D only. Text baselines are Free4D Liu et al. (2025), CAT4D Wu et al. (2025b), 4Real Yu et al. (2024), 4Dfy Bahmani et al. (2024b), and Dream-in-4D Zheng et al. (2024); image baselines are Free4D, GenXD Zhao et al. (2025), and DimensionX Sun et al. (2025). GenXD videos are passed through the Free4D reconstruction pipeline (†), so that row is not directly comparable. Diff4Splat Pan et al. (2026) is a missing image-to-4D baseline.

Tables 2 and 3 report VBench scores. VidLift4D has the highest reported Aesthetic score in both protocols. For text-to-4D, CAT4D leads Consistency and Dream-in-4D leads Dynamic Degree; a single operating point cannot establish a geometry–perception Pareto trade-off. The static-background diagnostics are isolated in Table 9 because their valid-pixel coverage is unknown and their cameras and depths are method-specific. A 25-participant study reports preference over Free4D, 4Real, 4Dfy,

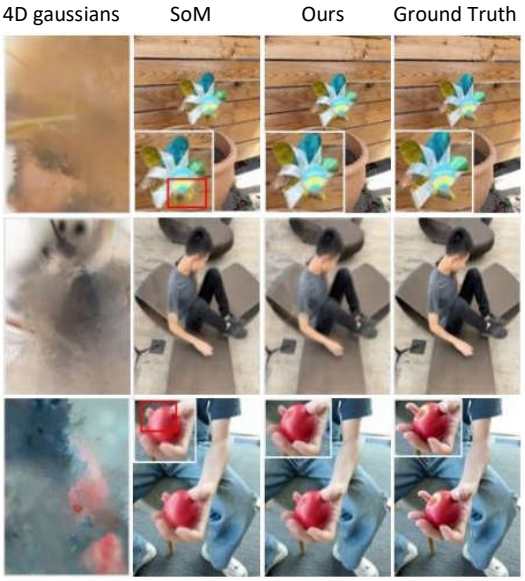


Figure 3: Qualitative iPhone comparison Gao et al. (2022). Rows show matched examples and columns show methods. VidLift4D uses the 60% SDEdit prior.

Method	TextAl. ↑	Cons. ↑	Dyn. ↑	Aes. ↑
Free4D	**26.2**	95.5	33.3	60.5
4Real	26.0	95.7	33.7	51.9
4Dfy	25.8	91.3	<u>52.4</u>	55.6
Dream4D	25.3	91.6	**53.4**	56.2
CAT4D	25.9	**97.7**	52.2	61.9
Ours	**26.2**	<u>96.8</u>	49.6	**62.7**

Table 2: Text-to-4D VBench scores on 30 prompts and five requested trajectories. Achieved paths and baseline seed counts are unverified. Best in bold, second best underlined; scores in %. Dream4D denotes Dream-in-4D Zheng et al. (2024).

Method	Cons. ↑	Dyn. ↑	Aes. ↑
Free4D	96.9	40.4	60.5
GenXD†	90.2	**98.3**	38.2
DimensionX	**97.7**	22.2	55.0
Ours	<u>97.2</u>	<u>52.1</u>	**63.4**

Table 3: Image-to-4D VBench scores on 30 images and five requested trajectories. Achieved paths and baseline seed counts are unverified. †GenXD Zhao et al. (2025) uses Free4D Liu et al. (2025) reconstruction, so its row is not directly comparable.

	Kubric		DAVIS	
Method	AbsRel ↓	EPE (m) ↓	Cont. ↑	Nov. ↑
MoSca	0.118	0.071	–	–
SoM	0.103	0.058	–	–
CAT4D	0.096	0.053	0.83	0.71
Free4D	–	–	0.81	0.68
Ours	0.082	0.041	0.88	0.74

Table 4: Reported point estimates. Kubric-derived synthetic protocol Greff et al. (2022): 50 scenes, with scale alignment, matching, input parity, and uncertainty undocumented. DAVIS Perazzi et al. (2016): 20 sequences; VidLift4D uses full video while Free4D/CAT4D use one frame and a caption.

and Dream-in-4D and a tie with CAT4D (supplement); the unit of analysis and multiplicity correction still require clarification.

4.3 Synthetic 4D Ground Truth (Kubric)

The supplied experiment record reports 50 Kubric-derived scenes Greff et al. (2022), each described as containing 1–6 dynamic rigid objects on a textured ground plane, an 80-frame monocular training video, and 8 held-out novel views with ground-truth depth and per-object 3D trajectories. These are custom settings rather than the released MOVi-F split; the rendering configuration and scene IDs are unavailable. The reported absolute relative depth error (AbsRel) and object-trajectory end-point error (EPE) are lower for VidLift4D than for MoSca Lei et al. (2024), SoM Wang et al. (2025b), and CAT4D Wu et al. (2025b) (Table 4, left). The point-estimate differences from SoM are 20% and 29%, respectively. However, the available record does not specify whether Stage I modified the Kubric input, how monocular scale was aligned, how objects were matched to predicted trajectories, or how errors were aggregated over views, time, and scenes. It also lacks per-scene dispersion. These omissions prevent a causal attribution to the two-stage coupling or an uncertainty assessment of the reported gaps.

4.4 Video-to-4D Re-rendering (DAVIS)

We additionally evaluate VidLift4D in a video-to-4D setting: given a real video, we reconstruct a 4D Gaussian scene along a new camera trajectory. On 20 DAVIS sequences Perazzi et al. (2016), we

report content preservation (CLIP-I Radford et al. (2021) between input and reconstructed video at the original camera) and cross-trajectory novel-view CLIP-I. Since DAVIS lacks multi-view ground truth, CLIP-I measures semantic/appearance fidelity rather than 4D geometry directly. Free4D and CAT4D use only the first frame as image input with a CLIP-T Radford et al. (2021) caption; VidLift4D receives the full video. The larger VidLift4D values in Table 4 (right) are therefore a capability illustration under unequal information, not evidence of a method advantage at matched input.

4.5 Ablation Study

The four cells in Table 5 separate GF from the VGGT frontend in the prior-assisted iPhone setting. With GF enabled, replacing VGGT by the SoM cascade changes PSNR from 16.23 to 15.91 dB (-0.32 dB). With the VGGT frontend fixed, removing GF changes it to 15.95 dB (-0.28 dB). The neither-component cell is 15.78 dB: the 0.45 dB difference from full VidLift4D is a joint point-estimate difference, not a frontend-only effect. The four rounded point estimates yield a 0.15 dB difference-in-differences; without per-sequence uncertainty this is not evidence of a nonzero population interaction. This grid does not compare against a modern unified geometry estimator such as ViPE. Removing per-Gaussian residual motion changes PSNR by -0.35 dB; removing the twist-space residual penalty changes it by -0.06 dB. Table 6 compares motion modules under common reported prior-assisted inputs, but an exact trainable-parameter count and compute parity require the coefficient time-parameterization. GF improves reported FVD by 9% and reduces VGGT-estimated video reprojection error by 14%; the latter is teacher-dependent.

Stage-I	VGGT	SoM cascade
GF off	15.95	15.78
GF on	**16.23**	15.91

Table 5: Prior-assisted iPhone PSNR (dB). The 0.15 dB GF-by-frontend interaction is a rounded point estimate without sequence-level uncertainty; the cascade is not an equally strong independent frontend.

Motion model	PSNR (dB)
Per-Gaussian MLP	15.81
HexPlane	15.96
Rigid group only	15.62
Group + point residuals	**16.23**

Table 6: Reported motion comparison on prior-assisted iPhone inputs with common canonical Gaussians and masks. Parameter counts, temporal coefficients, and optimization parity remain unaudited.

Noise Level	Held-out view		Input view: Stage-I versus original			
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	LPIPS ↓	CLIP-I ↑
0% (raw)	15.91	0.60	∞	1.00	0.00	1.00
20%	16.02	0.60	27.4	0.93	0.06	0.97
40%	16.14	0.61	23.1	0.86	0.11	0.94
60%	**16.23**	**0.62**	19.8	0.78	0.18	0.91
80%	16.04	0.60	16.5	0.69	0.27	0.84

Table 7: Reported iPhone SDEdit sweep. Held-out-view rendering improves through 60% noise while the video entering Stage II departs increasingly from the original input. Only the 0% row has input parity with raw-video baselines.

4.6 Applications and Analysis

The 4D Gaussian representation supports novel-view rendering (Figure 1) and produces 3D trajectories, rendered depth, and surface normals. Track queries, ground truth, visibility rules, and per-sequence results are not reported here, so the trajectory visualizations are qualitative examples.

Trade-off analysis. On text-to-4D, CAT4D exceeds VidLift4D's VBench Consistency by 0.9 percentage points and Dream-in-4D exceeds its Dynamic Degree by 3.8 points. The historical ReprErr/MVC scores have no known coverage or independent geometry reference. On Kubric, the reported depth AbsRel and object-trajectory EPE point estimates are about 20% and 29% below SoM's, but the scale and matching protocol is unavailable. The iPhone 0.34 dB numerical margin uses altered observations; the same-input margin is 0.02 dB. Common inputs and an independent geometry evaluator are needed to establish a geometry–perception trade-off.

Failure modes. The supplement discusses two observed artifacts and one structural limit. *(i) Topology change:* fixed group motion biases the parts of a split object to move together; point

residuals may separate them, but no split-object stress test is reported. *(ii) Large unobserved regions:* unreliable VGGT depth can cause ghosting on previously unseen surfaces, as in DAVIS `breakdance-flare`. *(iii) Mask identity flicker:* on iPhone `teddy`, later Track-Anything masks sometimes exchange arm identities. The fixed group assignment does not change, but the mask supervision can then conflict with that assignment and produce boundary artifacts; this is the lowest reported group-purity sequence (82.1%).

Reproducibility. We use released baseline implementations and report three-seed averages where available. The protocols do not always supply the same information: prior-assisted iPhone alters only VidLift4D's input, and DAVIS video-to-4D baselines use a first frame and caption while VidLift4D uses the full video. Requested camera trajectories and image resolutions are held fixed where the baseline supports them; output camera adherence remains unevaluated. Stage-I training-set filtering is described in the supplement.

4.7 Evaluation Scope and Reproducibility

Identity of the synthetic benchmark. The released MOVi-F description Google Research (2022) specifies two-second clips at 12 fps, 10–20 static and 1–3 dynamic objects, moving cameras, and uint16 ray-distance depth. The reported 80-frame, 1–6-object, eight-held-out-view protocol thus requires a custom renderer. Public MOVi-F clips cannot substitute for the historical 50 scenes: ray distance must first be converted to camera-z depth, and scene seeds, camera scripts, assets, and held-out views must match.

