Generative Sparse-View Gaussian Splatting

Hanyang Kong Xingyi Yang Xinchao Wang*
National University of Singapore
{hanyang.k, xyang}@u.nus.edu, xinchao@nus.edu.sg

a) Qualitative comparisons with 3 training views: the vanilla 3D/4DGS v.s. ours. b) Comparisons with SOTA methods on the LLFF dataset.

Figure 1. **Our proposed Generative Sparse-view Gaussian Splatting (GS-GS) achieves high-fidelity quality with only three training views.** 1) GS-GS is a general pipeline for static and dynamic scene reconstruction with sparse camera views (left: vanilla GS model, right: ours). 2) Quantitative comparisons with other state-of-the-art methods on the LLFF [22] dataset.

Abstract

Novel view synthesis from limited observations remains a significant challenge due to the lack of information in under-sampled regions, often resulting in noticeable artifacts. We introduce Generative Sparse-view Gaussian Splatting (GS-GS), a general pipeline designed to enhance the rendering quality of 3D/4D Gaussian Splatting (GS) when training views are sparse. Our method generates unseen views using generative models, specifically leveraging pre-trained image diffusion models to iteratively refine view consistency and hallucinate additional images at pseudo views. This approach improves 3D/4D scene reconstruction by explicitly enforcing semantic correspondences during the generation of unseen views, thereby enhancing geometric consistency—unlike purely generative methods that often fail to maintain view consistency. Extensive evaluations on various 3D/4D datasets—including Blender, LLFF, Mip-NeRF360, and Neural 3D Video—demonstrate that our GS-GS outperforms existing state-of-the-art methods in rendering quality without sacrificing efficiency.

1. Introduction

Gaussian splatting [15], known for its efficiency in representing 3D scenes using Gaussian primitives, has achieved impressive results in generating high-quality renderings from dense input views. However, when input views are sparse, this approach struggles to maintain scene fidelity. The lack of sufficient view constraints leads to severe ambiguities in reconstructing scene geometry and appearance, causing degraded performance with visible artifacts, such as incorrect depth estimation, floating structures, and inconsistent colors.

This problem is fundamentally ill-posed due to the under-constrained nature of sparse-view scenarios. Without enough observations, it becomes exceedingly difficult to accurately infer the 3D structure and appearance of a scene. As a result, existing methods either rely heavily on the large multi-view training datasets [5–7, 36, 41, 43, 48] or introduce regularization strategies [8, 18, 27, 51] that are often insufficient to address the inherent limitations posed by sparse views. The challenge is exacerbated in textureless or color-inconsistent areas [15, 23], where traditional reconstruction methods fail to establish reliable correspondences between views.

Our intuition is to leverage generative models to compensate for the missing information in under-sampled regions. Specifically, we propose **GS-GS**, a pipeline designed to enhance the quality of 3D/4D Gaussian splatting reconstructions from sparse inputs.

At the core of our approach is the integration of pre-trained image diffusion models, which are used to generate unseen views and iteratively refine the view consistency of

*Corresponding author.

the reconstructed scene. Unlike conventional generative solutions that often struggle with maintaining geometric consistency, our method incorporates explicit geometric constraints to guide the generation process. By enforcing semantic correspondences when generating new views, we ensure that the synthesized content aligns well with the underlying scene structure, preserving both geometric and photometric accuracy.

GS-GS introduces three key innovations: First, we propose a joint optimization approach for both the pre-trained diffusion model and the Gaussian Splatting model. The training datasets for both models are iteratively updated, with the images generated by the diffusion model at novel views improving the quality of Gaussian Splatting, and the rendered images from Gaussian Splatting enhancing the diffusion model to generate more scene-specific outputs. Second, we introduce a geometry-aware diffusion fine-tuning strategy to ensure geometry consistency across camera views, which is essential for producing consistent and realistic reconstructions by aligning the generative model's outputs with the true scene structure. Finally, we incorporate a depth regularization term when optimizing the Gaussian Splatting model, enabling more detailed and accurate geometry information.

To evaluate the effectiveness of our approach, we conducted extensive experiments on a variety of benchmarks, including the Blender, LLFF, Mip-NeRF360, and Neural 3D Video datasets. Our results demonstrate that GS-GS outperforms state-of-the-art methods in terms of both visual quality and reconstruction accuracy. This performance is achieved without sacrificing efficiency, as our method retains the speed and scalability benefits of Gaussian splatting.

In summary, our contributions are as follows:

1. We introduce GS-GS, a novel pipeline that significantly improves 3D/4D Gaussian splatting reconstructions from sparse input views by leveraging pre-trained diffusion models for view generation.
2. We enforce geometric consistency through semantic correspondences, ensuring that generated views align accurately with the scene's structure, which addresses common issues of view inconsistency in generative methods.
3. We demonstrate that GS-GS achieves state-of-the-art performance across multiple datasets, producing high-quality novel view synthesis with minimal input views and achieving results comparable to models trained with dense input data.

2. Related Works

Novel view synthesis. Recent advancements in neural rendering, such as Neural Radiance Fields (NeRFs)[23], have significantly improved novel view synthesis. Many studies[1–3, 25] have focused on enhancing the quality and efficiency of differentiable volume rendering. More recently, 3D Gaussian Splatting (3DGS)[15] has enabled real-time, high-fidelity view synthesis for various scenes, including objects and unbounded environments. Extensions of 3DGS [9, 17, 20, 35, 42, 46, 49] have been developed for dynamic scenes. Despite these advances, reconstructing scenes from sparse camera views remains challenging, and creating a general pipeline for 3DGS/4DGS methods in sparse-view settings is still an open area of research.

