FixAnything: 3D-Consistent Rendering Refinement via Video Generative Priors

Khiem Vuong, Deva Ramanan*, and Srinivasa Narasimhan*

Carnegie Mellon University, USA
https://fix-anything.github.io

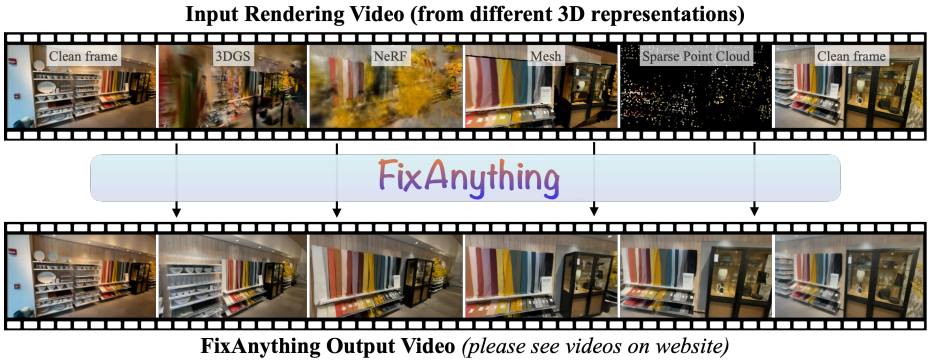

Fig. 1: FixAnything takes a rendering video from any 3D representation — 3DGS, NeRF, mesh, or sparse point cloud — and produces a photorealistic, 3D-consistent video that preserves the camera trajectory and scene content, using a single model finetuned from a pretrained video diffusion model with minimal adaptation. *Top:* One frame selected from each of the four videos rendered using different 3D representations. Clean frames are the input training views (they do not need to be at the start/end). *Bottom:* Corresponding clean output frames showing quality, consistency & generality.

Abstract. Rendering views using 3D scene representations such as Gaussian Splatting (3DGS), Neural Radiance Fields (NeRF), meshes, or even point clouds produces artifacts when input views are sparse or target views lie far from the input. Recent work mitigates these artifacts using diffusion-based generative priors, but is specialized to individual representations and require custom architectures or extensive retraining. We present FixAnything, a single model for fixing a wide range of rendering artifacts. It does so by repurposing a pretrained video generative model, leveraging its implicit multi-view priors with only minimal modification and lightweight finetuning. Our key insight is that even noisily-rendered sequences preserve camera motion and coarse scene structure, allowing cleanup to be formulated as video-to-video translation. To control what scene structure should be preserved, we introduce a binary mask denoting the clean pixels, enabling the model to anchor its output to high-quality inputs (e.g. training views) while refining the rest. To encourage FixAnything to produce 3D-consistent renderings that support downstream reconstruction, we use camera pose accuracy (recovered via

* denotes equal contribution/advising

structure-from-motion) as a reward signal for direct preference optimization (DPO). Across four distinct 3D representations, FixAnything consistently improves rendering quality with lightweight finetuning, demonstrating that a single generalist video prior can replace multiple specialist refinement pipelines. The simplicity of the framework enables immediate adoption of stronger future video models without architectural redesign.

Keywords: Novel View Synthesis · Video Diffusion Models · 3D Reconstruction

1 Introduction

The quality of 3D reconstruction has improved dramatically in recent years, but a gap remains: when input views are sparse or novel viewpoints lie far from training views, every 3D representation produces artifacts. 3DGS [13] produces floaters across the scene, NeRF [23] hallucinates foggy geometry, mesh reconstructions suffer from texture distortions, and point clouds leave holes. These artifacts directly limit downstream applications, where the visual quality of novel views may degrade to the point where they are unusable for content creation or robotics. The dominant approach to improve the quality of these renderings has been to build a specialist generative pipeline that removes the artifacts of that particular representation: 3D or per-image diffusion priors for NeRF [34–36], video diffusion models tailored to 3DGS [20, 37], and camera-controlled video generation conditioned on explicit geometry [9, 28, 43]. While effective individually, such a bespoke approach is difficult to scale: each time a new representation emerges or an existing one is improved, a new pipeline must be built, with its own architecture, training data, and conditioning mechanism.

This raises a natural question: can a single generalist replace this growing family of specialists? We show the answer is yes. Despite looking visually distinct, artifacts from different representations share a common property: they all deviate from the manifold of natural videos, while preserving the underlying camera trajectory and coarse scene layout. A pretrained video model, with minimal adaptation, can exploit this shared structure to translate (or project) degraded renderings onto the manifold of natural videos.

We present **FixAnything**, a framework that realizes this generalist approach. FixAnything takes as input a video rendered along a camera trajectory and directly produces a cleaned version, naturally preserving temporal coherence across views (see Fig. 1). We make use of a video diffusion model (Wan2.1 [32]) with minor architectural changes: the rendering video is concatenated as conditioning input in latent space, and only a lightweight LoRA [10] is trained to adapt the model for this task. A per-frame binary mask distinguishes clean reference frames (which the model should trust and preserve) from degraded frames (which need fixing), anchoring each restoration to known high-quality views (e.g. training views). We generate training pairs from DL3DV-10K [17] using videos rendered from four different 3D representations including 3DGS, NeRF, meshes,

and sparse point clouds. Because the pretrained model already understands natural videos, as few as 20 paired videos suffice for effective training.

Interestingly, we find that different 3D representations achieve comparable quality after cleanup. This is particularly remarkable for sparse (COLMAP [29]) point clouds, which often act as a prerequisite for other 3D representations such as 3DGS and NeRF (since they require *posed* input images). Our results suggest that such "intermediate" 3D representations may not be needed given a sufficiently powerful generative model. Indeed, video generation can already interpolate between two input frames (e.g., "first-last-frame-to-video" generation [32]). However, such models generate the most-likely camera path rather than conditioning on one provided by a rendering engine. We show that sparse point cloud renders are sufficient for exposing this camera path, avoiding the need to teach the generative model about SE(3) camera coordinates (e.g., using Plücker coordinates [26,44] which typically requires far more than 20 finetuning videos [46]).

