# GenFusion: Closing the Loop between Reconstruction and Generation via Videos

Sibo Wu[1,2] Congrong Xu[1,3] Binbin Huang[4] Andreas Geiger[5] Anpei Chen[1,5,†]

[1]Westlake University [2]Technical University of Munich [3]ShanghaiTech University
[4]The University of Hong Kong [5]University of Tübingen, Tübingen AI Center [†]Corresponding author


Figure 1. **GenFusion** introduces a reconstruction-driven generative model to enable artifact-free 3D asset generation and view synthesis for both view interpolation and extrapolation.

## Abstract

*Recently, 3D reconstruction and generation have demonstrated impressive novel view synthesis results, achieving high fidelity and efficiency. However, a notable conditioning gap can be observed between these two fields, e.g., scalable 3D scene reconstruction often requires densely captured views, whereas 3D generation typically relies on a single or no input view, which significantly limits their applications. We found that the source of this phenomenon lies in the misalignment between 3D constraints and generative priors. To address this problem, we propose a reconstruction-driven video diffusion model that learns to condition video frames on artifact-prone RGB-D renderings. Moreover, we propose a cyclical fusion pipeline that iteratively adds restoration frames from the generative model to the training set, enabling progressive expansion and addressing the viewpoint saturation limitations seen in previous reconstruction and generation pipelines. Our evaluation, including view synthesis from sparse view and masked input, validates the effectiveness of our approach. More details at https://genfusion.sibowu.com.*

## 1. Introduction

Generating 3D assets is a fundamental task in computer vision and computer graphics, with broad applications in AR/VR, autonomous driving and robotics. Recent advances in Neural Radiance Fields (NeRF [36], Mildenhall et al. in 2020) and Gaussian Splatting (GS [27], Kerbl et al. in 2023) have enabled high-fidelity 3D scene reconstruction and novel view synthesis. They employ MLP or Gaussian primitives to represent scenes and optimize 3D representation through photometric loss. However, this line of work inherits a key limitation: faithful reconstruction relies on abundant viewpoint coverage; under-observed regions or viewpoints may lead to significant artifacts or missing content.

This is primarily because reconstructing NeRFs or GSs from multi-view images is inherently underconstrained, as an infinite number of photo-consistent explanations may exist for the input images [41, 64]. Consequently, reconstruction models tend to generate "floaters" or "background collapse" artifacts to fake view-dependent effects, even when supplied with dense and well-captured high-quality images [47]. This observation has motivated a series of regularization techniques to constrain neural field training, including sparsity regularizers [22, 57], smoothness losses [37, 56, 63] and monocular geometric cues [62]. For example, ReconFusion [49] regularizes a NeRF-based 3D reconstruction pipeline by introducing a sample loss between novel random camera poses and images predicted by a PixelNeRF-style image diffusion model, yielding significant performance improvements over previous NeRF reconstruction methods in sparse setting. In contrast, feed-

forward reconstruction methods [5, 6, 10, 14, 58, 61] learn inductive biases directly from the dataset. While recent advances enable 3D reconstruction from as few as a single image, existing feedforward reconstruction methods exhibit performance saturation when processing more than 4-8 images. This limitation arises primarily from the architectural constraints of conventional feedforward networks in effectively aggregating and utilizing information from multiple viewpoints.

Meanwhile, generative methods have demonstrated the potential of obtaining 3D assets without multi-view capture. Leveraging large-scale datasets and scalable architectures, models like Stable Diffusion (SD) have achieved remarkable progress in image and video generation [3, 21, 23]. Recent work has applied these approaches to generate 3D assets. For example, DreamFusion [39] introduces Score Distillation Sampling (SDS) to perform text-to-3D synthesis using a pre-trained 2D text-to-image diffusion model, while another line of research [4, 30, 60] explores single-view scene extrapolation by progressively in/outpainting layered depth image.

Despite these advances in 3D reconstruction and generation, a notable conditioning gap remains: scalable 3D reconstruction typically requires dense view coverage, whereas generation methods often operate with single or even no input view. Our paper explores how 3D reconstruction and generation can complement each other in a scalable manner, relaxing the constraints on the number of input views.

We introduce *GenFusion*, a novel reconstruction method that leverages video generative model to achieve artifact-free 3D scene reconstruction and content expansion along novel trajectories by leveraging the proposed reconstruction pipeline, as illustrated in Figure 2. The core of our approach is a simple and scalable reconstruction-driven video generation architecture that predicts realistic video from artifact-prone renderings. Specifically, we first fine-tune DynamiCrafter [52] using RGB-D videos reconstructed from a large-scale, real-world scene-level video dataset [34]. We patchify the capture videos into patches, then randomly select a patch sequence to perform 3D scene reconstruction, rendering full-frame RGB-D videos as input to our video diffusion model, which is subsequently supervised by the original video capture and its monocular depth. Our key insight is that masked 3D reconstruction enables flexible pre-training of video models. As shown in Figure 3, masking 75% of the input pixels during 3D reconstruction produces artifacts and missing regions sharing similar artifact patterns with far-field viewpoint rendering. The resulting artifact-prone video is encoded into latent space, with diffusion guided by a scene description token processed by a CLIP model on a randomly sampled frame. In addition, the baseline video diffusion model, conditioned only on text and RGB, lacks sequence input handling and geometry constraints for view consistency. To address this, we embed input views as latent sequences and incorporate depth by replacing the RGB VAE with an RGB-D VAE. Once trained, we introduce a cyclic reconstruction-generation fusion scheme for scalable 3D scene generation with artifact correction.

Our *GenFusion* learns very high-capacity models that generalize well, we make the following contributions:

- We introduce a reconstruction-driven video diffusion model that efficiently repairs reconstruction artifacts and generates new content in invisible regions.
- We design a masked 3D reconstruction for artifact-GT video pair generation, serving as a new novel view synthesis evaluation protocol for far-field viewpoints.
- Experiments on challenging benchmark datasets [1, 28, 34] demonstrate the effectiveness of GenFusion in synthesizing views distant from the training views.
- Our approach to bridging 3D reconstruction and generation through videos is principled and straightforward, picking the best of these two fields.

## 2. Related Work

**Regularization Techniques:** Optimizing 3D scene representations, e.g., NeRF [36] and 3DGS [27], from 2D images inherently involves an ill-posed inverse problem, where multiple solutions may exist for a given set of observations. To address these challenges, various techniques have been proposed to constrain the optimization of scene representations. In general, unsupervised regularization techniques are based on the assumption that the 3D representation should be sparse [22, 57], smooth [37, 56, 63], low rank [7–9]; rendering weights to be compact [1]; or geometry/texture to be consistent with nearby views in image space [15, 16, 18]. In addition to the multitude of regularization strategies available, many optimization techniques have been proposed to enhance training procedures. For example, gradient scaling [38] is applied for NeRF to address the issue of high-magnitude gradients in regions close to the camera, where more points are sampled, often leading to artifacts known as floaters. One of the most impactful approaches is the "coarse-to-fine" training [31, 32], which modulates the frequency band of positional encodings or hash grid resolution. Another line of research leverages pseudo-observations by using off-the-shelf models to predict cues that enhance reconstruction quality, such as sparse point clouds [17] and monocular depth/normal maps [62]. These approaches are also employed in 3DGS representations [29, 65]. Recently, combining generation models to generate novel views as regularization has proved to be effective in reconstruction objects [55] and scene [35, 49]. Our work aligns closely with ReconFusion [49] and the con-

