See Through the Occlusions: Few-Shot Gaussian Splatting with Layered Amodal Supervision

Gwon Jung Kim
Hanyang University
Seoul, Republic of Korea
rnjswnd1501@gmail.com

Du Yeol Lee
Hanyang University
Seoul, Republic of Korea
adan910.dyl@gmail.com

Jae Hong Yang
Hanyang University
Seoul, Republic of Korea
jaehongyang@hanyang.ac.kr

Chae Eun Rhee
Hanyang University
Seoul, Republic of Korea
crhee@hanyang.ac.kr

Figure 1: Visualization of STO-GS. Given sparse modal views (left), we generate layered amodal views (middle) to reveal occluded content and guide opacity-aware training for reconstructing hidden regions. As shown in the rendered comparisons (right), STO-GS outperforms prior methods (FSGS [43], CoR-GS [41]) in both geometric and visual consistency, especially for occluded, non-central areas.

Abstract

High-quality three-dimensional (3D) reconstruction from sparse views is critical for applications such as virtual and augmented reality, robotics, and digital content creation. While methods like Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS) have shown strong performance in novel view synthesis, they struggle in few-shot settings, especially when scenes contain large occluded or unseen regions. The lack of explicit supervision for hidden content limits reconstruction completeness and realism. We propose See-Through-the-Occlusion Gaussian Splatting (STO-GS), a novel framework that rethinks occlusion modeling in static scenes. Drawing inspiration from four-dimensional Gaussian Splatting (4DGS), we reinterpret time as a proxy for occlusion depth and apply deformation-based opacity modulation to recover hidden layers. To provide supervision, we generate amodal views via diffusion-based inpainting, exposing occluded structures for training. A two-stage layered training pipeline further refines the reconstruction, with a multi-layer perceptron (MLP) adjusting Gaussian opacity across occlusion layers. STO-GS improves occlusion-aware reconstruction and achieves superior performance over existing few-shot 3DGS baselines, including a 0.51 dB gain on challenging datasets.

CCS Concepts

• **Computing methodologies** → **Image-based rendering**; **Reconstruction**.

Keywords

Novel View Synthesis, 3D Gaussian Splatting, Few-Shot Novel View Synthesis, Amodal Completion, Stable Diffusion

ACM Reference Format:
Gwon Jung Kim, Du Yeol Lee, Jae Hong Yang, and Chae Eun Rhee. 2025. See Through the Occlusions: Few-Shot Gaussian Splatting with Layered Amodal Supervision. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3746027.3755801

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755801

1 Introduction

With the rapid advancement of virtual reality (VR), augmented reality (AR), and the metaverse, the importance of three-dimensional

(3D) scene reconstruction has grown significantly. These applications require real-time generation and interactive manipulation of high-quality 3D content, supported by accurate spatial understanding and efficient rendering. Beyond VR/AR, 3D reconstruction is essential in domains such as autonomous driving, robotics, digital twins, and medical imaging, where precision and efficiency are critical. Traditional methods often rely on expensive sensors or dense image collections, limiting their scalability. Therefore, developing techniques that can produce high-fidelity 3D scenes from minimal input remains a core research challenge.

3D reconstruction approaches can be broadly classified into image-based, sensor-based, graphics-based, and deep learning-based methods. Image-based techniques like structure-from-motion (SfM) [26] and multi-view stereo (MVS) [28] use standard RGB cameras but are sensitive to occlusions and viewpoint changes. Sensor-based methods, using light detection and ranging (LiDAR), RGB-depth (RGB-D) cameras, or time-of-flight (ToF) sensors, offer higher accuracy but suffer from noise, cost, and scalability issues [25, 35]. Graphics-based approaches model geometry using meshes, voxels, or splines, but often involve heavy computation [6, 12, 27]. More recently, deep learning-based methods such as Neural Radiance Fields (NeRF) [2, 5, 7, 9, 22–24, 34, 37] and 3D Gaussian Splatting (3DGS) [10, 14, 20, 29, 40, 41, 43] have enabled high-quality 3D reconstruction by implicitly modeling scenes as continuous functions. In particular, 3DGS represents scenes using multiple Gaussian primitives—parameterized by position, size, orientation, and color—offering efficient training and fast rendering suitable for real-time applications. However, in few-shot settings with limited training images, 3DGS struggles to capture the full scene structure—particularly in occluded regions and under large viewpoint variations where background areas often remain unobserved.

To tackle the limitations of few-shot 3DGS under severe occlusion, we propose See-Through-the-Occlusion Gaussian Splatting (STO-GS)—a novel framework that brings temporal modeling techniques from 4D Gaussian Splatting (4DGS) [8, 16, 18, 33, 36, 38] into the spatial domain. Unlike previous approaches, STO-GS leverages amodal views with multiple occlusion layers, enabling the model to reason about hidden structures and improve occlusion-aware reconstruction. As illustrated in Fig.1, our method generates layered amodal views via image inpainting and uses them to guide opacity-aware learning across occlusion depths. This allows the model to effectively utilize contextual cues beyond visible regions and progressively learn layered scene representations from sparse inputs. Our major contributions are:

• **4DGS-inspired occlusion modeling in static scenes.** For the first time, we transfer temporal modeling techniques from 4DGS to the spatial domain, enabling occlusion layers to be treated analogously to time, which improves reconstruction of hidden structures from sparse views. This forms the basis of our See-Through-the-Occlusion (STO) strategy, which models occlusion as a learnable spatial dimension.

• **Amodal image integration for explicit supervision.** We generate amodal views via deep image inpainting to expose occluded regions in given sparse input images, providing direct supervision that outperforms conventional augmentation strategies in few-shot 3DGS.

• **Layered opacity-aware training scheme.** We introduce a two-stage pipeline that first learns from sparse input images, then refines using randomly sampled amodal views. A dedicated multilayer perceptron (MLP), referred to as STO-MLP, modulates Gaussian opacity per occlusion layer, enabling the model to reveal hidden structures and improve overall scene completeness.