However, one final challenge is that the generated videos may not be 3D consistent, which may prevent downstream applications such as 3D reconstruction. Rather than building geometric constraints into the architecture, we treat geometric consistency as a preference optimization problem: for each training video, we sample multiple outputs with different random seeds, rank them by how accurately structure-from-motion [29] recovers their camera poses, and apply Flow-DPO [19] to steer the model toward geometrically coherent outputs. This bakes the geometric consistency into the model during training, improving pose estimation accuracy by 7.2% (AUC@5°) with no additional inference cost.

Despite its simplicity, FixAnything matches or exceeds specialist methods while generalizing across representations. Because the adaptation requires no architectural changes and updates less than 1% of parameters, the same recipe can be applied to newer video foundation models as they become available, requiring only a new LoRA training run on limited (academic-scale) compute.

In summary, our contributions are:

- A generalist framework that takes rendering videos along camera trajectories of a 3D representation as input and fixes artifacts with a single model, without architectural changes to the base video model.
- A mask-aware conditioning mechanism that distinguishes *frames to trust* from *frames to fix*, combined with data-efficient training that achieves effective results with limited paired data.
- A geometry-aware preference optimization that bakes multi-view consistency into the model using camera pose accuracy as a reward signal via Flow-DPO.

2 Related Work

Sparse-view novel view synthesis. NeRF [23] and 3DGS [13] achieve high-fidelity rendering when dense input views are available, but their quality degrades sharply under sparse-view settings due to the lack of multi-view supervision. Many

methods address this by introducing regularization during reconstruction. RegNeRF [24] penalizes rendered patches at unobserved viewpoints to smooth geometry, while FreeNeRF [38] applies frequency and occlusion regularization to prevent overfitting. DSNeRF [5] and SparseNeRF [33] leverage depth cues to stabilize geometry under sparse supervision. On the 3DGS side, DNGaussian [15] and FSGS [47] also introduce similar geometric constraints to improve stability in sparse-view settings. While these approaches reduce artifacts, they remain constrained by the available observations. This faithfulness to input views is desirable when sufficient multi-view coverage exists, but it also limits their ability to infer plausible content in regions that are weakly observed or unseen. In such cases, incorporating generative priors can provide reasonable completions while maintaining consistency with observed views.

Generative priors for novel view synthesis and 3D enhancement. Recent work leverages pretrained generative models to improve 3D reconstructions beyond what the input views alone can support. Nerfbusters [34] trains a 3D diffusion prior to directly regularize NeRF geometry, removing floater artifacts in 3D space. ReconFusion [36] generates novel views one at a time, conditioned on PixelNeRF [42] features from nearby cameras, and uses them to regularize NeRF training. Difix3D+ [35] applies a single-step image diffusion model conditioned on a reference view to fix individual renderings, then progressively distills the cleaned images back into the 3D representation. FlowR [8] trains a multi-view flow matching model that jointly processes multiple views to map degraded renderings from sparse reconstructions to their dense-reconstruction counterparts. These methods process views independently or in small multi-view sets, and rely on the underlying 3D representation to enforce global consistency.

Other works repurpose *video diffusion models* [1, 3, 32, 39] for 3D and novel view synthesis tasks. ViewCrafter [43] and GEN3C [28] use camera-controlled video generation conditioned on rendered point clouds to synthesize novel views from sparse inputs. 3DGS-Enhancer [20] trains a video latent diffusion model with a custom spatial-temporal decoder to restore view-consistent renderings, then finetunes the 3DGS model on the enhanced views. Xu et al. [37] recast sparse-view NVS as test-time video completion, using a pretrained video diffusion model with uncertainty-aware modulation to generate pseudo-views that densify 3DGS supervision. A common pattern across all these methods is specialization: each targets a specific representation, introduces custom architecture components, and requires large-scale paired or annotated data. FixAnything instead adapts a single pretrained video model to handle four representation types with no architectural changes and orders of magnitude less training data.

Preference optimization for diffusion models. Direct Preference Optimization (DPO) [27] was originally proposed for aligning language models with human preferences without training an explicit reward model. The idea has since been extended to generative vision models. For instance, Diffusion-DPO [31] adapts the framework to noise-prediction diffusion models for image generation. For modern video generators based on rectified flow [18, 21], Flow-DPO [19] refor-

mulates the preference loss in terms of velocity prediction and demonstrates improvements in visual quality and text alignment for text-to-video models. These prior works optimize for human aesthetic preferences or prompt fidelity. As concurrent works, Epipolar-DPO [14] uses the Sampson distance from epipolar geometry as a reward signal to improve 3D consistency in text-to-video and image-to-video generation, and VideoGPA [7] distills dense geometric priors from reconstruction foundation models into video diffusion models via DPO. Our work shares the motivation of using geometric consistency as a preference signal but applies it to a different setting: rather than improving general-purpose video generation from text or a single image, we target rendering cleanup for 3D reconstructions, using camera pose accuracy from structure-from-motion as the reward to ensure that refined videos support downstream 3D tasks.

3 Method

Fig. 2 provides an overview of FixAnything. We first formulate the problem as representation-agnostic rendering cleanup (Sec. 3.1), then describe how a pretrained video diffusion model is adapted for this task with minimal changes (Sec. 3.2). Sec. 3.3 details the training data, and Sec. 3.4 introduces a preference optimization stage that further encourages geometric consistency for the model.

3.1 Representation-Agnostic Rendering Cleanup

Given any 3D representation, we can render a video along an arbitrary camera trajectory. Some frames along this trajectory, at exact or near training viewpoints, will look clean, while others contain artifacts. Rather than fixing frames independently, FixAnything processes the entire rendering video at once, leveraging temporal context from clean frames to guide the cleanup of degraded ones.

Concretely, let $\mathbf{x} \in \mathbb{R}^{T\times 3\times H\times W}$ be a rendering video along a camera trajectory, and let $\mathbf{m} \in \{0,1\}^T$ be a binary mask where $\mathbf{m}_i = 1$ marks frames rendered from training viewpoints (clean) and $\mathbf{m}_i = 0$ marks the rest (degraded). FixAnything produces a clean video $\mathbf{y} \in \mathbb{R}^{T\times 3\times H\times W}$ that preserves scene content and camera motion while fixing degraded frames. This formulation is representation-agnostic as renderings from NeRF, 3DGS, meshes, and point clouds look different, but all preserve the camera trajectory and coarse scene layout, providing the structure a pretrained video model can leverage to remove artifacts (Fig. 3).