Sparse-view novel view synthesis. Few-shot novel view synthesis aims to generate novel views from a sparse set of input views. Various methods [18, 27, 39, 44, 51] address this challenge by adding regularization to NeRF [23] and 3DGS [15] to enhance geometry under sparse inputs. For example, the depth regularization [8, 18, 26] is one of the most widely used techniques. In addition, feed-forward models [5–7, 36, 41, 43, 48] have been developed for sparse-view reconstruction, often trained on large multi-view datasets [21, 50]. Despite promising results, these models often struggle when input images largly differ from the training data, leading to reduced performance.

Lifting 2D diffusion models for 3D/4D generation. Leveraging 2D diffusion priors for generating 3D/4D content [28, 34, 37, 47] has seen significant advancements with pre-trained text-to-image/video diffusion models [4, 31]. For instance, DreamFusion [28] uses Score Distillation Sampling (SDS) to generate 3D objects from text prompts. Later methods [10, 11, 40, 45] enhance temporal and multi-view modeling by adding control signals to diffusion models. Although these approaches achieve high-quality results, they are designed for content generation and lack precision for accurate scene reconstruction. In contrast, our method focuses on improving 3D/4D reconstruction from sparse views, offering greater generalizability and reliability.

3. Preliminary

3.1. Gaussian Splatting

3D Gaussian Splatting [15] (3DGS) optimizes a set of 3D Gaussians through differentiable rasterization to represent a static 3D scene using images captured from multiple camera views with known extrinsic parameters. The process starts with 3D point clouds reconstructed through Structure-from-Motion (SfM). Each Gaussian primitive $G(\boldsymbol{x})$ is described by a position $\boldsymbol{\mu}_i$, a scaling factor $\boldsymbol{s}$, and a rotation quaternion $\boldsymbol{q}$. The basic function of the i-th Gaussian primitive is defined as:

$$G(\boldsymbol{x}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \tag{1}$$

where $\boldsymbol{x}$ is a 3D point location within the 3D scene. $\boldsymbol{\Sigma}$ is formulated using a scaling matrix $\boldsymbol{S}$ and rotation matrix $\boldsymbol{R}$:

$$\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{R}^T. \tag{2}$$

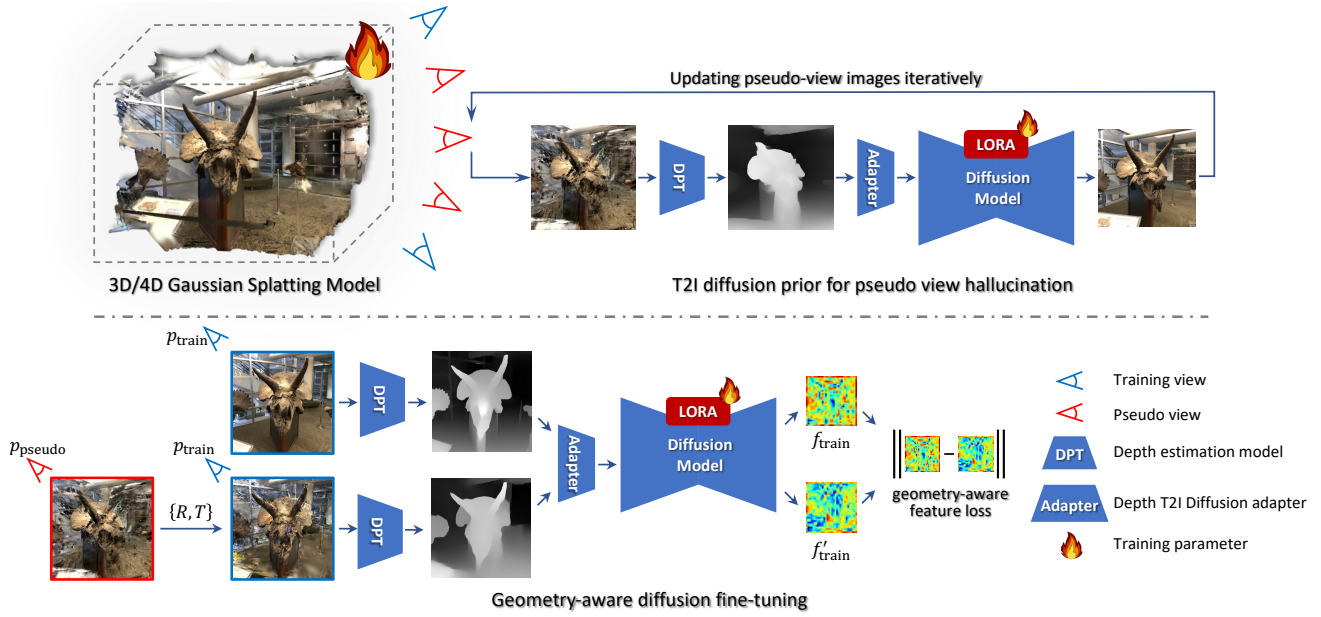

Figure 2. **The pipeline of our method.** Our method jointly optimizes the vanilla 3D/4D Gaussian Splatting model and the LoRA modules inserted to the pre-trained diffusion model by iteratively generating the pseudo-view images. 1) an image is rendered from an arbitrary pseudo viewpoint, 2) the depth map of the image is estimated by a depth estimation model, 3) depth-guided hallucination image at the pseudo view is generated by the pre-trained diffusion model and the depth adpater, 4) the hallucinated images are set as additional training dataset for training 3D/4DGS model and the parameters of inserted LoRA module of diffusion model are optimized based on the images at the training and pseudo views. For generating images with geometry consistency across various camera views, we constrain the diffusion features to be the same by warping images with known camera extrinsics. Specifically, an image rendered at a pseudo camera view is firstly warped to a training camera view with known camera poses $\{R, T\}$. The warped image and the ground-truth image at the training view are fed into the pseudo view hallucination pipeline to obtain the their corresponding diffusion feature f_{train} and f'_{train}. The parameters of LoRA module are optimized with the additional geometry-aware feature loss.