2 Related Work

2.1 Few-Shot Gaussian Splatting

3DGS [14, 20, 29, 40] represents 3D scenes using Gaussian primitives derived from training images, enabling efficient and high-quality rendering from novel viewpoints. However, its performance drops significantly in few-shot settings, where reconstructing unobserved or occluded regions becomes difficult. This limitation has led to growing interest in adapting 3DGS to sparse-view scenarios. An early approach is DNGaussian [17], which reduces dependence on SfM by initializing with randomly placed Gaussians. It introduces hard and soft depth regularization—selectively freezing or adjusting opacity based on depth—and global-local depth normalization to align depth distributions across views, improving stability and accuracy with limited data. Building on synthetic augmentation, FSGS [43] generates pseudo views—intermediate viewpoints synthesized between inputs. Although these views lack ground truth, FSGS enforces depth consistency between predicted and rendered depths. It also uses mesh subdivision and CUDA-based rasterization to place Gaussians in geometrically meaningful regions, improving coverage and visual quality. This method performs well across various few-shot benchmarks. Maintaining color consistency remains challenging in sparse settings. CoR-GS [41] tackles this with color co-regularization between pseudo view pairs rendered from separate Gaussian sets. It also introduces co-pruning to remove redundant or inconsistent Gaussians, improving both efficiency and fidelity. These strategies yield stable reconstructions even in severely under-constrained conditions. More recently, CoMapGS [10] enhances few-shot performance through covisibility map-based adaptive supervision, balancing learning across well- and under-observed regions. It further improves initialization using scaling-aware depth alignment based on monocular depth prediction and triangulation, enabling denser Gaussian distributions in occluded areas while maintaining stability with visibility-guided supervision.

In summary, enhancing initialization, enforcing geometric and photometric consistency, and pruning uninformative elements have proven effective for few-shot Gaussian splatting. However, prior methods primarily focus on visible regions, with limited attention to fully occluded content—an area we aim to address in this work.

2.2 4D Gaussian Splatting

4DGS [8, 16, 18, 33, 36, 38] extends the 3DGS framework by introducing a temporal dimension, enabling dynamic scene reconstruction through time-aware modeling of Gaussian attributes. Unlike earlier NeRF-based methods that regress color and density from spatiotemporal coordinates, 4DGS uses an explicit representation for faster and more efficient rendering, making it suitable for real-time applications. Most 4DGS approaches start with a static Gaussian field and apply temporal deformations to model motion. For example, 4DGS [33] employs voxel-plane-based feature encoding and a

lightweight MLP to predict changes in position, rotation, and scale, allowing a single set of Gaussians to represent multiple frames. SC-GS [8]further reduces complexity by modeling motion via sparse control points, separating appearance from dynamics to improve efficiency and stability. Spacetime-GS [18] takes a fully explicit approach, using parametric functions such as polynomials to describe Gaussian trajectories and opacity over time, reducing parameters while retaining fine control. Ex4DGS [16]combines static-dynamic decomposition with keyframe interpolation to lower memory and computation while preserving quality.

While prior 4DGS works focus on temporal deformation, our key insight is to draw a parallel between time in 4DGS and occlusion depth in 3DGS. Both deal with information invisible from the current view—due to either motion or occlusion. In this work, we reinterpret the 4DGS deformation mechanism to operate across occlusion layers instead of time, enabling few-shot novel view synthesis that reconstructs hidden structures behind foreground objects.

2.3 Amodal Completion

Amodal completion focuses on recovering scene elements that are not directly visible, offering a key advantage in understanding occluded structures for 3D reconstruction. Early work began with amodal segmentation, which extends traditional segmentation by inferring the full extent of partially visible objects. For example, BCNet [13] combines graph-based reasoning with instance-level segmentation to handle both occluders and occludees, improving object boundary inference under occlusion. While segmentation provides semantic cues, geometric understanding requires deeper reasoning—leading to amodal depth estimation, which predicts occluded depth from monocular images. Methods like ADE [11, 19] use occlusion masks and dedicated depth networks to reconstruct hidden geometry, producing denser 3D point clouds from limited views. More recent approaches explore image inpainting as a practical form of amodal completion. Diffusion-based pipelines like Inpaint Anything [39], which integrate segmentation (e.g., SAM [15]) and generative models (e.g., LAMA [30]), remove foreground occluders and synthesize plausible background content. However, single-view methods often suffer from multi-view inconsistency, limiting their use in 3D reconstruction. To address this, multi-view inpainting methods such as MVInpainter [4] refine occluded regions across views while preserving geometric coherence. Yet, these still struggle when reference images lack sufficient background—especially in few-shot scenarios.

In our work, we adopt an inpainting-based amodal view generation strategy to provide explicit supervision for occluded areas. This enables more robust learning of hidden structures from sparse inputs and significantly enhances the quality of occlusion-aware 3D reconstruction.

3 Amodal Layer Learning for Few-Shot 3D Gaussian Splatting

3.1 Amodal Layering

Removing a specific object from an image and restoring the background is a challenging task. Traditional non-learning-based image inpainting methods rely on surrounding pixel colors and textures to fill missing regions, but they often fail to preserve global scene structure. To address this, we generate segmentation-based masks using SAM [15] to identify occluded regions and apply a diffusion-based inpainting model, such as Adobe Firefly [1], guided by text prompts. As illustrated in Fig. 2a, the diffusion model removes the pixels within the masked regions and synthesizes plausible background content based on surrounding context. By iteratively applying this process with segmentation masks targeting different occlusion depths, we generate a sequence of amodal views, each corresponding to a different occlusion layer. For example, the original input image serves as the modal view, containing all visible foreground objects. The first amodal layer removes the nearest occluders to reveal the immediately hidden background, and subsequent layers progressively eliminate deeper occluding objects, uncovering farther parts of the scene in order of occlusion depth. This layered decomposition of the scene via amodal completion enables us to build structured supervision across occlusion depths. The resulting amodal views are not required as direct inputs, but instead serve as data augmentation to complement the original modal views during training. By exposing occluded regions, they enhance occlusion-aware learning and contribute to more complete 3D scene reconstruction from sparse inputs.