3.2 Lightweight Adaptation of a Pretrained Video Model

FixAnything builds on Wan2.1-I2V-14B [32], a DiT-based [25] image-to-video diffusion model trained with rectified flow. We repurpose it for rendering cleanup by channel-wise concatenating the rendering video as conditioning signal and training a lightweight LoRA [10] adapter, keeping the architecture unchanged.

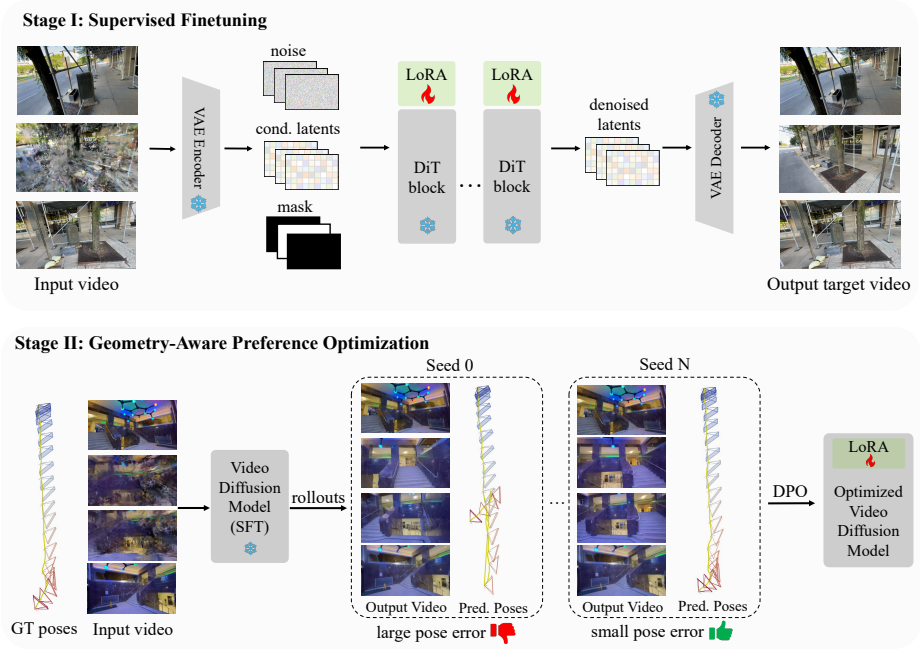

Fig. 2: Overview of FixAnything. Given a rendered video from any 3D representation (3DGS, NeRF, mesh, sparse point cloud) and a binary mask indicating clean frames (e.g., training views), FixAnything produces a cleaned video that preserves scene content and camera trajectory. *Top:* the rendered video is encoded into latent space via a frozen VAE, which is channel-concatenated with the binary mask and noise. These inputs are denoised with a Wan2.1 DiT, adapted only through LoRA [10], without any architectural modification. *Bottom:* after supervised finetuning (SFT), we further optimize the model by generating multiple candidate outputs per video with different random seeds and evaluate their geometric consistency by using COLMAP to recover camera poses. The resulting pose accuracy is used to construct preference pairs for Flow-DPO, which steers the model toward outputs that are geometrically consistent.

Conditioning via channel concatenation. We operate in the latent space of the pretrained VAE. Let $\mathbf{z}_{\text{cond}} = \mathcal{E}(\mathbf{x})$ denote the VAE-encoded latent of the degraded video and $\mathbf{z}_0 = \mathcal{E}(\mathbf{y})$ the latent of the clean target. At each timestep $t \in [0, 1]$, we form the noised latent via the rectified flow interpolation:

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

The degraded video latent $\mathbf{z}_{\text{cond}}$ carries the camera trajectory and coarse scene layout that the model should preserve; we inject it by concatenating with $\mathbf{z}_t$ and a spatially broadcast version of the mask $\mathbf{m}$ along the channel dimension:

$$\hat{\mathbf{z}}_t = [\mathbf{z}_t\,;\,\mathbf{z}_{\text{cond}}\,;\,\mathbf{m}], \tag{2}$$

where $[\cdot\,;\,\cdot]$ denotes channel-wise concatenation. Given this augmented input, the model $\mathbf{v}_\theta$ predicts the velocity field $\mathbf{v} = \boldsymbol{\epsilon} - \mathbf{z}_0$ and is trained with the flow

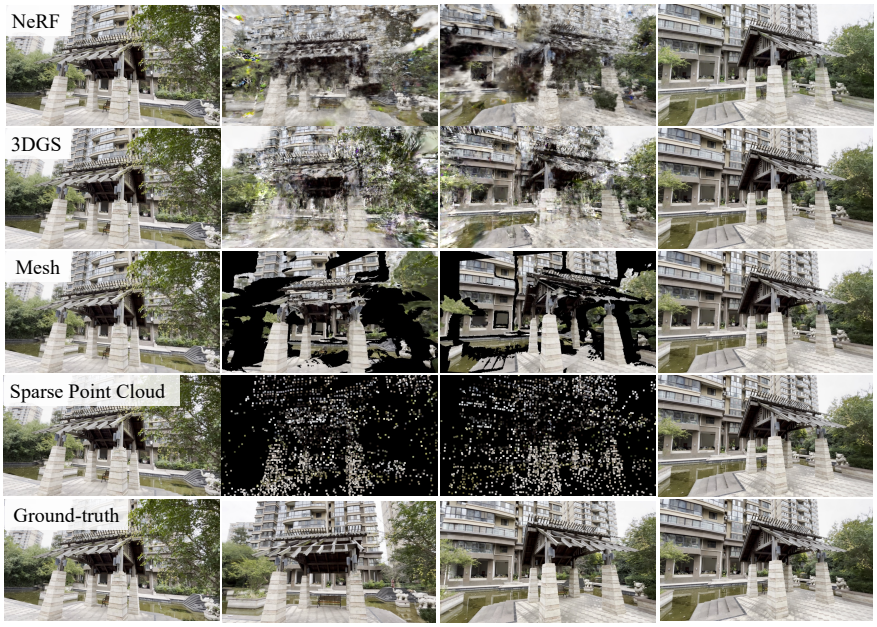

Fig. 3: Different types of degraded renderings from the same scene, paired with ground truth (last row). First and last columns are clean training views, while intermediate frames show artifacts of varying severity. NeRF produces blur and fog (row 1), 3DGS produces floaters (row 2), meshes have holes at ambiguous depth regions (row 3), and sparse point clouds render as scattered patches at detected keypoints (row 4). Despite their visual differences, all representations preserve the underlying camera trajectory and coarse scene layout, and FixAnything trains on all four types jointly.