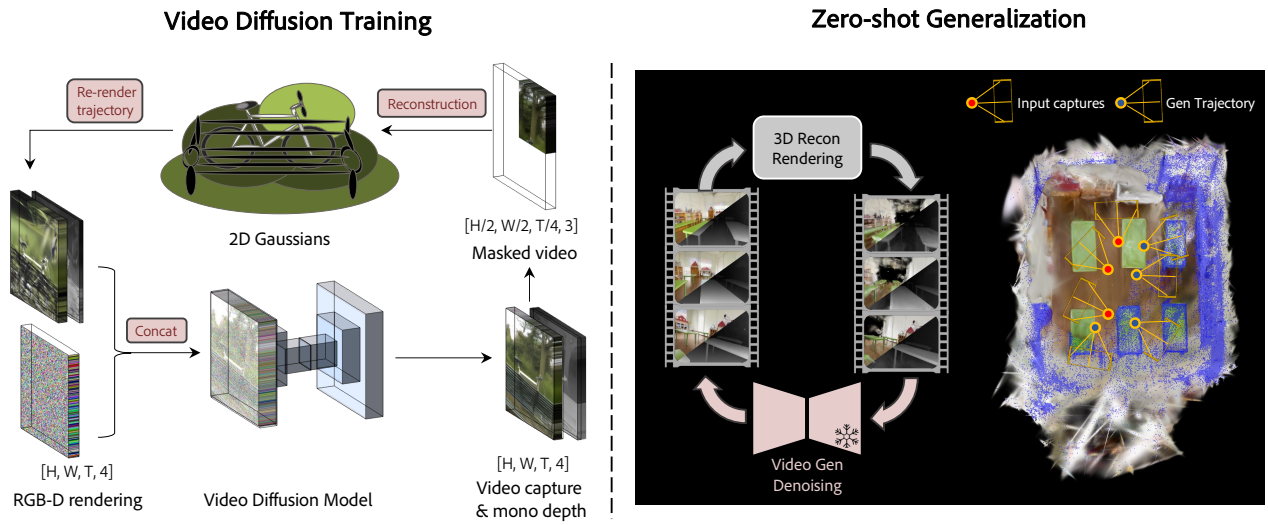


Figure 2. **GenFusion pipeline.** Our approach contains two stages: video diffusion pre-training (left) and zero-shot generalization (right). In pre-training, we first fine-tune DynamiCrafter [52] on RGB-D videos from a large-scale real-world scene video dataset [34]. Captured videos are patchified, and a random patch sequence is selected for 3D scene reconstruction, rendering full-frame RGB-D videos as input to our video diffusion model, supervised by the original video capture and its monocular depth. During generalization, we treat reconstruction and generation as a cyclical process, iteratively adding restoration frames from the generative model to the training set for artifact removal and scene completion.

current 3DGS-Enhancer [35], both of which utilize generative priors to guide the optimization of 3D representations. Despite these methods achieve impressive view interpolation results in sparse-view scenarios, they still struggle with rendering trajectories that deviate significantly from the input views.

**Feed-forward 3D Reconstruction:** In contrast to the per-scene optimization required by neural fields, recent research has explored feed-forward architectures capable of regressing 3D scene representations from a sparse set of input images. These methods learn 3D representations from input images and directly predict novel views in a feed-forward manner. PixelNeRF [59] predicts neural radiance representations from input images, using a convolutional network for efficient feature extraction. MVSNeRF [6] pioneers the paradigm of leveraging cost volumes to regress realistic images from novel viewpoints. Subsequent works [12, 26, 33, 53] further improve performance through enhanced feature matching architectures. In the context of 3DGS, pixelSplat [5] directly regresses scene-level 3D representations from paired images, incorporating an epipolar transformer module to effectively capture view-dependent geometric correspondences. MVSplat [14] builds a cost volume representation to learn cross-view feature similarity, achieving high-quality scene generation with improved efficiency. Recently, DepthSplat [54] further boosts the performance using a pre-train multi-view depth estimator, and Long-LRM [66] utilizes Mamba2 blocks to handle many input views. To handle single view inputs, Flash3D [45] extend monocular depth estimation to 3D shape and appearance reconstruction. Recently, SplatFormer [13] proposes a learning-based model to refine Gaussian splats, enabling out-of-distribution novel-view synthesis. However, existing feed-forward methods are limited to a small number of views - typically fewer than 10 - which significantly restricts their application.

**View-Conditioned Generation:** Generative models have emerged as a promising solution to synthesize plausible content for regions without observations. Thanks to their success in image generation, diffusion models are widely used for multi-view synthesis. Early work 3DiM [48] trains a pose-conditional image-to-image diffusion model for object-centric novel view synthesis. Follow-up research Zero-1-to-3 [20] advanced this approach by fine-tuning large-scale pre-trained diffusion models on synthetic datasets. SSDNeRF [11] jointly optimize diffusion and NeRF auto-decoder to synthesize novel views for object. Recently, ZeroNVS [42] and CAT3D [19] extended the paradigm by training diffusion models with camera pose and multi-view image conditioning. Nevertheless, relying solely on 2D features proves insufficient for maintaining 3D consistency across generated views. Several works [44, 51] inject 3D information into diffusion models for improved geometric understanding. For instance, ViewCrafter [61] explicitly utilizes 3D information from point clouds and expands it iteratively, facilitating consistent interpolation between two views. To enable generating long sequences of 3D scenes from a single input image, InfiniteNature-

Zero [30] and following work [4, 46, 60] explore single-view scene extrapolation by progressively in/outpainting layered depth images predicted by a monocular estimator. While these generative methods achieve visually appealing 3D asset generation, their reconstruction quality and view coverage, particularly at scene scale, remains far from that of 3D scenes reconstructed from densely captured multi-view data.

## 3. GenFusion

Our goal is to construct artifact-free 3D scenes with content augmentation given conditioning views $\{\mathbf{I}_i\}\,(i >= 1)$. The core idea to align 3D reconstruction and generation through video renderings. Specifically, we propose a cyclic fusion approach where 3D reconstruction and generation benefit each other in a virtuous cycle: during the reconstruction process, we iteratively leverage information from the generative model to improve reconstruction quality; meanwhile, more accurate reconstruction results help the generative model produce more realistic and consistent content. This bidirectional enhancement mechanism creates a positive feedback loop.

In the pre-training stage, we aim to train a model that learns from large-scale scenes, which then serves as guidance for optimizing general ill-posed 3D lifting tasks, such as reconstructing 3D from sparse views. We use a video diffusion model as guidance, pre-trained on video captures and their artifact-prone renderings of 3D reconstructions. Specifically, we train a generative model $G_\phi$ by maximizing the expected log-likelihood of generating the complete image given the reconstruction from a masked input:

$$\arg\max_{\phi} \mathbb{E}\left[\log p_\phi\left(I \mid R_\theta(\tilde{I})\right)\right] \tag{1}$$

where $R_\theta$ denotes a reconstruction and rendering function (e.g., NeRF or 3DGS) and $\tilde{I}$ denotes the masked version of target image $I$. To make the generation process 3D aware, our video diffusion model learns to capture the underlying distribution of images by conditioning on artifact-prone rendering video $R_\theta(\tilde{I})$ in our case.

With the video diffusion model, we propose optimizing the 3D scene representation, $\theta$, guided by both input image conditions and generated video for a new scene. Specifically, we maximize the similarity between the novel view videos rendered from 3D representation and the videos generated from our pre-train diffusion model, therefore the gradient of 3D representation is propagated from the photometric loss:

$$\arg\min_{\theta} \mathbb{E}\left[\left|G\phi(\hat{I}_{k+1} \mid R_\theta(\tilde{I}_k)) - R_\theta(\hat{I}_{k+1})\right|_2\right] \tag{2}$$

where $k$ denotes the iteration index.

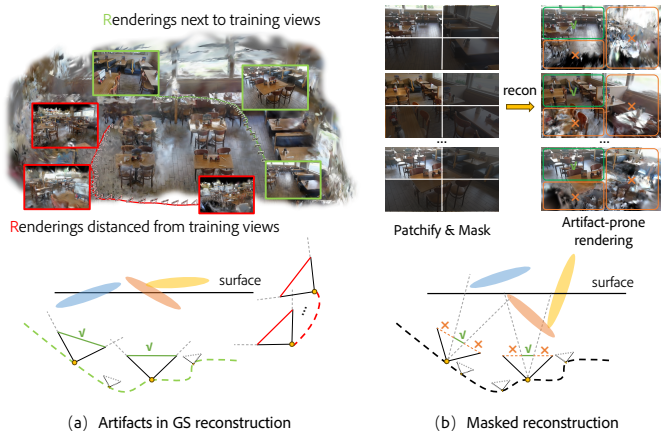


Figure 3. **Artifact-GT video pair generation using masked reconstruction.** a) current SOTA Gaussian Splatting methods render accurately near training views but produce artifacts for distant views due to limited angular supervision, like the red trajectory. b) we propose a masked reconstruction scheme to replicate such artifact patterns for training video diffusion models by masking 75% of pixels during 3D reconstruction and re-rendering the scene along the original trajectory, including the masked pixels.

In the following sections, we first introduce our reconstruction-driven generation architecture, i.e. $G_\phi(I|R_\theta(\tilde{I}))$, in Sec. 3.1. Next, we detail our cyclic generation and reconstruction fusion process in Sec. 3.2.

### 3.1. Reconstruction-driven Generation

We generate view-consistent video by conditioning the current fragment's generation on previous corrupted reconstructions. To generate corrupted reconstructions, we propose a masked 3D reconstruction approach for generating training data, along with a new video diffusion model that facilitates 3D-aware generation and regularization.