3.2 Learning to See Through with Opacity Modulation

We reinterpret the deformation network structure of 4DGS—originally designed to handle temporal changes in dynamic scenes—as a mechanism for modeling variations in occlusion depth. While 4DGS dynamically updates Gaussian attributes such as position, orientation, scale, and color over time, our method operates in a static 3DGS setting and modulates only the opacity α of each Gaussian. Specifically, occluded Gaussians behind foreground objects can be directly supervised through amodal views by progressively decreasing the opacity of the foreground Gaussians, allowing the model to access and learn from hidden structures. This effectively transforms temporal deformation into occlusion-aware spatial reasoning, leading to more complete scene understanding.

Fig. 2b illustrates the layered structure of Gaussians used to represent a 3D scene. The camera observes along the xy plane, and the scene is divided into three occlusion layers ($l = 0$, $l = 1$, and $l = 2$) based on depth. The blue Gaussians in the front layer ($l = 0$) are fully visible and thus easier to supervise, whereas the orange ($l = 1$) and green ($l = 2$) Gaussians are often occluded by foreground objects and may be only partially observed or completely invisible from certain viewpoints. As a result, these Gaussians receive limited supervision and are more difficult to learn. While this example is shown from a single view direction, in practice, occlusion relationships vary with the viewing angle, and multi-layer occlusion structures must be learned across different viewpoints.

Our proposed STO-GS adopts a two-stage training strategy consisting of a coarse stage and a fine stage. The coarse stage uses only modal input images to train the initial Gaussian attributes, focusing on unoccluded regions to establish the basic 3D structure. The fine stage refines the Gaussians learned during the coarse stage using both the original modal images (I_0) and a set of amodal views

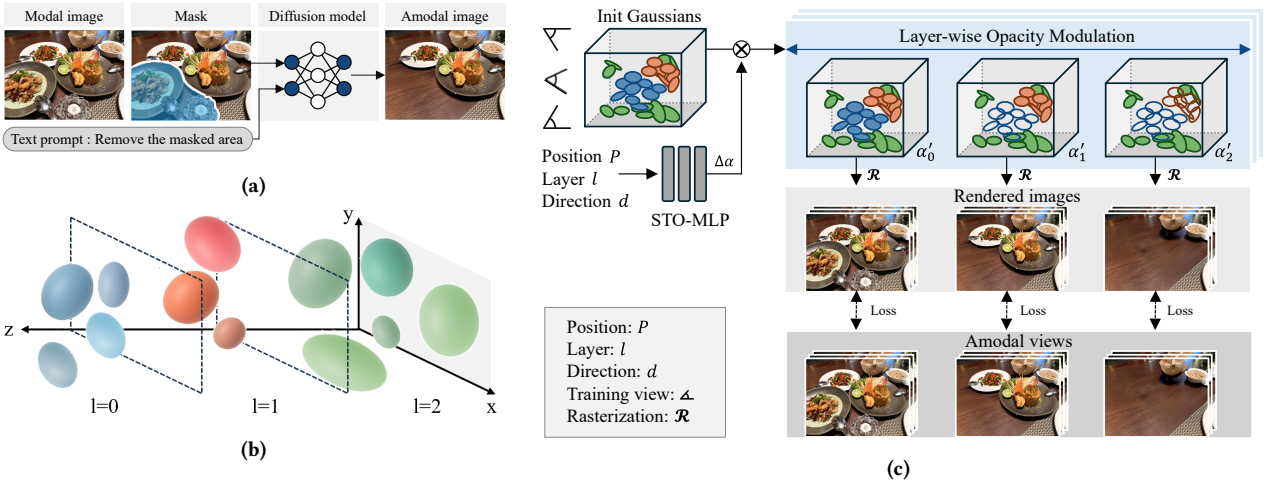

Figure 2: Overview of the proposed STO-GS framework. (a) We generate amodal views by applying segmentation-based masks and diffusion-based inpainting to the modal input image. (b) The scene is decomposed into occlusion layers ($l = 0, 1, 2$), with blue, orange, and green Gaussians representing different depth layers. (c) An STO-MLP modulates the opacity of each Gaussian based on its position P, viewing direction d, and occlusion layer index l, enabling the model to progressively reveal hidden content. Layer-specific rendered images are supervised using the corresponding amodal views, allowing the model to learn occlusion-aware scene structures across depth.

$(I_1, I_2, \ldots)$ generated via amodal completion. Fig. 2c illustrates this process, where the opacity α of each Gaussian is modulated depending on its assigned occlusion layer l. A lightweight network, STO-MLP, takes as input the Gaussian's position P, layer index l, and viewing direction d, and predicts an opacity adjustment factor $\Delta\alpha$, which is multiplied with the original opacity to yield the modulated value α'_l:

$$\alpha'_l = \text{STO-MLP}(P, l, d) \cdot \alpha \tag{1}$$

This layer-specific opacity modulation allows the model to reveal occluded regions by progressively reducing the opacity of foreground Gaussians. Using the modulated opacities, rasterization is performed to render amodal views corresponding to each occlusion layer. The final pixel color in each rendered image is computed as:

$$C_l = \sum_{i=1}^{n} c_i \, \alpha'_{i,l} \prod_{j=1}^{i-1} \left(1 - \alpha'_{j,l}\right) \tag{2}$$

Here, n denotes the number of Gaussians contributing to the pixel, c_i is the color computed using the Spherical Harmonics (SH) coefficients of the i-th Gaussian, and $\alpha'_{i,l}$ is the modulated opacity for layer l. The rendered image for each occlusion layer is supervised by its corresponding amodal image through photometric loss.