Independent geometry and motion checks. A verifiable comparison must fix the input clip, masks, requested path, train/test views, and optimization budget across methods. Estimate one depth scale from input views and apply it unchanged to held-out depth; test-view alignment would leak the target. Object EPE requires matched physical IDs, visibility masks, and one similarity transform fitted to input cameras. Camera orientations resolve the rotation gauge of straight-line paths, while a stationary input camera cannot determine object scale from poses alone. Retain scene-level AbsRel, RMSE, EPE, coverage, and failures, and compute paired intervals over scenes. Failed scenes and empty visibility sets must be counted and reported explicitly rather than silently removed from the denominator. Our evaluation code implements this contract, but absent original predictions and scene IDs prevent new VidLift4D scores.

Requested versus achieved camera path. The requested path, the path inferred from the generated video, and the final render path may differ. Each achieved path needs one clip-wide Sim(3) alignment to the request; per-frame alignment would erase drift. Report rotation, center ATE, relative pose error, and failures by path type. Still paths have no translation scale, so normalized translation error is undefined. The requested and achieved arrays are absent; VBench and internal reprojection scores do not establish camera control.

5 Conclusion and Limitations

VidLift4D combines camera-conditioned video diffusion and per-scene 4D Gaussian reconstruction through two distinct uses of VGGT: training-time GF feature supervision and post-generation pose/depth/point-map estimation. Fixed semantic groups and per-Gaussian residual transforms structure the motion model, although its parameter efficiency is not established. The reported Kubric depth and trajectory point estimates favor VidLift4D, but missing input and scale-alignment details prevent attribution to the common teacher. Text/image scores describe visual quality; internal static-background diagnostics are presented separately and requested-camera error remains unmeasured. On iPhone, the same-input result is effectively tied with SoM at reported precision, while the larger prior-assisted result changes the input video. Fixed grouping, uncertain depth in unobserved regions, and mask flicker limit the method. Stronger geometry frontends, matched-input baselines, dynamic-region ground truth, valid-pixel coverage, and reproducible per-scene outputs are required before claiming a general advantage for the common teacher or motion hierarchy.

AI USE STATEMENT

Generative AI tools assisted with language editing, literature discovery, and consistency checks. The authors remain responsible for the technical content, citations, and reported results.

REFERENCES

Sherwin Bahmani, Xian Liu, Wang Yifan, Ivan Skorokhodov, Victor Rong, Ziwei Liu, Xihui Liu, Jeong Joon Park, Sergey Tulyakov, Gordon Wetzstein, et al. Tc4d: Trajectory-conditioned text-to-4d generation. In *European Conference on Computer Vision*, pp. 53–72. Springer, 2024a.

Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B. Lindell. 4d-fy: Text-to-4d generation using hybrid score distillation sampling. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22875–22889, 2025.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. Technical report, OpenAI, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 130–141, 2023.

Yukang Cao, Jiahao Lu, Zhisheng Huang, Zhuowen Shen, Chengfeng Zhao, Fangzhou Hong, Zhaoxi Chen, Xin Li, Wenping Wang, Yuan Liu, et al. Reconstructing 4d spatial intelligence: A survey. *arXiv preprint arXiv:2507.21045*, 2025.

Zhaoxi Chen, Tianqi Liu, Long Zhuo, Jiawei Ren, Zeng Tao, He Zhu, Fangzhou Hong, Liang Pan, and Ziwei Liu. 4dnex: Feed-forward 4d generative modeling made easy. *arXiv preprint arXiv:2508.13154*, 2025.

Wen-Hsuan Chu, Lei Ke, and Katerina Fragkiadaki. Dreamscene4d: Dynamic multi-object scene generation from monocular videos. *Advances in Neural Information Processing Systems*, 37:96181–96206, 2024.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10061–10072, 2023.

Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022.

Google Research. Kubric MOVi-F dataset builder. https://github.com/google-research/kubric/blob/main/challenges/movi/movi_f.py, 2022.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3749–3761, 2022.

Fengzhi Guo, Chih-Chuan Hsu, Sihao Ding, and Cheng Zhang. Uncertainty matters in dynamic gaussian splatting for monocular 4d reconstruction. In *International Conference on Learning Representations*, 2026.

Yuwei GUO, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo DAI. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Representation Learning*, volume 2024, pp. 34630–34648, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/94894cf9bab20c00f6eaaff3b7f2be70-Paper-Conference.pdf.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation, 2024.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixè, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yan-Bin Jia. Plücker coordinates for lines in the space. *Problem Solver Techniques for Applied Computer Science, Com-S-477/577 Course Handout*, 3, 2020.

Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. Consistent4d: Consistent 360-degree dynamic object generation from monocular video. *arXiv preprint arXiv:2311.02848*, 2023.

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

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Lingdong Kong, Wesley Yang, Jianbiao Mei, Youquan Liu, Ao Liang, Dekai Zhu, Dongyue Lu, Wei Yin, Xiaotao Hu, Mingkai Jia, et al. 3d and 4d world modeling: A survey. *arXiv preprint arXiv:2509.07996*, 2025.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. In *European Conference on Computer Vision*, 2023. URL https://api.semanticscholar.org/CorpusID:265551457.

Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas J Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 16240–16271. Curran Associates, Inc., 2024. doi: 10.52202/079017-0518. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1d49235669869ab737c1da9d64b7c769-Paper-Conference.pdf.

Junmyeong Lee, Hoseung Choi, and Minsu Cho. Space-time forecasting of dynamic scenes with motion-aware gaussian grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41033–41043, 2026.

Jiahui Lei, Yijia Weng, Adam W. Harley, Leonidas J. Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6165–6177, 2024. URL https://api.semanticscholar.org/CorpusID:270062699.

Bing Li, Cheng Zheng, Wenxuan Zhu, Jinjie Mai, Biao Zhang, Peter Wonka, and Bernard Ghanem. Vivid-zoo: Multi-view video generation with diffusion model. *Advances in Neural Information Processing Systems*, 37: 62189–62222, 2024.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Hanwen Liang, Yuyang Yin, Dejia Xu, Hanxue Liang, Zhangyang Wang, Konstantinos N Plataniotis, Yao Zhao, and Yunchao Wei. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models. *arXiv preprint arXiv:2405.16645*, 2024.

Yiming Liang, Tianhan Xu, and Yuta Kikuchi. Himor: Monocular deformable gaussian reconstruction with hierarchical motion representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 886–895, 2025.

Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8576–8588, 2024.

Tianqi Liu, Zihao Huang, Zhaoxi Chen, Guangcong Wang, Shoukang Hu, Liao Shen, Huiqiang Sun, Zhiguo Cao, Wei Li, and Ziwei Liu. Free4d: Tuning-free 4d scene generation with spatial-temporal consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25571–25582, October 2025.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pp. 800–809. IEEE, 2024.

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.

Zijie Pan, Zeyu Yang, Xiatian Zhu, and Li Zhang. Efficient4d: Fast dynamic 3d object generation from a single-view video. *arXiv preprint arXiv:2401.08742*, 2024.

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.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732, 2016.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=FjNys5c7VyY`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023.

Jiawei Ren, Cheng Xie, Ashkan Mirzaei, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, Huan Ling, et al. L4gm: Large 4d gaussian reconstruction model. *Advances in Neural Information Processing Systems*, 37:56828–56858, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, and Yaniv Taigman. Text-to-4d dynamic scene generation. *arXiv:2301.11280*, 2023.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhu, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with decoupled video diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13695–13706, October 2025.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

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

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025a.

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.

Yikai Wang, Xinzhou Wang, Zilong Chen, Zhengyi Wang, Fuchun Sun, and Jun Zhu. Vidu4d: Single generated video to high-fidelity 4d reconstruction with dynamic gaussian surfels. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. *arXiv preprint arXiv:2507.07982*, 2025a.

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 (CVPR)*, pp. 26057–26068, June 2025b.

Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos, 2023.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10371–10381, 2024a.

Zeyu Yang, Zijie Pan, Chun Gu, and Li Zhang. Diffusion^2: Dynamic 3d content generation via score composition of video and multi-view diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025a.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Xu Bin, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Representation Learning*, volume 2025, pp. 83048–83077, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ce31378e9f41d8907e97dab172b6c559-Paper-Conference.pdf.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20331–20341, 2024b. doi: 10.1109/CVPR52733.2024.01922.

Yuyang Yin, Dejia Xu, Zhangyang Wang, Yao Zhao, and Yunchao Wei. 4dgen: Grounded 4d content generation with spatial-temporal consistency. *arXiv preprint arXiv:2312.17225*, 2023.

Heng Yu, Chaoyang Wang, Peiye Zhuang, Willi Menapace, Aliaksandr Siarohin, Junli Cao, László Jeni, Sergey Tulyakov, and Hsin-Ying Lee. 4real: Towards photorealistic 4d scene generation via video diffusion models. *Advances in Neural Information Processing Systems*, 37:45256–45280, 2024.

David Junhao Zhang, Roni Paiss, Shiran Zada, Nikhil Karnad, David E. Jacobs, Yael Pritch, Inbar Mosseri, Mike Zheng Shou, Neal Wadhwa, and Nataniel Ruiz. Recapture: Generative video camera controls for user-provided videos using masked video fine-tuning. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2050–2062, 2024. URL https://api.semanticscholar.org/CorpusID:273877842.

Xuankai Zhang, Junjin Xiao, Shangwei Huang, Wei-shi Zheng, and Qing Zhang. Learning explicit continuous motion representation for dynamic gaussian splatting from monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Yuyang Zhao, Zhiwen Yan, Enze Xie, Lanqing Hong, Zhenguo Li, and Gim Hee Lee. Animate124: Animating one image to 4d dynamic scene. *arXiv preprint arXiv:2311.14603*, 2023.

Yuyang Zhao, Chung-Ching Lin, Kevin Lin, Zhiwen Yan, Linjie Li, Zhengyuan Yang, Jianfeng Wang, Gim Hee Lee, and Lijuan Wang. Genxd: Generating any 3d and 4d scenes. In *ICLR*, 2025.

Yufeng Zheng, Xueting Li, Koki Nagano, Sifei Liu, Otmar Hilliges, and Shalini De Mello. A unified approach for text-and image-guided 4d scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7300–7309, 2024.

Jensen (Jinghao) Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint*, 2025.

Protocol	VidLift4D video entering Stage II	Baseline information parity	Independent target
iPhone raw	Original monocular video; Stage I skipped	Same video as reconstruction baselines	Held-out RGB views
iPhone prior	Video after 60% Stage-I SDEdit	Baselines use the original video	Held-out RGB views of original scene
Text/image	Stage-I generated video, then Stage-II scene; VBench described as rendered clips	Same requested paths only where supported; achieved paths unknown	No external 4D geometry target
Kubric	80-frame synthetic video; whether Stage I modifies it is undocumented	Input parity cannot be verified from available record	Held-out depth and object trajectories; scale/matching rules undocumented
DAVIS	Full input video passed through Stage I	Free4D/CAT4D use first frame plus caption	No multiview or 3D ground truth

Table 8: Protocol ledger for the reported results. "Undocumented" means the information was absent from the available experiment record. A metric win across unequal inputs is not a same-input method comparison.