**Masked 3D Reconstruction.** We now discuss how to learn novel view generalization capability in a reconstruction paradigm from open-world large-scale videos. Given video captures of a scene, a straightforward approach is to downsample frames at uniform intervals or split the input sequence at the midpoint into train/test segments for scene reconstruction, then render the reconstructed scene from the testing camera viewpoints to generate artifact-prone data. These rendered sequences and original video captures then serve as input and output pairs for video diffusion model training. However, we observe that these sampling schemes limit models to either view interpolation or pure generation. Specifically, scenes are often fully covered by sampled views, with target views located adjacent to these, while the alternative approach often leaves most content unseen in the training segment. Our work aims to learn regularization and generate new content in trajectories that deviate significantly far from the capture trajectories.

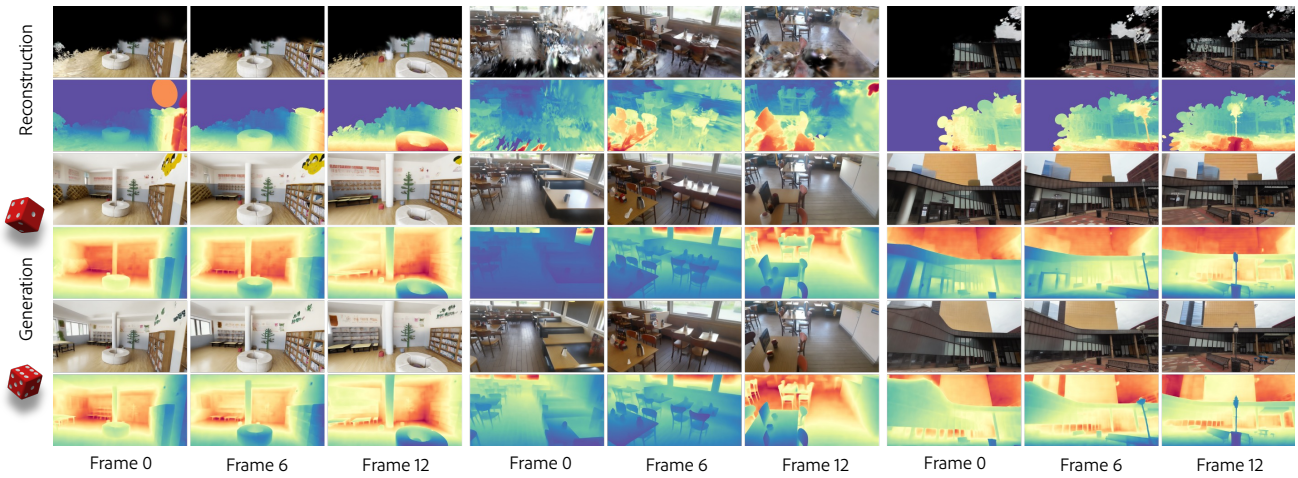


Figure 4. **Reconstruction-driven Video Generation.** Our video diffusion model is able to generate realistic RGB-D video from artifact-prone RGB-D renderings, which is then used as photometric guidance in our cyclic fusion period.

To this end, we propose a masked 3D reconstruction technique to obtain corrupted 3D scenes from observations across both spatial and temporal dimensions, then render images for the full input sequence to generate the training dataset. In specific, we divide input image captures into 4 regular non-overlapping patches, i.e. top-left, top-right, bottom-left and bottom-right, as shown in Figure 3. Then we sample the top-left or bottom-right patch region and mask out (i.e., remove) the remaining three for each scene. The sampled patch sequence is used to conduct 3D reconstruction using the standard approaches, i.e. 2D Gaussian Splatting (2DGS) [25] in this paper. Note that the mask is applied per scene rather than per view, forcing the reconstruction process to have limited view coverage. In practice, since we use open-world large-scale video sequence captures $\mathbf{I}$ for training, which further increases sparsity – views are dense only on the trajectory but extremely sparse in angle coverage – unlike standard multi-view datasets such as Mip-NeRF 360 [1].

With masked reconstruction, we render full-view videos along the same camera poses as the input to form real capture and reconstruction rendering pairs for our video diffusion model training. Our insight is that reconstructing from masked images simulates a narrower field-of-view camera, requiring that context outside the mask be inferred from views deviating from the current viewpoint, which promotes view extrapolation. Moreover, rendering to full view introduces unconstrained regions with extensive black backgrounds, facilitating content outpainting.

**Video Diffusion.**

In essence, we build our video generation model upon the foundation of pre-trained DynamiCrafter [52] - an image-to-animation model - and adapt the model to reconstruction-related video restoration tasks.

More specifically, we enhance frame consistency by incorporating geometric information into the generation process. This is achieved by replacing the VAE in the baseline model with a pre-trained RGB-D VAE, where encoder and decoded are denoted as $\mathcal{E}$ and $\mathcal{D}$ respectively. It allows depth to be integrated without altering the diffusion architecture. In the training process, the ground truth RGBD video $I_{RGB-D}$ are encoded into latent space $z := \mathcal{E}(I_{RGB-D})$, which we add noise in different timestep $t$ and obtain $z_t$. To guide the generation process under reconstruction result, We provide two conditions $c$. The artifact-prone RGB-D video $I'_{RGB-D}$ is encoded and concatenated with per-frame initial noise to support sequence conditioning. This allows rich visual details from the rendered videos to condition the generation process effectively. Additionally, we select the input view closest to the rendering trajectory and embed its ground-truth image into the CLIP feature space. This high-level conditioning provides global information about the scene content. Thus the video denoising network $\epsilon_\theta$ is optimized by:

$$\mathcal{L} = \mathbb{E}_{\mathcal{E}(x),c,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c)\|_2^2\right], \quad (3)$$

The refined latent is subsequently decoded into the final RGB-D video through a pre-trained VAE decoder $\hat{V} = \mathcal{D}(z)$ [44].

### 3.2. Cyclic Fusion

We build our fusion process upon the popular 2D Gaussian Splatting and use a number of 2D oriented planar Gaussian disks to represent 3D scenes. Gaussian disks are parameterized by center position $p \in \mathbb{R}^{3\times1}$, opacity (scale) $\alpha \in [0, 1]$, two principal tangential vectors $\mathbf{t}_u$ and $\mathbf{t}_v$ for orientation, a calling vector $\mathbf{S} = (s_u, s_v)$, and Spherical Harmonices

(SH) coefficients. We refer the reader to the original paper for representation and splatting details [25]. In the following, we introduce how to initialize and progressively update the Gaussian primitives by fusing the 3D reconstruction and video diffusion output.

**Fusion.** We follow the original 3DGS approach by reusing the calibration point cloud for initialization. We initialize Gaussians and their attributes, which are then updated end-to-end based on rendering losses. The optimization process operates as a reconstruction and generation cycle, supervising Gaussians with both input conditioning and novel generated views. More concretely, for every $K$ iteration, we begin with sample new trajectories and render RGB-D videos based on the current reconstruction. We feed these rendering sequences to our video diffusion model to generate artifact-free videos, which are then added to the supervision set, as shown in Figure 2. The cyclic process enables artifact correction in under-observed regions and generates new content for areas that are invisible within the input views.

Among these, we find that novel trajectory sampling is the most critical component. To ensure comprehensive view and angle coverage, we employ two types of trajectories: view interpolation between neighboring input views and a spiral/spherical path generated across all input camera poses.

**Content Expansion.** Large unobserved areas can appear as black or noisy pixels when the sampling trajectories are away from the input views. Although we use generation outputs as supervision for these regions, we found it challenging to split and clone new Gaussians [27] due to the absence of surrounding Gaussians. We solve this issue by adaptively adding new Gaussian points to the scene during optimization, using an unreliable depth-based mapping. Pixels are considered unreliable when

$$T < \tau_T \text{ or } |D - \hat{D}| > \tau_D \tag{4}$$

where $T$ is cumulative opacity, $D$ and $\hat{D}$ are rendered depth and aligned generated depth D respectively. $\tau_T$ and $\tau_D$ are hyperparameter thresholds.

For these unreliable areas, we add new Gaussians by back-projecting the generated RGB-D points into 3D space, similar to the initialization stage. Note that for the newly added Gaussians, position and color values are directly obtained from the RGB-D video, while other attributes are initialized as in 2DGS.

**Loss function.** During cyclic fusion, we freeze the video diffusion model and optimize the 3D representation end-to-end, using simple photometric losses between rendered RGB-D images and input (i.e. $\mathcal{L}_{recon}$) and generated views (i.e. $\mathcal{L}_{gen}$):

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda \mathcal{L}_{gen} \tag{5}$$

where $\mathcal{L}_{recon} = \lambda_{l_1} \cdot \mathcal{L}_{l1} + \lambda_{SSIM} \cdot \mathcal{L}_{SSIM} + \lambda_{mono} \cdot \mathcal{L}_{mono}$, and the $\mathcal{L}_{mono}$ is a scale-invariant depth loss, as used in [62] that enforces consistency between our rendered rendered depth $\hat{D}$ and the monocular depth $D$ predicted by our video diffusion model. The generation loss $\mathcal{L}_{gen}$ shares components but is instead applied to generated images.