To render an image from the given view, 3D Gaussians are first projected to 2D, and the rendered pixel value $\boldsymbol{C}$ is formulated as:

$$\boldsymbol{C} = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_i), \tag{3}$$

where c_i and α_i denote the color and density of the point, and N is the number of sorted Gaussians contributing to the rendering process.

To represent 4D dynamic scenes, various strategies have been proposed to model the attributes of each Gaussian over time [20, 35, 42, 49]. In this work, we use vanilla 3DGS [15] for static scenes and SpacetimeGS [20], a state-of-the-art multi-view 4DGS method, as the base models for dynamic scenes.

3.2. Diffusion Models

A diffusion model [12] consists of a forward noising process, which gradually adds noise to an image, and a denoising process that iteratively removes the noise to reconstruct a valid image. In this work, we leverage Stable Diffusion [31] (SD), a text-conditioned latent diffusion model (LDM). SD employs an autoencoder [16] to map images to a latent space and a modified UNet [32] denoising network, $\boldsymbol{\epsilon}_{\boldsymbol{\Theta}}$, parameterized by $\boldsymbol{\Theta}$, to perform denoising in the latent space. The encoder $\mathcal{E}$ embeds an input image I into a latent representation $z = \mathcal{E}(I)$, and the decoder $\mathcal{D}$ reconstructs the image from z as $I' = \mathcal{D}(z)$. The LDM is optimized using the following equation:

$$\mathcal{L}_{\text{LDM}}(I, y; \boldsymbol{\Theta}) = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0,1), t} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\Theta}}(z_t, t, \tau(y)) \right\|_2^2, \tag{4}$$

where y is the text prompts, τ is the text encoder (*e.g.*, CLIP [29]), and t is the timestep.

DreamBooth [33] is a subject-driven method for SD, which fine-tuning all parameters of the UNet or insert low rank adaptation [13] (LoRA) to generate personalizied images.

T2I-adapter [24] is a lightweight module to align internal knowledge in SD with external control signals, achieving rich control and editing effects in the color and structure of the generation results. In this work, we apply a pre-trained depth T2I-adapter [24] as the foundation module of our pipeline.

4. Generative Sparse-View Gaussian Splatting

4.1. Overview

Problem Definition. Given a set of input images $\mathcal{I} = \{I_1, I_2, \cdots, I_N\}$ with known camera poses $\mathcal{P} = \{P_1, P_2, \cdots, P_N\}$, the goal is to generate a set of images $\tilde{\mathcal{I}} = \left\{\tilde{I}_1, \tilde{I}_2, \cdots, \tilde{I}_{\tilde{N}}\right\}$ from pseudo-novel views $\tilde{\mathcal{P}} = \left\{\tilde{P}_1, \tilde{P}_2, \cdots, \tilde{P}_{\tilde{N}}\right\}$. These pseudo-novel views are interpolated from the original camera poses $\mathcal{P}$. Using a pre-trained text-to-image diffusion model, we generate the images $\tilde{\mathcal{I}}$ for these interpolated views, which are then used to update the 3D/4D scene representation.

Alternating Optimization. Consider a conditional diffusion model from which we can sample images using $I = \texttt{sample}[\boldsymbol{\epsilon}_{\boldsymbol{\Theta}}(y)]$, where y is the conditioning input and $\boldsymbol{\epsilon}_{\boldsymbol{\Theta}}$ represents the diffusion model parameterized by $\boldsymbol{\Theta}$. Our ultimate objective is to optimize the 3D/4D representation parameterized by $\boldsymbol{\theta}$. The above problem can be formularized as a bi-level optimization problem as:

$$\begin{aligned} &\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \mathcal{L}_{\text{GS}}\big(R(P_i;\boldsymbol{\theta}), I_i\big) + \sum_{j=1}^{\tilde{N}} \mathcal{L}_{\text{GS}}\big(R(\tilde{P}_j;\boldsymbol{\theta}), \tilde{I}_j\big) && \text{[Scene Reconstruction]} \\ &s.t. \quad \boldsymbol{\Theta}^* = \arg\min_{\boldsymbol{\Theta}} \sum_{i=1}^{N} L_{\text{LDM}}(I_i, y_i; \boldsymbol{\Theta}), && \text{[Model Adaptation]} \\ &\tilde{I}_j = \texttt{sample}[\boldsymbol{\epsilon}_{\boldsymbol{\Theta}^*}(\tilde{y}_j)] && \text{[View Generation]} \end{aligned} \tag{5}$$

where $R(\cdot;\boldsymbol{\theta})$ is the rendering function, and $\mathcal{L}_{\text{GS}}$ is the reconstruction loss between the rendered images and the observations. The variables P_i and $\tilde{P}_j$ denote the original and pseudo-novel camera poses, respectively.

To solve the optimization problem above, we alternating between 3 steps. We start by adapting the diffusion model to the available views. Using the pre-trained depth adapter [24], we generate new training images at pseudo-novel views. These images are then used to optimize the parameters of the 3D/4D GS model.

We loop over these three steps iteratively. In each iteration, the 3D/4D representation renders new images, which serve as supplementary data for further fine-tuning the diffusion model. This iterative process gradually improves the quality of the 3D/4D reconstruction as more data becomes available.

4.2. Geometry-aware Pseudo View Hallucination

Our goal is to hallucinate the images for novel pseudo views, which facilitates the optimization of 3D/4D scene reconstruction under sparse-view setting. To achieve this, we need to guarantee that the pre-trained diffusion model can generate scene-specific images with camera-view-related condition. Specifically, we use LoRA [13] and a pre-trained depth adapter [24] to train the diffusion model. This setup allows us to generate scene-specific images based on the given depth image. An overview of the entire pipeline is shown in Fig. 2.