A Overview of Supplement

This supplement covers internal-consistency diagnostics (Section B); preliminaries (Section C); Plücker cameras (Section D); motion derivations (Section E); the inference algorithm (Section F); implementation and artifact availability (Section G); additional experiments (Section H); and qualitative examples (Section J). The user-study and visual-ablation figures are collected separately in Section I.

B Historical Internal Diagnostics and an Audit Definition

The historical ReprErr and MVC values in Table 9 use each method's own rendered camera and depth on its static-background region, but the original implementation and valid-pixel counts are unavailable. For a future reproducible audit, define $\mathcal{S}_t = \{\mathbf{u} : \mathbf{M}_t(\mathbf{u}) = 0\}$, $\mathcal{P} = \{(t, t') : 1 \le |t - t'| \le 4\}$, and $\mathcal{U}_{t,t'} \subseteq \mathcal{S}_t$ as pixels with finite positive depth and in-bounds forward and backward projections. For nonempty pairs $\mathcal{P}_R = \{(t, t') \in \mathcal{P} : |\mathcal{U}_{t,t'}| > 0\}$, a proposed per-valid-pixel reprojection statistic is

$$\textsc{ReprErr}_{\text{audit}} = \frac{1}{|\mathcal{P}_R|} \sum_{(t,t') \in \mathcal{P}_R} \frac{1}{|\mathcal{U}_{t,t'}|} \sum_{\mathbf{u} \in \mathcal{U}_{t,t'}} \|\mathbf{u} - \mathcal{C}_{t \to t' \to t}(\mathbf{u})\|_2, \tag{8}$$

where $\mathcal{C}_{t \to t' \to t}$ back-projects through rendered depth and camera in frame t, projects to t', and returns using depth at t'. Define $\mathcal{V}_{t,t'} = \{\mathbf{u} \in \mathcal{U}_{t,t'} : \|\mathbf{u} - \mathcal{C}_{t \to t' \to t}(\mathbf{u})\|_2 < 1 \text{ px}\}$ and $\mathcal{P}_M = \{(t, t') \in \mathcal{P} : |\mathcal{V}_{t,t'}| > 0\}$. With RGB in $[0, 1]$ and $\widetilde{\mathbf{I}}_{t \leftarrow t'}$ reprojected from t', the color statistic is

$$\text{MVC}_{\text{audit}} = 1 - \frac{1}{|\mathcal{P}_M|} \sum_{(t,t') \in \mathcal{P}_M} \frac{\|\mathbf{I}_t - \widetilde{\mathbf{I}}_{t \leftarrow t'}\|_{1, \mathcal{V}_{t,t'}}}{3|\mathcal{V}_{t,t'}|}. \tag{9}$$

These are proposed audit quantities, not the verified definitions of the historical table values. A pair with no valid pixels is excluded, not scored as failure. Source and target masks may differ; target-frame foreground masking and an explicit z-buffer occlusion rule were not recorded in the historical protocol. MVC's cycle-error selection favors easy pixels. Neither historical statistic reports valid-pixel or pair coverage, so the old values are descriptive within each method, not an independently grounded ranking of 4D geometry or camera control. The accompanying audit script makes its own explicit visibility and mask choices and cannot reproduce the historical table without raw arrays.

C Preliminaries

C.1 Latent Video Diffusion

We employ a latent diffusion model (LDM) Rombach et al. (2022) as the generative backbone. Given a video $\mathbf{x}$ encoded into a latent $\mathbf{z}_0 = \mathcal{E}_{\text{VAE}}(\mathbf{x})$ by a pretrained VAE encoder, the variance-preserving forward process is

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\, \mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{10}$$

Text-to-4D			Image-to-4D		
Method	RErr (px)	MVC	Method	RErr (px)	MVC
Free4D	6.21	0.81	Free4D	6.05	0.82
4Real	6.95	0.79	GenXD†	7.94	0.71
4Dfy	8.13	0.74	DimensionX	5.13	0.84
Dream-in-4D	8.04	0.75	–	–	–
CAT4D	4.93	0.86	–	–	–
VidLift4D	4.42	0.89	VidLift4D	4.61	0.88

Table 9: Historical static-background internal-consistency values, not recomputed with Equations (8) and (9). Their original code, valid-pixel and valid-pair coverage are unavailable, so cross-method ranking is not warranted. †GenXD video uses Free4D reconstruction.

with $t \in \{1, \dots, T\}$ and a fixed noise schedule $\bar{\alpha}_t$. Conditioned on $\boldsymbol{c}$, the reverse denoising process is modeled as a Gaussian whose mean $\boldsymbol{\mu}_\theta(\mathbf{z}_t, t, \boldsymbol{c})$ is predicted by a spatiotemporal U-Net. Training minimizes the standard noise-prediction loss

$$\mathcal{L}_{\text{VDM}} = \mathbb{E}_{\mathbf{z}_0, \boldsymbol{\epsilon}, t}\big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \boldsymbol{c})\|_2^2\big]. \tag{11}$$

At inference, the model iteratively denoises $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to obtain $\mathbf{z}_0$, decoded by the VAE decoder $\mathcal{D}_{\text{VAE}}$.

C.2 Geometry Forcing

Geometry Forcing (GF) Wu et al. (2025a) aligns the diffusion latent features $\boldsymbol{h} \in \mathbb{R}^{N \times P' \times D'}$ with intermediate features $\boldsymbol{y} \in \mathbb{R}^{L \times N \times P \times D}$ produced by a pretrained geometric foundation model, where L, N, P, D are the number of layers, views, patches, and feature dimensions. A projector f_ϕ maps $\boldsymbol{h}$ into the target space, $\boldsymbol{m}_{n,p} = f_\phi(\boldsymbol{h}_{n,p})$. We use the same Angular and Scale alignment objectives as in the main paper. Here $\boldsymbol{m}$ does not depend on the layer index ℓ: a single projected feature is aligned to all selected VGGT layers, and the prediction head g_ϕ is also layer-shared. The indices of those layers and a comparison with layer-conditioned projectors are not recoverable from the available implementation record.

C.3 3D Gaussian Splatting

3DGS Kerbl et al. (2023) represents the scene as a set of 3D Gaussians $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with influence $f(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}))$. Given camera extrinsics $\mathbf{E}$ and intrinsics $\mathbf{K}$, each Gaussian is projected as $\boldsymbol{\mu}' = \Pi(\mathbf{K}\mathbf{E}\boldsymbol{\mu})$, $\boldsymbol{\Sigma}' = \mathbf{J}\mathbf{W}\boldsymbol{\Sigma}\mathbf{W}^\top\mathbf{J}^\top$, with $\mathbf{W}$ the rotation block of $\mathbf{E}$ and $\mathbf{J}$ the Jacobian of perspective projection. Rasterization composes per-pixel color and depth as $\hat{I}(\boldsymbol{p}) = \sum_{i \in \mathcal{H}(\boldsymbol{p})} T_i \alpha_i \boldsymbol{c}_i, \quad \hat{D}(\boldsymbol{p}) = \sum_{i \in \mathcal{H}(\boldsymbol{p})} T_i \alpha_i d_i$, with $\alpha_i = o_i \cdot f(\boldsymbol{p} \mid \boldsymbol{\mu}_i', \boldsymbol{\Sigma}_i')$ and $T_i = \prod_{j=1}^{i-1}(1 - \alpha_j)$.

D Plücker Camera Representation

We follow CameraCtrl He et al. (2024) and use a Plücker embedding as a global positional encoding for camera parameters. Under a world-to-camera convention with extrinsics $(\mathbf{R}, \mathbf{t})$, the camera center in world coordinates is $\boldsymbol{o} = -\mathbf{R}^\top\mathbf{t}$. For each pixel (u, v), the embedding is

$$\boldsymbol{p}_{u,v} = \langle \boldsymbol{o} \times \hat{\boldsymbol{d}}_{u,v},\, \hat{\boldsymbol{d}}_{u,v} \rangle \in \mathbb{R}^6, \tag{12}$$

where $\hat{\boldsymbol{d}}_{u,v} = \boldsymbol{d}_{u,v}/\|\boldsymbol{d}_{u,v}\|_2$ and $\boldsymbol{d}_{u,v} = \mathbf{R}^\top\mathbf{K}^{-1}[u, v, 1]^\top$. The translation $\mathbf{t}$ enters only through the camera center $\boldsymbol{o}$. The full trajectory is $\mathcal{P} \in \mathbb{R}^{T \times H \times W \times 6}$ for T frames.

E Hierarchical Motion: Derivations

We elaborate on the hierarchical motion decomposition introduced in Section 3.3. The group-level transformation is $\boldsymbol{T}_{0:t}^{(b)} = \exp(\sum_{k=1}^{K_g} c_{0:t}^{(b,k)} \boldsymbol{e}_g^{(k)})$ with bases shared across all Gaussians of group b. The per-Gaussian residual is $\boldsymbol{T}_{0:t}^{(j)} = \exp(\sum_{k=1}^{K_p} c_{0:t}^{(j,k)} \boldsymbol{e}_p^{(j,k)})$ with K_p bases per Gaussian and a regularizer that keeps $\|c^{(j,k)}\|$ small.

Noncommutative composition. Write the group and residual twists as $\boldsymbol{a}, \boldsymbol{b} \in \mathfrak{se}(3)$. The implemented transform is $\exp(\boldsymbol{a})\exp(\boldsymbol{b})$. Its Baker–Campbell–Hausdorff expansion contains $\boldsymbol{a} + \boldsymbol{b} + \frac{1}{2}[\boldsymbol{a}, \boldsymbol{b}] + \frac{1}{12}[\boldsymbol{a}, [\boldsymbol{a}, \boldsymbol{b}]] + \cdots$. Terms linear in $\boldsymbol{b}$ can remain when the group motion $\boldsymbol{a}$ is large, so small residual coefficients alone do not justify replacing the composition by $\exp(\boldsymbol{a} + \boldsymbol{b})$. All reported results use the composed transform.

Identifiability and scale. The coefficients and learned bases are not individually identifiable: replacing one basis $\boldsymbol{e}$ by $a\boldsymbol{e}$ and its coefficient c by c/a leaves its twist and rendered motion unchanged. Coefficient penalties such as $\mathcal{L}_{\text{point}}$ and $\mathcal{L}_{\text{group}}$ change under this rescaling unless basis norms are constrained. Also, the Euclidean norm of an $\mathfrak{se}(3)$ twist mixes a rotation measured in radians with a translation measured in scene-length units; a similarity change of the VGGT reconstruction changes their relative weight. The available implementation record does not specify basis normalization, rotation/translation weighting, a logarithm branch, or an explicit $\boldsymbol{T}_{0:0} = \mathbf{I}$ anchor. Consequently we interpret the motion regularizers as the reported parameterization, not as gauge-invariant physical penalties, and make no parameter-efficiency claim.

