World from Motion: Generative Dynamic Gaussian Reconstruction from Monocular Video

Liyuan Zhu[1,2] **Shengyu Huang**[2] **Amrita Mazumdar**[2] **Tianye Li**[2] **Zan Gojcic**[2]
Gordon Wetzstein[1] **Iro Armeni**[1] **Shalini De Mello**[2] **Alex Trevithick**[2]
[1]Stanford University [2]NVIDIA
https://research.nvidia.com/labs/amri/projects/world-from-motion/

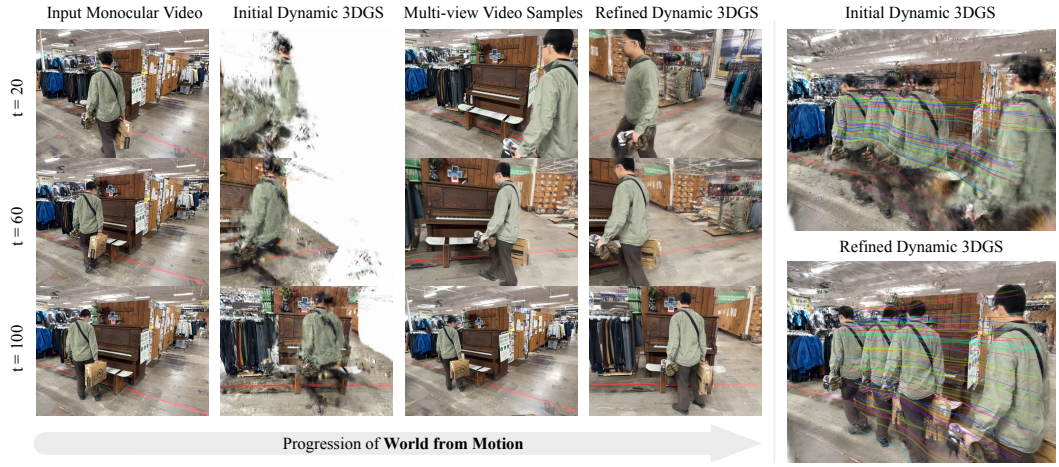

Figure 1: **World from Motion reconstructs a dynamic 3DGS world from the camera and scene motion in a single monocular video.** From an input video and an initial reconstruction produced by MoSca [25], our video model generates novel views that are distilled into a refined reconstruction. Our method faithfully recovers observed structure and synthesizes plausible novel-view dynamics, in turn improving the underlying scene motion, as visualized on the right.

Abstract

We present World from Motion, a method for generating freely renderable dynamic 3D Gaussian representations from monocular videos. Our approach conditions a video model on dense, pixel-aligned renderings that encode appearance, geometry, and 3D scene motion along both input and target camera trajectories to correct rendering artifacts and fill in missing regions from an initial reconstruction. To train this model, we construct a dataset of aligned multiview video pairs and dynamic 3DGS representations, with simulated artifacts characteristic of monocular reconstruction. At test time, we distill the model's generations, including newly observed regions and motions, back into a single consistent, high-quality dynamic 3DGS, improving both novel-view synthesis and the underlying 3D motion. Our method sets a new state of the art in PSNR, SSIM, and LPIPS on the DyCheck and NVIDIA-LS benchmarks and seamlessly generalizes to in-the-wild videos with large viewpoint changes and dynamic motions.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

The human visual system excels at inferring the 4D structure of the dynamic world from limited 2D observations. This internal "world model" combines parallax and stereo cues with learned priors, allowing humans to reason about precise geometry and resolve inherent ambiguities in shape, scale, and motion. As a computational realization of this capability, monocular 4D reconstruction enables important applications across robotic simulation, spatial perception, and immersive AR/VR. This process aims to reconstruct a model of the dynamic world's geometry, appearance, and motion for rendering at arbitrary viewpoints and times.

To replicate these human capabilities, classical 3D vision techniques [38, 47, 16] exploit geometric constraints that are highly effective under sufficient parallax and static scene assumptions. Notably, the outputs of these methods generally improve with more observations. Building on these techniques, the state-of-the-art 4D reconstruction methods [54, 25, 55] accurately recover well-observed static regions and motion estimates with dynamic 3D Gaussian splatting (3DGS) [24, 36]. However, they struggle to infer novel views and unobserved regions of dynamic scene elements.

Generative models offer the opposite tradeoff: they can resolve underconstrained regions by filling in missing appearance, geometry, and motion using learned priors. However, while state-of-the-art video generative models [2, 15] can synthesize high-fidelity frames, they often lack the multiview consistency required for precise 4D reconstruction, leading to inconsistent structure when lifted into 3D. Recent hybrid methods [39, 61] attempt to combine the strengths of both paradigms, but their results generally lack sufficient multiview consistency for precise dynamic reconstruction (see Figure 4). Consequently, existing frameworks are forced to choose between geometric rigor and generative expressivity, failing to capture the fluid yet persistent nature of the dynamic world. This motivates the central question of our work:

> *How can a monocular reconstruction method achieve the fidelity of geometry-driven methods where structure is well determined, while simultaneously deferring to generative priors where it is not—especially for scene dynamics?*

While recent work [32, 12, 59, 74] has made significant progress in leveraging generative priors to enhance *static* scene renderings, extending these successes to *dynamic* environments remains an open challenge. Dynamic world modeling is fundamentally more difficult, requiring separation of dynamic and static scene elements and estimation of dense 3D scene flow for the dynamic parts. Precise, generative 4D reconstruction therefore presents two unique challenges: (i) imbuing a generative model with precision and consistency across space-time; and (ii) lifting the generated samples into a unified, freely renderable 4D representation.

To address these challenges, we propose *World from Motion* (WfM), a 4D-consistent framework that unifies reconstruction and generative modeling. An homage to classical Structure from Motion (SfM) [47], WfM reconstructs a persistent and traversable snapshot of the world from both camera and scene motion. Prior methods [3, 7, 65] generate only short videos by conditioning on RGB or depth along the target trajectory. In contrast, WfM generates a high-quality 4D asset. We show how to condition a video model on dense, pixel-aligned renderings of a reconstructed dynamic 3DGS, including explicit 3D scene flow, and distill the generations into the underlying 4D representation. We choose dynamic 3DGS as the conditioning scaffold for two reasons. First, it compactly encodes priors across appearance, geometry, and 3D motion that can be rendered into screen space for dense, aligned video conditioning. Second, its explicit and differentiable form allows it to be consistently updated with novel observations and dynamics.

Put together, our conditional video model sets a new state of the art on the community-standard DyCheck [14] benchmark in perceptual quality without any further optimization (see Table 1). The dynamic 3DGS optimized using these samples provides even higher-quality results than the generations. In general, our method can handle in-the-wild videos, including large viewpoint changes and 3D motions (see Figure 1). Concretely, our contributions are as follows:

1. We show how to condition a video generative model on dynamic 3DGS through dense, pixel-aligned appearance, geometry, and 3D-motion buffers generated along both the input *and* target camera trajectories (Section 3.2).

2. We design an optimization procedure to distill the generated samples into a unified dynamic 3DGS representation (Section 3.3), which enhances motion quality (see Table 3) and improves with more generations (see Table 2).
3. We construct a carefully designed dataset of paired dynamic 3D Gaussian representations and ground-truth multiview videos, simulating the artifacts of monocular capture (Section 4).

2 Related Work

4D reconstruction from monocular video. Casual monocular videos usually capture the dynamic nature of the real world. Yet classical methods for camera pose and depth estimation typically assume a static capture [47, 38, 8]. Early non-rigid shape-from-motion methods relaxed this assumption using low-rank trajectory priors [5]. While recent systems improve robustness through learned optimization [51], long-term tracks [28], or dense point maps [19, 69, 56, 53, 30, 72, 23], inferring a unified representation including unobserved content remains challenging.

Beyond camera and depth estimation, monocular 4D reconstruction must infer geometry and scene flow in a unified frame. NeRF-based methods [34, 27, 71, 41, 44, 11, 40] briefly dominated dynamic view synthesis before dynamic 3DGS optimization-based methods [54, 25, 55, 48] became state-of-the-art. These methods leverage long-range motion supervision but struggle to infer unseen dynamics and content. CAT4D [61] and ViDAR [39] incorporate generative models into this optimization but require SDEdit-style [37] noising and denoising, resulting in inconsistent samples and blurry output. In parallel, feedforward approaches aim to predict 4D structure directly: 4DGT [64], MoVieS [29], and Lyra [1] directly predict dynamic 3D Gaussian parameters. While these feedforward methods improve speed, they retain a limited context window and therefore lag behind optimization-based reconstruction in geometric fidelity over longer videos.

3D-Conditioned Visual Generative Models. Early work in 3D reconstruction with diffusion priors regularized synthesis from text or sparse images [43, 57, 73, 49, 60, 32]. A line of image-space "fixers" seeks to enhance 3DGS via conditioning on imperfect renderings. Difix3D [59] repurposes a single-step image generator for rendering enhancement. FlowR [12] employs a multiview flow-matching model to fix renderings from multiple viewpoints. GaussFusion [74] proposes to condition video diffusion models on 3DGS for enhanced 3D reconstruction. However, although these methods perform well in static settings, they do not target the more challenging 4D regime.