During training, both modal and amodal views are randomly sampled from a unified image pool. For modal images ($l = 0$), opacity modulation is skipped, and the Gaussian attributes are updated using the original opacities. In contrast, for amodal views ($l > 0$), STO-MLP is used to modulate opacities per layer, allowing direct supervision of Gaussians that are occluded in the original inputs. During inference, opacity modulation is not applied, as occluded Gaussians have already been sufficiently optimized during training. All Gaussians are rendered as if they belong to layer $l = 0$, without invoking STO-MLP, enabling fast rendering while maintaining high-quality occlusion-aware scene reconstruction.

3.3 Occlusion-Layer Guided Extension of Training Pipelines

To support the STO learning framework described in Section 3.2, we adopt a two-stage training pipeline that consists of a coarse initialization stage using only modal images and a fine refinement stage incorporating amodal views. Building upon this structure, we further extend the training process through an occlusion-layer guided strategy, which introduces co-regularization between dual Gaussian sets and a layer-aware extension of pseudo view sampling.

Inspired by CoR-GS [41], we construct two independent Gaussian primitive sets, θ^1 and θ^2, both initialized from the same input images. Each set is trained independently, and for each occlusion layer l, we render the corresponding images I'^1_l and I'^2_l from the two sets. We then compute a photometric loss between these paired images to enforce consistency in color and structure across sets. This dual-set training approach mitigates ambiguity and local minima that may arise when training a single set and, when combined with the STO learning strategy, significantly improves the stability and accuracy of occlusion prediction. In this way, the two Gaussian sets serve as regularizers for each other within the multi-layer supervision framework, leading to more expressive and generalizable scene representations. In addition, we extend the pseudo view sampling strategy introduced in prior works [41, 43] to operate at the occlusion-layer level. As defined in (3), camera parameters for a pseudo view are sampled as:

$$\mathcal{P}'_l = (t + \epsilon, ; q, ; l), \quad l \sim \mathcal{U}(0, 1, \ldots, L) \tag{3}$$

where t is the camera position of a training image, ϵ is random noise sampled from a normal distribution, and q denotes the average rotation between two cameras in quaternion form. By introducing the occlusion layer index l into the sampling process, each pseudo view

is associated with a specific occlusion layer, allowing the generation of corresponding amodal views. This formulation extends the multi-scene learning strategy of CoR-GS [41] into a layer-aware occlusion modeling pipeline, enabling STO-GS to supervise occluded content more explicitly and effectively.

3.4 Loss Function

Adaptive Amodal Loss Weight. To reflect the different importance of modal and amodal views during training, we apply a layer-specific weight λ_l to each rendered view. For the original modal image (I_0), the weight is fixed to $\lambda_l = 1$. In contrast, amodal views (I_l, $l > 0$) start with an initial weight of 0.5, which gradually decreases as training progresses. The rate of this decay is controlled by a hyperparameter d, which is set to 3 in our implementation. Let $total\_iters$ denote the total number of training iterations, and $iter$ the current iteration. The value of λ_l is defined as follows:

$$\lambda_l = \begin{cases} 1, & \text{if } l = 0, \\ \max\left(0.01,\ 0.5 \cdot \left(1 - \dfrac{\text{iter}}{\text{total\_iters}}\right)^d\right). & \text{if } l > 0. \end{cases} \tag{4}$$

Here, λ_l is lower-bounded by 0.01 to prevent vanishing contributions from amodal views during late training.

Photometric loss. To compute the difference between the rendered image I'_l and the GT image I_l, we employ a photometric loss that combines L1 loss $\mathcal{L}1$ and structural similarity index (SSIM)-based loss $\mathcal{L}_{D-SSIM}$. A weighting factor $\lambda_{lp} = 0.2$ is used to balance the two terms. The overall photometric loss is defined as (5).

$$L^l_{photo} = (1 - \lambda_{lp})\mathcal{L}_1(I'_l, I_l) + \lambda_{lp}\mathcal{L}_{D-SSIM}(I'_l, I_l) \tag{5}$$

Color Co-Regularization. Using layered sampling pseudo views, we construct two Gaussian primitive sets, θ^1 and θ^2, and train the model to maintain color consistency between the rendered images I'^1_l and I'^2_l at the same occlusion layer l. The corresponding photometric consistency loss is defined as in (6). Here, I'^1_l and I'^2_l are the images rendered from θ^1 and θ^2, respectively, at layer l. A weighting factor $\lambda_{rp} = 0.2$ is used to balance the two terms.

$$\mathcal{R}^l_{photo} = (1 - \lambda_{rp})\mathcal{L}_1(I'^1_l, I'^2_l) + \lambda_{rp}\mathcal{L}_{D-SSIM}(I'^1_l, I'^2_l) \tag{6}$$

Total Loss. The Total loss function is defined as the sum of the two loss terms described above, weighted by the adaptive factor λ_l for each layer l.

$$\mathcal{L}_{total} = \lambda_l\left(\mathcal{L}^l_{photo} + \mathcal{R}^l_{photo}\right) \tag{7}$$

4 Experiments

4.1 Experimental Settings

We compare our STO-GS with prior few-shot Gaussian Splatting methods—FSGS [43], CoR-GS [41], and CoMapGS [10]—under the same experimental settings as reported in their original publications. The performance is evaluated both quantitatively, using peak signal-to-noise ratio (PSNR), SSIM [31], and learned perceptual image patch similarity (LPIPS) [42], and qualitatively through visual comparison. We conduct experiments on three publicly available datasets: LLFF [21], Shiny [32], and Mip-NeRF360 [3], each with distinct viewpoint configurations and occlusion characteristics. Mip-NeRF360 is a 360-degree panoramic dataset that allows scenes to be observed from all directions. However, when the number of training views is limited, structures or regions outside the visible range remain unobserved due to occlusion, making reconstruction particularly difficult in sparse-view settings. To evaluate this, we conducted experiments using 12 and 24 training views at both 1/4 and 1/8 resolution levels. In contrast, LLFF and Shiny follow a forward-facing configuration, where the field of view is restricted and observations are centered around the main object. This setup results in less overlap between views and presents challenges in reconstructing background regions occluded by foreground objects. For both datasets, we use only 3 sparse training views and conduct experiments at 1/4 and 1/8 resolutions, consistent with the Mip-NeRF360 setting.