F Inference Algorithm Details

Inference steps. Given a multimodal condition $\mathcal{C}$ and requested trajectory $\{\mathcal{P}_t\}_{t=1}^{T}$:

1. Resolve the condition. For text/image generation and prior-assisted video-to-4D, generate $\{\mathbf{I}_t\}_{t=1}^{T}$ with the camera-conditioned diffusion model. In the same-input iPhone protocol, set $\{\mathbf{I}_t\}$ to the original frames and skip Stage I. The saved record does not identify which branch produced the Kubric table.
2. Run VGGT in overlapping 24-frame windows (stride 12); align the windows to estimate $(\hat{\mathbf{P}}_t, \hat{\mathbf{D}}_t, \hat{\mathbf{X}}_t)$.
3. Initialize canonical Gaussians from the first-frame point map and assign them to persistent instance-mask IDs.
4. Optimize Gaussian attributes and the composed group/residual transforms by rendering at $\hat{\mathbf{P}}_t$ and minimizing Equation (7).

The condition resolver handles the three input types as described in main Section 3.1: for text input, we generate a first-frame anchor with a public text-to-image model; for image input, we use it directly; for prior-assisted video input, we apply SDEdit-style partial-noise inversion. Raw-input reconstruction uses the observed video without this step. The 24-frame VGGT window with stride 12 makes interior frames appear in overlapping geometry windows, and we stitch outputs by similarity-aligning overlapping camera poses to minimize RMS pose error. Initial Gaussian centers are sampled from the canonical-frame point map $\hat{\mathbf{X}}_0$ at uniformly random pixels with mask-aware density (more samples on dynamic regions). Group assignment $\pi(j)$ is computed once and held fixed; no dynamic re-grouping ablation is available (Section H.7).

G Implementation Details, Runtime, and Artifact Availability

G.1 Stage-I Fine-Tuning

Stage I is described as a publicly released camera-controllable I2V backbone He et al. (2024); Bahmani et al. (2025), but the exact checkpoint and backbone variant are not identified in the available record. The pretrained text and image encoders, as well as the VAE, are frozen; the temporal-attention adapters of the diffusion U-Net, camera encoder $\mathcal{E}_{\text{cam}}$, and GF projectors f_ϕ and g_ϕ are fine-tuned. Each projector is a 4-layer MLP with hidden width 1024, aligning to three intermediate Transformer layers of VGGT Wang et al. (2025a); their exact layer indices are unavailable. Fine-tuning uses 49-frame clips at 480×720, AdamW with learning rates 5×10^{-5} for the diffusion adapters and 1×10^{-4} for auxiliary modules, batch size 8, $\lambda_{\text{Ang}} = 1.0$, $\lambda_{\text{Sca}} = 0.1$, $\lambda_{\text{GF}} = 0.5$. Sampling is reported as 50 DDIM steps with classifier-free guidance scale 5.0. The method for extending 49-frame training clips to 80-frame generation or 200–500-frame input videos is not specified; the generation-time numbers cannot be reproduced without the temporal windowing policy.

Stage-I training data scale. We curate a subset of the public SEVA Zhou et al. (2025) camera-annotated video data: about 320K clips of 49 frames each at 480×720 ($\approx$ 145 hours of video at

30 fps), filtered to retain clips with non-trivial dynamics (mean optical-flow magnitude > 2 px) and well-defined camera trajectories. The duration is calculated as $320{,}000 \times 49/(30 \times 3600)$ hours; clip overlap, if any, is not removed from this total.

Train/eval disjointness. The reported protocol excludes the 30 text prompts and 30 image conditions used in VBench from the Stage-I fine-tuning subset. For the image set, it hash-matches each evaluation image against SEVA clips at scene level and removes a clip whose first frame matches an evaluation image (Hamming distance < 8 on 64-bit perceptual hashes). For text, it excludes clips whose captions have token overlap > 0.6 with an evaluation prompt. A manual check reportedly excluded Kubric seeds, DAVIS sequences, and iPhone validation sequences. The actual clip/split manifest is absent from the current source bundle, so disjointness cannot be independently verified.

Stage-I fine-tuning runs on 8 NVIDIA H100 GPUs for approximately 5 days. Stage-II per-scene optimization uses the resulting frozen backbone and runs on a single NVIDIA RTX 3090.

G.2 Stage-II Per-Scene Optimization

Stage II is per-scene optimization. We use Adam with learning rate 1×10^{-4} for Gaussian attributes and motion coefficients; the procedure consists of $1{,}000$ canonical-frame iterations followed by 600 joint optimization epochs over all frames. We use $K_g = 15$ group-level bases shared across $B = 15$ semantic groups, and $K_p = 2$ point-level bases per Gaussian; the reconstruction loss weights are $\lambda_{\text{D}} = 0.5$, $\lambda_{\text{M}} = 0.1$, $\lambda_{\text{g}} = 10^{-2}$, $\lambda_{\text{p}} = 10^{-3}$, $\lambda_{\text{h}} = 10^{-2}$. The scene is initialized with 50K Gaussian primitives sampled from $\hat{\mathbf{X}}_0$, and we use the adaptive density-control strategy of 3D-GS Kerbl et al. (2023).

We emphasize that Stage-I and Stage-II are not jointly trained: the Stage-II reconstruction loss is not back-propagated through the Stage-I diffusion model, so VidLift4D is best described as an integrated pipeline with two trained components rather than a single end-to-end network.

G.3 Runtime

For an 80-frame clip at 960×720 on a single RTX 3090: video generation (50 sampling steps) $\approx$ 110 s; VGGT inference with overlapping 24-frame windows $\approx$ 18 s; Gaussian initialization $\approx$ 6 s; canonical-frame fitting $\approx$ 3 min; joint hierarchical-motion optimization $\approx$ 23 min. Rendering at 20 fps requires $\approx$ 18 GB of GPU memory.

G.4 Artifact Availability

The current source bundle does not include Stage-I/II code, trained weights, per-scene predictions, prompt/path manifests, Kubric scene seeds, or raw user-study responses. Their absence prevents independent reconstruction of several reported tables and the protocol details identified in Table 8. The separate metric-audit script in the accompanying revision bundle defines coverage-aware diagnostics on saved depth and camera arrays, but cannot recover those missing outputs. Reproducibility requires releasing the exact artifacts and evaluation commands with a subsequent experimental revision; a release promise is not a substitute for them in this submission.

G.5 Geometry-Aware Metrics: Estimator Dependence

ReprErr and MVC use each method's own rendered depth and camera and evaluate only static background. They test internal consistency, are sensitive to method-specific visibility coverage, and can be low even when geometry or the requested camera path is wrong. The Stage-I-only ReprErr in Table 11 instead uses VGGT-predicted depth and camera and is teacher-dependent. Kubric AbsRel and 3D EPE use synthetic ground truth and are the independent geometry measures here.

H Additional Experiments

H.1 Per-Sequence iPhone Results

Table 10 reports the prior-assisted iPhone protocol by sequence. The Ours subscripts are reported standard deviations across three seeds. All seven rounded PSNR comparisons with SoM are positive, giving a two-sided exact sign-test value of $p = 0.0156$ for the altered-input setting. One rounded SSIM pair is tied, and unrounded values are needed for its exact test. Frame-level tests would treat

correlated frames as independent and are not reported. We additionally compute an equal-sequence mean directly from the displayed rounded cells; this transparent descriptive summary differs from the previously reported aggregate, whose weighting remains unknown. These results cannot establish a same-input reconstruction advantage.

	4D Gaussian			MoSca			SoM			Ours		
Seq.	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
apple	16.34	0.61	0.43	16.78	0.62	0.44	17.05	0.63	0.43	**17.41**$_{0.06}$	**0.65**$_{0.01}$	**0.43**$_{0.01}$
block	18.92	0.65	0.42	18.85	0.65	0.43	19.10	0.66	0.42	**19.16**$_{0.05}$	**0.66**$_{0.01}$	**0.42**$_{0.01}$
paper-windmill	13.21	0.50	0.49	13.55	0.51	0.49	13.69	0.52	0.49	**14.34**$_{0.09}$	**0.55**$_{0.01}$	**0.48**$_{0.01}$
spin	15.07	0.55	0.46	15.40	0.56	0.46	15.55	0.57	0.45	**15.97**$_{0.07}$	**0.59**$_{0.01}$	**0.45**$_{0.01}$
teddy	14.70	0.53	0.47	15.02	0.54	0.47	15.21	0.55	0.47	**15.86**$_{0.08}$	**0.58**$_{0.01}$	**0.46**$_{0.01}$
haru-sit	15.58	0.59	0.45	15.81	0.60	0.45	16.04	0.60	0.45	**16.42**$_{0.06}$	**0.62**$_{0.01}$	**0.45**$_{0.01}$
wheel	14.20	0.55	0.46	14.59	0.56	0.45	14.59	0.57	0.46	**14.85**$_{0.07}$	**0.59**$_{0.01}$	**0.45**$_{0.01}$
Equal-seq. mean	15.43	0.57	0.45	15.71	0.58	0.46	15.89	0.59	0.45	16.29	0.61	0.45
Reported agg.	15.42	0.58	0.45	15.71	0.59	0.46	15.89	0.60	0.46	**16.23**	**0.62**	0.46

Table 10: Prior-assisted iPhone PSNR / SSIM / LPIPS on 7 sequences. Subscripts are reported across-seed standard deviations. The equal-sequence row is calculated from the displayed rounded cells, so it is approximate. The aggregate row is transcribed from the original result; its weighting cannot be audited without raw outputs.

H.2 2×2 GF × Shared-Geometry Ablation

Table 5 varies GF and the inference frontend in the prior-assisted setting. The VGGT effect is 0.32 dB with GF and 0.17 dB without it; the difference-in-differences is 0.15 dB. This grid cannot distinguish teacher matching from the intrinsic quality of VGGT because it lacks a comparably strong independent frontend.

H.3 Motion-Module Comparison

Table 6 compares motion modules with the same reported canonical Gaussians and Track-Anything masks. A trainable-parameter comparison is omitted because the temporal parameterization of per-Gaussian coefficients and the basis constraints have not been documented well enough to verify the previously stated 2.3M count. With 50K Gaussians, two independent coefficients per Gaussian per frame, and 80 frames, point coefficients alone would number 8M; an explicit compression or interpolation scheme must be specified before making an efficiency claim.