matching objective [18, 21]:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\mathbf{z}_0,\,\boldsymbol{\epsilon},\,t}\left[\left\|\mathbf{v} - \mathbf{v}_\theta(\hat{\mathbf{z}}_t, t)\right\|^2\right]. \tag{3}$$

Mask-aware conditioning: what to trust vs. what to fix. A rendering video from a sparse reconstruction is not uniformly degraded. Frames near training viewpoints may look nearly perfect, while frames further away can be severely corrupted. Naively asking a generative model to "clean up" the entire video has a problem: the model cannot easily distinguish frames that are already correct from those that need fixing, and may hallucinate over content that should be preserved.

The mask $\mathbf{m}$ resolves this by making the distinction explicit: entries corresponding to training poses are set to 1 (trust) and all others to 0 (fix). This provides two signals. First, it knows which frames to leave untouched, preventing unnecessary hallucination over already-clean content. Second, the clean frames become anchors that provide context for the degraded ones: the model can propagate appearance, lighting, and scene structure from trusted/clean frames to their neighbors, rather than guessing from scratch. Although the mask itself is binary, its temporal arrangement is informative: degradation severity typically increases with distance from the nearest anchor, so the model implicitly learns to mod-

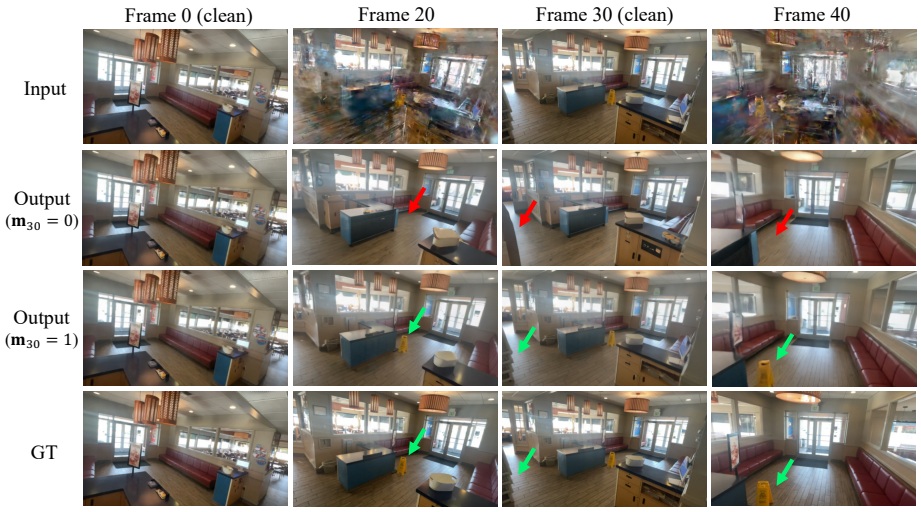

Fig. 4: Effect of mask-aware conditioning in cleaning up 3DGS inputs. The input video (row 1) has clean frames at positions 0 and 30. Without the mask (row 2, $\mathbf{m}_{30}$=0), the model treats frame 30 as degraded and hallucinates over it: objects like the wet floor sign disappear (red arrows). This corruption propagates to nearby frames, causing frame 40 to lose content that should be preserved. With the mask (row 3, $\mathbf{m}_{30}$=1), the model preserves frame 30 and uses it as context for neighboring ones: the wet floor sign and details are retained (green arrows), matching ground truth (row 4).

ulate the strength of its refinement accordingly. Fig. 4 illustrates both effects, and quantitatively, removing the mask degrades PSNR by 1.3 dB (Tab. 3).

LoRA finetuning. We adapt the model using LoRA [10] with rank 64, updating less than 1% of the total parameters while keeping the base model weights and the VAE frozen. This minimal adaptation is sufficient because the task is narrower than general video generation: the model only needs to learn to condition on the degraded input, not to generate videos entirely from scratch.

3.3 Training Data

We build training data of paired videos of degraded renderings and their clean ground-truth counterparts from DL3DV-10K [17], which provides diverse videos with precomputed COLMAP [29] reconstructions. For each scene, we uniformly sample $k \in [3, 12]$ frames as training views and extract 61-frame trajectories that pass through at least two of them. We then render each trajectory from four representation types to produce diverse degraded-clean pairs (Fig. 3):

- *NeRF [23]*: we run Nerfacto [30] on the sparse training views, with characteristic blur and fog artifacts at novel viewpoints.
- *3DGS [13]*: we initialize from a random point cloud and fit a model with gsplat [40] for 7K iterations, deliberately underfitting so that novel viewpoints exhibit visible artifacts while training viewpoints render cleanly.

- *Meshes*: we run MapAnything [12], a feed-forward 3D reconstruction model, on the training views and fit a triangular mesh from the predicted depths. Frames at training viewpoints are replaced with the original images because depth is unreliable at sky regions and occlusion boundaries.
- *Sparse point clouds*: we retain only COLMAP keypoints visible in the selected training views. Because these points exist only at detected keypoints and lack coverage in textureless regions (e.g., sky or walls), we replace frames at training viewpoints with the captured images to provide richer context.