STO-GS is implemented using the PyTorch framework and trained and evaluated on an NVIDIA RTX A6000 GPU. We use the same initial point cloud as in previous works, and the training process is divided into two stages: coarse and fine. The coarse stage is conducted for 3,000 iterations equally across all datasets. The fine stage uses different numbers of iterations depending on the dataset: 10,000 iterations for LLFF and Shiny, and 30,000 iterations for Mip-NeRF360.

4.2 Quantitative Results

Mip-NeRF360. Table 1 presents the quantitative results on the Mip-NeRF360 dataset. Compared to CoR-GS [41], the proposed STO-GS achieves improvements of approximately 0.48 dB and 0.51 dB in PSNR for the 12-view setting at 1/4 and 1/8 resolutions, respectively. For the 24-view setting, STO-GS yields gains of 0.21 dB and 0.31 dB at the corresponding resolutions. In comparison to CoMapGS [10] under the 12-view, 1/4 resolution setting, STO-GS shows an improvement of approximately 0.03 dB in PSNR and 0.02 in SSIM. Across all experimental settings, STO-GS consistently achieves the best performance in both PSNR and SSIM. Also, STO-GS achieves the lowest LPIPS scores across all settings, with the exception of the 12-view, 1/4 resolution case, where CoMapGS [10] performs slightly better. Mip-NeRF360, being a 360-degree panoramic dataset, is particularly susceptible to occlusion-induced information loss in sparse-view settings, which significantly affects reconstruction quality. STO-GS addresses this challenge by generating amodal views and applying opacity modulation, allowing the model to infer and learn from occluded scene content. As a result, it substantially enhances the reconstruction quality, especially for occluded or non-central regions.

LLFF. Table 2 presents the evaluation on the LLFF dataset. Although LLFF consists of forward-facing scenes that are relatively less affected by occlusion, its sparse view configuration frequently causes unobserved regions—particularly behind foreground objects—due to limited camera coverage. In these scenarios, STO-GS improves reconstruction by generating amodal views to reveal occluded structures and applying opacity modulation for layered learning. While CoMapGS [10] achieves higher accuracy under certain LLFF settings through covisibility map-based adaptive supervision and point cloud initialization, it requires complex preprocessing and supervision. In contrast, STO-GS attains competitive performance using a simpler and more intuitive strategy based on amodal layering and opacity modulation.

Table 1: Quantitative results on the Mip-NeRF360 dataset under 12 and 24 training views at 1/4 and 1/8 resolutions. STO-GS achieves the best performance across most metrics, outperforming previous methods in both PSNR and SSIM. LPIPS scores also demonstrate improved perceptual quality, particularly in sparse-view settings. The best results are highlighted in bold, and second-best results are underlined.

Methods	1/4 Resolution						1/8 Resolution					
	12 views			24 views			12 views			24 views		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS[14]	18.52	0.523	0.415	22.80	0.708	0.276	18.77	0.531	0.408	19.93	0.588	0.401
FSGS[43]	18.83	0.559	0.418	23.37	0.734	0.266	19.03	0.554	0.372	23.85	0.762	0.216
CoR-GS[41]	19.23	0.577	0.414	23.29	0.731	0.263	19.28	0.571	0.382	23.77	0.756	0.215
CoMapGS[10]	19.68	0.591	**0.394**	23.46	0.734	0.264	-	-	-	-	-	-
Ours	**19.71**	**0.593**	0.411	**23.50**	**0.766**	**0.262**	**19.79**	**0.581**	**0.371**	**24.08**	**0.763**	**0.213**

Table 2: Quantitative results on the LLFF dataset with 3 training views at 1/4 and 1/8 resolutions. STO-GS shows strong overall performance across all metrics, while CoMapGS slightly outperforms in some 1/8 cases.

Methods	1/4 Resolution			1/8 Resolution		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS[14]	16.94	0.488	0.402	19.22	0.649	0.229
FSGS[43]	19.88	0.612	0.340	20.43	0.682	0.248
CoR-GS[41]	19.74	0.676	0.268	20.45	0.712	0.196
CoMapGS[10]	-	-	-	**21.10**	**0.747**	**0.182**
Ours	**20.02**	**0.680**	**0.264**	20.78	0.741	0.190

Table 3: Quantitative results on the Shiny dataset with 3 training views at 1/4 and 1/8 resolutions. STO-GS achieves the best performance across all metrics, showing robust reconstruction even in scenes with reflective surfaces.

Methods	1/4 Resolution			1/8 Resolution		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS[14]	15.73	0.497	0.328	17.83	0.547	0.385
FSGS[43]	18.40	0.551	0.317	19.85	0.653	0.217
CoR-GS[41]	18.80	0.569	0.311	20.20	0.662	0.223
Ours	**19.23**	**0.576**	**0.309**	**20.51**	**0.683**	**0.210**

Shiny. Table 3 reports results on the Shiny dataset, which, like LLFF, follows a forward-facing configuration and contains relatively fewer occlusions across the entire scene. However, in sparse-view settings, parts of the scene remain outside the field of view, which still negatively impacts reconstruction performance. The proposed STO-GS achieves the best performance across all metrics—PSNR, SSIM, and LPIPS—surpassing existing methods. In particular, at 1/4 resolution, STO-GS achieves 19.23 dB in PSNR, 0.576 in SSIM, and 0.311 in LPIPS, demonstrating stable and consistent reconstruction even in low-occlusion scenarios. At 1/8 resolution, STO-GS achieves 20.51 dB in PSNR, 0.683 in SSIM, and 0.210 in LPIPS, indicating not only accurate structural recovery but also superior perceptual quality. These results quantitatively demonstrate that STO-GS is not limited to occlusion-centric improvements, but also delivers consistent performance across various view configurations and resolutions, showing strong generalization in forward-facing sparse-view environments.