H.4 Stage-I Video Quality (without Stage-II)

Table 11 reports FVD Unterthiner et al. (2018), CLIP-T Radford et al. (2021), and a cross-frame depth-reprojection error on the generated videos themselves, before invoking Stage-II. The reprojection error here uses VGGT-predicted depth (because there is no rendered depth at this point); we mark this dependence explicitly in the table caption.

Stage-I configuration	FVD ↓	CLIP-T ↑	ReprErr (px) ↓
AC3D backbone, no GF	168.4	0.292	4.71
+ Angular Alignment only	158.5	0.293	4.32
+ Scale Alignment only	156.9	0.291	4.18
+ Full GF (*Ours*)	**152.7**	**0.293**	**4.06**

Table 11: Stage-I-only video quality. ReprErr uses VGGT-predicted depth and is therefore estimator-dependent.

H.5 Sensitivity to K_g and K_p

Table 12 reports a joint sensitivity sweep on the number of group-level bases K_g and the number of point-level bases K_p, decoupling K_g from the number of groups B (we fix $B = 15$). Performance is stable; we keep $(K_g, K_p) = (15, 2)$ as the default.

H.6 SDEdit Noise-Level Sensitivity

Table 7 shows that the 60% SDEdit prior raises reported held-out-view PSNR while increasingly changing the input video. The 0% row is the same-input comparison with SoM and differs by

$K_g \setminus K_p$	1	2	3
8	15.91	16.05	16.02
15	16.04	**16.23**	16.18
30	16.06	16.18	16.20

Table 12: $K_g \times K_p$ sweep on iPhone PSNR ($B = 15$ fixed).

only 0.02 dB, with no sequence-level uncertainty supplied. The 60% point maximizes the reported PSNR in this sweep, not fidelity to the original observations. Future comparisons should give all reconstruction methods identical raw and transformed videos.

H.7 Group-Assignment Purity and Failure Cases

We measure semantic-group purity as the fraction of Gaussians whose 2D projection across all frames falls into a single Track-Anything instance. The seven listed iPhone validation-sequence values are `apple` 96.4%, `block` 98.7%, `paper-windmill` 95.1%, `spin` 92.8%, `teddy` 82.1%, `haru-sit` 96.0%, and `wheel` 91.0%. Their arithmetic equal-sequence mean is 93.16% (93.2% rounded). A pooled Gaussian-weighted statistic would require each sequence's Gaussian count and is not reported. The lowest purity occurs on `teddy`, where a fast-moving fluffy boundary is occasionally split by Track-Anything across adjacent masks; we observe minor ghosting at the corresponding boundary.

Qualitative failure reports and structural risk. The latter two cases are reported qualitative artifacts; the first is a representation-level risk without a measured split-object stress test:

1. **Rapid topology change.** When an object splits into independently moving parts, fixed group assignments encourage them to share group motion; per-Gaussian residuals can still move existing points apart. The concern is optimization bias, residual regularization, and newly exposed surfaces rather than strict representational impossibility. This behavior was not measured on the 50 official rigid-object Movi-F scenes; a separate split-object stress test is needed.
2. **Large unobserved regions / strong parallax.** On DAVIS `breakdance-flare`, the camera moves rapidly while the dancer rotates; VGGT's depth becomes unreliable in the back-of-body region that is never directly observed, and the shared-geometry interface inherits this unreliability. VidLift4D's reconstruction shows ghosting around the back of the dancer in novel views.
3. **Mask flicker between adjacent instances.** On iPhone `teddy`, Track-Anything occasionally swaps left- and right-arm mask identities across frames. Canonical Gaussian group assignments remain fixed; the later mask targets can instead become inconsistent with those assignments and produce ghosting at the joint. This is the lowest-purity sequence and a qualitative iPhone failure.

The two reported artifacts are qualitative observations; paired renderings are needed to assess their frequency and severity. The topology case remains an unmeasured structural risk.

H.8 Depth-Evaluation Scope

No quantitative iPhone depth or flicker result is used as evidence here: the available protocol does not identify a coherent evaluation subset, depth reference, scale alignment, or camera-aware flicker computation. The qualitative depth renderings in this supplement illustrate an output type only.

H.9 Datasets

iPhone Gao et al. (2022). 14 monocular sequences, 200–500 frames, with held-out validation views. We use the official validation split (7 evaluation sequences).

DAVIS Perazzi et al. (2016). Real-world video clips, 30–100 frames. We use the 20 sequences of the 2017 `val` split for video-to-4D evaluation.

Kubric-derived protocol Greff et al. (2022). The supplied record describes 50 custom synthetic scenes with 1–6 dynamic rigid objects, 80-frame monocular training videos, and 8 held-out novel

views with ground-truth depth and per-object 3D trajectories. These settings differ from the released MOVi-F split. The scene IDs, random seeds, asset split, camera and object-motion scripts, held-out views, depth conversion, and raw predictions are absent, so the reported aggregate cannot be reproduced from the public builder alone.

User study. 25 participants, paired comparison, randomized order, 30 prompts per participant; each participant evaluates each prompt under three axes (dynamic realism, aesthetic, and overall realism). The original analysis reports two-sided Wilcoxon signed-rank tests at $p < 0.05$ and Krippendorff $\alpha = 0.71$; the pairing unit, exact p-values, and multiple-comparison correction require verification from raw responses. We treat these as descriptive observations, not inferential evidence of superiority over each baseline.

I User Study and Qualitative Ablation

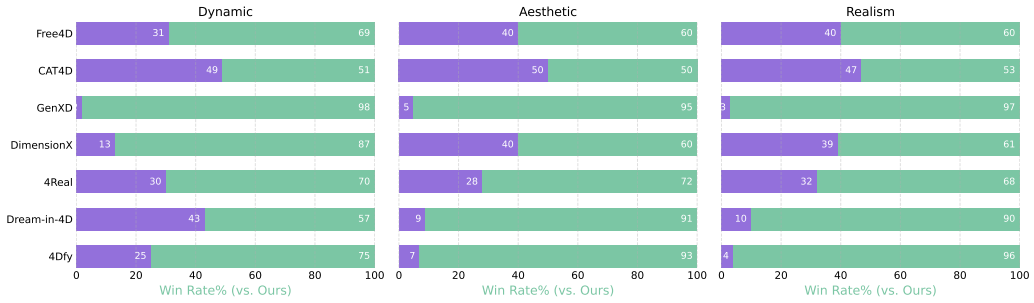


Figure 4: User-study win rates against each baseline along three axes (dynamic realism, aesthetic, and overall realism); error bars are 95% bootstrap intervals over 25 participants.

Figure 4 displays the reported participant preferences along three perceptual axes. Several displayed win rates are high, but the raw paired responses and multiplicity-adjusted confidence intervals are needed before drawing a statistical conclusion about each baseline. The figure measures subjective preference, not geometric accuracy or requested-camera adherence.

Figure 5 illustrates visible appearance and boundary differences among variants. The examples are diagnostic only; attributing temporal stability or geometric accuracy requires video-level or ground-truth measurements. The figure's "w/o shared geometry" label denotes removal of the VGGT frontend, not evidence that per-scene geometry was shared across stages.

J Additional Qualitative Results

We show additional text-to-4D and image-to-4D renderings under the same requested paths where supported, a gallery of varied prompts, qualitative depth and normal maps, and DAVIS trajectory visualizations. These figures illustrate output types and failure modes; they do not independently validate 3D accuracy or camera control.

J.1 Baseline Qualitative Comparisons

Figure 7 compares representative rendered frames under common requested trajectories where supported. The examples show background and texture differences; their static appearance does not determine which reconstruction has more faithful geometry. Requested-path adherence must be evaluated separately.

J.2 Diverse 4D Generation Results

The gallery covers text prompts about animals, objects, vehicles, and fantasy characters and shows renderings from the resulting 4D Gaussian scenes (Figures 8 to 13). The image-conditioned examples use one reference image with several requested camera paths (Figure 14). These examples show that the system produces renderable outputs under diverse conditions; they do not establish generalization rates, geometric correctness, or compliance with the specified paths.

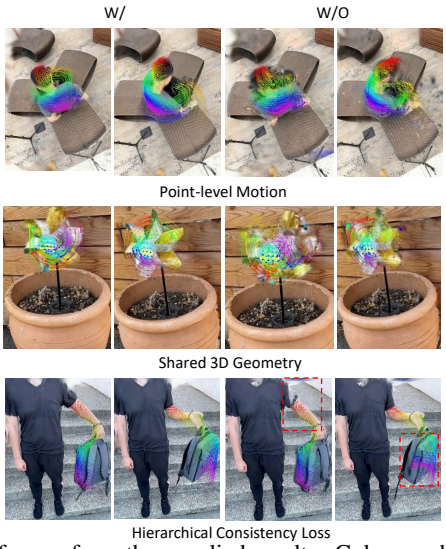


Figure 5: Illustrative ablation frames from the supplied results. Columns show the full model and variants labeled without GF, the VGGT frontend, point-level motion, and the residual-transform penalty. The still frames do not quantify temporal behavior.

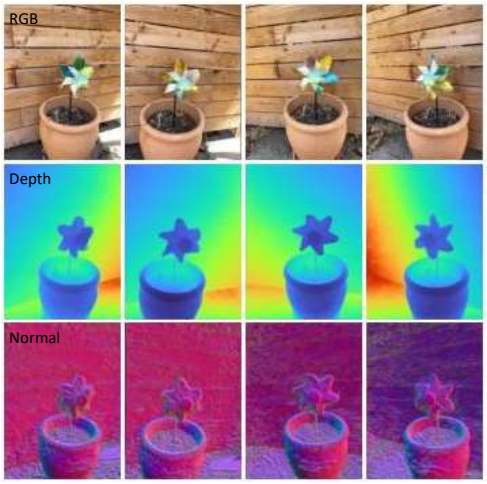


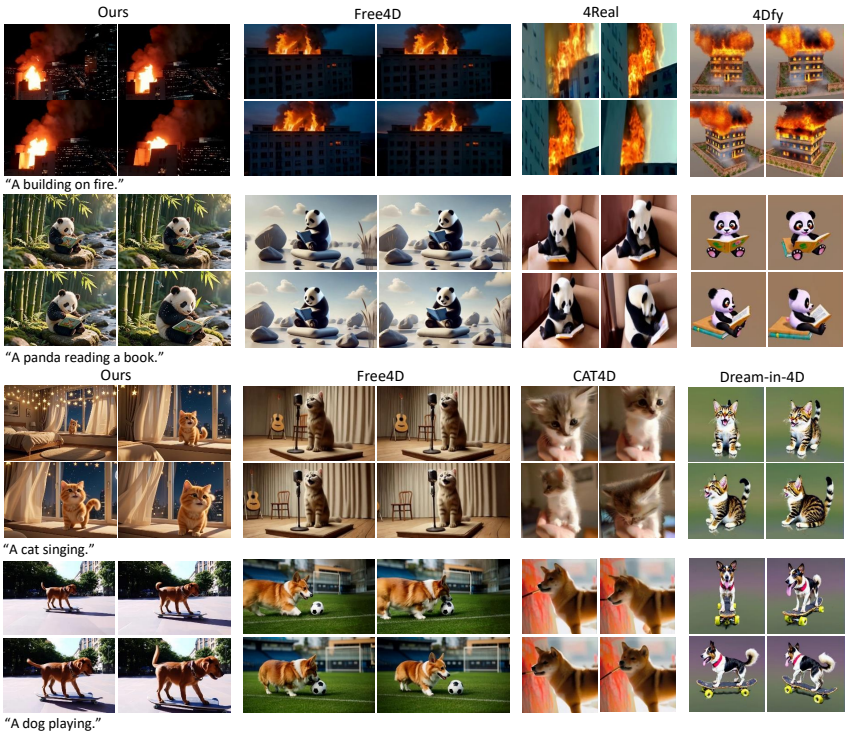
Figure 6: Qualitative examples of rendered depth and surface normals from an RGB video. No surface-normal ground truth is available for these frames.