As shown in Sec. 4.1, our pipeline involves a joint optimization, in which the training dataset for LoRA and 3D/4DGS model are iteratively updated by each other. This iterative process allows for gradual percolation of the diffusion priors to static or dynamic scene and personalize the diffusion model to the specific scene.

When optimization begins, we optimize the GS model based on the sparse training view to initialize a coarse 3D/4D scene. Then we update the training images for LoRA model and GS model, iteratively. Specifically, 3D/4DGS models firstly render all images at training views $\{P_1, P_2, \cdots, P_N\}$ and pseudo views $\left\{\tilde{P}_1, \tilde{P}_2, \cdots, \tilde{P}_{\tilde{N}}\right\}$. All the rendered images serves as the training dataset for fine-tuning LoRA model. After the LoRA parameters optimized for several iterations, we generate the images $\left\{\tilde{I}_1, \tilde{I}_2, \cdots, \tilde{I}_{\tilde{N}}\right\}$ by the fine-tuned diffusion model given the estimated depth maps at pseudo views $\left\{\tilde{P}_1, \tilde{P}_2, \cdots, \tilde{P}_{\tilde{N}}\right\}$. The GS model are further optimized by the ground truth images $\{I_1, I_2, \cdots, I_N\}$ and the hallucinated images $\left\{\tilde{I}_1, \tilde{I}_2, \cdots, \tilde{I}_{\tilde{N}}\right\}$. The overall procedure is illustrated in Alg. 1.

Algorithm 1 Joint Optimization for 3D/4DGS with Diffusion Prior

Input: Sparse-view images $\mathcal{I}$ at views $\mathcal{P}$, pseudo views $\tilde{\mathcal{P}}$, pre-trained diffusion model, depth adapter, LoRA module.
for initialize iteration **do**
  Train GS with $\mathcal{I}$ at views $\mathcal{P}$
end for
for training iteration **do**
  Render images $\mathcal{I}$ and $\tilde{\mathcal{I}}$ from GS at views $\mathcal{P}$ and $\tilde{\mathcal{P}}$.
  for fine-tuning LoRA **do**
    Optimize LoRA parameters with $\mathcal{I}$ and $\tilde{\mathcal{I}}$.
  end for
  Update refined pseudo-view images $\tilde{\mathcal{I}}$ at views $\mathcal{P}$ by the fine-tuned diffusion model.
  for optimizing GS **do**
    Optimize GS model with $\mathcal{I}$ and $\tilde{\mathcal{I}}$.
  end for
end for
Output: Optimized 3D/4DGS model.

4.3. Geometry-aware Diffusion Fine-Tuning

The main challenge in hallucinating images at pseudo views is maintaining geometry consistency across views. While the LoRA module personalizes the pre-trained diffusion model

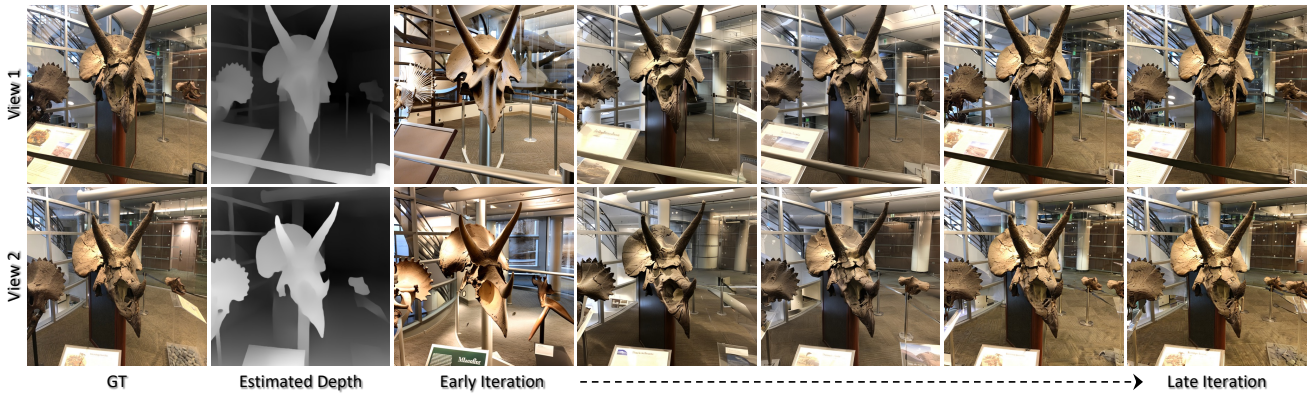

Figure 3. **Dataset Evolution.** We illustrate the dataset evolution process for the horn scene using the LLFF [22] dataset. Starting with two camera views, the diffusion model generates geometry-consistent image pairs as the training dataset is updated iteratively.

to a specific scene, ensuring geometry consistency remains difficult, especially with depth maps from two camera views. Recent work by DIFT [38] demonstrates that the intermediate features of diffusion models contain strong semantic information and can establish correspondences between images. To address this, we propose a geometry-aware fine-tuning strategy for updating the LoRA parameters of the diffusion model.