A second paradigm, camera-controlled video generation, encodes pose information to synthesize sequences along specified trajectories [17, 2], with PlenopticDreamer [13] introducing camera-memory banks for long-term consistency. While visually plausible, these models typically prioritize 2D video quality over persistent 4D assets. Finally, recent work has sought to anchor generation in explicit 3D/4D representations [67, 68, 65]. Gen3C [45] conditions on a point-based representation; VMem [26] retrieves past observations through surfel-indexed memory and SPMem [62] combines 3D structure with 2D memory. Vivid4D [20] employs depth-guided inpainting. Concurrent work Vista4D [31] conditions on dynamic point clouds; however, due to this weak conditioning, its results fall short of the precision required for high-fidelity 4D reconstruction (see Figure 4). GS-DiT [3], CogNVS [7], and NeoVerse [65] condition video generators on RGB (and depth) renderings of a dynamic reconstruction along the target trajectory, but output per-trajectory videos without updating the underlying 4D representation. In contrast, our approach conditions a video generator on geometry, appearance, and explicit 3D scene-flow cues from dynamic 3DGS renderings along both reference and target views, and distills the generations back into that same representation. This process allows us to generate not only videos, but renderable 4D worlds with improved 3D motion.

3 Method

Overview. Given a monocular input video $\mathbf{V} = \{\mathbf{I}_t\}_{t=1}^{N}$ with $\mathbf{I}_t \in \mathbb{R}^{H \times W \times 3}$, our goal is to recover a dynamic 3DGS representation $\mathcal{G}$ that can produce novel-view renderings from arbitrary viewpoints at arbitrary times. Our approach begins by estimating an initial dynamic 3DGS representation $\mathcal{G}_0$ with an existing method (second part of Section 3.1). We then proceed in two stages. First, we show how to generate virtual observations of the evolving scene from fixed cameras by conditioning a video generator on the initial reconstruction's rendering (Section 3.2). Then, we show how to distill the generated, high-quality videos back into the 4D representation through re-optimization (Section 3.3).

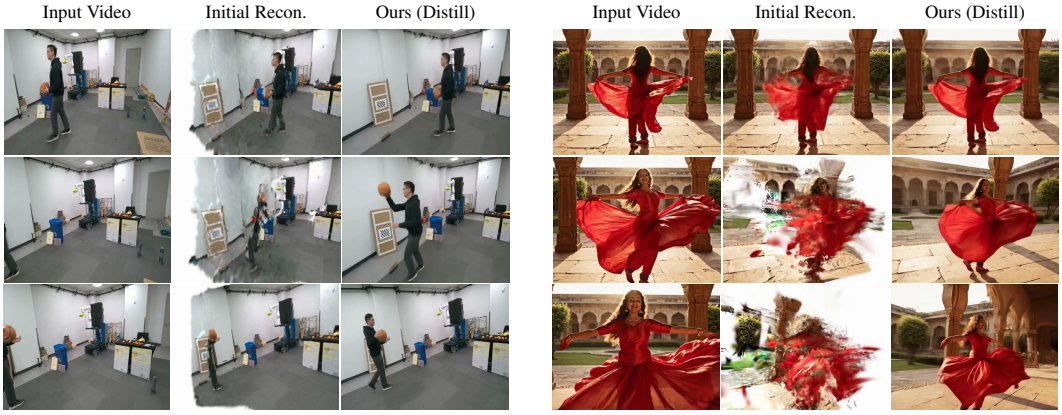

Figure 2: **Qualitative results on challenging in-the-wild videos**. Compared to the input reconstruction [25], our method performs visual outpainting, fixes degraded dynamics, and infers out-of-frustum dynamics while respecting the static region.

3.1 Preliminaries

Video generation model. Latent video diffusion models [4] generate videos in the latent space of a variational autoencoder (VAE). Given a video $\mathbf{V} = \{\mathbf{I}_t\}_{t=1}^N$, the VAE encoder $\mathcal{E}$ maps it to a latent tensor $\mathbf{z}_0 = \mathcal{E}(\mathbf{V})$, and a decoder maps sampled latents back to RGB frames. These models are commonly trained with flow matching [33]. Given Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and an interpolation $\mathbf{z}_\tau = (1-\tau)\boldsymbol{\epsilon} + \tau\mathbf{z}_0$ for $\tau \in [0, 1]$, the model learns a velocity field $\mathbf{v}_\theta$ that predicts the direction from noise to the data latent by minimizing $\mathcal{L}_{\text{fm}} = \|\mathbf{v}_\theta(\mathbf{z}_\tau, \tau; \mathbf{c}) - (\mathbf{z}_0 - \boldsymbol{\epsilon})\|_2^2$, conditioned on text, images, or other controls $\mathbf{c}$. Modern video generators usually adopt a Diffusion Transformer (DiT) backbone [42] to parameterize this velocity field.

Monocular 4D reconstruction. Given the input monocular video $\mathbf{V}$, we first obtain an initial dynamic 3DGS representation by running an off-the-shelf monocular 4D reconstruction method [54, 25, 55]. These methods start by estimating depth $\mathcal{D}$ and camera parameters $\mathcal{P}^{\text{in}} = \{(\mathbf{K}_t, \mathbf{T}_t)\}_{t=1}^N$, where $\mathbf{K}_t$ denotes intrinsics and $\mathbf{T}_t \in SE(3)$ denotes the camera pose [30, 28]. They also employ priors to predict dynamic masks [6, 25], and long-term tracks [9, 63]. The extracted camera poses and depth maps are then used to lift 2D tracks to 3D to initialize a low-rank motion representation, *e.g.*, motion bases [54], motion scaffolds [25], and temporal tree structures [55]. The dynamic scene $\mathcal{G}$ is represented by static and dynamic 3D Gaussians with M Gaussian primitives and the underlying motion representation that maps canonical Gaussian attributes to their state at time t.

At any given time t, the state of each Gaussian is evaluated as $g_i(t) = (\boldsymbol{\mu}_i(t), \mathbf{R}_i(t), \mathbf{s}_i, \alpha_i, \mathbf{c}_i)$. In this formulation, the 3D center $\boldsymbol{\mu}_i(t) \in \mathbb{R}^3$ and orientation $\mathbf{R}_i(t) \in SO(3)$ are time-dependent, mapped from canonical attributes to their current state by the motion model. The remaining properties including scale $\mathbf{s}_i \in \mathbb{R}^3$, opacity $\alpha_i \in \mathbb{R}$, and view-dependent color coefficients $\mathbf{c}_i$, are stored as constant, per-Gaussian attributes. Once evaluated at time t, the standard rasterization pipeline [24] projects each 3D Gaussian to a 2D splat and alpha-composites the visible splats in depth order from a camera viewpoint $(\mathbf{K}, \mathbf{T})$ to form the rendered image $\hat{\mathbf{I}}_t = \mathcal{R}_{\text{rgb}}(\mathcal{G}, t, \mathbf{K}, \mathbf{T})$. Along with the photometric loss, the extracted priors supervise a depth loss $\mathcal{L}_{\text{depth}}$ and a 3D track loss $\mathcal{L}_{\text{track}}$, and an as-rigid-as-possible term $\mathcal{L}_{\text{arap}}$ regularizes the motion. We refer readers to [54, 25, 55] for details.

Off-the-shelf methods provide an initial dynamic 3DGS $\mathcal{G}_0$ which exhibits floaters, missing regions, blurry texture, and incorrect dynamics due to heavy regularization to compensate for the ill-posedness of the problem. Nevertheless, it still provides a persistent 4D scene scaffold that can be rendered from novel camera trajectories. We now describe how to repurpose a video generator to condition on this 4D scene representation for view synthesis across space and time.

3.2 Dynamic 3DGS-conditioned Target Video Generation

We choose to condition our video generator on pixel-aligned multi-modal renderings (4D buffers) of a dynamic 3DGS rather than on point clouds [45, 31] or posed images [2, 67] because dynamic 3DGS

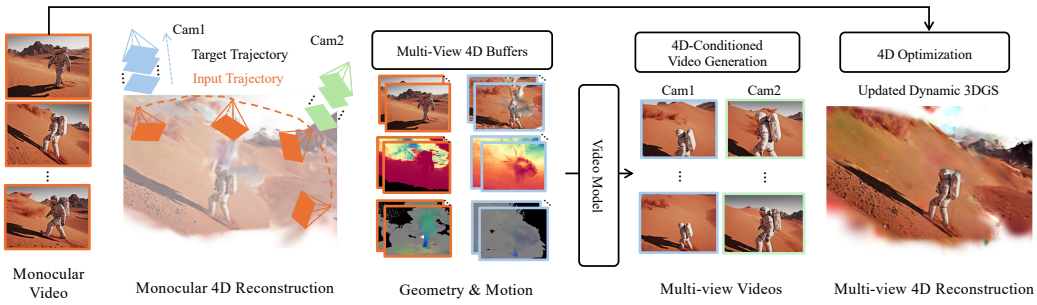

Figure 3: **Overview of 4D reconstruction with World from Motion**. Given a monocular video, our method first generates an initial 4D reconstruction of the input video. Along the target trajectory, our method then renders the corresponding appearance, geometry and motion to condition a video generative model. The generated samples are used to create a higher-quality 4D reconstruction.

provides a persistent, continuous, and differentiable 4D scene scaffold. While recent works [74, 12, 59] have demonstrated the effectiveness of conditioning generative models on 3DGS renderings for static 3D reconstruction, extending this paradigm to dynamic 4D settings is nontrivial. Monocular 4D reconstruction is inherently more ambiguous, making rendered buffers from novel target views less reliable, especially for dynamic regions.