J.3 Depth and Normal

Figure 6 shows RGB frames, rendered depth, and rendered normals for a flower-and-pot sequence. The visible maps appear smooth in these selected frames, but no normal ground truth or temporal error measurement is reported; downstream robotics or physical-interaction utility is therefore not evaluated.

J.4 3D Tracking on DAVIS

Figures 15 and 16 show 3D tracking visualizations on DAVIS Perazzi et al. (2016), including sequences such as bear, blackswan, breakdance-flare, bus, car-turn, drift-straight, elephant, flamingo, gold-fish, hike, horsejump-high, judo, motorbike, rhinoceros, soccerball, stroller, swing, and train. These DAVIS visualizations show trajectories produced by the optimized motion field for several scene types. DAVIS provides no 3D trajectory ground truth in this protocol, so these images cannot establish tracking accuracy, robustness, or generalization quantitatively. A reproducible evaluation needs documented queries, visibility, scale alignment, and per-sequence ground-truth errors.



(a) Text-to-4D qualitative comparison on three prompts under the same requested trajectory where supported. Rows show corresponding examples across methods. The frames illustrate background and appearance differences; 3D accuracy is not measured here.



(b) Image-to-4D qualitative comparison. The same single-image prompt is given to each method; we show held-out novel-view frames.

Figure 7: Text-/Image-to-4D qualitative comparison.

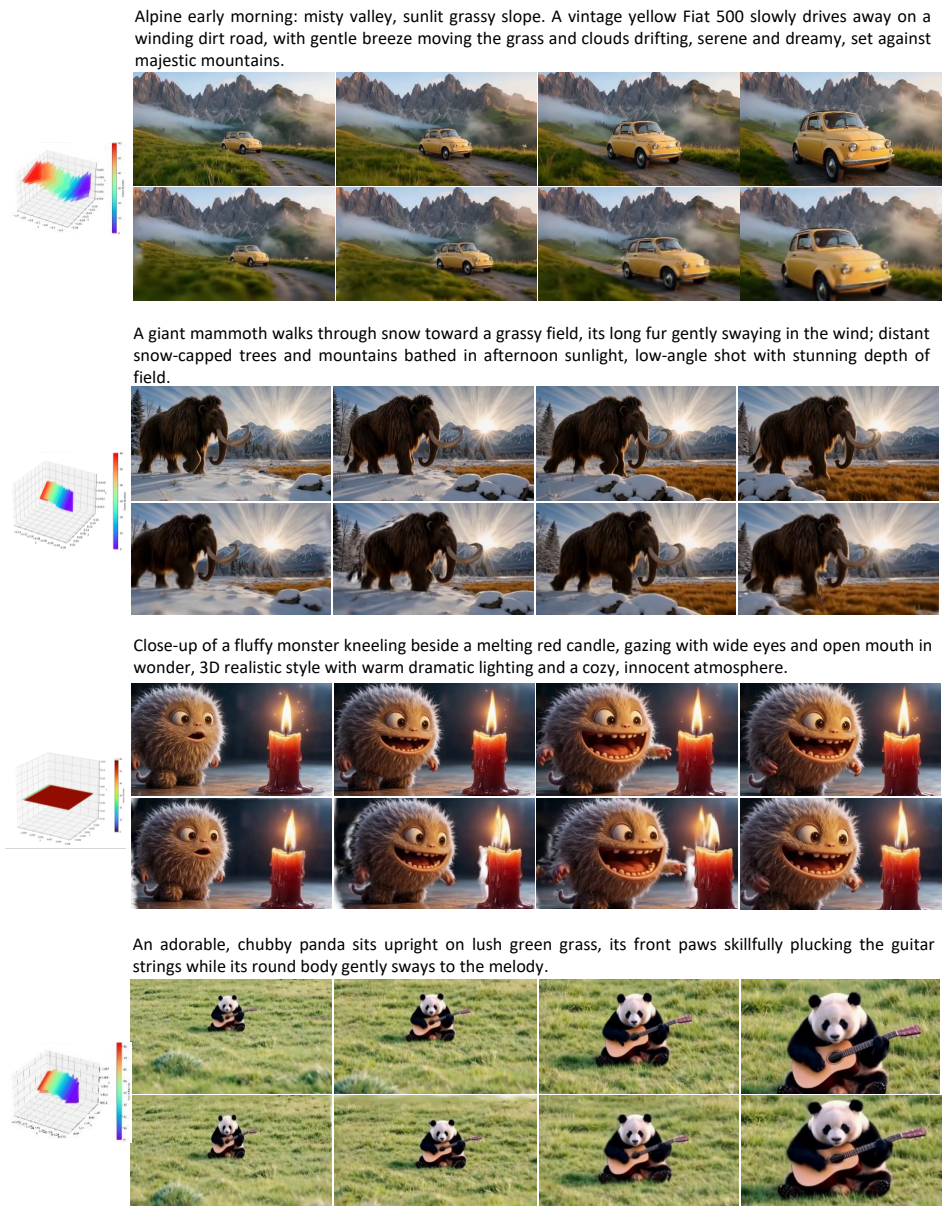


Figure 8: Text-to-4D examples under varied prompts and requested camera paths. Each row pairs a generated video frame with a rendering from the optimized 4D Gaussian scene. The figure illustrates appearance and renderability, not camera-path or 3D accuracy.



Figure 9: Diverse 4D generation results: generated videos and reconstructed scenes under identical trajectories and varying text prompts.

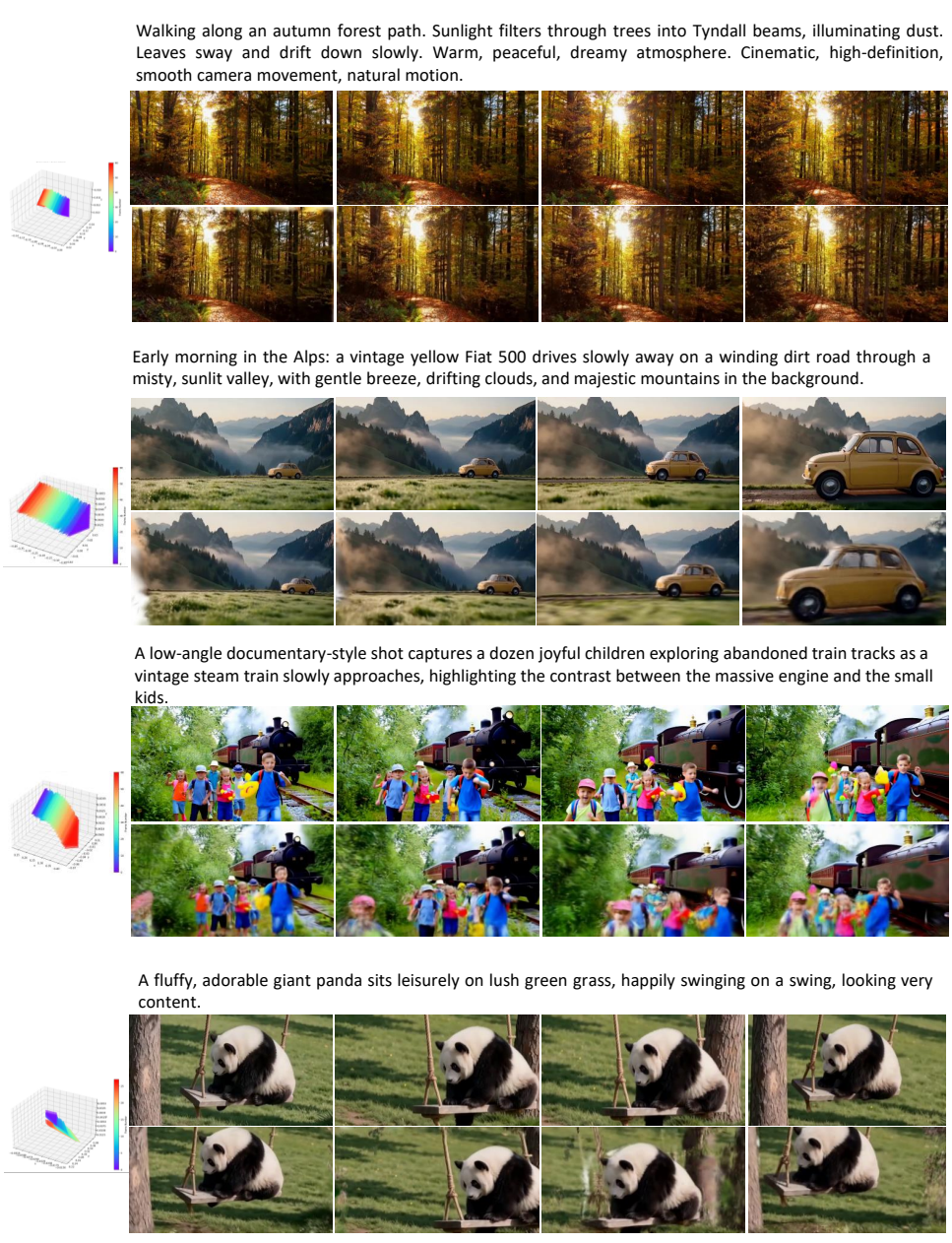


Figure 10: Diverse 4D generation results under different camera trajectories and text prompts.