Data efficiency. As the pretrained video model already encodes strong priors about videos, the training data only needs to teach it to condition on degraded input, a simpler task than learning video generation from scratch. Therefore, LoRA finetuning requires remarkably little data: 20 paired videos already produce effective rendering cleanup, and scaling to 500 yields further improvements (Tab. 4). By comparison, prior methods require 80K–150K image pairs [20, 35].

3.4 Geometry-Aware Preference Optimization

The flow matching loss (Eq. (3)) encourages per-frame visual quality but does not explicitly enforce multi-view geometric consistency. In practice, the supervised fine-tuning (SFT) model sometimes hallucinates structures that look plausible in individual frames but are geometrically inconsistent across views. Fig. 5 (row 2) shows a typical failure: the model generates a tree-like structure (red boxes) that shifts position and shape across frames. When SfM (e.g., COLMAP [29]) is run on such outputs, it tracks these inconsistent features and recovers incorrect camera poses, a signal that we can use to improve the model.

Pose accuracy as a reward. We quantify geometric consistency by measuring how well camera poses can be recovered from a generated video. For each output, we run COLMAP [29] with SuperPoint [6] features and LightGlue [16] matching to estimate their camera poses, and compare against the known ground-truth. We report the Area Under the Curve at a 5° threshold (AUC@5°) over both relative rotation accuracy (RRA) and relative translation accuracy (RTA) [11]. Geometrically consistent videos yield high AUC because SfM reliably recovers their poses; videos with hallucinated or inconsistent geometry produce low scores.

Constructing preference pairs. For a separate set of 1,000 DL3DV scenes, we generate five candidate outputs per scene using different random seeds and rank them by AUC@5°. We construct preference pairs $(\mathbf{y}_w, \mathbf{y}_l)$ by pairing higher-ranked outputs against lower-ranked ones, retaining only pairs with an AUC gap of at least 0.2 to ensure a clear preference signal.

Flow-DPO training. We optimize the model using Flow-DPO [19], which adapts DPO [27] to rectified flow models. Let $\mathbf{v}^w = \boldsymbol{\epsilon}^w - \mathbf{z}_0^w$ and $\mathbf{v}^l = \boldsymbol{\epsilon}^l - \mathbf{z}_0^l$ denote the target velocity fields for the preferred and dispreferred samples. The loss is:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}\left[\log \sigma\left(-\frac{\beta}{2}\left(\Delta_w - \Delta_l\right)\right)\right], \tag{4}$$

Fig. 5: Why geometric-aware preference optimization helps. The SFT-only model (row 2) hallucinates structures (red boxes) that shift across frames, causing SfM to recover wrong poses. After Flow-DPO (row 3), the model avoids such hallucinations and closely matches the ground truth (row 4), improving pose AUC@5° by 7.2%.

where $\Delta_w = \|\mathbf{v}^w - \mathbf{v}_\theta(\mathbf{z}_t^w, t)\|^2 - \|\mathbf{v}^w - \mathbf{v}_{\text{ref}}(\mathbf{z}_t^w, t)\|^2$ and Δ_l is defined analogously for the dispreferred sample, with $\mathbf{v}_{\text{ref}}$ being the SFT (LoRA finetuned) checkpoint. This loss steers the model toward outputs that SfM methods can reconstruct accurately. Because the geometric prior is baked into the learned LoRA adapter during training, no pose estimation is needed at inference.

3.5 Inference

At inference time, the user provides a rendering video $\mathbf{x}$ and a mask $\mathbf{m}$ indicating which frames are clean. FixAnything produces a cleaned video by sampling from Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ at $t = 1$ and integrating the learned velocity field $\mathbf{v}_\theta$ toward $t = 0$, with the rendering video and mask provided as conditioning via channel concatenation (Eq. (2)). The process follows the standard flow matching ODE, and we use 50 denoising steps by default. For scenes with more frames than the model's temporal window, we split the video into overlapping chunks of 61 frames. Notably, we observe that reducing the number of denoising steps to 5 still produces reasonable results with a 10× speedup (Tab. 5).

4 Experiments

4.1 Experimental Setup

Implementation details. We finetune Wan2.1-I2V-14B [32] using LoRA with rank 64 on 500 paired DL3DV-10K [17] videos at 288×512 resolution first, then upgrade to 480×832 with T=61 frames per video. SFT training runs for 3000 iterations on a single H100 GPU. Flow-DPO training uses the SFT checkpoint

as reference and further optimizes the LoRA adapter for 2000 additional iterations. At inference, we use 50 denoising steps by default and process videos in overlapping chunks of 61 frames.

Dataset and evaluation protocol. Following the test splits of 3DGS-Enhancer [20] and Xu et al. [37], we evaluate on 20 held-out scenes from DL3DV-10K dataset [17], where we uniformly select 3, 6, or 9 frames as training views from each test scene. The remaining frames (excluding training views) are sampled at every 8th frame to form the query set. We report standard metrics PSNR, SSIM, and LPIPS [45] to measure synthesized image quality, and AUC@5° of relative rotation accuracy (RRA) and relative translation accuracy (RTA) to measure geometric consistency of the generated videos.

Baselines. We compare against two groups of methods. *Sparse-view reconstruction methods* train a 3D representation using few input views: 3DGS [13], RegNeRF [24], FreeNeRF [38], DNGaussian [15], and FSGS [47]. *Post-hoc enhancement methods* refine the output of an existing reconstruction (3DGS rendering): 3DGS-Enhancer [20], Xu et al. [37], and Difix3D+ [35]. For FixAnything, we report results using four different input representations, all built from the same sparse training views following the protocol in Sec. 3.3. All four use the same single model, and only the rendering input changes.

4.2 Comparison with Prior Methods

Tab. 1 compares FixAnything against prior methods on DL3DV under 3-view, 6-view, and 9-view settings. Sparse-view reconstruction methods struggle in this setting: 3DGS produces visible floater artifacts, while regularization-based approaches (RegNeRF, FreeNeRF, DNGaussian) improve geometry but cannot complete content in under-observed regions. Among post-hoc enhancement methods, 3DGS-Enhancer [20], Xu et al. [37], and Difix3D+ [35] demonstrate the value of generative priors, achieving substantial gains over all sparse-view baselines. FixAnything achieves competitive or superior performance on the 3DGS input while being simpler to implement and easier to adapt to future video backbones. Fig. 7 shows a qualitative comparison with Difix3D and Difix3D+ which produce sharper individual images but struggle with cross-view consistency, whereas FixAnything produces temporally coherent output.