The geometry-aware diffusion fine-tuning pipeline is illustrated in Fig. 2. Given a image $\tilde{I}$ rendered by 3D/4DGS at the pseudo view $\tilde{P} \in \left\{\tilde{P}_1, \tilde{P}_2, \cdots, \tilde{P}_{\tilde{N}}\right\}$, we first warp it to the known training view $P \in \{P_1, P_2, \cdots, P_N\}$ with known camera entrinsic parameter $\{R, T\}$, denoting $\tilde{I}_{R,T}$. The depth maps of the ground-truth image I at view P and the warped image $\tilde{I}_{R,T}$ are estimated by the pre-trained Dense Prediction Transformer (DPT) [30], serving the condition of the diffusion model. The diffusion features of images I and $\tilde{I}_{R,T}$, denoting f_{train} and f'_{train} are extracted by the diffusion model with the inserted fine-tuned LoRA module. Ideally, the diffusion features f_{train} and f'_{train} should be the same since the camera extrinsic parameters of I and $\tilde{I}_{R,T}$ are the same. To this end, we calculate the L1 loss for f_{train} and f'_{train} as the regularization term when fine-tuning LoRA parameters. The loss function for fine-tuning LoRA module formulate as:

$$\begin{aligned}\mathcal{L}_{\mathbf{\Theta}} = \mathcal{L}_{LDM}(I, y; \mathbf{\Theta}) + \mathcal{L}_{LDM}(\tilde{I}_{R,T}, y; \mathbf{\Theta}) \\ + \lambda_{geo} \left\| f_{train}, f'_{train} \right\|,\end{aligned} \tag{6}$$

where $\mathcal{L}_{LDM}(\cdot, y; \mathbf{\Theta})$ is the optimization loss for LDM with optimized parameters $\mathbf{\Theta}$, as depict in Eq. (4). λ_{geo} represents the geometric regularization term for fine-tuning LoRA module. We set $\lambda_{geo} = 0.1$ in all experiments.

We visualize the evolution of hallucinated images generated by the fine-tuned diffusion model in Fig. 3. Specifically, we show the ground-truth images, estimated depth maps, and hallucinated images at different training iterations for two camera views. Initially, the generated images at pseudo views are inconsistent across camera views. However, as training progresses, the hallucinated images become more geometry-consistent. The high-fidelity, 3D-consistent images at pseudo views facilitate the optimization of 3D and 4D scene representations in a sparse-view setting.

4.4. Depth Regularization for Gaussian Optimization

Although the pseudo-view hallucination pipeline generates additional images with geometry awareness, the resulting images are not perfectly aligned with the scene, leading to blurring in the rendered results. To inject richer geometric information into the Gaussian optimization, we incorporate pixel-level geometric correspondences as regularization during 3D/4DGS training. Specifically, for each camera view P in the training views $\mathcal{P}$ and pseudo views $\tilde{\mathcal{P}}$, we first render an image I at view P. We then compute the structural similarity between the rasterized depth D_{render} at view $\mathcal{P}$ and the estimated monocular depth of the rendered image I. This structural similarity serves as a regularization term when optimizing the 3D/4DGS model. We now detail this procedure.

Similar to the differentiable rasterization for RGB images in Eq. (3), the rendered depth map D_{GS} at camera views $\mathcal{P}$ and $\tilde{\mathcal{P}}$ are calculated by:

$$D_{\text{GS}} = \sum_{i \in N} d_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_i), \tag{7}$$

where d_i and α_i denote the z-buffer and density of the Gaussian point, and N is identical to that in Eq. (3).

For each image I at either the ground-truth training views $\mathcal{P}$ or the generated pseudo views $\tilde{\mathcal{P}}$, we first estimate its monocular depth map D_{DPT} by the pre-trained Dense Prediction Transformer (DPT) [30]. To mitigate depth ambiguity between the estimated depth D_{DPT} and the rasterized depth

map D_{GS}, D_{DPT} and D_{GS} are further normalized by the following equation:

$$D_{\text{GS}}^{norm} = \frac{D_{\text{GS}} - \mu_{D_{\text{GS}}}}{\sigma_{D_{\text{GS}}}}, \tag{8}$$

and

$$D_{\text{DPT}}^{norm} = \frac{D_{\text{DPT}} - \mu_{D_{\text{DPT}}}}{\sigma_{D_{\text{DPT}}}}, \tag{9}$$

where μ_{D^*} and the σ_{D^*} are the mean and variance values of D_{DPT} and D_{GS}, respectively.

Rather than previous methods [18, 51] which calculate L1 or L2 distance between D_{DPT} and D_{GS}, we calculate the multi-scale structural similarity (MS-SSIM) between D_{DPT} and D_{GS} by

$$\mathcal{L}_{\text{reg}} = -\text{MS-SSIM}(D_{\text{DPT}}^{norm}, D_{\text{GS}}^{norm}). \tag{10}$$

4.5. Optimization

The 3D/4DGS model is optimized by the regular photometric loss terms with depth regularization.

$$\begin{aligned}\mathcal{L}_{\text{GS}}\big(R(\cdot;\boldsymbol{\theta})\big) = \mathcal{L}_1(R(\mathcal{P};\boldsymbol{\theta}),\mathcal{I}) + \mathcal{L}_1(R(\tilde{\mathcal{P}};\boldsymbol{\theta}),\tilde{\mathcal{I}}) \\ + \lambda\mathcal{L}_{\text{D-SSIM}}(R(\mathcal{P};\boldsymbol{\theta}),\mathcal{I}) + \lambda\mathcal{L}_{\text{D-SSIM}}(R(\tilde{\mathcal{P}};\boldsymbol{\theta}),\tilde{\mathcal{I}}) \\ + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}},\end{aligned} \tag{11}$$

where, $R(\cdot;\boldsymbol{\theta})$ is the rendering function, $\boldsymbol{\theta}$ is the parameter of 3D/4D Gaussian representation. $\mathcal{L}_1$ and $\mathcal{L}_{\text{D-SSIM}}$ are the L1 reconstruction loss and D-SSIM loss, respectively.

5. Experiments

5.1. Datasets

We conduct our experiments on four datasets: the NeRF Blender Synthetic dataset (Blender) [23], the LLFF dataset [22], the Mip-NeRF-360 dataset [2], and the Neural 3D Video dataset [19].