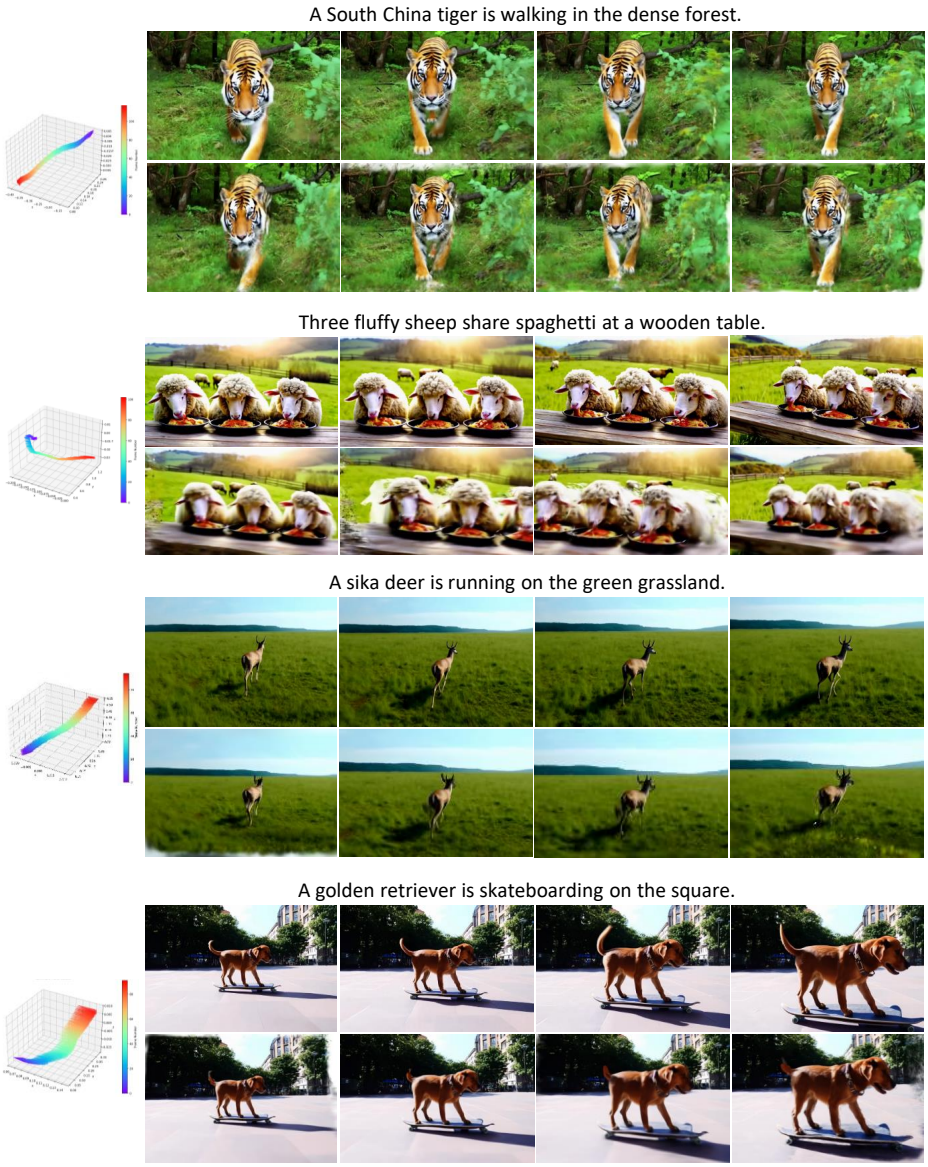


Figure 11: Diverse 4D generation results under different camera trajectories and text prompts.

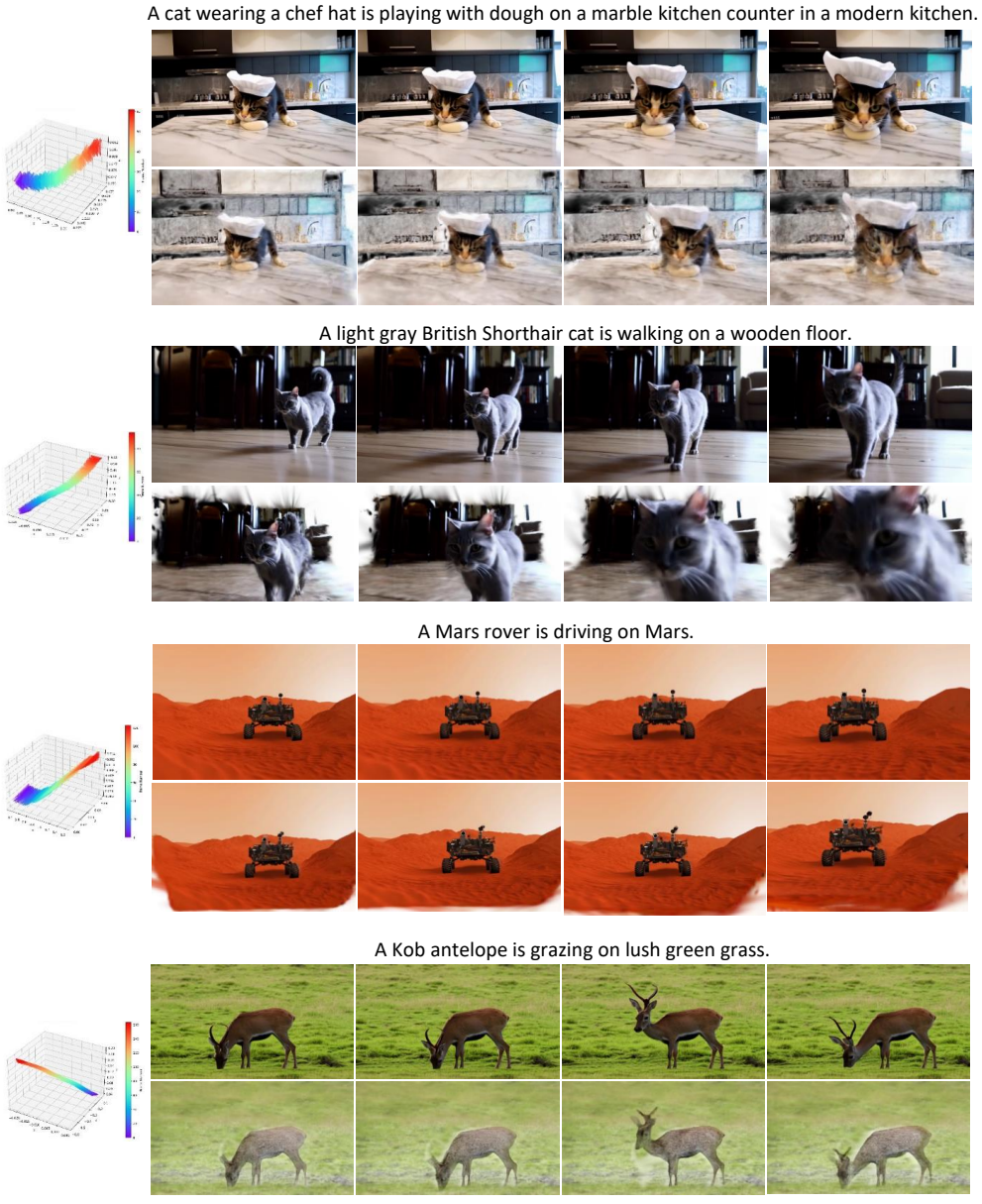


Figure 12: Diverse 4D generation results under different camera trajectories and text prompts.

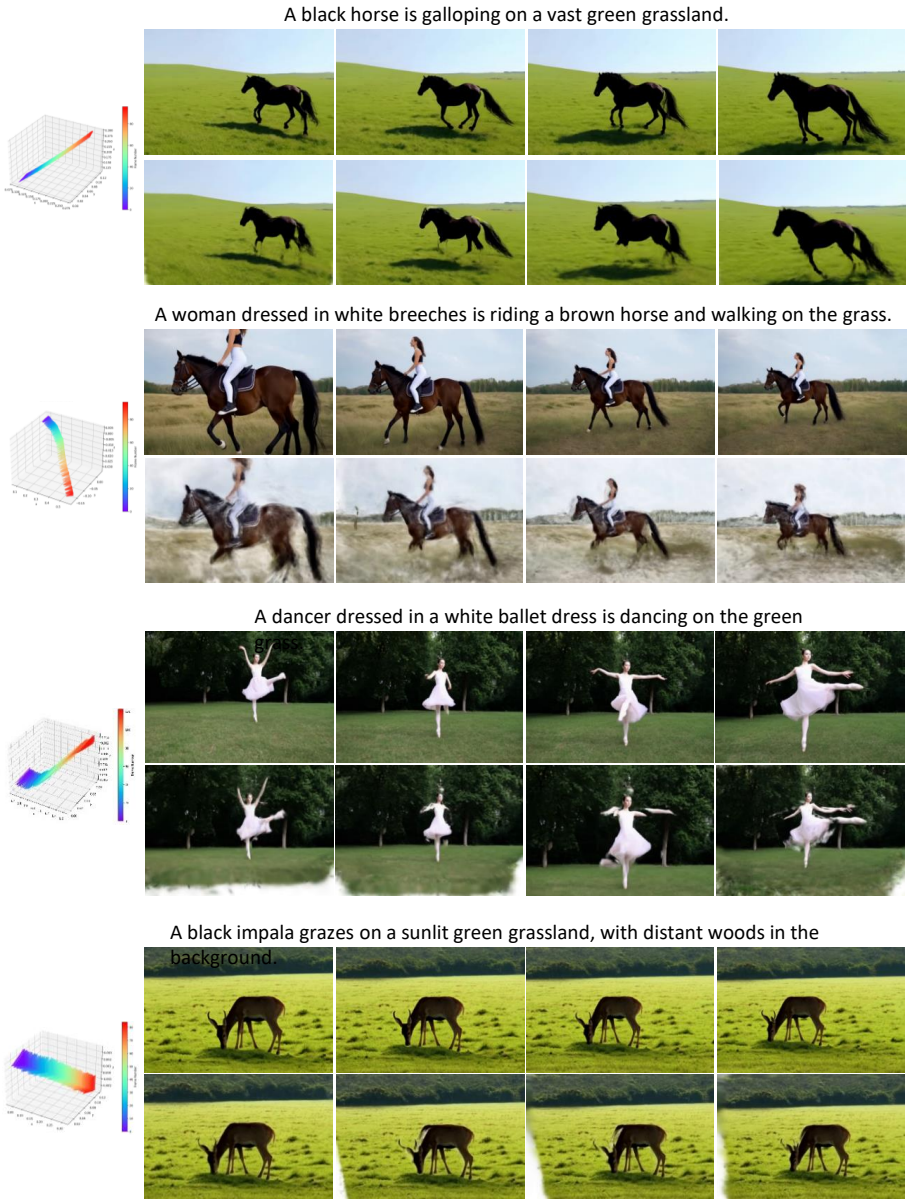


Figure 13: Diverse 4D generation results under different camera trajectories and text prompts.