Notably, the same model also cleans up mesh renderings and sparse point clouds to comparable quality (Tab. 1, bottom rows), despite these inputs providing far less visual information than 3DGS. Fig. 6 shows qualitative results: even when the input consists of sparse COLMAP keypoints with clean training views interspersed along the trajectory, the model produces dense, photorealistic output. This supports the finding from Sec. 3.1 that the rendering primarily serves as a structural scaffold while the video prior fills in the rest.

In supplementary materials, we further evaluate our model on MipNeRF-360 [2] and LLFF [22], where FixAnything (using 3DGS input) achieves comparable performance to SOTA methods [37, 43] with a notable improvement in LPIPS, showing strong cross-dataset generalization.

Table 1: Quantitative comparison on DL3DV. FixAnything uses a *single model* to clean up renderings from four different input representations (NeRF, 3DGS, mesh, sparse SfM points), achieving comparable quality across all four. Notably, sparse SfM point renderings perform on par with or better than 3DGS and NeRF inputs, despite providing only scattered keypoints and the sole dense visual information comes from the few clean training views that the trajectory passes through. We color each cell as best and second best. († indicates numbers reported by [20, 37])

	3 Views			6 Views			9 Views		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Sparse-view reconstruction									
3DGS [13]†	10.97	0.248	0.567	13.34	0.332	0.498	14.99	0.403	0.446
RegNeRF [24]†	11.46	0.214	0.600	12.69	0.236	0.579	12.33	0.219	0.598
FreeNeRF [38]†	10.91	0.211	0.595	12.13	0.230	0.576	12.85	0.241	0.573
DNGaussian [15]†	11.10	0.273	0.579	12.67	0.329	0.547	13.44	0.365	0.539
FSGS [47]†	12.22	0.296	0.535	13.73	0.429	0.540	15.52	0.468	0.416
Post-hoc enhancement (3DGS rendering)									
3DGS-Enhancer [20]†	14.33	0.424	0.464	16.94	0.565	0.356	18.50	0.630	0.305
Xu et al. [37]†	14.62	0.471	0.491	17.35	0.566	0.396	19.19	0.616	0.335
Difix3D [35]	12.85	0.392	0.557	14.84	0.445	0.462	16.76	0.520	0.399
Difix3D+ [35]	12.37	0.363	0.512	14.41	0.424	0.400	16.39	0.498	0.330
FixAnything (Ours) – single model									
w/ NeRF rendering	14.22	0.427	0.451	17.01	0.522	0.329	18.86	0.605	0.297
w/ 3DGS rendering	15.18	0.452	0.408	17.65	0.561	0.289	19.76	0.632	0.269
w/ mesh rendering	15.74	0.482	0.366	17.95	0.583	0.269	19.86	0.646	0.233
w/ sparse SfM points	15.52	0.463	0.381	17.74	0.568	0.271	19.72	0.624	0.241

Table 2: Effect of Flow-DPO. Fine-tuning with Flow-DPO improves geometric consistency at no extra inference cost.

Method	PSNR↑	SSIM↑	LPIPS↓	AUC@5°↑
SFT only	17.51	0.554	0.296	61.12
+DPO	**17.65**	**0.561**	**0.289**	**68.32**

Table 3: Mask-aware conditioning. Without the mask, quality degrades due to hallucination over clean frames.

Variant	PSNR↑	SSIM↑	LPIPS↓
No mask	16.37	0.525	0.311
With mask	**17.65**	**0.561**	**0.289**

4.3 Effect of Geometry-Aware Preference Optimization

Tab. 2 compares the SFT-only model against the full model after Flow-DPO training, with camera pose accuracy as a reward signal to construct preference pairs for Flow-DPO (Sec. 3.4). While image quality metrics (PSNR, SSIM, LPIPS) improve modestly, the primary gain is in geometric consistency: AUC@5° improves by 7.2%, so COLMAP recovers more accurate camera poses from the DPO-refined outputs. Fig. 5 illustrates why: the SFT-only model can produce frames that look clean but contain hallucinations (red boxes), causing SfM to fail. After Flow-DPO training, these hallucinations are suppressed and the model produces geometrically coherent output closely matching the ground truth. Crucially, this improvement comes at no additional inference cost as the geometric prior is baked into the model weights (LoRA adapter) during DPO training.

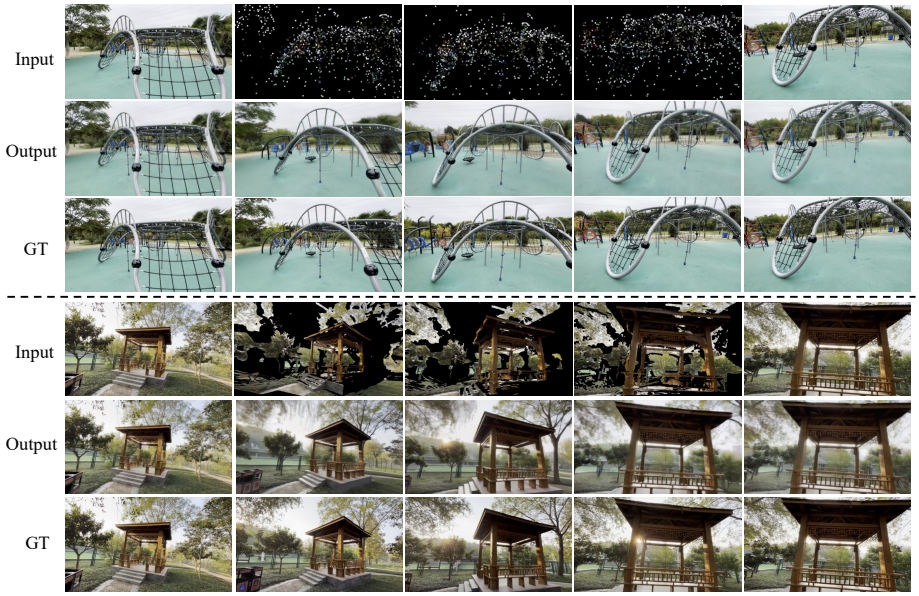

Fig. 6: FixAnything generalizes across input representations. The same model cleans up sparse COLMAP point clouds (top) and meshes (bottom), producing photo-realistic output despite minimal visual input except for the clean first and last views.

