Splat-based Gradient-domain Fusion for Seamless View Transition

Dongyoung Choi Jaemin Cho Woohyun Kang Hyunho Ha James Tompkin‡ Min H. Kim
KAIST ‡Brown University

Abstract

In sparse novel view synthesis with few input views and wide baselines, existing methods often fail due to weak geometric correspondences and view-dependent color inconsistencies. Splatting-based approaches can produce plausible results near training views, but they frequently overfit and struggle to maintain smooth, realistic appearance transitions in novel viewpoints. We introduce a splat-based gradient-domain fusion method that addresses these limitations. Our approach first establishes reliable dense geometry via two-view stereo for stable initialization. We then generate intermediate virtual views by reprojecting input images, which provide reference gradient fields for gradient-domain fusion. By blending these gradients, our method transfers low-frequency, view-dependent colors to the rendered Gaussians, producing seamless appearance transitions across views. Extensive experiments show that our approach consistently outperforms state-of-the-art sparse Gaussian splatting methods, delivering robust and perceptually plausible view synthesis. A comprehensive user study further confirms that our results are perceptually preferred, with significantly smoother and more realistic color transitions than existing methods.

1. Introduction

Given a set of multi-view images with their camera poses, novel view synthesis creates images of the scene from new viewpoints. To achieve high-quality results, we require many input images to produce a natural view-dependent appearance [15, 23]. In sparse settings with few views, many traditional approaches fail due to overfitting the color and geometry to the few training images.

One key problem is handling changes in appearance. Beyond view-dependent color changes, such as specular reflections, factors like lens shading, exposure differences, or minor illumination variations in outdoor scenes cause additional *frame-dependent* color variations across training images. These variations have a greater impact on reconstruction in scenarios with sparse input views, leading to overfitting artifacts that make achieving novel view synthesis with smooth and consistent transitions more challenging.

In sparse settings with limited input information, one natural choice is to blend adjacent input images, reprojected into the novel view, so that the reconstructed frame-dependent appearance is smooth and less perceptually objectionable [6, 19, 29]. Some image-based rendering methods [1, 17] achieve this by blending target views onto known proxy geometry using Poisson blending [27], a *gradient-domain fusion* (GF) technique that smoothly blends source and target regions by transferring its gradients.

Rather than use Poisson blending to produce the final image, we present an approach to use gradient-domain fusion within a Gaussian Splatting (GS) optimization as a regularizer to produce better appearance. This works by reprojecting and seamlessly fusing the gradients of input images at intermediate-generated virtual novel views to create reference gradient fields (Figure 1 middle). By comparing the image gradients of the rendered image with those of the reference gradient fields, we can induce smooth frame-

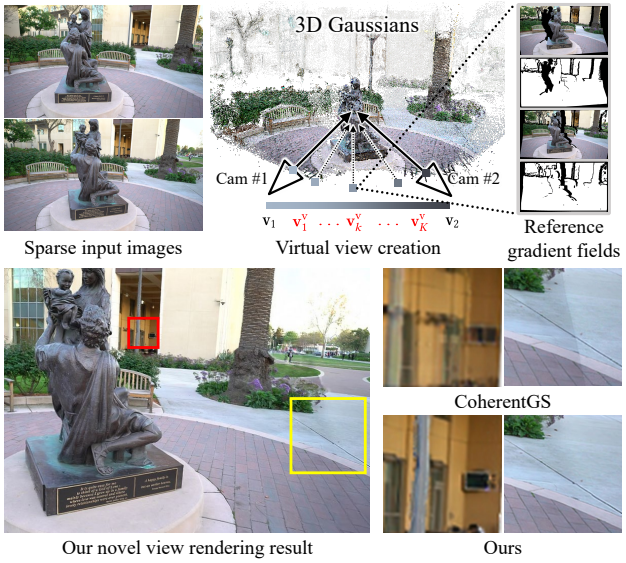


Figure 1. Alleviating overfitting in sparse settings for 3D Gaussian splatting. Given sparse input, Gaussians are adjusted to match specific input training views, resulting in unreliable geometry and inconsistent color appearance through novel view transitions. We integrates *gradient-domain fusion* (GF) of reprojected input views at intermediate virtual poses into the GS rendering framework. This creates reference gradient fields to regularize the rendered result, producing a smoother appearance variation during view transitions. Our method achieves higher-fidelity and perceptually pleasing view synthesis in sparse scenarios than existing work.