Rendering 4D buffers. Due to the ambiguities of 4D reconstruction, we condition our video model on as much scene information as possible. Prior work in the static regime conditions video generative models only on information from the target trajectory [74], which can fail to anchor generated content to the observed input. To address this, we explicitly condition on 4D buffers rendered along both the input trajectory $\mathcal{P}^{\text{in}}$ and the target trajectory $\mathcal{P}^{\text{tgt}}$, providing stronger and more stable guidance for dynamic scene synthesis.

Concretely, given the initial representation $\mathcal{G}_0$, we render per-frame conditioning signals for any camera viewpoint $(\mathbf{K}_t, \mathbf{T}_t)$ at time t by evaluating the dynamic Gaussian state $\mathcal{G}_0(t)$. Standard rasterization produces the RGB image $\hat{\mathbf{I}}_t$ along with spatial G-buffers including opacity $\mathbf{A}_t$, depth $\mathbf{D}_t$, and surface normals $\mathbf{N}_t$, which provide dense geometric and appearance cues. To further encode scene dynamics, we augment these signals with 3D scene flow defined in the global coordinate system of $\mathcal{G}_0$. Specifically, we compute per-Gaussian displacements $\mathbf{u}_i(t) = \boldsymbol{\mu}_i(t+1) - \boldsymbol{\mu}_i(t)$ and rasterize them into a per-pixel flow map $\mathbf{F}_t = \mathcal{R}_{\text{flow}}(\mathcal{G}_0, t, \mathbf{K}_t, \mathbf{T}_t)$. Each pixel thus stores a full 3D displacement in world coordinates rather than a 2D optical flow. We aggregate these signals into a per-frame conditioning bundle $\mathbf{b}_t = [\hat{\mathbf{I}}_t, \mathbf{A}_t, \mathbf{D}_t, \mathbf{N}_t, \mathbf{F}_t]$, and render sequences $\mathbf{B}^{\text{in}} = \{\mathbf{b}_t^{\text{in}}\}_{t=1}^N$ and $\mathbf{B}^{\text{tgt}} = \{\mathbf{b}_t^{\text{tgt}}\}_{t=1}^N$ along the input and target trajectories, respectively.

Model architecture. Our goal is to enable the generator to preserve reliable regions of the initial 4D reconstruction while correcting its failure modes, including inaccurate dynamics and missing content, in a temporally and multiview-consistent manner. Given the buffer sequences $\mathbf{B}^{\text{in}}$ and $\mathbf{B}^{\text{tgt}}$, we first encode each modality independently using the video VAE encoder $\mathcal{E}$. The encoded modalities are concatenated along the channel dimension to form per-frame 4D tokens, and then concatenated along the temporal dimension across input and target trajectories, yielding the full conditioning sequence $\mathbf{z}_{\mathbf{B}}$. This sequence is processed by a VACE-style [21] DiT adapter (the *4D buffer adapter*), where self-attention over the 4D tokens produces residual updates that are injected into each layer of the base DiT. The base DiT takes in the encoded input and target videos separately, applies noise only to the target latents, and concatenates clean input latents with noisy target latents along the temporal dimension before the DiT blocks. In addition to the 4D buffer adapter, we incorporate explicit camera conditioning by encoding each camera pose $\tilde{\mathbf{T}}$ into a frame-level embedding, which is broadcast to all spatial tokens before self-attention, providing geometric guidance for both input and target sequences.

We train the model using the flow-matching loss defined in Section 3.1, applied only to the target latents. Figure S5 in the supplement illustrates the full architecture. Since the conditioning is derived from rendered 4D buffers rather than representation-specific parameters, the model naturally generalizes across different underlying 4D representations.

Reconstruction guidance. We observe that 4D buffer-based guidance improves alignment between generated samples and the underlying 4D scene representation. To enable this, we introduce a classifier-free-style dropout strategy on the 4D buffer tokens. During training, we randomly replace the encoded 4D-buffer tokens $\mathbf{z_B}$ (for both input and target trajectories) with a learned null embedding $\bar{\mathbf{z}}_\mathbf{B}$. Importantly, the base conditioning from the input video and camera poses is always retained, so the resulting "unconditional" branch corresponds to a ReCamMaster-like model [2] without 4D buffer guidance.

At inference time, we evaluate the model twice, with and without the 4D buffer condition, producing velocities $\mathbf{v}_{\theta,c}$ and $\mathbf{v}_{\theta,u}$, respectively. We then apply a guidance update analogous to classifier-free guidance to steer the generation toward the reconstruction-conditioned prediction:

$$\mathbf{v}_{\theta,\text{geo}} = \mathbf{v}_{\theta,u} + \gamma(\tau)\big(\mathbf{v}_{\theta,c} - \mathbf{v}_{\theta,u}\big), \tag{1}$$

where $\gamma(\tau)$ is a timestep-dependent guidance scale. In practice, we also support more general guidance operators such as APG [46]. This guidance improves geometric consistency by explicitly controlling the influence of the 4D reconstruction prior during generation.

3.3 Refining the 4D Scene Representation

Our goal is to leverage the trained video generator to synthesize novel observations of the 4D scene $\mathcal{G}_0$ and use them to refine the underlying representation. We sample a set of target viewpoints and generate temporally consistent videos from these cameras. These synthesized observations are then used together with the input video to re-optimize $\mathcal{G}_0$.

Sampling virtual cameras. To ensure good spatial coverage, we select J target viewpoints via farthest-point sampling over the input camera trajectory and keep them fixed over time. For each sampled trajectory $\mathcal{P}^{\text{tgt},j}$, we generate a video $\mathbf{V}^{\text{tgt},j} = \{\mathbf{I}_t^{\text{tgt},j}\}_{t=1}^N$ using the procedure described in Section 3.2. Virtual cameras are sampled from the input trajectory only; test trajectories are never used for conditioning or supervision. For long sequences, generation is performed in a temporal sliding-window manner.

We empirically find that this simple sampling strategy is sufficient to produce target videos with sufficient spatial coverage and strong spatio-temporal consistency. This is enabled by our 4D-conditioned generator, which maintains coherence across viewpoints without requiring complex cross-view coordination. In contrast, prior work such as CAT4D [61] relies on more involved multi-view and temporal sampling procedures to achieve consistent generation.

Reoptimization. We refine the initial dynamic 3D Gaussian representation $\mathcal{G}_0$ by re-optimizing the initialized model with supervision from both the input video $\mathbf{V}$ along $\mathcal{P}^{\text{in}}$ and the generated videos $\{\mathbf{V}^{tgt,j}\}_{j=1}^J$ along $\{\mathcal{P}^{tgt,j}\}_{j=1}^J$. During initialization, we jointly estimate the depth of input and generated videos together with DAv3 [30], then backproject the depth maps of the generated views to initialize additional Gaussians to fill in newly-generated content in the scene. We jointly optimize the Gaussian variables, including 3D centers, rotations, scales, opacities, and motion parameters. We find re-optimization of the motion representation critical because it allows our synthesized samples to correct potentially inaccurate estimates from the initial 3D tracks of the input view. Overall, we optimize the following loss:

$$\mathcal{L}_{\text{reopt}} = \mathcal{L}_{\text{in}} + \lambda_{\text{tgt}}\mathcal{L}_{\text{tgt}} + \lambda_{\text{motion}}\mathcal{L}_{\text{motion}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{track}}\mathcal{L}_{\text{track}} + \lambda_{\text{arap}}\mathcal{L}_{\text{arap}} \tag{2}$$

where each $\lambda \in \mathbb{R}$ controls the weight of its respective loss component. $\mathcal{L}_{\text{depth}}$ and $\mathcal{L}_{\text{track}}$ are supervised by the estimated depth and 3D tracks extracted in the initial reconstruction (Section 3.1). The input term $\mathcal{L}_{\text{in}}$ keeps the representation faithful to the observed input video, while the generated-view term $\mathcal{L}_{\text{tgt}}$ uses generated videos as additional supervision. Specifically, $\mathcal{L}_{\text{in}}$ sums a per-frame appearance loss ℓ_{app} between renderings from $\mathcal{G}_0$ and the input video $\mathbf{V}$ along $\mathcal{P}^{\text{in}}$, and $\mathcal{L}_{\text{tgt}}$ sums ℓ_{app} over all generated videos $\{\mathbf{V}^{tgt,j}\}_{j=1}^J$ along their sampled camera trajectories $\{\mathcal{P}^{tgt,j}\}_{j=1}^J$. This appearance loss consists of photometric and perceptual terms:

$$\ell_{\text{app}}(\hat{\mathbf{I}}, \mathbf{I}) = \|\hat{\mathbf{I}} - \mathbf{I}\|_1 + \lambda_{\text{ssim}}\mathcal{L}_{\text{SSIM}}(\hat{\mathbf{I}}, \mathbf{I}) + \lambda_{\text{lpips}}\mathcal{L}_{\text{LPIPS}}(\hat{\mathbf{I}}, \mathbf{I}). \tag{3}$$

$\mathcal{L}_{\text{in}}$ and $\mathcal{L}_{\text{tgt}}$ supervise both the underlying motion representation and the appearance parameters of the dynamic Gaussians, while $\mathcal{L}_{\text{arap}}$ is defined in [25].