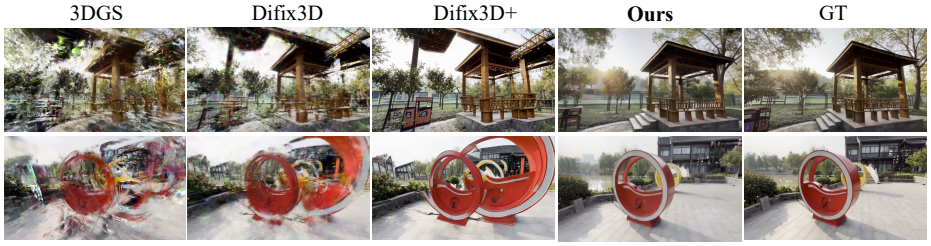

Fig. 7: Qualitative comparison with Difix3D and Difix3D+ [35]. Difix3D enhances each frame independently using the nearest training view then distills results into 3DGS, introducing inconsistencies. Difix3D+ applies an additional enhancement step at render time, sharpening details but amplifying these inconsistencies. By processing the full video at once, FixAnything produces temporally coherent output.

4.4 Ablation Studies

We ablate the remaining design choices on DL3DV-10K with 6 training views.

Mask-aware conditioning. Tab. 3 compares a variant with all mask entries set to 1 (no trust/fix distinction) against the full model. Providing the mask improves PSNR by 1.3 dB: without it, the model must infer degradation severity from the visual signal alone, which is ambiguous since some clean frames resemble mildly degraded ones, causing hallucination over content that should be preserved.

Table 4: Training data volume. As few as 20 paired training videos produce effective 3DGS rendering cleanup, with diminishing returns beyond 100 videos.

Training vids.	PSNR↑	SSIM↑	LPIPS↓
20	16.70	0.531	0.309
50	17.20	0.548	0.297
100	17.45	0.556	0.292
500	17.65	0.561	0.289

Table 5: Inference denoising steps vs. runtime. Quality remains comparable even at 5 denoising steps, enabling a significant speedup.

Steps	PSNR↑	SSIM↑	LPIPS↓	Time (s)
5	18.02	0.574	0.313	31
10	17.91	0.570	0.296	62
25	17.75	0.564	0.289	155
50	17.65	0.561	0.289	309

Data efficiency and inference speed. Tab. 4 shows that 20 paired videos already produce effective cleanup, with diminishing returns beyond 100, showing that the pretrained model already captures most necessary priors and the paired data primarily teaches the conditioning mechanism. Tab. 5 shows that reducing denoising steps from 50 to 5 yields comparable quality across all metrics while providing a 10× speedup, generating a 61-frame clip at 480×832 resolution in 31 seconds on a single H100. This opens the door to further acceleration through distillation [41], potentially enabling real-time rendering cleanup.

5 Discussion: Hallucination and Uncertainty

A natural concern with generative cleanup is *hallucination*. We define *hallucination* as content the model must invent when it is not seen in the input views. However, we emphasize that hallucination is not inherently a failure: it is what makes generative cleanup useful by plausibly filling in unobserved regions, and it becomes a failure only when it contradicts the existing observations.

This raises a practical question: can we tell where the model is hallucinating? As a preliminary analysis, we run inference N=5 times with different random seeds and use the per-pixel standard deviation as an uncertainty estimate (Fig. 8; black pixels denote occluded or unobserved regions). Sky- and ground-like regions show *low* uncertainty, where the model confidently propagates the input texture, while regions with multiple plausible completions (e.g., buildings) show *high* uncertainty. This uncertainty measurement also correlates well with regions of high reconstruction error: on DL3DV (6 views), the mean PSNR over the most-confident 25% of pixels is 25.7 dB versus 14.4 dB over the least-confident 25%. This points to a simple, training-free way to quantify uncertainty, a promising step toward making generative cleanup more reliable.

6 Conclusion

We present a single model for removing artifacts across multiple 3D representations, including 3DGS, NeRF, meshes, and point clouds, by finetuning a pretrained video diffusion model on a modest set of paired videos. Our results suggest that many representations can be cleaned up equally well, raising questions about common reconstruction pipelines. Given N input images, existing

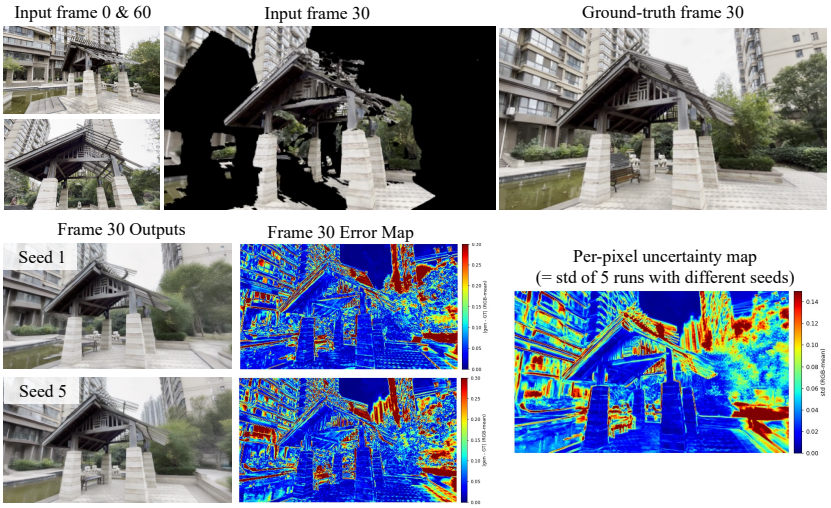

Fig. 8: Uncertainty estimation. Uncertainty maps of mesh rendering cleanup correlates well with reconstruction error and with unobserved (black) input regions.

workflows typically estimate camera poses with COLMAP, learn a representation such as NeRF or 3DGS from the posed views, and finally refine rendered views with a generative model. Our experiments indicate that comparable results can be obtained by instead rendering sparse COLMAP point clouds (optionally meshed via multi-view stereo) and applying generative cleanup directly.