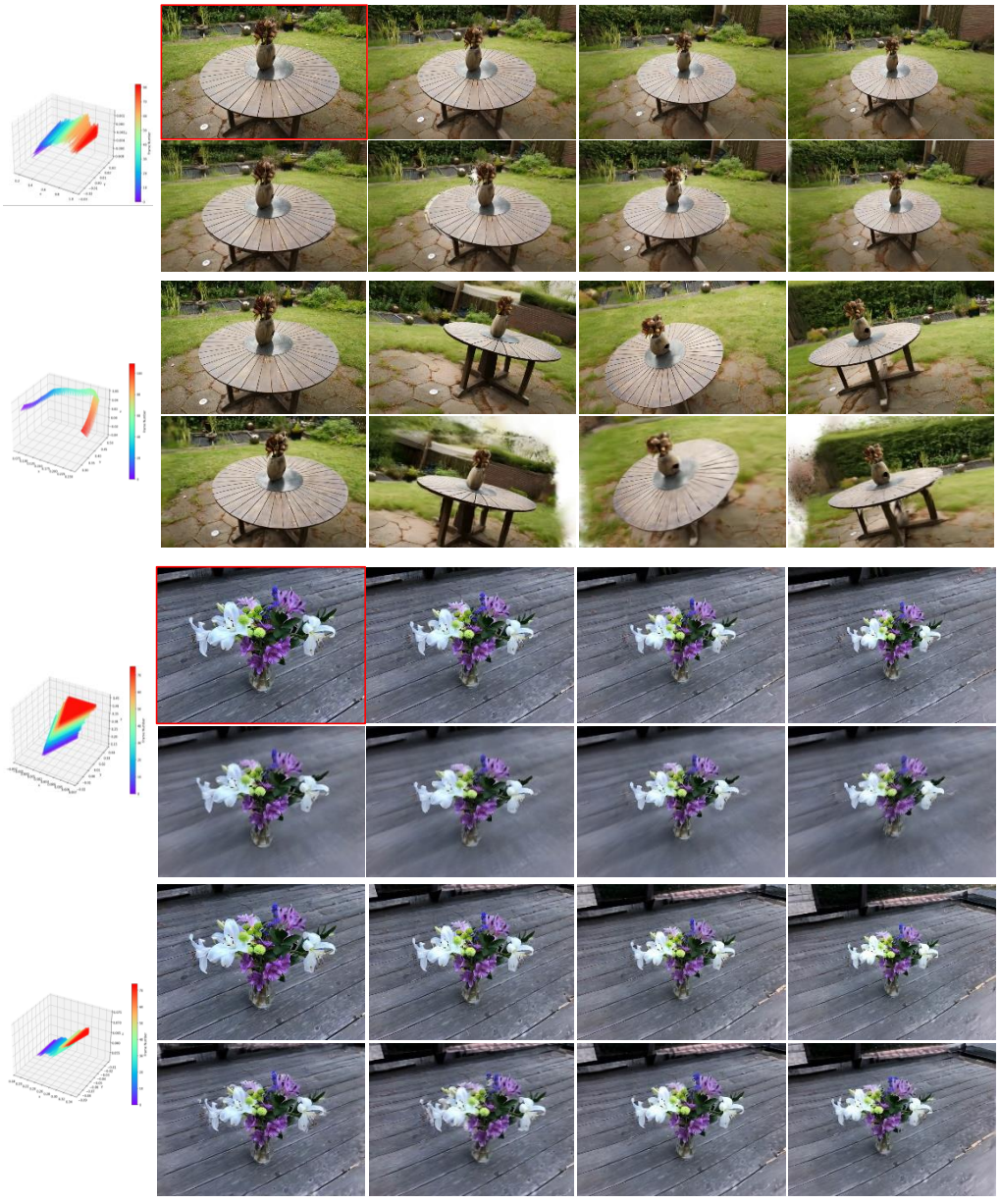


Figure 14: Image-to-4D: videos and reconstructed scenes from the same input image under different camera trajectories. Input image from SEVA Zhou et al. (2025).

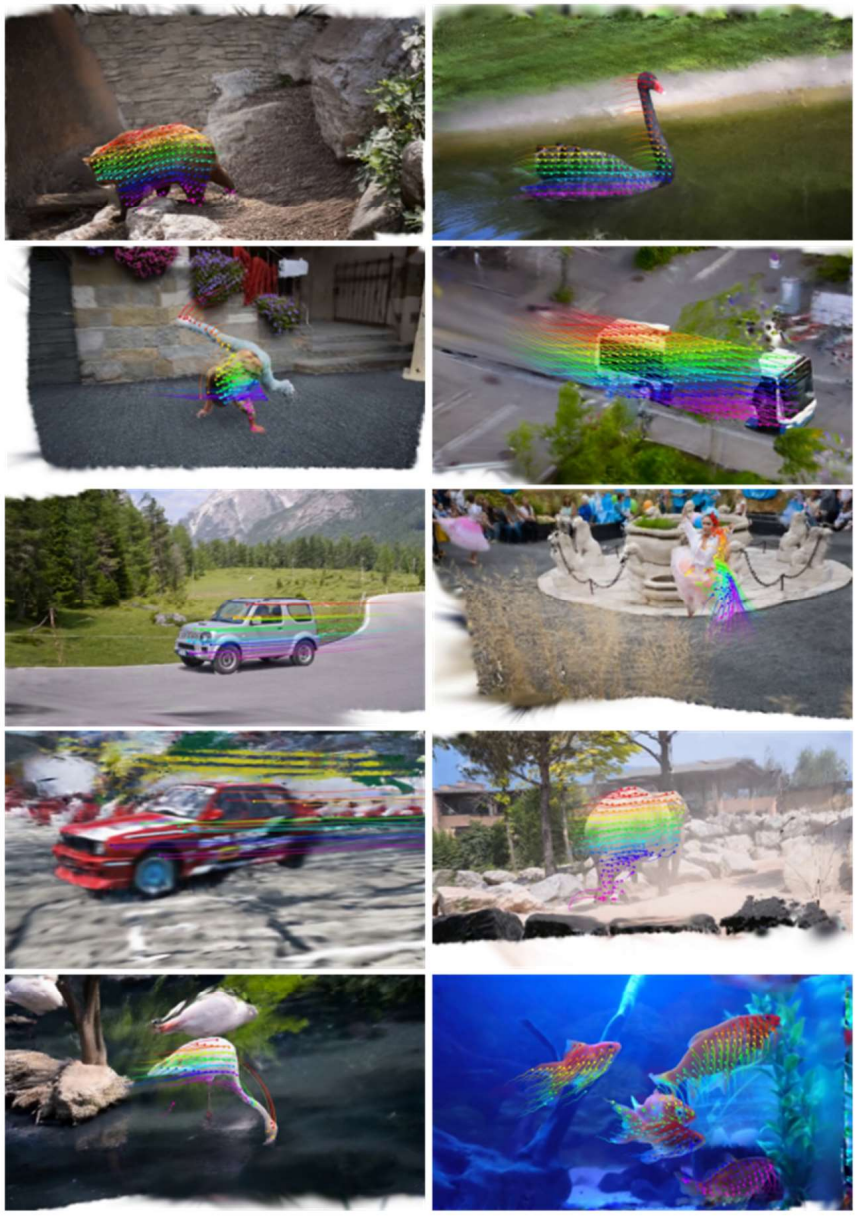


Figure 15: 3D tracking visualization on DAVIS Perazzi et al. (2016), in world coordinates.

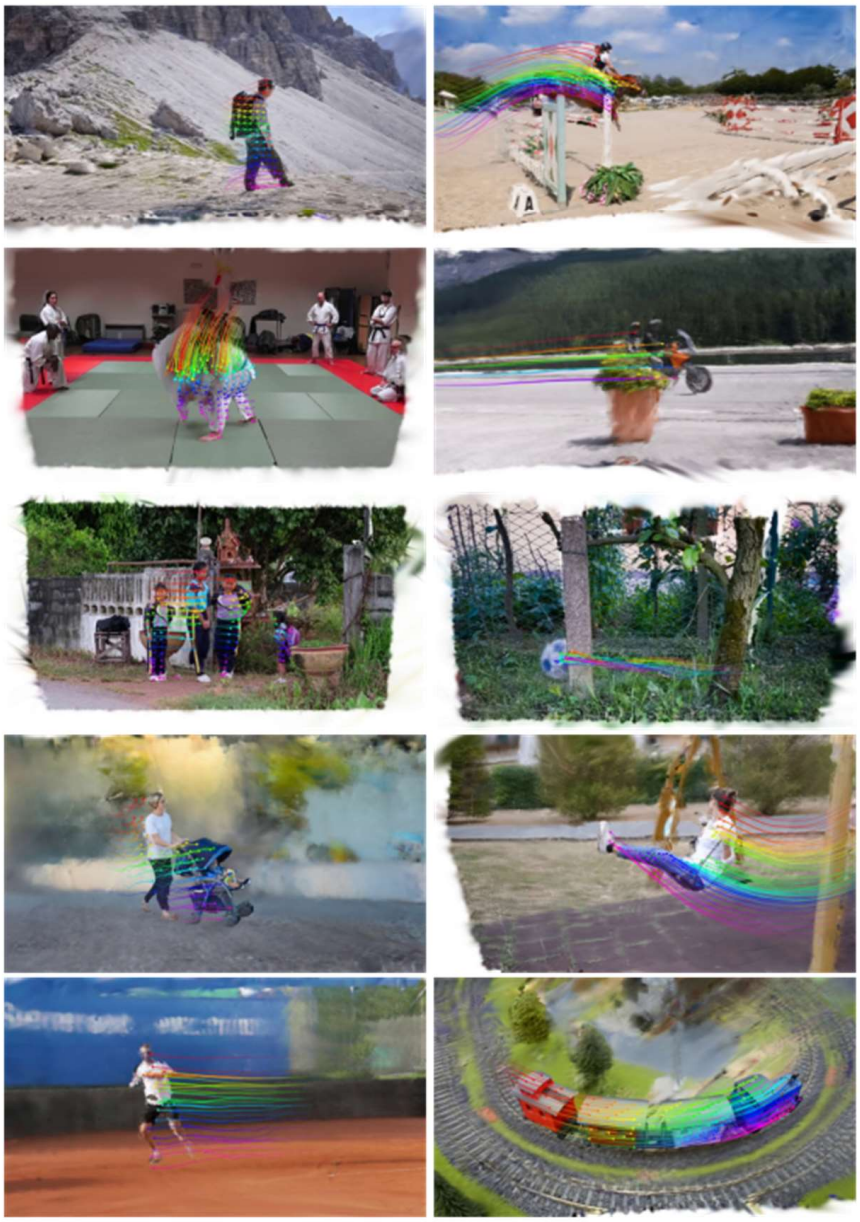


Figure 16: 3D tracking visualization on DAVIS Perazzi et al. (2016), in world coordinates.