4.3 Qualitative Results

Mip-NeRF360. Fig. 3 presents qualitative comparisons of different methods on the Mip-NeRF360 dataset. As shown in the first row, FSGS and CoR-GS produce stable reconstructions in central regions that are well-covered by training views. However, due to the 360-degree panoramic nature of the dataset and the sparse-view setting, many peripheral regions remain unobserved and are poorly reconstructed, especially under occlusion. In contrast, the proposed STO-GS demonstrates better detail and consistency in peripheral areas that are often missed by previous methods. In the third row (kitchen scene), FSGS [43] reconstructs the central region reasonably well but fails to capture the texture and structure of background elements. CoR-GS [41] improves color consistency through pseudo view-based co-regularization but still suffers from distortion and incomplete geometry. Our STO-GS, by leveraging amodal view augmentation to supervise occluded regions, successfully recovers structural information behind foreground objects. As a result, details such as wall textures and object boundaries in the background are more accurately and consistently reconstructed. These results visually demonstrate that STO-GS not only improves consistency but also enables the model to infer and recover structures in occluded regions more effectively than prior approaches.

LLFF. The LLFF dataset, composed of forward-facing scenes, contains relatively fewer occluded regions compared to Mip-NeRF360. However, in sparse-view settings with limited camera viewpoints, unobserved areas frequently occur, making occlusion recovery still a challenging problem. As shown in Fig. 4, the proposed STO-GS achieves higher structural accuracy and visual consistency even in regions that previous methods failed to reconstruct.

Shiny. The Shiny dataset, like LLFF, consists of forward-facing scenes but presents additional challenges due to the presence of reflective materials such as glass and metal. In particular, the Seasoning scene in Fig. 5 contains a combination of specular surfaces and transparent glass structures, making it prone to distortion not only in occluded regions but also in surrounding details. STO-GS addresses these challenges by using amodal views and opacity modulation to recover hidden structures. As a result, STO-GS improves fine-grained reconstruction around objects and enhances the recovery of background details behind transparent surfaces, leading to a noticeable improvement in overall visual quality compared to previous methods.

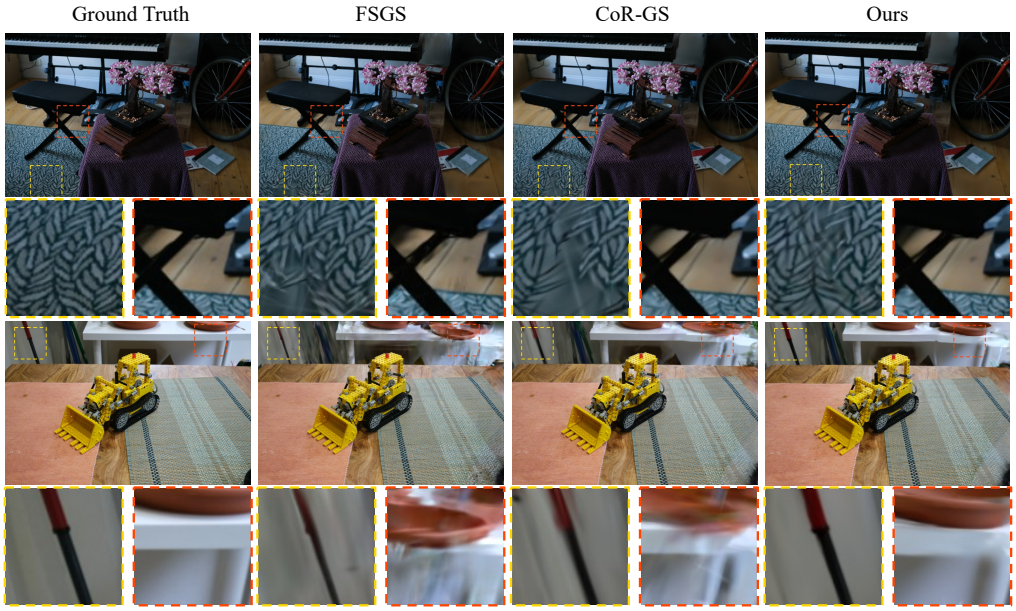

Figure 3: Qualitative results on the Mip-NeRF360 Bonsai and Kitchen scenes with 24 training views and a downscale factor of 4. STO-GS shows more accurate reconstruction, especially in occluded non-central regions, with improved color, texture, and structural consistency.