More broadly, our observations suggest that the main difficulty in novel-view rendering arises in regions that are weakly observed or unobserved. In these areas, classical reconstruction methods struggle because the problem becomes one of plausible completion rather than geometric inference. This points to an alternative strategy: reconstruct reliable geometry where observations exist, and use generative models to infer missing content where they do not. Future work could explore feeding such generative predictions back into reconstruction pipelines to produce more complete and consistent 3D scene models.

Acknowledgments We thank Shubham Tulsiani, Nikhil Keetha, Sriram Narayanan, Anurag Ghosh, and other members of Deva's and Srinivas' groups at CMU for their valuable feedback and suggestions at various stages of this project. This work used Bridges-2 [4] at Pittsburgh Supercomputing Center through allocation `cis240119p` from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. This work was supported by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number 140D0423C0074. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022)
3. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
4. Brown, S.T., Buitrago, P., Hanna, E., Sanielevici, S., Scibek, R., Nystrom, N.A.: Bridges-2: A platform for rapidly-evolving and data intensive research. In: Practice and Experience in Advanced Research Computing (2021)
5. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: CVPR (2022)
6. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised Interest Point Detection and Description. In: CVPR Deep Learning for Visual SLAM Workshop (2018)
7. Du, H., Ye, J., Cong, X., Li, R., Ni, J., Agarwal, A., Zhou, Z., Li, Z., Balestriero, R., Wang, Y.: Videogpa: Distilling geometry priors for 3d-consistent video generation. arXiv preprint arXiv:2601.23286 (2026)
8. Fischer, T., Bulò, S.R., Yang, Y.H., Keetha, N., Porzi, L., Müller, N., Schwarz, K., Luiten, J., Pollefeys, M., Kontschieder, P.: Flowr: Flowing from sparse to dense 3d reconstructions. In: ICCV (2025)
9. Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., et al.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In: SIGGRAPH (2025)
10. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: ICLR (2022)
11. Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E.: Image matching across wide baselines: From paper to practice. IJCV (2021)
12. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., Luiten, J., Lopez-Antequera, M., Bulò, S.R., Richardt, C., Ramanan, D., Scherer, S., Kontschieder, P.: MapAnything: Universal feed-forward metric 3D reconstruction. In: 3DV (2026)
13. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. SIGGRAPH (2023)
14. Kupyn, O., Manhardt, F., Tombari, F., Rupprecht, C.: Epipolar geometry improves video generation models. arXiv preprint arXiv:2510.21615 (2025)
15. Li, J., Zhang, J., Bai, X., Zheng, J., Ning, X., Zhou, J., Gu, L.: Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In: CVPR (2024)
16. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: LightGlue: Local Feature Matching at Light Speed. In: ICCV (2023)
17. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: CVPR (2024)
18. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)

19. Liu, J., Liu, G., Liang, J., Yuan, Z., Liu, X., Zheng, M., Wu, X., Wang, Q., Xia, M., Wang, X., et al.: Improving video generation with human feedback. In: NeurIPS (2025)
20. Liu, X., Zhou, C., Huang, S.: 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. In: NeurIPS (2024)
21. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: ICLR (2023)
22. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (ToG) (2019)
23. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
24. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S.M., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: CVPR (2022)
25. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)
26. Plucker, J.: Xvii. on a new geometry of space. Philosophical Transactions of the Royal Society of London (155), 725–791 (1865)
27. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. In: NeurIPS (2023)
28. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: CVPR (2025)
29. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
30. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: SIGGRAPH (2023)
31. Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., Naik, N.: Diffusion model alignment using direct preference optimization. In: CVPR (2024)
32. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
33. Wang, G., Chen, Z., Loy, C.C., Liu, Z.: Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In: ICCV (2023)
34. Warburg, F., Weber, E., Tancik, M., Holynski, A., Kanazawa, A.: Nerfbusters: Removing ghostly artifacts from casually captured nerfs. In: ICCV (2023)
35. Wu, J.Z., Zhang, Y., Turki, H., Ren, X., Gao, J., Shou, M.Z., Fidler, S., Gojcic, Z., Ling, H.: Difix3d+: Improving 3d reconstructions with single-step diffusion models. In: CVPR (2025)
36. Wu, R., Mildenhall, B., Henzler, P., Park, K., Gao, R., Watson, D., Srinivasan, P.P., Verbin, D., Barron, J.T., Poole, B., et al.: Reconfusion: 3d reconstruction with diffusion priors. In: CVPR (2024)
37. Xu, Y., Wang, Y., Yu, S.X.: Novel view synthesis from a few glimpses via test-time natural video completion. In: NeurIPS (2025)
38. Yang, J., Pavone, M., Wang, Y.: Freenerf: Improving few-shot neural rendering with free frequency regularization. In: CVPR (2023)

39. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)
40. Ye, V., Li, R., Kerr, J., Turkulainen, M., Yi, B., Pan, Z., Seiskari, O., Ye, J., Hu, J., Tancik, M., Kanazawa, A.: gsplat: An open-source library for gaussian splatting. JMLR (2025)
41. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: CVPR (2024)
42. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR (2021)
43. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. TPAMI (2025)
44. Zhang, J.Y., Lin, A., Kumar, M., Yang, T.H., Ramanan, D., Tulsiani, S.: Cameras as rays: Pose estimation via ray diffusion. In: ICLR (2024)
45. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
46. Zhu, H., Wang, Y., Zhou, J., Chang, W., Zhou, Y., Li, Z., Chen, J., Shen, C., Pang, J., He, T.: Aether: Geometric-aware unified world modeling. In: ICCV (2025)
47. Zhu, Z., Fan, Z., Jiang, Y., Wang, Z.: Fsgs: Real-time few-shot view synthesis using gaussian splatting. In: ECCV (2024)