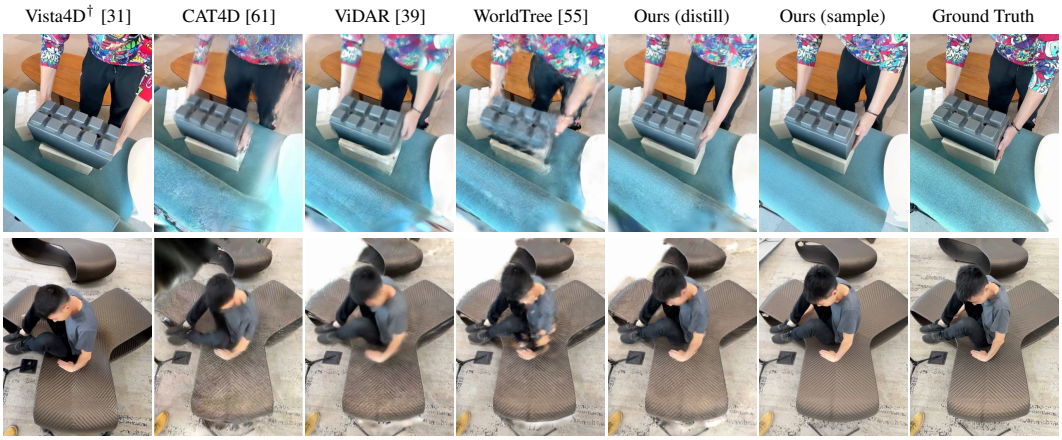

Figure 4: **Qualitative comparison on DyCheck [14].** Each row compares the predicted test view from each method with the ground-truth frame. Our method produces sharper details and consistency.

4 Generating Training Data from Multiview Videos

Training data. We train our model on synthetic multiview dynamic scenes from the MultiCamVideo dataset [2], which provides posed and time-synchronized multiview videos. For each scene, we randomly select one video as input and another as target to form training pairs. We reconstruct an initial dynamic 3DGS representation from the input video using an off-the-shelf monocular 4D reconstruction method, such as Shape of Motion [54] or WorldTree [55], and render it along both input and target trajectories to obtain the 4D conditioning buffers.

Joint depth alignment. Geometric alignment between rendered conditioning and ground-truth video is critical; scale ambiguity in monocular reconstruction often leads to inconsistencies that cause the generator to ignore the conditioning signal. We resolve this by enforcing global scale consistency through joint depth prediction across views. Specifically, we utilize anchor frames with large baselines to provide multi-view constraints, coupling otherwise independent monocular estimates within each prediction window. By aligning the resulting trajectory to the ground-truth and propagating the recovered metric scale via overlapping frames, we produce consistently scaled depth that ensures the conditioning remains spatially compatible with the target video.

5 Experiments

Baseline methods. We compare against non-generative, generative, and hybrid approaches for 4D reconstruction. Non-generative baselines include Shape of Motion [54], MoSca [25], and WorldTree [55]. Hybrid methods include CAT4D [61] and ViDAR [39]. We further consider generative approaches that do not optimize an explicit dynamic 3DGS representation, including ReCamMaster [2] and concurrent work Vista4D [31]. For our method, we report the sampled videos (*Ours (sample)*) and distilled dynamic 3DGS representation (*Ours (distill)*). *Ours (sample)* queries the generator directly on the test trajectory to benchmark generation against other generative methods on equal footing; these samples play no role in distillation. *Ours (distill)* is supervised only by the input video and by virtual cameras sampled from the input trajectory, so test camera trajectories are never used when generating virtual views, and all distilled results are evaluated on held-out views.

Evaluation datasets. We evaluate our method on three benchmarks. First, we report results on the *DyCheck* benchmark [14], a standard dataset for dynamic scene reconstruction. Second, we use a held-out validation set from *MultiCamVideo* [2], consisting of 5 synthetic multiview videos of dynamic human scenes. Third, we evaluate on the real-world NVIDIA-LS dataset [66] following the protocol of WorldTree [55]. In addition, we present qualitative results on challenging in-the-wild videos, including both real-world footage and videos synthesized by a generative model [15].

DyCheck benchmark [14]. We present quantitative results in Table 1 using two reconstruction backbones, WorldTree [55] and MoSca [25]. The DyCheck benchmark assumes static test cameras over time. However, several prior works [25, 55, 39] perform per-frame pose re-optimization during

Table 1: **Quantitative results on the DyCheck dataset.** We evaluate under the static-camera protocol using three regions (*Full, Valid, Covisible*), and additionally report results with per-frame pose optimization for comparison with prior work. All metrics are computed over the corresponding evaluation masks, except for the full (unmasked) evaluation.

	Static-camera evaluation									Per-frame pose opt.		
	Full			Valid			Covisible			Covisible		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓
Shape-of-Motion [54]	16.89	0.477	0.340	17.27	0.569	0.286	17.32	0.598	0.296	–	–	–
MoSca (MS) [25]	16.60	0.553	0.362	17.76	0.644	0.303	18.69	0.696	0.272	19.32	0.706	0.264
WorldTree (WT) [55]	17.08	0.574	0.332	18.38	0.662	0.278	19.28	0.714	0.246	19.75	0.728	0.240
ViDAR [39]	17.22	0.570	0.315	18.64	0.661	0.254	19.44	0.711	0.224	19.69	0.713	0.223
MS + Ours (sample)	18.64	0.556	**0.257**	19.01	0.646	0.229	19.45	0.694	0.214	–	–	–
MS + Ours (distill)	17.30	0.566	0.268	18.73	0.657	**0.206**	19.63	0.707	**0.184**	20.26	0.725	**0.176**
WT + Ours (sample)	**18.79**	0.563	0.259	**19.17**	0.652	0.230	19.63	0.701	0.214	–	–	–
WT + Ours (distill)	17.57	**0.583**	0.271	19.10	**0.672**	0.211	**20.06**	**0.721**	0.187	**20.38**	**0.730**	0.184

Table 2: **3DGS refinement with varying numbers of virtual cameras.**

# virtual cams	0	1	2	4	8	16
mPSNR ↑	18.69	19.29	19.39	19.49	**19.78**	19.63

Table 3: **DyCheck 3D track evaluation**. PCK@0.05 ↑ is reported.

CoTracker [22]	BootsTAPIR [10]	MoSca [25]	Ours
0.803	0.779	0.824	**0.862**

evaluation, which alters the camera trajectory and deviates from this assumption. To ensure a fair comparison, we recalibrate these methods to use fixed camera poses across time. For completeness, we also report results under the original protocol with per-frame pose optimization (see Section S1 for more analysis). Under the static-camera setting, we evaluate performance over three regions: (1) *covisible*, which includes only pixels visible in both reference and test views at each timestep [14]; (2) *valid*, defined as the union of covisible regions across all timesteps; and (3) the *full* image.

Across all settings, our method consistently outperforms prior approaches. The video generator alone achieves higher PSNR and lower LPIPS than all baselines, including when applied to out-of-distribution reconstructions from MoSca. After distillation, the reconstructed 4D representation further improves accuracy, yielding the best performance across all metrics and regions. We further compare against generative approaches in Table 4, where our method (sample) outperforms all baselines, including concurrent Vista4D [31], demonstrating the benefit of conditioning on a structured and geometrically aligned 4D representation (*i.e.* WorldTree). As seen in Table 3, compared to the input MoSca [25], the photometric constraints from our samples improve the underlying 3D tracks. Under the protocols of CogNVS [7] and NeoVerse [65], which differ from Table 1 in resolution, masks, and pose handling, our method outperforms both by large margins on all metrics (Table 5).

Qualitative comparisons in Figure 4 show that our method produces sharper, more coherent results with improved geometric alignment. Compared to generative methods such as Vista4D [31] and CAT4D [61], our results better preserve structure while reducing temporal inconsistencies and geometric drift. Compared to reconstruction-based methods such as WorldTree [55] and ViDAR [39], our method significantly improves fidelity, reducing blurring and high-frequency noise.

NVIDIA-LS [66] and MultiCamVideo [2] benchmarks. On the real-world NVIDIA-LS dataset, we sample 4 virtual cameras and outperform all baselines on all metrics (Table 7). On MultiCamVideo, Table 6 presents the quantitative comparison with baselines, where our method achieves the best performance across all metrics. The results show that our framework effectively recovers disoccluded regions and stabilizes ambiguous motion in monocular inputs. We refer readers to the supplement and video for qualitative comparisons.

In-the-wild data. We finally show qualitative results of our method on challenging in-the-wild real videos and those generated by an external method [15]. Figure 2 shows our method compared to the input reconstruction [25]. The first sample is taken from a dynamic SLAM dataset WildGS [70] and the second example is a generated video by [15]. Notably, our model can successfully perform visual outpainting, fix highly-degraded dynamics, and even infer out-of-frustum dynamics while respecting the reconstruction of the static region. Please see the supplement for more video results.

Table 4: **Comparison against generative methods on DyCheck.**

Method	mPSNR ↑	mSSIM ↑	mLPIPS ↓
CAT4D [61]	18.24	0.666	0.227
Ours	**19.89**	**0.715**	**0.197**
ReCamMaster [2]	10.96	0.262	0.755
TrajectoryCrafter [67]	13.06	0.320	0.656
GEN3C [45]	12.06	0.260	0.679
Vista4D [31]	14.14	0.310	0.514
Ours (eval at full-res)	**18.45**	**0.635**	**0.362**

Table 5: **Comparison under the DyCheck protocols of CogNVS [7] (top) and NeoVerse [65] (bottom).** †As reported in [65].