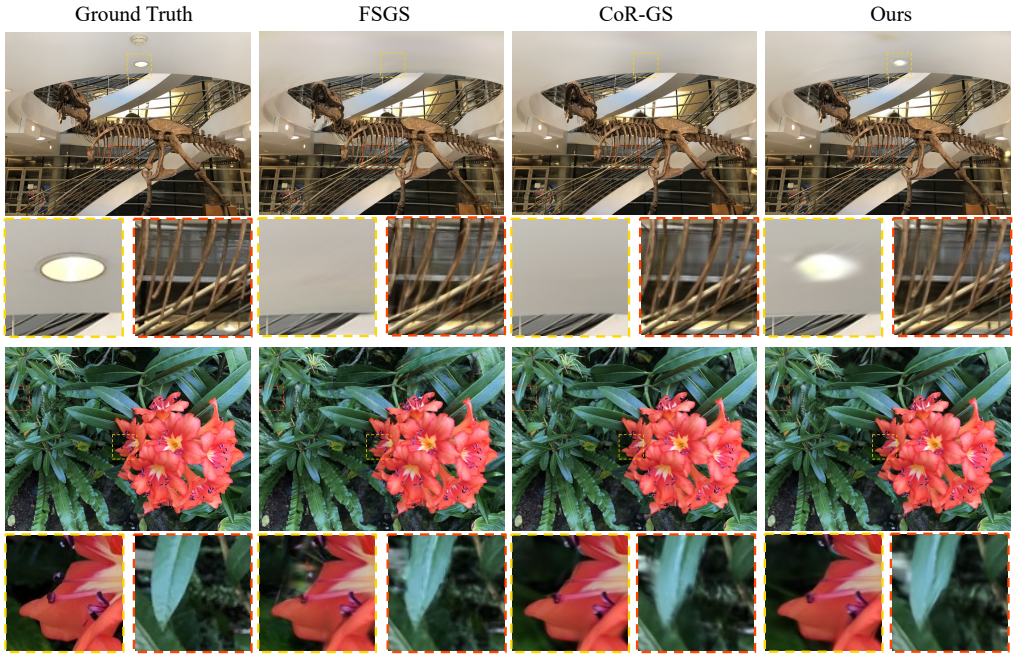

Figure 4: Qualitative results on the Trex and Flower scenes from the LLFF dataset with 3 training views at $1/4$ resolution. STO-GS shows improved color restoration, structural consistency, and detail around occluded regions compared to prior methods.

Layer-Wise Effect of Opacity Modulation. To analyze the effect of our proposed method, which modulates Gaussian opacity across occlusion layers, we visualize inference results on the Crest scene from the Shiny dataset (Fig. 6). Each row corresponds to a different amodal layer ($l = 0, 1, 2$). Fig. 6 shows the inpainted amodal views, while Fig. 6 presents the inference results using opacity modulation to control occlusion-causing Gaussians in each layer. As the layer index increases, the opacity of the corresponding Gaussians decreases, revealing previously hidden structures and background regions. This demonstrates that our opacity modulation strategy

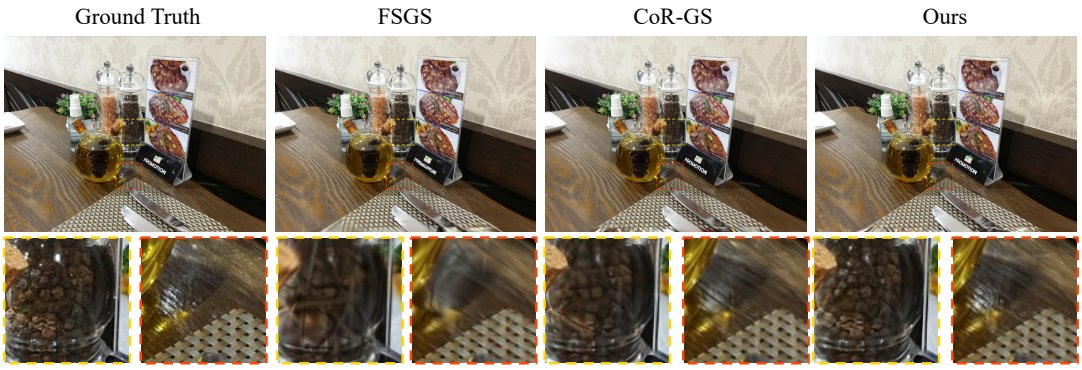

Figure 5: Qualitative results on the Seasoning scene from the Shiny dataset with 3 training views at 1/4 resolution. STO-GS outperforms prior methods in color restoration, structural consistency, and detail around occluded regions.

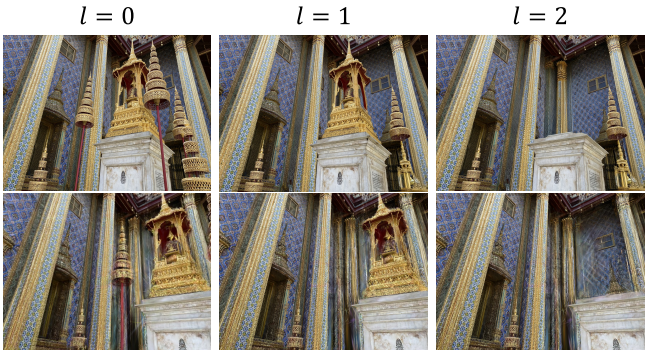

Figure 6: Layer-wise results on the Crest scene from the Shiny dataset. Each column shows an amodal layer; the first row displays generated amodal views, and the second shows inference results on a validation view.

Two-stage Training	Layer-aware Sampling	PSNR ↑	SSIM ↑	LPIPS ↓
–	–	19.61	0.569	0.376
✓	–	19.71	0.578	0.373
–	✓	19.62	0.569	0.374
✓	✓	**19.75**	**0.579**	**0.372**

Table 4: The effect of two-stage training and layer-aware pseudo view sampling.

Amodal Loss Weight λ_l	PSNR ↑	SSIM ↑	LPIPS ↓
0.01	19.75	0.578	0.372
0.50	19.77	0.579	0.373
adaptive	**19.79**	**0.581**	**0.371**

Table 5: The effect of adaptive amodal loss weight λ_l.

effectively disentangles occluded Gaussians and enables learning of the hidden space. The differences across layers are also reflected in the validation view, indicating successful generalization of amodal space across varying occlusion levels. Overall, this experiment confirms that STO-GS can effectively recover occluded regions through layer-aware structural learning.