NeRF Blender Synthetic dataset (Blender) [23] consists of 8 object with realistic images rendered by Blender. Aligned with [14, 18, 51], we use 8 views for training and 25 views for testing.

LLFF dataset [22] consists of eight real-world scenes. Following baseline methods [18, 27, 51], we select every 8^{th} image as the test set, and sample 3 views from the remaining views for training evenly.

Mip-NeRF360 dataset [2] consists of 9 real-world complex ourdoor scenes. Following [51], the test images are selected the same as the LLFF dataset [22] and the training views are 24.

Neural 3D Video dataset [19] contains six indoor multi-view video sequences captured by 18 to 21 cameras. Following other methods [20, 35], the first camera is set for evaluation. Since there are no method designed specifically for sparse dynamic scene generation, we evenly sample 3 views from the all other views for training.

5.2. Experimental Settings

Baselines. We compare our method with several state-of-the-art methods on these datasets, including Mip-NeRF [1], DietNeRF[14], RegNeRF [27], FreeNeRF [44], SparseNeRF [39], 3DGS [15], DNGaussian [18], and FSGS [51]. We report PSNR, SSIM, and LPIPS scores to evaluate the reconstruction performance quantitatively. For sparse view dynamic scene reconstruction, we conduct experiments based on SpacetimeGS [20] under sparse-view setting.

Implementation details. We build our pipeline based on the official 3DGS [15] and SpacetimeGS [20] for sparse-view 3D and 4D scene reconstruction, respectively. We train the overall pipeline with $10,000$ iteration for all datasets. We apply pre-trained DPT [30] to estimate monocular depth for all training and pseudo views. The initial point clouds for each scene are reconstructed by Structure-from-Motion (SfM) with the sparse training views. Please refer to the supplementary material for details.

5.3. Few-shot 3D Reconstruction

Table 1. **Quantitative comparisons on the Blender, LLFF, and Mip-NeRF360 datasets.** We color each cell as the best, second best, and third best.

Method	Blender			LLFF			Mip-NeRF360		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF [1]	20.89	0.830	0.168	16.11	0.401	0.460	19.51	0.517	0.413
3DGS [15]	21.56	0.847	0.130	17.43	0.522	0.321	20.89	0.588	0.401
DietNeRF [14]	22.50	0.823	0.124	14.94	0.370	0.496	20.21	0.482	0.452
RegNeRF [27]	23.86	0.852	0.105	19.08	0.587	0.336	22.19	0.546	0.398
FreeNeRF [44]	24.26	0.883	0.098	19.63	0.612	0.308	22.78	0.587	0.377
SparseNeRF [39]	24.04	0.876	0.113	19.86	0.328	0.328	22.85	0.600	0.389
DNGaussian [18]	24.31	0.886	0.088	19.12	0.591	0.132	23.09	0.637	0.322
FSGS [51]	24.64	0.895	0.095	20.31	0.652	0.288	23.70	0.693	0.293
Ours	28.57	0.923	0.055	24.82	0.737	0.105	25.87	0.745	0.182

Blender dataset. The quantitative results on the Blender dataset [23] with 8-view setting are reported in Tab. 1. Our method outperforms all other approaches on the Blender dataset [23] across all three evaluation metrics—PSNR, SSIM, and LPIPS. Specifically, it achieves a substantial improvement in PSNR with a value of 28.57, surpassing the second-best method, FSGS, by more than 3 dB. Additionally, our method demonstrates the highest SSIM score of 0.923, reflecting superior structural similarity between the reconstructed and ground-truth images. These results highlight the effectiveness of our approach in accurately reconstructing 3D scenes with high fidelity, both in terms of geometry and visual quality.

LLFF dataset. The evaluation results on the LLFF dataset [22] are shown in Tab. 3. Our method leads in all three metrics, achieving a PSNR of 24.82, SSIM of 0.737, and LPIPS of 0.105, significantly outperforming the 2^{nd}-best method, FSGS [51]. The PSNR improvement over FSGS is approximately 4.5 dB, demonstrating the robustness of our approach in sparse-view reconstruction. Additionally,