Method	PSNR ↑	SSIM ↑	LPIPS ↓
TrajectoryCrafter [67]	12.94	0.279	0.590
CogNVS [7]	15.36	0.356	0.426
Shape-of-Motion [54]	15.38	0.483	0.396
MoSca [25]	16.28	0.495	0.413
Ours	**17.69**	**0.556**	**0.292**
MonST3R† [69]	9.32	0.103	0.710
4DGT† [64]	9.94	0.208	0.639
NeoVerse† [65]	11.56	0.293	0.558
Ours	**18.09**	**0.554**	**0.309**

Table 6: **Results on MultiCamVideo [2].**

Method	PSNR ↑	SSIM ↑	LPIPS ↓
Shape-of-Motion [54]	21.17	0.750	0.245
MoSca [25]	21.88	0.736	0.256
WorldTree [55]	20.76	0.792	0.285
ReCamMaster [2]	19.48	0.596	0.255
ViDAR [39]	22.14	0.754	0.198
Ours (sample)	**27.43**	**0.918**	**0.083**
Ours (distill)	24.36	0.855	0.118

Table 7: **Results on NVIDIA-LS [66].** Baselines from [55].

Method	PSNR ↑	SSIM ↑	LPIPS ↓
4DGS [58]	23.34	0.790	0.128
MoSca [25]	23.72	0.792	0.109
WorldTree [55]	24.06	0.804	0.100
Ours (distill)	**24.49**	**0.807**	**0.090**

Ablation of dynamic 3DGS-conditioned video model. In Table 8 and the top row of Figure 5, we present ablations on our video model design and data pipeline. The first three rows of the second block highlight our conditioning signals. Removing 3D scene flow degrades motion coherence, removing camera injection for the input and target trajectories costs over 0.5 dB, while removing reference-view G-buffers significantly degrades performance because the model loses reliable input-trajectory context under imperfect 4D reconstruction. We next examine the data pipeline. Joint depth alignment across multiview videos substantially outperforms independent per-video depth prediction because it preserves geometric consistency between the reconstructed 4D representation and target views. We also find that mitigating dataset bias [2] through temporal trimming and reversal is important because it prevents overfitting to fixed starting frames. Finally, we study model scaling by replacing the Wan 2.1 14B model with its 1.3B variant [52]. The resulting performance drop indicates that our approach benefits from larger-capacity video generators. Please see the supplement for more visual ablation comparisons.

Ablation of dynamic 3DGS refinement. In Table 9 and the bottom of Figure 5, we ablate key choices in our dynamic 3DGS refinement stage. Fixing the motion scaffold (*No motion opt.*) degrades performance, showing that re-optimizing motion is necessary to incorporate constraints from virtual views. Directly continuing optimization from the monocular reconstruction without reinitialization (*No reinit.*) also leads to worse results, as the initial representation often overfits ill-posed single-view observations. Removing depth-based reinitialization (*No depth reinit.*), where we augment the scene with multi-view depth from generated samples, reduces performance, indicating its importance for recovering missing regions. Finally, treating generated views with equal importance as input views (*No sample weight*) slightly hurts performance, suggesting that down-weighting synthesized observations improves robustness. We further evaluate the number of sampled virtual cameras in Table 2. Reconstruction quality improves monotonically with additional virtual viewpoints, with gains largely saturating beyond eight cameras.

6 Discussion

We have presented World from Motion, a 4D reconstruction framework for real-world monocular videos. By conditioning on reconstructed dynamic 3DGS, we show how to attain highly precise and physically plausible video samples and how to re-optimize those samples into a higher-quality

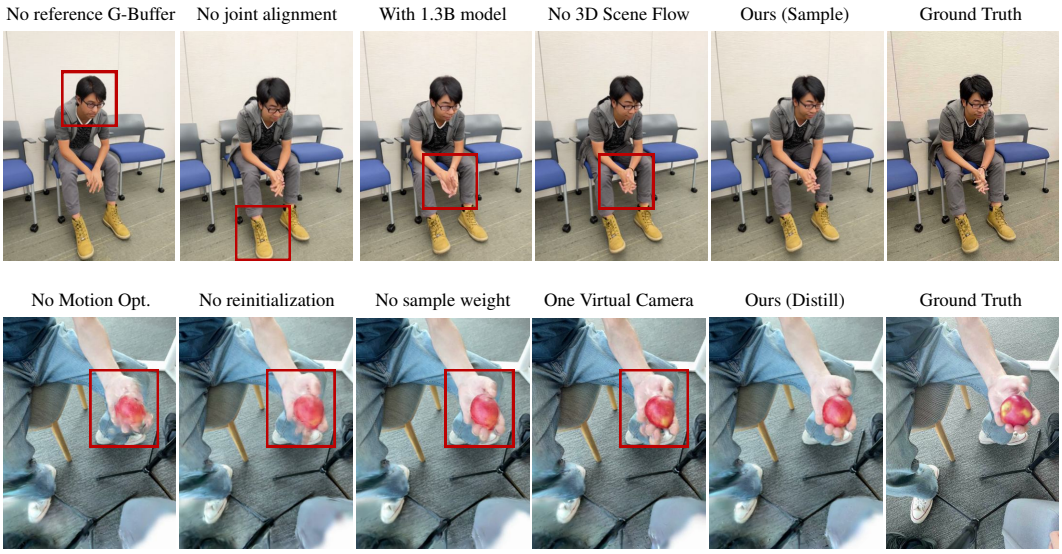

Figure 5: **Ablation study of video model components (top) and re-optimization process (bottom).**

Table 8: **Ablation study of video model.**

Variant	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	**19.50**	**0.701**	0.224
No Recon. Guidance	19.43	0.690	**0.223**
No 3D scene flow	19.30	0.688	0.226
No camera injection	18.93	0.693	0.229
No reference G-Buffer	18.61	0.678	0.265
No joint alignment	15.58	0.598	0.347
No debiased data	18.57	0.665	0.250
With 1.3B model	18.64	0.671	0.244

Table 9: **Ablation study of scene refinement.**

Variant	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	**19.49**	0.704	**0.177**
No motion opt.	18.64	0.680	0.209
No reinit.	19.15	0.695	0.218
No depth reinit.	19.40	**0.705**	0.195
No sample weight	19.36	0.700	0.190

dynamic 3DGS. Our approach significantly reduces the burden of high-quality 4D capture. However, it has some limitations. When the initial scene reconstruction fails completely, our method struggles to recover. The video model may also struggle to align the reconstruction with the input when the target trajectory moves extremely far from the input trajectory. Please see the supplement for representative failure cases.

We believe there are many promising future directions for our method. Incorporating multiple rounds of alternation between refinement and sampling of the dynamic 3DGS reconstruction may produce higher-quality results. We also hope that our method may be useful in downstream robotic applications, reducing the camera burden and improving planning. Finally, we hope that our approach can form a basis for memory and persistence in video generative modeling.

Acknowledgments

We thank Yang Zheng, Zhengfei Kuang, Lior Yariv, and Jianhao Zheng for fruitful discussions. We also thank Yijia Weng and Jiahui Lei for providing evaluation details for MoSca, Kuan Heng Lin for providing Vista4D evaluation details, and Michal Nazarczuk and Eduardo Pérez-Pellitero for providing evaluation details for ViDAR.

References

[1] 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 *ICLR*, 2026.

[2] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *ICCV*, 2025.

[3] Weikang Bian, Zhaoyang Huang, Xiaoyu Shi, Yijin Li, Fu-Yun Wang, and Hongsheng Li. GS-DiT: Advancing video generation with dynamic 3d gaussian fields through efficient dense 3d point tracking. In *CVPR*, 2025.

[4] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023.

[5] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, 2000.

[6] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL `https://arxiv.org/abs/2511.16719`.

[7] Kaihua Chen, Tarasha Khurana, and Deva Ramanan. Reconstruct, inpaint, test-time finetune: Dynamic novel-view synthesis from monocular videos. In *NeurIPS*, 2025.

[8] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001. doi: 10.1109/70.938381.

[9] 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 *ICCV*, 2023.

[10] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, and Andrew Zisserman. BootsTAP: Bootstrapped training for tracking-any-point. *ACCV*, 2024.

[11] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

[12] Tobias Fischer, Samuel Rota Bulò, Yung-Hsu Yang, Nikhil Keetha, Lorenzo Porzi, Norman Müller, Katja Schwarz, Jonathon Luiten, Marc Pollefeys, and Peter Kontschieder. Flowr: Flowing from sparse to dense 3d reconstructions. In *ICCV*, 2025.

[13] Xiao Fu, Shitao Tang, Min Shi, Xian Liu, Jinwei Gu, Ming-Yu Liu, Dahua Lin, and Chen-Hsuan Lin. Plenoptic video generation. In *CVPR*, 2026.

[14] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *NeurIPS*, 2022.

[15] Google DeepMind. Veo: A text-to-video generation system. Technical report, Google DeepMind, 2025. URL `https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf`.

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[17] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL `https://openreview.net/forum?id=qw8AKxfYbI`.

[19] 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-Taixé, and Sanja Fidler. ViPE: Video pose engine for 3d geometric perception. In *arXiv preprint arXiv:2508.10934*, 2025.

[20] Jiaxin Huang, Sheng Miao, Bangbang Yang, Yuewen Ma, and Yiyi Liao. Vivid4d: Improving 4d reconstruction from monocular video by video inpainting. In *ICCV*, 2025.