To stabilize the optimization process, we use an Sinusoidal warm-up and annealing strategy for the generation loss weight $\lambda$, which is defined as:

$$\lambda(k) = 1.0 \cdot \sin\left(\frac{k - K_{\text{start}}}{K_{\text{end}} - K_{\text{start}}} \cdot \pi\right) \tag{6}$$

where $k$ is the iteration, and $K_{\text{start}}$ and $K_{\text{end}}$ are the start and end iteration of the diffusion term.

## 4. Experiments

We begin with the experimental setup for training and evaluating our GenFusion. Next, we present our reconstruction-driven video diffusion model and compare it quantitatively/qualitatively with state-of-the-art methods for sparse-view 3D reconstruction and view synthesis. Finally, we demonstrate the generative capabilities of our method with scene completion.

### 4.1. Experimental Setup

**Training set:** We train our diffusion model on DL3DV-10K [34], a large-scale dataset containing 10,510 videos, including 140 benchmark scenes. To prepare reconstruction-gt video pairs for diffusion model training, we optimize each scene for 7K steps using our masked 3D reconstruction scheme introduced in Section 3.1. The number of training views are uniformly downsampled to $1/4$ frames for each video to ensure sufficient sparsity, and we render the reconstructed scenes along the original trajectories at a resolution of $960 \times 540$ and augment the dataset with depth information using the current SOTA monocular depth estimator [24].

**Evaluation Dataset:** We evaluate our method on diverse scenes from three datasets, including 24 scenes from the DL3DV-Benchmark [34], 7 scenes from Tanks and Temples (TnT) [28], and 9 scenes from the Mip-NeRF360 [1] dataset, see supplement for details.

**Implementation Details:** In this work, we adopt DynamiCrafter [52] as the video diffusion backbone and fine-tune the model on artifact-GT RGB-D video pairs. The fine-tuning includes both coarse and fine stages: in the coarse stage, the video resolution is set to $16 \times 320 \times 512 \times 4$ with

| Method | RGB VAE | RGB-D VAE | RGB-D VAE | RGB-D VAE |
|---|---|---|---|---|
| Frames | 16 | 16 | 48 | 16 |
| Resolution | 512×320 | 512×320 | 512×320 | 960×512 |
| FID↓ | 26.1607 | 25.4006 | 29.3545 | 22.5526 |

Table 1. **Analysis on reconstruction-driven video diffusion model.**

| | PSNR ↑ | | | | SSIM ↑ | | | | LPIPS ↓ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 3-view | 6-view | 9-view | Avg. | 3-view | 6-view | 9-view | Avg. | 3-view | 6-view | 9-view | Avg. |
| Zip-NeRF [2] | 12.77 | 13.61 | 14.30 | 13.56 | 0.271 | 0.284 | 0.312 | 0.289 | 0.705 | 0.663 | 0.633 | 0.667 |
| DiffusioNeRF [50] | 11.05 | 12.55 | 13.37 | 12.32 | 0.189 | 0.255 | 0.267 | 0.237 | 0.735 | 0.692 | 0.680 | 0.702 |
| FreeNeRF [56] | 12.87 | 13.35 | 14.59 | 13.60 | 0.260 | 0.283 | 0.319 | 0.287 | 0.715 | 0.717 | 0.695 | 0.709 |
| SimpleNeRF [43] | 13.27 | 13.67 | 15.15 | 14.03 | 0.283 | 0.312 | 0.354 | 0.316 | 0.741 | 0.721 | 0.676 | 0.713 |
| ZeroNVS [42] | 14.44 | 15.51 | 15.99 | 15.31 | 0.316 | 0.337 | 0.350 | 0.334 | 0.680 | 0.663 | 0.655 | 0.666 |
| ReconFusion [49] | 15.50 | 16.93 | 18.19 | 16.87 | 0.358 | 0.401 | 0.432 | 0.397 | 0.585 | 0.544 | 0.511 | 0.547 |
| 3DGS [27] | 13.06 | 14.96 | 16.79 | 14.94 | 0.251 | 0.355 | 0.447 | 0.351 | 0.576 | 0.505 | 0.446 | 0.509 |
| 2DGS [25] | 13.07 | 15.02 | 16.67 | 14.92 | 0.243 | 0.338 | 0.423 | 0.335 | 0.580 | 0.506 | 0.449 | 0.512 |
| FSGS [63] | 14.17 | 16.12 | 17.94 | 16.08 | 0.318 | 0.415 | 0.492 | 0.408 | 0.578 | 0.517 | 0.468 | 0.521 |
| GenFusion (Ours) | 15.29 | 17.16 | 18.36 | 16.93 | 0.369 | 0.447 | 0.496 | 0.437 | 0.585 | 0.500 | 0.465 | 0.517 |

Table 2. **Quantitative evaluation of sparse view 3D reconstruction methods on Mip-NeRF360 dataset.** Our approach demonstrates strong performance across a variety of domains, surpassing baseline methods in most cases. We color each column as: best, second best, and third best. The NeRF baseline results above are taken from Reconfusion.

a latent space dimension of $16\times40\times64\times4$. We train this stage for 30K steps with a learning rate of $1\times10^{-5}$ and a batch size of 2 on four H100 GPUs. The model is then fine-tuned to a higher resolution of $16\times512\times960\times4$ for another 34K steps in the fine stage. To handle the RGB-D format, we use a frozen RGB-D VAE from LDM3D [44]. During inference, we apply DDIM sampling with 25 steps and set the classifier-free guidance scale to 3.2.

For zero-shot generalization, we use 2DGS as the 3D representation and initialize with the COLMAP point cloud. In our experiments, views are masked and frames are downsampled; we filter and retain only the points visible from the training views for point cloud initialization.

## 4.2. Video Generation

In this paper, we introduce a reconstruction-driven video diffusion architecture that enables novel view regularization and content generation. We report the VAE design and its impact on video resolution in Table 1. Surprisingly, we find that our model fine-tuned with RGB-D VAE achieves a better FID score than the RGB VAE, even though the diffusion backbone was originally pre-trained on the RGB VAE latent space. Significant improvement can be observed when increasing the spatial resolution from $512\times320$ to $960\times512$. We show the visualization results of our diffusion model in Figure 4, our video diffusion model effectively removes "floaters" from the input video while generating realistic content in black-pixel regions.

## 4.3. View Interpolation

Next, we evaluate our method in a view interpolation scenario where the target scene is fully covered by the input

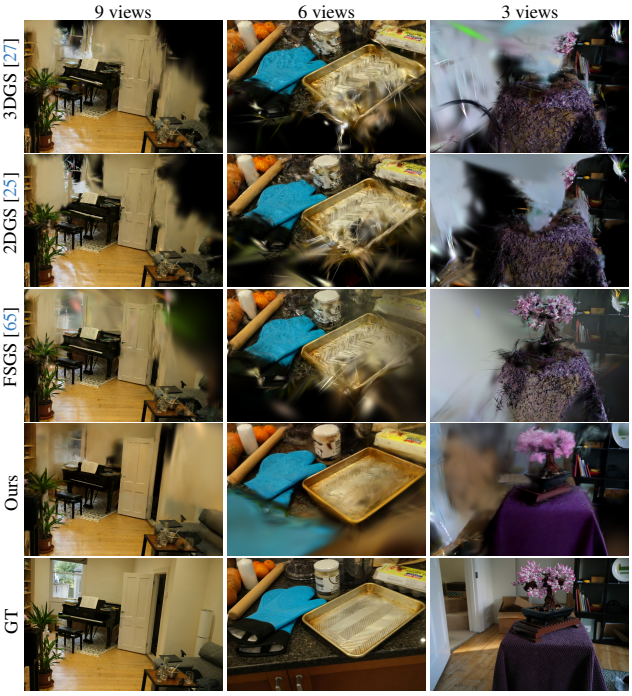


Figure 5. **Qualitative comparison of novel view synthesis using sparse view input on Mip-NeRF360 scenes [1].**

| | DL3DV 1/2 fps. | | | TnT 1/2 fps. | | | DL3DV 1/4 fps. | | | TnT 1/4 fps. | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| 3DGS [27] | 17.22 | 0.740 | 0.314 | 15.95 | 0.653 | 0.414 | 16.90 | 0.728 | 0.321 | 14.75 | 0.609 | 0.440 |
| 2DGS [25] | 16.56 | 0.717 | 0.323 | 15.46 | 0.640 | 0.409 | 16.02 | 0.693 | 0.336 | 14.38 | 0.589 | 0.440 |
| FSGS [63] | 18.25 | 0.722 | 0.362 | 16.72 | 0.625 | 0.465 | 17.83 | 0.710 | 0.370 | 16.04 | 0.607 | 0.473 |
| Ours | 20.47 | 0.788 | 0.284 | 17.45 | 0.662 | 0.427 | 20.01 | 0.780 | 0.292 | 16.29 | 0.630 | 0.447 |

Table 3. **Quantitative comparison on DL3DV [34] and TnT [28] datasets.** Each method is trained for 7,000 steps. Our method outperforms baselines by a significant margin.

views, and testing views lie between these inputs— a common setup in prior regularization evaluations. We compare our method with previous regularization techniques on the Mip-NeRF 360 test set using 3, 6, and 9 input views, as shown in Table 2.