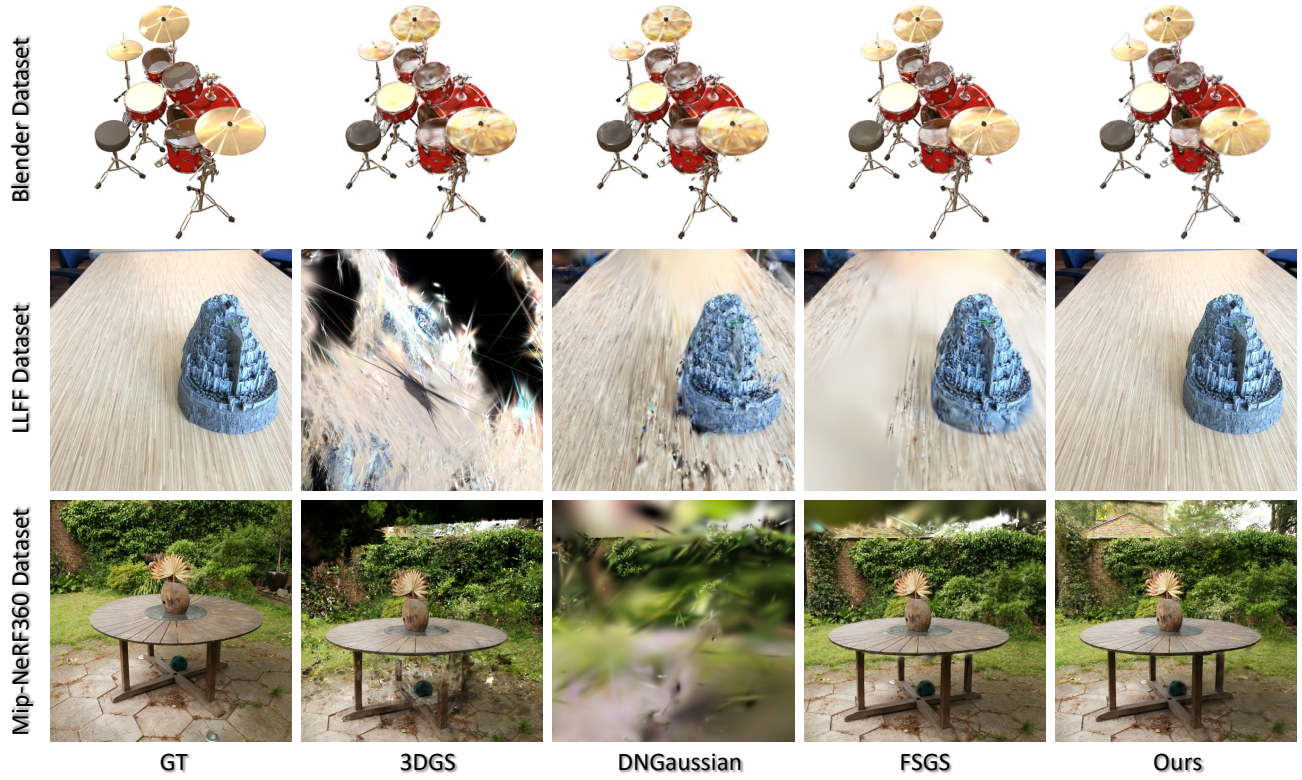

Figure 4. **The visualization results on the Blender [23], LLFF [22], and Mip-NeRF360 datasets.** Our method produces detailed foreground geometry and renders high-quality novel views with sparse camera views. Please refer to the supplementary material for more visualization results.

our SSIM score of 0.737 indicates superior structural alignment, while our LPIPS score of 0.105 reflects the lowest perceptual discrepancy, highlighting the visual quality of our method. These results confirm that our method excels in both geometric accuracy and perceptual realism on the LLFF dataset.

Mip-NeRF360 dataset. The performance of our method on the Mip-NeRF360 dataset [2] is also exceptional, with a PSNR of 25.87, SSIM of 0.745, and LPIPS of 0.182. These results mark the highest scores across all three metrics, outperforming the second-best method, FSGS [51], by a substantial margin. Notably, our PSNR score is the highest, indicating that our method produces the sharpest reconstructions. The SSIM score further confirms the high level of structural consistency with ground-truth images, and our LPIPS score of 0.182 is the best, showing minimal perceptual deviation. These results reinforce the effectiveness of our method in handling the complex, large-scale reconstructions in the Mip-NeRF360 dataset. We visualize the rendering results in Fig. 4. We observe that both 3DGS [15] and DNGaussian [18] fail to capture the geometry and the intricate details of scenes. In comparisons, our method recovers the fine-grained details, such as the Lego in the kitchen scene, aligning well with the ground truth.

5.4. Few-shot Dynamic Scene Reconstruction

Due to there are no proper baselines for sparse-view dynamic scene reconstruction, we compare our method with the baseline 4DGS method, *i.e.*, SpacetimeGS [20]. The evaluation results are listed in Tab. 2. Our method outperforms the baseline SpacetimeGS across all camera view configurations in the Neural 3D Video dataset. Even with just 3 views, our method achieves a PSNR of 27.13, which is over 12 dB higher than SpacetimeGS's 14.98. The SSIM score also reflects this improvement, with our method scoring 0.907 compared to SpacetimeGS's 0.774. Furthermore, our LPIPS score of 0.135 demonstrates a significant reduction in perceptual discrepancy. As the number of views increases to 6 and 9, the gap between our method and SpacetimeGS continues to widen, with our method achieving a PSNR of 30.21, an SSIM of 0.928, and an LPIPS of 0.082 at 9 views. Notably, even when comparing our results with just 3 views to SpacetimeGS's performance with 9 views, our method still outperforms them in all metrics, highlighting the efficiency and effectiveness of our approach in sparse view 4D scene generation. We visualize the rendering results for both SpacetimeGS [20] and our method with 3, 6, and 9 camera views in Fig. 5. SpacetimeGS suffers from

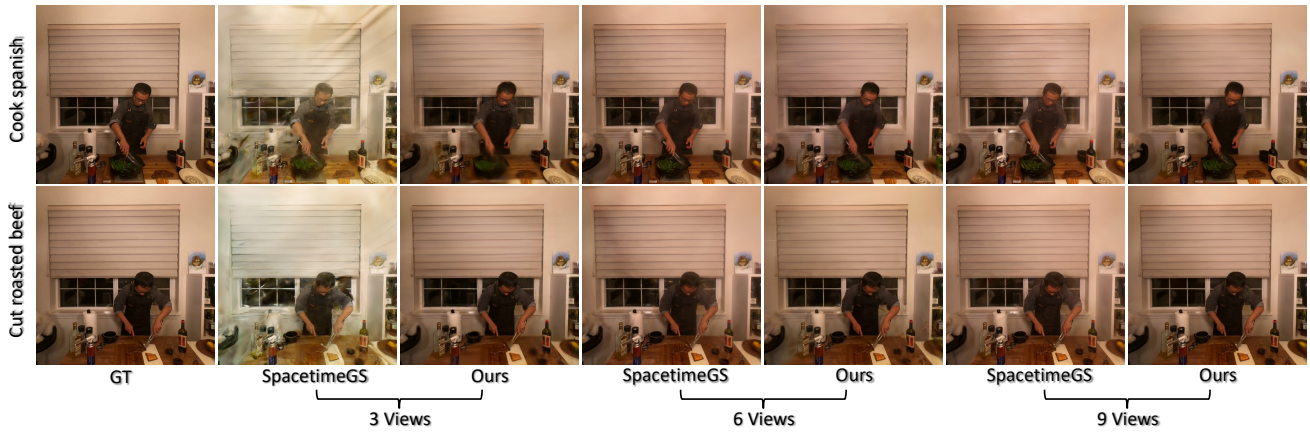

Figure 5. **The visualization results on the Neural 3D Video dataset.** Comparisons are conducted with SpacetimeGS [20] with 3, 6, and 9 training views. Please refer to the supplementary material for more results.

poor results and inaccurate RGB values with only 3 views. Our method achieves much better rendering results, with a high-fidelity appearance. Though both SpacetimeGS and our method improve the rendering results with more training views, our method still achieves higher results with various camera views.