[21] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *ICCV*, 2025.

[22] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024.

[23] Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4D: Unified feed-forward metric 4d reconstruction. In *CVPR*, 2026.

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

[25] Jiahui Lei, Yijia Weng, Adam W Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In *CVPR*, 2025.

[26] Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In *ICCV*, 2025.

[27] 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*, pages 6498–6508, 2021.

[28] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, Fast and Robust Structure and Motion from Casual Dynamic Videos. In *CVPR*, 2025.

[29] 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 *CVPR*, 2026.

[30] Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *ICLR*, 2026.

[31] Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, et al. Vista4d: Video reshooting with 4d point clouds. *arXiv preprint arXiv:2604.21915*, 2026.

[32] Xi Liu, Chaoyi Zhou, and Siyu Huang. 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. In *NeurIPS*, 2024.

[33] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[34] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13–23, 2023.

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[36] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024.

[37] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=aBsCjcPu_tE`.

[38] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[39] Michal Nazarczuk, Sibi Catley-Chandar, Thomas Tanay, Zhensong Zhang, Gregory Slabaugh, and Eduardo Pérez-Pellitero. Vidar: Video diffusion-aware 4d reconstruction from monocular inputs. *arXiv preprint arXiv:2506.18792*, 2025.

[40] 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*, pages 5865–5874, 2021.

[41] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[42] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[43] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[44] 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*, pages 10318–10327, 2021.

[45] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, 2025.

[46] Seyedmorteza Sadat, Otmar Hilliges, and Romann M. Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=e2ONKX6qzJ`.

[47] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[48] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[49] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. *arXiv preprint arXiv:2310.16818*, 2023.

[50] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*. Springer, 2020.

[51] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *NeurIPS*, 2021.

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

[53] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

[54] 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 *ICCV*, 2025.

[55] Qisen Wang, Yifan Zhao, and Jia Li. Worldtree: Towards 4d dynamic worlds from monocular video using tree-chains. *arXiv preprint arXiv:2602.11845*, 2026.

[56] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024.

[57] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36:8406–8441, 2023.

[58] 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 *CVPR*, 2024.

[59] Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *CVPR*, 2025.

[60] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*, 2024.

[61] 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 *CVPR*, 2025.

[62] Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *arXiv preprint arXiv:2506.05284*, 2025.

[63] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In *ICCV*, 2025.

[64] 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 *NeurIPS*, 2025.

[65] Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. *CVPR*, 2026.

[66] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *CVPR*, 2020.

[67] Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *ICCV*, 2025.

[68] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

[69] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.

[70] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In *CVPR*, 2025.

[71] Kaichen Zhou, Jia-Xing Zhong, Sangyun Shin, Kai Lu, Yiyuan Yang, Andrew Markham, and Niki Trigoni. Dynpoint: Dynamic neural point for view synthesis. *Advances in Neural Information Processing Systems*, 36:69532–69545, 2023.

[72] Kaichen Zhou, Yuhan Wang, Grace Chen, Gaspard Beaudouin, Fangneng Zhan, Paul Pu Liang, and Mengyu Wang. Page-4d: Vggt-4d perception via disentangled pose and geometry estimation. In *ICLR*, 2026.

[73] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *CVPR*, 2023.

[74] Liyuan Zhu, Manjunath Narayana, Michal Stary, Will Hutchcroft, Gordon Wetzstein, and Iro Armeni. Gaussfusion: Improving 3d reconstruction in the wild with a geometry-informed video generator. In *CVPR*, 2026.

Supplementary Material for World from Motion

Abstract

This supplementary document provides additional technical details, experimental analyses, and qualitative results to complement the main paper. We present a detailed analysis of test-time pose optimization for consistent comparison, comprehensive implementation and evaluation protocols, and a granular breakdown of performance across dynamic and static scene components. Furthermore, we include additional qualitative ablations and a discussion of representative failure cases to characterize the current limitations of our framework.

S1 Pose optimization analysis.

MoSca [25] and methods that build on it [39, 55] perform optimization through their output dynamic 3DGS to refine the test poses. They use both dynamic and static pixels to optimize a different camera pose per frame, as shown on the right of Figure S1b. However, the test camera poses of DyCheck are *static*. These methods therefore trade off camera pose accuracy for better metrics in the dynamic region, as seen on the top left of Figure S1a. As seen in the middle left of Figure S1a, the metrics in the static region degrade significantly while the dynamic region improves, slightly improving the overall metrics. The aggregate flow of the test image should be approximately zero in the non-dynamic region outlined in white, as seen in the ground-truth aggregate flow (predicted by RAFT-large [50]) on the bottom of Figure S1b, but this is violated in their per-timestep pose-optimization procedure.

To rectify this, we optimize one camera pose for the whole test sequence using only the static region. The comparison between the former pose optimization and our corrected version in Table S1 shows that this improves MoSca's performance in the static region, but decreases its overall metrics.

Table S1: MoSca results with our corrected pose optimization.

	Covisible			Static Intersection			Dynamic Intersection		
Pose Opt.	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
All Pixels	**19.33**	**0.708**	**0.264**	20.64	0.803	0.258	**15.93**	**0.909**	**0.242**
Static Only	18.69	0.696	0.272	**20.90**	**0.806**	**0.251**	14.71	0.895	0.266

Virtual camera count qualitative comparison. Figure S2 provides a qualitative companion to Table 2, showing held-out target views on DyCheck as the number of synthesized virtual cameras varies. MoSca, which uses no virtual cameras, exhibits the most artifacts; reconstruction quality improves steadily as additional virtual cameras are added.

MultiCamVideo qualitative comparison. Figure S3 provides a qualitative companion to Table 6, showing held-out novel-view renderings on our MultiCamVideo [2] validation split. Compared to MoSca [25] and ViDAR [39], our samples recover sharper details and reconstruct more coherent motion under large viewpoint changes.

Dynamic and static valid breakdown. We further analyze our model's performance by decomposing the improvements across the dynamic and static components of the scene. To do so, we obtain dynamic segmentation masks for all test images using SAM3 [6]. Metrics are then computed over the intersection of the valid evaluation region with the static and dynamic masks, respectively. As shown in Table S2, our method consistently improves rendering quality across both static and dynamic regions, demonstrating robust refinement capabilities regardless of scene motion.

Figure S1: **Pose optimization analysis.** Left: accumulated optical flow of the static test camera with and without test-time per-frame optimization. Right: test camera trajectories under different optimization strategies.

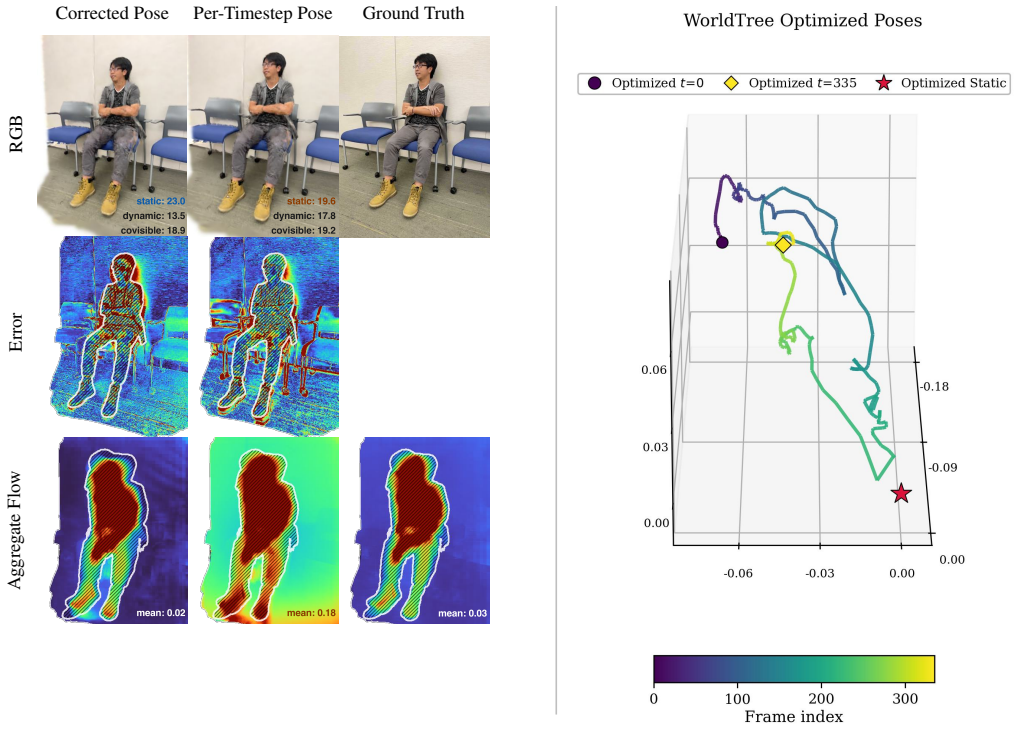

Table S2: **Dynamic and static valid metrics for the same comparison methods.**

	Dynamic Valid			Static Valid		
Method	mPSNR ↑	mSSIM ↑	mLPIPS ↓	mPSNR ↑	mSSIM ↑	mLPIPS ↓
MoSca	14.58	0.881	0.303	19.52	0.775	0.271
WT	15.15	0.887	0.315	20.00	0.786	0.227
ViDAR	15.79	0.893	0.280	20.10	0.780	0.218
Ours+MS	**15.84**	**0.894**	**0.191**	**20.56**	0.779	**0.194**
Ours+WT	15.69	0.891	0.264	20.16	**0.787**	0.222

Test mask visualization. Figure S4 shows a representative DyCheck test pair: the reference image used as model input alongside the corresponding masked test image. We define specific evaluation regions to isolate different aspects of reconstruction quality: the green mask identifies instantaneous covisible pixels between the input and target viewpoints, while the union of the red and green regions constitutes the valid region. This latter region is used to evaluate the fidelity of spatiotemporal accumulation within a persistent 4D reconstruction.