Our results demonstrate that our method achieves more realistic novel view synthesis than the 3DGS and 2DGS baselines, as well as recent FSGS, by a significant margin. It is worth noting that Gaussian Splatting is known to be more challenging to train than NeRF, especially in sparse view settings. We significantly narrow this gap, and, for the first time, show that Gaussian Splatting achieves performance comparable to state-of-the-art NeRF on the challenging Mip-NeRF360 dataset in sparse view settings.(see Table 2 and Figure 5).

## 4.4. View Extrapolation

Unlike sparse view reconstruction, our paper focuses on a more practical scenario: generating complete, artifact-free 3D scenes from video captures. In reality, video offers dense sampling along the trajectory but highly sparse view-

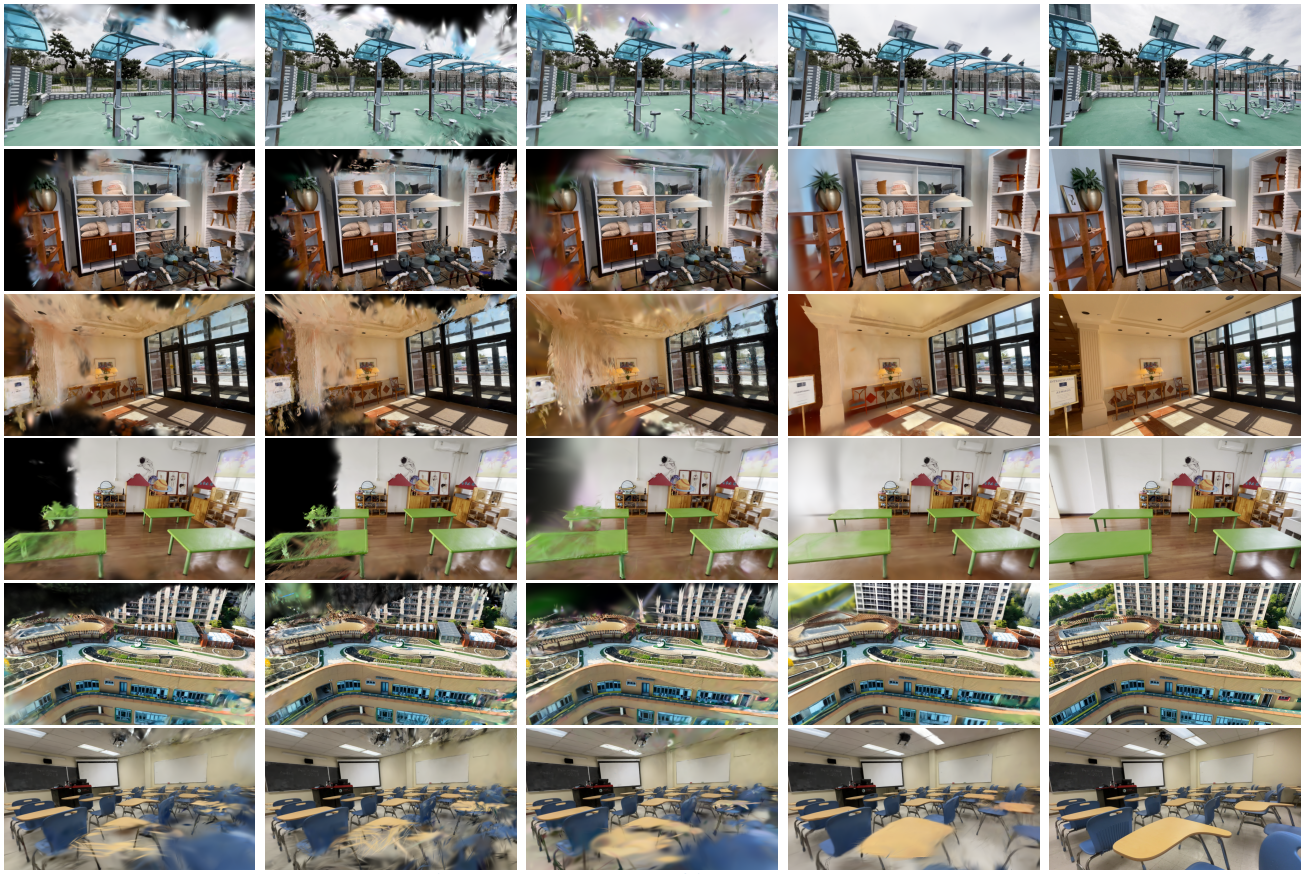


Figure 6. **Qualitative comparison of novel view synthesis using masked input on DL3DV scenes [34].** Gaussian Splatting methods can easily overfit to training views, producing holes and needle-like artifacts when viewed from distant viewpoints. Our method effectively repairs these artifacts while augmenting invisible regions.

point coverage. While video is easy to capture and calibrate, it's challenging for 3D reconstruction, even reconstruction with state-of-the-art reconstruction methods can easily overfit to training views and produce serious artifacts for the viewpoints that are far away from the training views, see Figure 3. Therefore, we care about the novel view synthesis quality from the far field.

To this end, we propose a new evaluation protocol and reuse the masked reconstruction to mime the far-field rendering while preserving the reference, as introduced in Section 3.1. Specifically, we downsample the video sequence at different time ratios for the train/test split (e.g., $1/2$ and $1/4$ in Table 3) and select a patch covering about $25\%$ of the pixels in the training viewpoint for reconstruction, while evaluating on the full field of view of test viewpoints. Note that fixing the path location can result in many invisible and black regions, leading to bias in our generative methods, where the baselines may have significant errors in these areas. To ensure a reasonable comparison, we assign a trajectory to the masking path to provide more complete content coverage, see supplementary video.

Table 3 and Figure 6 show quantitative and qualitative results comparing our method with baselines on the DL3DV-benchmark and TnT scenes. GenFusion achieves significantly better rendering quality than the baselines, thanks to the strong prior from the diffusion model. It effectively removes needle-like artifacts and augments realistic 3D content for the invisible regions.

**Scene Completion.** Furthermore, GenFusion not only provides content augmentation at the 3D scene boundary but also achieves scene-level completion, as shown in Figure 1. Please refer to our webpage for more results.

## 5. Discussion

We have presented GenFusion, a novel model for artifact-free 3D asset generation. First, we adapt an existing well-trained 2D video diffusion model to drive a powerful 3D guidance with minimal modifications. Second, cyclical fusion enables scalable and robust 3D lifting, efficiently closing the loop between 3D reconstruction and generation through video synthesis. Several limitations are evident: our method requiring additional denoising steps and slightly increasing training time (about 40 minutes per scene). Ad-

ditionally, filling large invisible regions can cause blurriness due to inconsistency between video fragments. Modeling and addressing this inconsistency in the fusion module will be a key step toward achieving the next level of quality.

**Acknowledgements:** This work was supported by the Westlake Education Foundation, supported by the Natural Science Foundation of Zhejiang province, China (No. QKWL25F0301), by the ERC Starting Grant LEGO-3D (850533), by the DFG EXC number 2064/1 - project number 390727645.

## References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 5, 6, 7

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 7

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv.org*, 2023. 2

[4] Shengqu Cai, Eric Ryan Chan, Songyou Peng, Mohamad Shahbazi, Anton Obukhov, Luc Van Gool, and Gordon Wetzstein. Diffdreamer: Towards consistent unsupervised single-view scene extrapolation with conditional diffusion models. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 2, 4

[5] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. Pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3

[6] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 2, 3

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. 2022. 2

[8] Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. Factor fields: A unified framework for neural fields and beyond. *arXiv preprint arXiv:2302.01226*, 2023.