Table 2. **Quantitative comparisons on the Neural 3D Video dataset.** Our proposed pipeline is general for both 3D and 4D scene reconstructions. We set SpacetimeGS [20] as the baseline model for dynamic scene reconstruction and evaluate the performance with different number of camera views.

Method	3 Views			6 Views			9 Views		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
SpacetimeGS [20]	14.98	0.774	0.327	25.15	0.895	0.163	26.72	0.913	0.165
Ours	27.13	0.907	0.135	29.20	0.916	0.117	30.21	0.928	0.082

5.5. Ablation Studies and Analysis

Figure 6. **Ablation studies for the proposed modules.** We ablate for each component of our GS-GS.

Diffusion Hallucination. To demonstrate the effectiveness of our diffusion hallucination strategy, we conduct an ablation study by removing the diffusion prior from the pseudo-view hallucination pipeline. As shown in Tab. 3 and Fig. 6, without the diffusion prior, the 3DGS model struggles to render high-quality novel views due to the limited number of camera views.

Geometry-Aware Fine-Tuning. Introducing the diffusion hallucination pipeline into the optimization of the 3DGS model leads to improved rendering results. As shown in Tab. 3, the PSNR score increases from 17.43 to 22.71 on the LLFF dataset [22]. The flowers depicted in Fig. 6 exhibit better appearance. However, compared to the full model, the rendering results still appear blurry, and the details are not as refined. This is because, although the diffusion process helps hallucinate images at pseudo views, the generated images lack geometry consistency, which negatively impacts scene reconstruction.

Depth Regularization. As shown in Fig. 6, incorporating depth regularization further enhances geometric details, such as the edges of the flowers. With the addition of the depth regularization term, the PSNR score on the LLFF dataset increases from 24.09 to 24.82.

Table 3. **Ablation studies on the proposed components.**

	w/o diffusion hallucination	diffusion w/o geometry-aware fine-tuning	w/o depth reg.	full model
Mip-NeRF360 [2]	20.89	23.23	25.28	25.87
LLFF [22]	17.43	22.71	24.09	24.82

6. Conclusion

In this work, we present Generative Sparse-view Gaussian Splatting (GS-GS), a general pipeline designed to enhance the quality of 3D/4D Gaussian Splatting (GS) with sparse-view inputs. By leveraging pre-trained diffusion models to hallucinate additional views while maintaining semantic and geometric consistency, GS-GS enhances the quality of 3D/4D Gaussian splatting even in under-sampled regions. Through extensive experiments on diverse datasets, we demonstrate that our method significantly outperforms existing state-of-the-art techniques in both reconstruction accuracy and rendering performance, offering a promising solution for high-quality view synthesis in sparse-view settings.

Acknowledgement

This project is supported by the National Research Foundation, Singapore, under its Medium Sized Center for Advanced Robotics Technology Innovation, and the Singapore Ministry of Education Academic Research Fund Tier 1 (WBS: A-0009440-01-00).

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 2, 6

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 6, 7, 8

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 2

[4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[5] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19467, 2024. 1, 2

[6] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *European Conference on Computer Vision*, pages 338–355. Springer, 2025.

[7] 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. In *European Conference on Computer Vision*, pages 370–386. Springer, 2025. 1, 2

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 1, 2

[9] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024. 2

[10] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 2

[11] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024. 2

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3, 4

[14] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 6

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2, 3, 6, 7

[16] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[17] Hanyang Kong, Xingyi Yang, and Xinchao Wang. Efficient gaussian splatting for monocular dynamic scene rendering via sparse time-variant attribute modeling. *arXiv preprint arXiv:2502.20378*, 2025. 2

[18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20775–20785, 2024. 1, 2, 6, 7

[19] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 6

[20] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024. 2, 3, 6, 7, 8

[21] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021. 2

[22] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019. 1, 5, 6, 7, 8

[23] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 6, 7

[24] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4296–4304, 2024. 3, 4

[25] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 2

[26] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 2

[27] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 1, 2, 6

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

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[30] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 5, 6

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 3

[33] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 3

[34] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023. 2

[35] Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20675–20685, 2024. 2, 3, 6

[36] 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. 1, 2

[37] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 2

[38] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023. 5

[39] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9065–9076, 2023. 2, 6

[40] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[41] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. *arXiv preprint arXiv:2403.16292*, 2024. 1, 2

[42] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 2, 3

[43] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21551–21561, 2024. 1, 2

[44] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8254–8263, 2023. 2, 6

[45] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024. 2

[46] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 2

[47] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6796–6807, 2024. 2

[48] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2025. 1, 2

[49] Haoyu Zhao, Chen Yang, Hao Wang, Xingyue Zhao, and Wei Shen. Sg-gs: Photo-realistic animatable human avatars with semantically-guided gaussian splatting. *arXiv preprint arXiv:2408.09665*, 2024. 2, 3

[50] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 2

[51] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, pages 145–163. Springer, 2025. 1, 2, 6, 7