4.4 Ablation Study

To evaluate the effectiveness of the core components in the proposed STO-GS framework, we conduct an ablation study using the Mip-NeRF360 dataset at 1/8 resolution. The opacity modulation introduced in Section 3.2 is used as a default component across all experiments. Table 4 compares the impact of two-stage training and layer-aware sampling. The baseline model, which removes both components, shows the lowest performance across all metrics. Incorporating two-stage training improves convergence and performance by enabling more stable initialization of Gaussians. Similarly, applying layer-aware sampling alone facilitates occlusion-aware learning and yields performance gains. The best results are achieved when both components are combined, demonstrating their complementary effects. Table 5 presents the results of varying the weighting strategy λ_l used for amodal loss. When using fixed weights, the supervision on occluded regions can become either too weak or too strong, leading to unstable training. In contrast, the adaptive weighting strategy gradually reduces the contribution of higher-layer losses as training progresses, which improves generalization to occluded areas. As a result, the adaptive setting achieves the most balanced performance across all metrics.

5 Conclusion

We introduced STO-GS, a novel framework for few-shot 3D Gaussian Splatting that enables occlusion-aware reconstruction through layered amodal supervision. By leveraging amodal view generation and opacity modulation across occlusion layers, our method effectively learns hidden structures behind foreground objects. Extensive experiments demonstrate that STO-GS consistently outperforms existing methods in both visual quality and structural completeness, especially under sparse-view and occluded scenarios. However, STO-GS relies on segmentation-based diffusion inpainting to generate amodal views, which can struggle when foreground objects are extremely large. In such cases, background regions may not be plausibly completed, limiting the effectiveness of occlusion recovery. Future work may explore correction mechanisms or alternative inpainting strategies that are more robust to object size and scene complexity.

Acknowledgments

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-ITRC (Information Technology Research Center) grant (RS-2021-II21052) and the Artificial Intelligence Semiconductor Support Program (IITP-2025-RS-2023-00253914) and also the National Research Foundation of Korea (NRF) grant (RS-2024-00414230) funded by the Korea government (MSIT).

References

[1] Adobe Inc. 2023. Adobe Firefly: Generative AI by Adobe. https://firefly.adobe.com. Generative AI tools for images, text effects, and design. Accessed: 2025-04-11.

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

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

[4] Chenjie Cao, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. 2024. Mvinpainter: Learning multi-view consistent inpainting to bridge 2d and 3d editing. *arXiv preprint arXiv:2408.08000* (2024).

[5] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. 2021. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14124–14133.

[6] Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 303–312.

[7] Guangcong, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. 2023. SparseNeRF: Distilling Depth Ranking for Few-shot Novel View Synthesis. *IEEE/CVF International Conference on Computer Vision (ICCV)* (2023).

[8] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. 2024. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4220–4230.

[9] Ajay Jain, Matthew Tancik, and Pieter Abbeel. 2021. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 5885–5894.

[10] Youngkyoon Jang and Eduardo Pérez-Pellitero. 2025. CoMapGS: Covisibility Map-based Gaussian Splatting for Sparse Novel View Synthesis. *arXiv preprint arXiv:2503.20998* (2025).

[11] Seong-Uk Jo, Du Yeol Lee, and Chae Eun Rhee. 2024. Occlusion-aware Amodal Depth Estimation for Enhancing 3D Reconstruction from a Single Image. *IEEE Access* (2024).

[12] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, Vol. 7.

[13] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. 2021. Deep occlusion-aware instance segmentation with overlapping bilayers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4019–4028.

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

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

[16] Junoh Lee, ChangYeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. 2024. Fully explicit dynamic gaussian splatting. *Advances in Neural Information Processing Systems* 37 (2024), 5384–5409.

[17] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. 2024. DNGaussian: Optimizing Sparse-View 3D Gaussian Radiance Fields with Global-Local Depth Normalization. *arXiv preprint arXiv:2403.06912* (2024).

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

[19] Zhenyu Li, Mykola Lavreniuk, Jian Shi, Shariq Farooq Bhat, and Peter Wonka. 2024. Amodal Depth Anything: Amodal Depth Estimation in the Wild. *arXiv preprint arXiv:2412.02336* (2024).

[20] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20654–20664.

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

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[23] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. doi:10.1145/3528223.3530127

[24] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. 2022. RegNeRF: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[25] Radu Bogdan Rusu and Steve Cousins. 2011. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*. IEEE, 1–4.

[26] Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4104–4113.

[27] Thomas W Sederberg, Jianmin Zheng, Almaz Bakenov, and Ahmad Nasri. 2003. T-splines and T-NURCCs. *ACM transactions on graphics (TOG)* 22, 3 (2003), 477–484.

[28] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, Vol. 1. IEEE, 519–528.

[29] Xiangyu Sun, Joo Chan Lee, Daniel Rho, Jong Hwan Ko, Usman Ali, and Eunbyung Park. 2024. F-3dgs: Factorized coordinates and representations for 3d gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 7957–7965.

[30] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. 2022. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2149–2159.

[31] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[32] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. 2021. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8534–8543.

[33] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 2024. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 20310–20320.

[34] Jamie Wynn and Daniyar Turmukhambetov. 2023. DiffusioNeRF: Regularizing Neural Radiance Fields with Denoising Diffusion Models. In *CVPR*.

[35] Hantong Xu, Jiamin Xu, and Weiwei Xu. 2019. Survey of 3D modeling using depth cameras. *Virtual Reality & Intelligent Hardware* 1, 5 (2019), 483–499.

[36] Jinbo Yan, Rui Peng, Luyang Tang, and Ronggang Wang. 2024. 4D Gaussian Splatting with Scale-aware Residual Field and Adaptive Optimization for Real-time rendering of temporally complex dynamic scenes. In *ACM Multimedia 2024*.

[37] Jiawei Yang, Marco Pavone, and Yue Wang. 2023. FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[38] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. 2024. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting. In *International Conference on Learning Representations (ICLR)*.

[39] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. 2023. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790* (2023).

[40] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. 2024. Mip-Splatting: Alias-free 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 19447–19456.

[41] Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. 2024. CoR-GS: sparse-view 3D Gaussian splatting via co-regularization. In *European Conference on Computer Vision*. Springer, 335–352.

[42] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

[43] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. 2024. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European conference on computer vision*. Springer, 145–163.