[9] Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. Dictionary fields: Learning a neural basis decomposition. *ACM Trans. Graph.*, 2023. 2

[10] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 2

[11] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *ICCV*, 2023. 3

[12] Yuedong Chen, Haofei Xu, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Explicit correspondence matching for generalizable neural radiance fields. *arXiv preprint arXiv:2304.12294*, 2023. 3

[13] Yutong Chen, Marko Mihajlovic, Xiyi Chen, Yiming Wang, Sergey Prokudin, and Siyu Tang. Splatformer: Point transformer for robust 3d gaussian splatting. *arXiv preprint arXiv:2411.06390*, 2024. 3

[14] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024. 2, 3

[15] Zheng Chen, Chen Wang, Yuanchen Guo, and Song-Hai Zhang. Structnerf: Neural radiance fields for indoor scenes with structural hints. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2022. 2

[16] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. 2022. 2

[17] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[18] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[19] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. 3

[20] Weibo Gao, Qi Liu, Hao Wang, Linan Yue, Haoyang Bi, Yin Gu, Fangzhou Yao, Zheng Zhang, Xin Li, and Yuanjing He. Zero-1-to-3: Domain-level zero-shot cognitive diagnosis via one batch of early-bird students towards three diagnostic objectives. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2024. 3

[21] Xun Guo, Mingwu Zheng, Liang Hou, Yuan Gao, Yufan Deng, Pengfei Wan, Di Zhang, Yufan Liu, Weiming Hu, Zhengjun Zha, et al. I2v-adapter: A general image-to-video adapter for diffusion models. In *SIGGRA*, 2024. 2

[22] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *ICCV*, 2021. 1, 2

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 2

[24] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024. 6, 1

[25] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH Asia*, 2024. 5, 6, 7, 8, 3

[26] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. on Graphics*, 2023. 1, 2, 6, 7, 8, 3

[28] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 2, 6, 7, 3

[29] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[30] Zhengqi Li, Qianqian Wang, Noah Snavely, and Angjoo Kanazawa. Infinitenature-zero: Learning perpetual view generation of natural scenes from single images. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022. 2, 4

[31] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[32] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 2

[33] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *ACM Trans. on Graphics*, 2022. 3

[34] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3d vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 6, 7, 8

[35] Xi Liu, Chaoyi Zhou, and Siyu Huang. 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 3

[36] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. 2020. 1, 2

[37] Michael Niemeyer, Jonathan Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. 2022. 1, 2

[38] Julien Philip and Valentin Deschaintre. Floaters No More: Radiance Field Gradient Scaling for Improved Near-Camera Training. In *Eurographics Symposium on Rendering*, 2023. 2

[39] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2023. 2

[40] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. 2020. 2

[41] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[42] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. Zeronvs: Zero-shot 360-degree view synthesis from a single image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3, 7

[43] Nagabhushan Somraj, Adithyan Karanayil, and Rajiv Soundararajan. SimpleNeRF: Regularizing sparse input neural radiance fields with simpler solutions. In *SIGGRAPH Asia*, 2023. 7

[44] Gabriela Ben Melech Stan, Diana Wofk, Scottie Fox, Alex Redden, Will Saxton, Jean Yu, Estelle Aflalo, Shao-Yen Tseng, Fabio Nonato, Matthias Müller, and Vasudev Lal. LDM3D: latent diffusion model for 3d. *arXiv preprint arXiv:2305.10853*, 2023. 3, 5, 7, 1

[45] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F. Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024. 3

[46] Haiping Wang, Yuan Liu, Ziwei Liu, Zhen Dong, Wenping Wang, and Bisheng Yang. Vistadream: Sampling multiview consistent images for single-view scene reconstruction. *arXiv preprint arXiv:2410.16892*, 2024. 4

[47] Frederik Warburg*, Ethan Weber*, Matthew Tancik, Aleksander Hołyński, and Angjoo Kanazawa. Nerfbusters: Removing ghostly artifacts from casually captured nerfs. 2023. 1

[48] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2023. 3

[49] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. Reconfusion: 3d reconstruction with diffusion priors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 7

[50] Jamie Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 7

[51] Jianfeng Xiang, Jiaolong Yang, Binbin Huang, and Xin Tong. 3d-aware image generation using 2d diffusion models. In *ICCV*, 2023. 3

[52] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 2, 3, 5, 6, 1

[53] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: Multi-baseline radiance fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[54] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862*, 2024. 3

[55] Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Gaussianobject: High-quality 3d object reconstruction from four views with gaussian splatting. *ACM Trans. Graph.*, 2024. 2

[56] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 7

[57] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 1, 2

[58] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[59] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[60] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T. Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, and Charles Herrmann. Wonderjourney: Going from anywhere to everywhere. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 4

[61] 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. 2, 3

[62] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2, 6

[63] Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric P. Xing. Fregs: 3d gaussian splatting with progressive frequency regularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 7

[64] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 1

[65] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: real-time few-shot view synthesis using gaussian splatting. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 2, 7, 8, 3

[66] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. *arXiv preprint 2410.12781*, 2024. 3

# GenFusion: Closing the Loop between Reconstruction and Generation via Videos

Supplementary Material

## A. Overview

In the supplementary materials, we provide comprehensive experimental details and extensive ablation studies to evaluate the contributions of our framework designs. Additionally, we present qualitative comparisons between our approach and baseline methods.

## B. Experimental Details

### B.1. Video Diffusion Model Details

Our diffusion model is built upon a pre-trained image-to-video latent diffusion model [52] which operates on RGB latent space. However, we found that relying only on RGB inputs fails to produce consistent video frames, particularly in regions with severe artifacts. Therefore, we leverage depth maps to inject geometry information into the diffusion model. To process RGB-D inputs, we utilize a pre-trained VAE from LDM3D [44], which is designed to encode RGB-D image into the latent space. Therefore, given an RGB-D video of size $4\times T\times 512\times 320$ ($T$: video length), we flatten it along the first two dimensions, encode it into latent features of shape $4T\times 64\times 40$, and reshape to $4\times T\times 64\times 40$ for diffusion. For CLIP feature embedding, we randomly sample a reference frame from the input sequence. During reconstruction, the nearest input frame to the target trajectory serves as the reference for the CLIP guidance. In the training process, each training example comprises an artifact-prone RGB-D video, a reference image, and one target RGB-D video. To obtain the temporally consistent depth map for training, we leverage the SOTA monocular depth estimatior [24] to augment the training data. During inference, we employ DDIM sampling with classifier-free guidance to modulate condition adherence strength. To do so, we implement random dropout of conditioning images with $10\%$ probability per sample during training.

In the video diffusion experiment section, we explore different designations of diffusion model to identify the optimal balance between model performance and computational efficiency. Therefore, four diffusion models are trained and analyzed in three aspects, input type, resolution, and video length. To this end, the base model that generates 16 frames of videos with a resolution of $512\times 320$ is trained for 30k iterations using a learning rate of $1e-5$ and a batch size of 2 on each GPU. To assess the impact of depth information, we conduct a comparative analysis by training two base models: one utilizing RGB-D inputs and another with RGB inputs only. Both models are trained under identical hyperparameter settings to ensure a fair comparison. To enhance the quality of generated videos, we fine-tune the base RGBD model for higher resolution inputs (16 frames at $960\times 512$) with an additional 34k iterations, maintaining the same learning rate and batch size configurations. To extend video generation capabilities, we fine-tune the temporal layers of our base model to produce 48-frame sequences for 30k iterations while maintaining the base model's batch size and learning rate.

### B.2. Masked 3D Reconstruction

In the main paper, we introduce a masked 3D reconstruction scheme to mime the far-field rendering artifacts. The marked 3D reconstruction is used in both video diffusion data generation and novel view synthesis evaluation. In practice, we use a patch mask of size $H/2\times W/2$ to enable narrow field-of-view inputs in both settings. But differently, we randomly select one of the four corner locations for training dataset generation, since fixing the mask location introduces diverse artifacts and under-observed regions, enriching the dataset's complexity. However, the extremely limited observation setup often produces large black regions near the boundaries. Using such data directly for evaluation can lead to unrealistically low quantitative metrics in these regions due to content ambiguity. To enable a fair comparison, we generate a trajectory to move the mask over time, rather than fixing its location as in video data generation. This ensures that most scene content is included in the input. Notably, all baselines and our method use the same sampling trajectories for each scene. To further reduce sparsity along the camera trajectory, we downsample the viewpoints by factors of 2 and 4, using these masked frames as our training input while using the remaining full frames for evaluation.

### B.3. Cyclic Fusion

We close the loop between reconstruction and generation through cyclic fusion that updates the 3D scene representation (i.e. 2D Gaussian primitives) using input captures and generated videos.