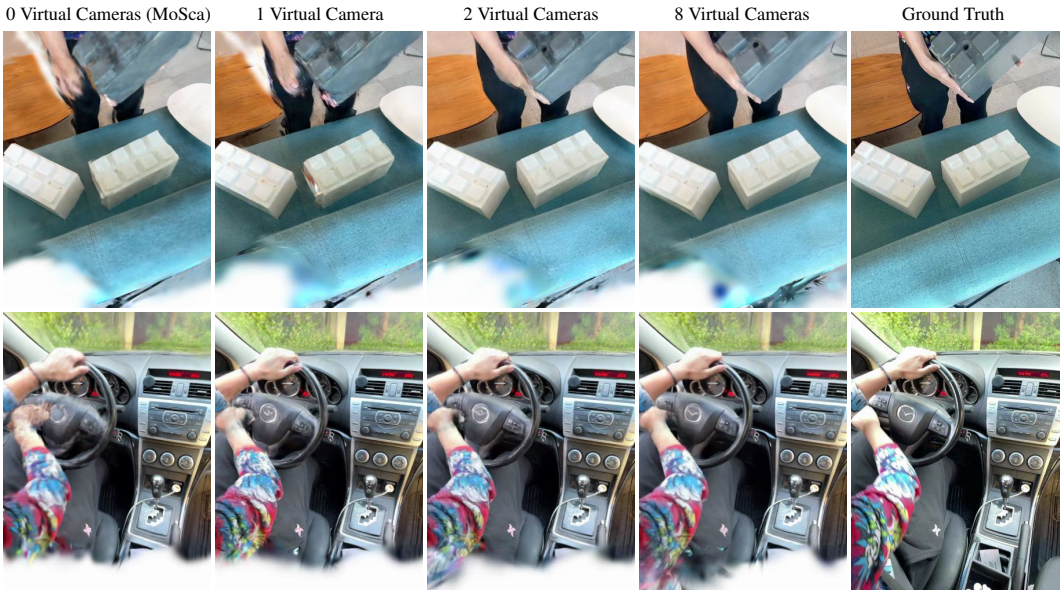

Figure S2: **Qualitative comparison on DyCheck as the number of virtual cameras varies.** MoSca corresponds to the no-virtual-camera setting and suffers the most degradation; quality improves as more virtual cameras are added.

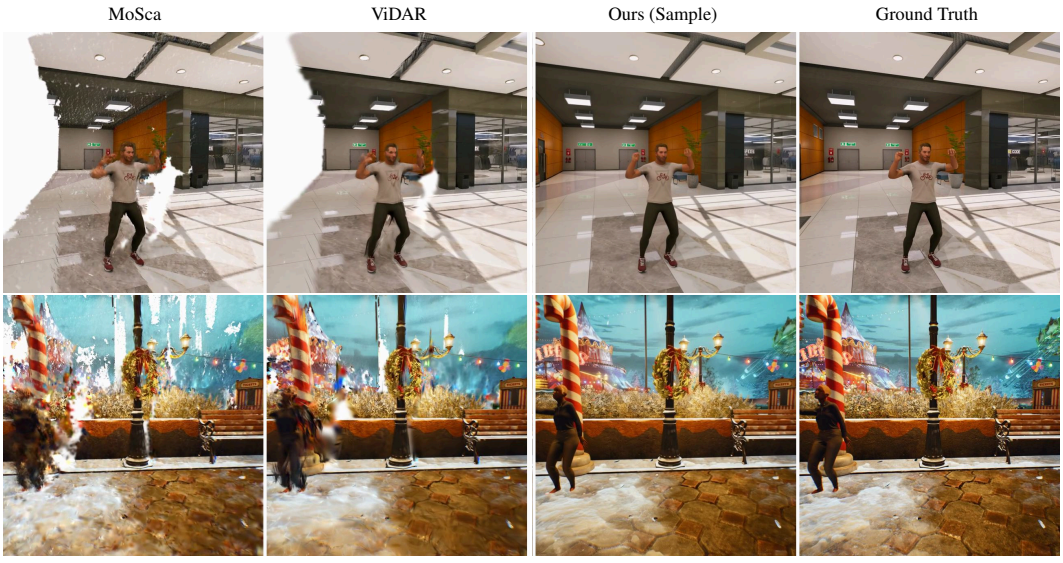

Figure S3: **Qualitative comparison on the MultiCamVideo [2] benchmark.** Each row shows a held-out novel view; our method produces sharper details and more coherent motion than baseline reconstructions.



Figure S4: **Evaluation mask visualization.** Reference image and corresponding masked test image used for evaluation on DyCheck.

Network architecture. Figure S5 illustrates the architecture of our 4D-conditioned video generator. The model builds on a video DiT backbone that denoises the concatenation of clean input-video latents and noisy target-video latents, with the flow-matching loss applied only to the target branch. In parallel, a VACE-style DiT adapter processes the rendered 4D buffers from the input and target trajectories. Its residual features are injected into the base DiT blocks, allowing the generator to use the initial dynamic 3DGS as a dense spatiotemporal scaffold while still relying on the pretrained video prior for photorealistic synthesis. Camera control is added through a pose encoder that maps relative camera trajectories to token-level conditioning before self-attention, giving the denoising network explicit geometric information for both observed and target views.

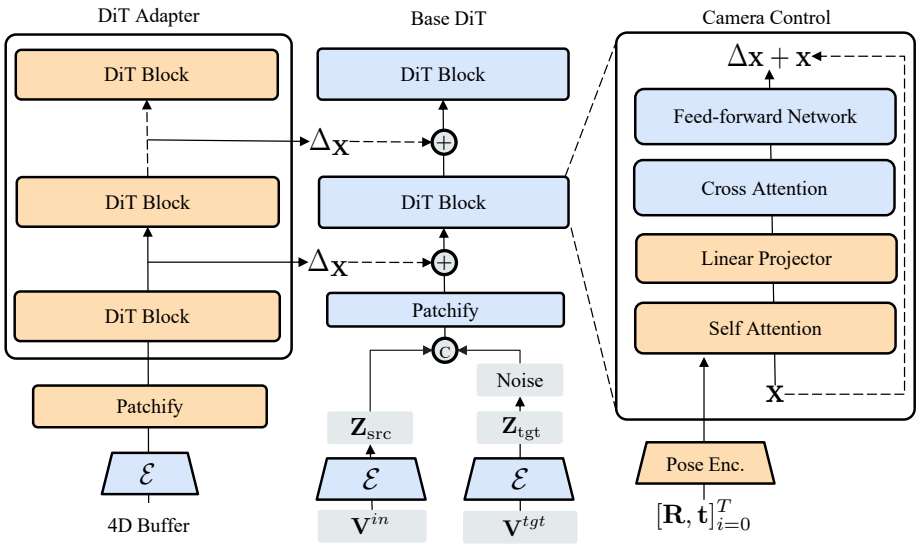

Figure S5: **Network architecture.** Our generator combines a 4D-buffer DiT adapter with a pretrained video DiT. The adapter encodes rendered 4D buffers and injects residual updates into the base DiT, while a camera-control module conditions self-attention on the relative input and target camera trajectories.

Implementation details. We employ Wan-2.1-14B T2V [52] as the backbone for our dynamic 3DGS-conditioned video generator. The DiT adapter is initialized from pretrained Wan-14B-2.1-VACE weights [21]. VACE's context embedding takes 96 latent channels; our five buffer modalities give $5 \times 16 = 80$, so we replace its patch-embedding layer with a new 3D convolution taking 80 channels. During training, we fine-tune the full DiT adapter blocks, the camera encoder, and the self-attention layers within the base DiT, while keeping all other parameters frozen. The training data consists of 150K 73-frame video pairs at a resolution of 512×384 derived from SoM [54] and WorldTree [55] reconstructions. We set the learning rate to 10^{-5} and the batch size to 64, and train our model for 12K steps using the AdamW optimizer [35]. At inference time, we set the number of denoising steps to 20 for all experiments. For the re-optimization stage, we set $\lambda_{\text{tgt}} = 0.5$, $\lambda_{\text{motion}} = 3.0$, $\lambda_{\text{depth}} = 0.05$, $\lambda_{\text{track}} = 0.01$, and $\lambda_{\text{arap}} = 3.0$. We run re-optimization for 15K steps in all experiments.

Dataset debiasing. In MultiCamVideo [2], all videos share the same starting viewpoint, introducing a bias uncommon in real-world capture. To address this, we apply temporal augmentations during training, including random temporal cropping and sequence reversal. This exposes the model to diverse trajectories and temporal orders, improving robustness to varied camera trajectories.