**Warm-up:** During the warm-up phase of the fusion process, the 3D representation is updated exclusively from input captures for the first 1000 iterations. Afterward, we apply our reconstruction-driven video diffusion every 1000 iterations to remove the artifacts and generate new content for the video renderings, which are then added to the training view set.

**Sparsity-aware Densification:** In the original Gaussian

| No. | Method | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|
| 1 | 2DGS baseline | 13.87 | 0.572 | 0.447 |
| 2 | +train view monocular depth | 13.89 | 0.575 | 0.442 |
| 3 | +sample view rgb | 15.33 | 0.602 | 0.442 |
| 4 | +sample view depth | 15.34 | 0.622 | 0.438 |
| 5 | +sparsity aware densification | 15.81 | 0.617 | 0.409 |

Table 1. Ablation studies using on Tanks and Temples dataset. ↑ indicates higher is better, while ↓ indicates lower is better.

Splatting [27], scene primitives are cloned and split based on the average magnitude of view-space position gradients, and the gradient for each primitive is reset every K steps (i.e., 100 steps in 3DGS and 2DGS). We find this strategy performs well in scenarios where the scene is densely captured. In such cases, primitives are typically observed for more than half of the reset steps ($> \frac{K}{2}$), making the averaged gradient over K steps a reliable indicator for deciding whether to add the primitive to the densification list. However, this strategy becomes unreliable for masked 3D reconstruction, as the visibility counts of each Gaussian primitive are significantly lower, resulting in unstable gradient accumulation. To address this, we propose a sparsity-aware densification strategy that maintains the densification list by incorporating minimal visibility counts. Specifically, we disable gradient resets and add a primitive to the densification list only if its gradient exceeds the threshold and its visibility count surpasses the minimal visibility requirement. Accordingly, we perform the densification process every 100 iterations to progressively refine the point cloud representation. We found this strategy is more robust for handling diverse input scenarios.

## C. Ablation Studies

In Table 1, we perform comprehensive ablation studies to validate the contributions of our model components using scenes from the Tanks and Temples dataset[28]. We begin with a vanilla 2D Gaussian Splatting (2DGS) model, following its original implementation, as the baseline. Building on this, we evaluate the effect of incorporating monocular depth supervision during training and view sampling using the ScaleAndShiftInvariant loss [40]. As shown in (2) of Table 1, this addition does not yield quantitative improvements. However, it encourages smoothness in the rendered depth, effectively reducing floating artifacts typically observed during initial reconstruction stages (visualized in Figure 1). Significant performance gains are observed in (3) and (5) of Table 1, attributed to our RGB regularization and sparsity-aware densification strategies, further confirming the effectiveness of our method.

## D. More Evaluation

### D.1. View Interpolation

Table 4 provides a per-scene break down for quantity metrics in Mip-NeRF360. These results showcase that our models consistently improve the baselines.

### D.2. View Extrapolation and Scene Completion

Here we present extensive experimental results on masked 3D reconstruction. Figure 2 demonstrate that our performance also outperforms baselines in far-field viewpoint renderings. Table 3 and Table 2 provide per-scene quantitative results.

## E. Conclusion

We have observed viewpoint saturation as a fundamental limitation in previous reconstruction and generation methods: high-quality reconstruction relies on dense captures, while generation methods are optimized for weak conditioning. To relax this constraint, we propose GenFusion, an efficient generative guidance framework that enables accurate 3D reconstruction and content generation for input conditions across varying densities. We achieve this by closing the loop between reconstruction and generation, creating a feedback loop where generation becomes aware of the reconstruction status through novel trajectory rendering, and reconstruction is further regularized using RGB-D videos generated by our video diffusion model. We evaluate the interpolation capability using a sparse view reconstruction setup and the extrapolation capability with a novel masked reconstruction mechanism. Both tasks demonstrate significant improvements over baseline methods. In addition, our approach achieves scene-level 3D completion, enabling 3D scene expansion. We hope our findings in bridging reconstruction and generation can inspire other novel view syntheses and 3D scene generation tasks.


Figure 1. From top to bottom: 2DGS baseline, with train view monocular depth added, with sample view RGB added, with sample view depth added, and finally with sparsity-aware densification.


Figure 2. Qualitative comparison of novel view synthesis using masked input on TnT scenes [28].

| | 2DGS | | | 3DGS | | | FSGS | | | Ours | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| 14eb48a50e | 16.44 | 0.690 | 0.384 | 17.40 | 0.730 | 0.360 | 17.64 | 0.690 | 0.434 | 19.92 | 0.767 | 0.351 |
| 0a1b7c20a9 | 15.74 | 0.733 | 0.278 | 16.38 | 0.760 | 0.263 | 18.17 | 0.752 | 0.310 | 19.27 | 0.811 | 0.230 |
| 06da796666 | 15.42 | 0.672 | 0.396 | 15.34 | 0.698 | 0.390 | 17.02 | 0.710 | 0.437 | 18.54 | 0.755 | 0.376 |
| 389a460ca1 | 18.04 | 0.810 | 0.309 | 18.24 | 0.824 | 0.311 | 18.48 | 0.799 | 0.367 | 21.11 | 0.861 | 0.273 |
| 2cbfe28643 | 16.09 | 0.782 | 0.257 | 16.79 | 0.799 | 0.254 | 19.50 | 0.790 | 0.321 | 22.03 | 0.850 | 0.227 |
| 374ffd0c5f | 19.85 | 0.780 | 0.256 | 21.16 | 0.803 | 0.250 | 20.98 | 0.763 | 0.327 | 22.35 | 0.842 | 0.224 |
| 5c3af58102 | 15.66 | 0.692 | 0.273 | 15.95 | 0.709 | 0.260 | 16.22 | 0.661 | 0.325 | 20.10 | 0.794 | 0.214 |
| 66fd66cbed | 21.42 | 0.855 | 0.235 | 22.15 | 0.873 | 0.224 | 22.29 | 0.867 | 0.246 | 23.27 | 0.897 | 0.191 |
| 3bb3bb4d3e | 16.89 | 0.795 | 0.266 | 17.85 | 0.810 | 0.253 | 18.84 | 0.780 | 0.319 | 22.48 | 0.883 | 0.198 |
| 91afb9910b | 19.18 | 0.765 | 0.274 | 19.91 | 0.773 | 0.278 | 20.86 | 0.776 | 0.304 | 22.76 | 0.820 | 0.240 |
| 7705a2edd0 | 16.74 | 0.698 | 0.398 | 16.78 | 0.712 | 0.396 | 18.89 | 0.715 | 0.440 | 21.71 | 0.792 | 0.350 |
| 71b2dc8a2a | 15.67 | 0.796 | 0.264 | 15.94 | 0.814 | 0.252 | 20.42 | 0.857 | 0.252 | 21.64 | 0.887 | 0.199 |
| a726c1112a | 18.60 | 0.804 | 0.321 | 19.45 | 0.832 | 0.295 | 17.02 | 0.726 | 0.423 | 20.00 | 0.83 | 0.297 |
| cbd44beb04 | 16.46 | 0.700 | 0.311 | 17.40 | 0.728 | 0.299 | 17.35 | 0.706 | 0.349 | 19.38 | 0.789 | 0.285 |
| df4f9d9a0a | 17.21 | 0.743 | 0.358 | 18.04 | 0.768 | 0.344 | 19.36 | 0.777 | 0.356 | 21.82 | 0.845 | 0.268 |
| 6d22162561 | 15.79 | 0.663 | 0.398 | 16.62 | 0.681 | 0.401 | 18.07 | 0.671 | 0.441 | 20.58 | 0.737 | 0.372 |
| 6d81c5ab0d | 13.19 | 0.540 | 0.448 | 14.22 | 0.601 | 0.425 | 14.47 | 0.573 | 0.492 | 16.42 | 0.634 | 0.448 |
| ec305787b7 | 16.75 | 0.751 | 0.286 | 16.98 | 0.763 | 0.278 | 16.78 | 0.688 | 0.381 | 22.122 | 0.846 | 0.211 |
| 85cd0e9211 | 18.17 | 0.758 | 0.285 | 18.45 | 0.767 | 0.289 | 18.90 | 0.688 | 0.372 | 22.73 | 0.814 | 0.269 |
| 95e4b24092 | 13.97 | 0.581 | 0.353 | 13.66 | 0.596 | 0.351 | 14.99 | 0.598 | 0.373 | 15.99 | 0.609 | 0.345 |
| 7da3db9905 | 16.51 | 0.737 | 0.309 | 18.69 | 0.778 | 0.285 | 19.98 | 0.765 | 0.314 | 22.09 | 0.831 | 0.231 |
| d3812aad53 | 15.09 | 0.607 | 0.454 | 16.16 | 0.654 | 0.438 | 16.80 | 0.662 | 0.451 | 17.43 | 0.684 | 0.428 |
| b0c4613d6c | 15.10 | 0.612 | 0.332 | 15.54 | 0.623 | 0.336 | 17.71 | 0.637 | 0.368 | 19.00 | 0.668 | 0.324 |
| b4f53094fd | 13.48 | 0.634 | 0.306 | 14.28 | 0.653 | 0.299 | 17.17 | 0.677 | 0.311 | 18.59 | 0.698 | 0.271 |
| average | 16.56 | 0.717 | 0.323 | 17.22 | 0.740 | 0.314 | 18.25 | 0.722 | 0.363 | 20.47 | 0.788 | 0.284 |

Table 2. Quantitative comparison on DL3DV datasets. Each method is trained on 7000 steps.

| | 2DGS | | | 3DGS | | | FSGS | | | Ours | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| barn | 16.80 | 0.675 | 0.371 | 17.64 | 0.685 | 0.377 | 18.47 | 0.677 | 0.405 | 17.84 | 0.672 | 0.402 |
| ignatius | 15.75 | 0.588 | 0.329 | 15.88 | 0.591 | 0.359 | 16.14 | 0.521 | 0.458 | 17.51 | 0.614 | 0.363 |
| meetingroom | 17.63 | 0.672 | 0.364 | 17.80 | 0.694 | 0.356 | 17.71 | 0.667 | 0.421 | 19.37 | 0.733 | 0.348 |
| truck | 14.66 | 0.646 | 0.357 | 15.39 | 0.663 | 0.361 | 16.69 | 0.654 | 0.407 | 16.80 | 0.673 | 0.383 |
| courthouse | 14.80 | 0.630 | 0.411 | 15.15 | 0.640 | 0.419 | 15.80 | 0.632 | 0.454 | 15.68 | 0.622 | 0.461 |
| caterpillar | 13.79 | 0.532 | 0.403 | 14.33 | 0.542 | 0.431 | 15.35 | 0.530 | 0.490 | 16.58 | 0.580 | 0.432 |
| train | 13.77 | 0.561 | 0.423 | 14.31 | 0.587 | 0.424 | 14.47 | 0.528 | 0.516 | 15.34 | 0.580 | 0.458 |
| average | 15.31 | 0.615 | 0.380 | 15.79 | 0.629 | 0.390 | 16.38 | 0.601 | 0.450 | 17.01 | 0.639 | 0.406 |

Table 3. Quantitative comparison on TnT datasets. Each method is trained on 7000 steps with 1/2 frames

| | 2DGS | | | 3DGS | | | FSGS | | | Ours | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| 3 Views | | | | | | | | | | | | |
| bicycle | 12.70 | 0.124 | 0.622 | 14.33 | 0.300 | 0.556 | 14.30 | 0.234 | 0.624 | 15.46 | 0.275 | 0.647 |
| bonsai | 11.60 | 0.300 | 0.568 | 10.92 | 0.301 | 0.736 | 13.75 | 0.376 | 0.524 | 14.12 | 0.418 | 0.534 |
| counter | 13.17 | 0.311 | 0.539 | 12.62 | 0.305 | 0.597 | 13.99 | 0.392 | 0.527 | 15.20 | 0.470 | 0.520 |
| garden | 13.06 | 0.184 | 0.575 | 12.08 | 0.145 | 0.649 | 14.33 | 0.274 | 0.586 | 16.65 | 0.305 | 0.580 |
| room | 13.79 | 0.410 | 0.490 | 13.04 | 0.342 | 0600 | 14.26 | 0.483 | 0.484 | 16.40 | 0.570 | 0.438 |
| stump | 14.63 | 0.171 | 0.593 | 14.10 | 0.196 | 0.626 | 15.93 | 0.276 | 0.607 | 17.13 | 0.317 | 0.640 |
| kitchen | 14.07 | 0.307 | 0.542 | 13.35 | 0.257 | 0.621 | 14.76 | 0.361 | 0.538 | 16.02 | 0.427 | 0.542 |
| flowers | 10.57 | 0.104 | 0.657 | 10.08 | 0.129 | 0.794 | 12.17 | 0.177 | 0.664 | 12.89 | 0.210 | 0.715 |
| treehill | 11.95 | 0.186 | 0.627 | 11.22 | 0.200 | 0.793 | 14.10 | 0.290 | 0.647 | 12.89 | 0.326 | 0.652 |
| average | 13.06 | 0.318 | 0.576 | 13.07 | 0.243 | 0.580 | 14.17 | 0.318 | 0.578 | 15.29 | 0.367 | 0.585 |
| 6 Views | | | | | | | | | | | | |
| bicycle | 14.35 | 0.188 | 0.576 | 12.92 | 0.181 | 0.663 | 15.76 | 0.294 | 0.597 | 16.52 | 0.311 | 0.604 |
| bonsai | 14.77 | 0.471 | 0.457 | 13.07 | 0.373 | 0.602 | 16.67 | 0.546 | 0.436 | 16.55 | 0.557 | 0.441 |
| counter | 15.09 | 0.428 | 0.467 | 13.77 | 0.352 | 0.535 | 16.02 | 0.495 | 0.449 | 16.99 | 0.545 | 0.428 |
| garden | 16.06 | 0.308 | 0.465 | 14.03 | 0.201 | 0.569 | 17.57 | 0.401 | 0.504 | 18.74 | 0.406 | 0.490 |
| room | 14.80 | 0.481 | 0.446 | 13.98 | 0.426 | 0.564 | 15.22 | 0.542 | 0.443 | 17.54 | 0.623 | 0.410 |
| stump | 16.13 | 0.229 | 0.556 | 14.62 | 0.201 | 0.609 | 17.58 | 0.323 | 0.582 | 18.36 | 0.343 | 0.585 |
| kitchen | 17.12 | 0.494 | 0.397 | 15.11 | 0.321 | 0.530 | 17.64 | 0.577 | 0.374 | 18.54 | 0.560 | 0.390 |
| flowers | 11.89 | 0.145 | 0.607 | 10.89 | 0.147 | 0.757 | 13.21 | 0.211 | 0.649 | 14.01 | 0.237 | 0.658 |
| treehill | 13.33 | 0.240 | 0.584 | 12.10 | 0.222 | 0.741 | 15.46 | 0.347 | 0.613 | 15.36 | 0.363 | 0.605 |
| average | 14.96 | 0.355 | 0.505 | 15.02 | 0.338 | 0.506 | 16.12 | 0.415 | 0.517 | 17.16 | 0.447 | 0.500 |
| 9 Views | | | | | | | | | | | | |
| bicycle | 15.30 | 0.237 | 0.536 | 13.53 | 0.213 | 0.648 | 17.15 | 0.343 | 0.577 | 17.10 | 0.332 | 0.578 |
| bonsai | 17.43 | 0.609 | 0.373 | 15.51 | 0.460 | 0.482 | 19.30 | 0.669 | 0.356 | 19.31 | 0.662 | 0.354 |
| counter | 16.42 | 0.516 | 0.406 | 14.54 | 0.391 | 0.493 | 17.63 | 0.572 | 0.391 | 18.23 | 0.607 | 0.379 |
| garden | 18.10 | 0.412 | 0.397 | 15.06 | 0.250 | 0.522 | 19.22 | 0.477 | 0.455 | 19.97 | 0.470 | 0.446 |
| room | 17.36 | 0.600 | 0.370 | 15.49 | 0.492 | 0.499 | 18.16 | 0.662 | 0.359 | 19.75 | 0.700 | 0.366 |
| stump | 17.45 | 0.300 | 0.514 | 15.69 | 0.237 | 0.548 | 18.72 | 0.386 | 0.555 | 19.40 | 0.392 | 0.553 |
| kitchen | 19.17 | 0.611 | 0.324 | 16.21 | 0.393 | 0.473 | 20.30 | 0.682 | 0.305 | 20.59 | 0.640 | 0.322 |
| flowers | 13.01 | 0.191 | 0.564 | 12.01 | 0.163 | 0.695 | 14.33 | 0.247 | 0.629 | 14.95 | 0.267 | 0.629 |
| treehill | 14.34 | 0.300 | 0.555 | 13.23 | 0.265 | 0.733 | 15.46 | 0.347 | 0.613 | 15.98 | 0.390 | 0.595 |
| average | 16.79 | 0.447 | 0.446 | 16.67 | 0.423 | 0.449 | 17.94 | 0.492 | 0.471 | 18.36 | 0.496 | 0.465 |

Table 4. Per-scene Quantitative comparison on sparse view reconstruction