Evaluation details. For the MultiCamVideo dataset, the original test set lacks ground-truth camera parameters. Since our task requires precise poses to evaluate 4D reconstruction accuracy, we instead hold out a custom test split from the source data. This split consists of 5 scenes; for each, we randomly select one camera as the input view and use the remaining 5 cameras for evaluation. We

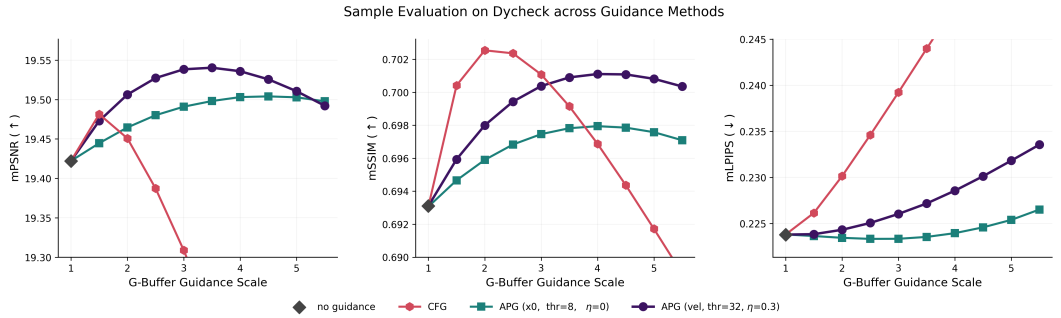

Figure S6: Guidance-scale sweep on DyCheck across no guidance, CFG, and two APG variants. Velocity APG is the most stable at higher scales and reaches the strongest mPSNR, while the x_0 APG variant yields the best mLPIPS.

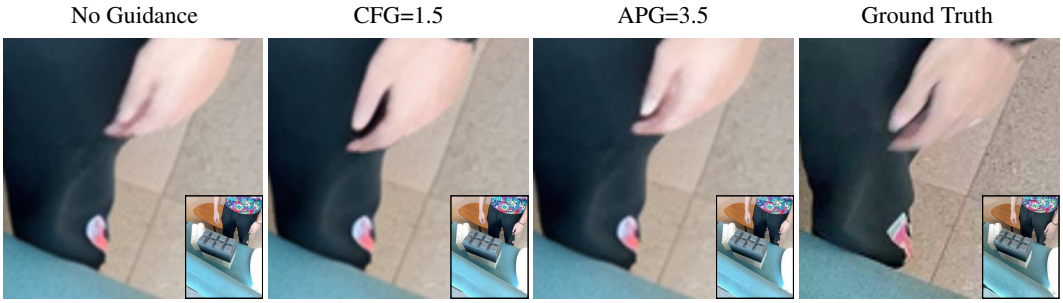

Figure S7: Qualitative guidance comparison for representative DyCheck outputs.

evaluate all methods on the first 73 frames at a spatial resolution of 640×640. On DyCheck, we run the initial reconstruction at a resolution of 480×360. For initial 4D reconstruction, we use RAFT [50] for optical flow extraction and Tapir [9] for 2D track extraction, following MoSca to ensure a consistent comparison across methods. To handle sequences exceeding 300 frames, we adopt a temporal sliding window approach to process the videos in chunks. Generation uses non-overlapping 73-frame windows (the training length), with the final window shifted back so that it also receives a full 73-frame context. Since every window is conditioned on buffers rendered from one global 4D reconstruction, windows cannot drift geometrically relative to each other, and residual per-window appearance variation is absorbed during distillation. For model inference, 4D buffers are resized to our training resolution of 512×384, with the resulting videos resized back to the target resolution to ensure a consistent comparison with baseline methods. For in-the-wild samples, we apply a temporal stride of 2 to the input videos, allowing the model to incorporate longer temporal context.

Camera sampling strategy. On DyCheck, we use our main method to sample the 8 farthest cameras along the input trajectory and fix them over time. In the MultiCamVideo [2] dataset, each scene contains ten camera trajectories; we randomly select one of these as the virtual camera for dynamic 3DGS re-optimization. For in-the-wild samples, we adopt the camera UI from [31] and manually sample novel trajectories to serve as virtual cameras.

Normal rendering. We derive normals from a rendered world-position map $\mathbf{P}_t = \sum_i \alpha_i T_i \boldsymbol{\mu}_i(t)$, obtained by alpha-compositing Gaussian centers, via central differences, $\mathbf{N}_t = \text{normalize}(\partial_x \mathbf{P}_t \times \partial_y \mathbf{P}_t)$, oriented toward the camera and zeroed at silhouettes and depth discontinuities. Normals are stored in world space, so they remain consistent as the camera moves.

Reconstruction guidance ablation. Figure S6 sweeps the guidance scale on DyCheck for no guidance, standard classifier-free guidance (CFG) [18], and two APG [46] variants. This complements Table 8: velocity APG gives the strongest mPSNR/mSSIM trade-off near the selected operating point, while the x_0 APG variant achieves the lowest mLPIPS.



Figure S8: **Representative failure cases of World from Motion.**

Generalization. To demonstrate the robustness of our design across different 4D representations, we evaluate a version of our model trained exclusively on SoM-derived data and test it using DyCheck reconstructed by MoSca [25] and WorldTree [55]. We report the performance of direct samples at test views with full image metrics. Table S3 shows each baseline on its own row, followed by a '+ Ours' row whose green parenthesized values denote the change relative to the baseline directly above. Our model trained on a different, more degraded representation can still faithfully improve the reconstruction fidelity (PSNR) and perceptual quality (LPIPS), with SSIM slightly falling behind.

Table S3: Our method trained on Shape of Motion [54] generalizes to improve other methods.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
MoSca [25]*	16.60	0.553	0.362
+ Ours	18.26 (+1.66)	0.540 (−0.013)	0.278 (−0.084)
WT [55]*	17.08	0.574	0.332
+ Ours	18.60 (+1.52)	0.550 (−0.024)	0.274 (−0.058)

Failure cases. Figure S8 illustrates representative failure cases of World from Motion. We observe that artifacts present in the initial input reconstruction can occasionally propagate to the final output despite our generative refinement process. Specifically, the model struggles with scenes containing complex stochastic or volumetric effects, such as the fluids and smoke shown in the first and third examples. Additionally, as seen in the second example, our model may fail to accurately hallucinate consistent dynamics when subjected to extreme viewpoint shifts.

Compute resources. Our conditional video generative model was trained on a cluster of 32 NVIDIA Blackwell GB200 GPUs (192 GB VRAM each). The final model reached convergence in approximately 48 hours of training. For the dynamic 3DGS re-optimization stage, all experiments were conducted on NVIDIA A100 (80 GB) GPUs. On average, the optimization of a 400-frame sequence requires 40 minutes to complete on a single A100.

Societal impacts. Our approach unifies geometric reconstruction with temporal priors from generative models. Its deployment and the advancement of 4D world modeling carry both positive and negative impacts. On the positive side, by providing more consistent and precise 4D reconstruction from monocular video, our work can enhance the "world models" used by autonomous systems. This has the potential to improve efficiency and safety in human-robot interaction and navigation in complex environments. Our method also lowers the barrier for creators in AR/VR and education, allowing for the preservation of dynamic heritage or personal moments in 4D. On the negative side, the generative capabilities of World from Motion present potential for misuse in the creation of deceptive 4D content. While our method is designed for scene reconstruction, the underlying temporal priors could be used to generate fraudulent "deepfake" videos that exhibit high spatiotempo-

ral consistency. Additionally, the significant energy consumption required to train 14B-parameter generative backbones poses environmental challenges.

Safeguards. To ensure responsible use, we plan to release our code and model weights under a standard academic license that restricts misuse.

Data licenses. MultiCamVideo [2], DyCheck [14], and WildGS [70] are under Apache 2.0.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, our method is the first (to our knowledge) dynamic 3DGS-conditioned video generative model and it allows 4D reconstruction of in-the-wild videos, while achieving state-of-the-art results on DyCheck and MultiCamVideo with both samples and reconstructions. We show how to construct training data and distill the samples of this model.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, we discuss the limitations in the conclusion. Additionally, we show failures in the supplemental PDF and the supplemental website.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A].

Justification: Our paper does not provide theoretical results/proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, we provide all the information needed to reproduce experimental results; details of the model architecture are given in the main paper and all hyperparameters, open-source models used, and reconstruction method details are given in the supplement. We also describe how to construct the training data for our method starting from open-source datasets and detailing the 4D reconstruction process.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We are unable to release the code and data at submission time, but plan to make the data and code available before the conference.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, the paper specifies all training and test details necessary to understand the results and how we chose them.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to training 14B-parameter video generative models [52], it is infeasible to run the training process multiple times to report error bars. For the optimization stage, a single run is expensive to run over every scene multiple times; indeed, prior 4D reconstruction work published at NeurIPS does not provide error bars [39].

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, the paper provides in the supplement sufficient information about the computer resources needed to reproduce experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, our work does not propose a dataset with real human participants, and we discuss the potential misuses and consider mitigation strategies.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes, the paper discusses the potential positive and negative societal impacts of the work performed.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: Yes, we discuss safeguards we plan to put in place for responsible release of data and models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, we credit the creators and state their license for data used in the paper, and respect those licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: We plan to release well-documeted assets (code and data) of this project before the 2026 NeurIPS conference if accepted; however, at this time, we are not providing assets, so this question is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: We do not do crowdsourcing experiments or research with human subjects (besides using images with specified licenses of existing datasets with human subjects).

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: We do not do crowdsourcing experiments or research with human subjects (besides using images with specified licenses of existing datasets with human subjects).

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.