dependent color for the Gaussians and avoid artifacts from overfitting color in novel view transitions. However, producing accurate reference gradient fields requires reprojecting input images to virtual novel views, which in turn demands precise scene geometry.

Recent GS-based methods have shown that obtaining good geometry under sparse input conditions necessitates a dense and well-initialized point cloud. To this end, they have used monocular depth priors [25] or multi-view depth priors [10, 26, 35]. However, monocular priors often suffer from poor multi-view consistency, while multi-view priors can fail to reconstruct regions that are visible in only a single training view. Instead, we additionally use dense geometric correspondence to initialize Gaussians via a hierarchical point cloud reconstruction method based on two-view stereo [34]. This initialization yields more accurate geometry, enabling precise depth reprojection and the generation of high-quality intermediate reference views for GF.

Our method demonstrates more robust and higher-fidelity view synthesis than current state-of-the-art sparse GS approaches [20, 25, 26, 35], achieving smooth color transitions with robust reconstruction. We further evaluate it against alternative strategies replacing the GF loss, consistently showing superior performance, particularly in perceptual metrics like LPIPS [40] and DISTS [9]. While these metrics are strong indicators, they do not fully capture human preference. To address this, we conduct a user study on natural color transitions and perceptual quality evaluation, where our method is consistently preferred over existing sparse view synthesis methods and other alternatives.

2. Related Work

Gaussian Splatting from sparse images. Existing Gaussian Splatting for sparse view synthesis mainly focuses on resolving geometry instability caused by limited input views. DNGaussian [20] uses random points with monocular depth priors, while CoR-GS [39] and FSGS [42] employ patch-match MVS [28] and appearance regularization, but often produce overly smooth geometry and floating artifacts due to sparse or inaccurate initialization. CoherentGS [25] improves initialization using monocular depth [37] and optical flow refinement [31], yet still suffers from multi-view inconsistency and color overfitting. Neural MVS models like MVPGS [35] and InstantSplat [10], and feature matching methods like SCGaussian [26], enhance multi-view consistency but struggle in singly observed regions.

Although these geometry-focused methods improve reconstruction quality, they largely overlook the problem of color overfitting and frame-dependent appearance variations, thereby degrading perceptual quality. In contrast, our work explicitly addresses this under-explored issue by introducing a gradient-domain fusion regularizer that encourages smooth and consistent frame-dependent colors without compromising fine details. Our method combines this regularizer with dense geometry initialization from two-view stereo [34], enabling more robust and smooth novel views.

Virtual view regularization. Recent view synthesis methods for sparse inputs have proposed various strategies to regularize unobserved views. FSGS [42] enhances geometric accuracy by regularizing pseudo views with mono-depth constraints, while CoR-GS [39] suppresses point and rendering disagreements across two different Gaussian radiance fields. These approaches regularize the geometry of virtual views rather than the color. Generative models have also been used. RegNeRF [24] uses a normalizing flow model to maximize the likelihood of virtual view colors, achieving smooth geometry and high-likelihood color generation. In contrast, we generate virtual views by rasterizing two paired meshes to obtain reference gradient fields, effectively transferring low-frequency, view-dependent color gradients to 3D Gaussians.

Gradient-domain fusion. Gradient-domain fusion has been widely adopted to manipulate image gradients to blend images, preserving important edge and texture details seamlessly in computational photography. Beyond seam removal, this enables the smooth integration of overlapping regions, and thus applies to image stitching [18], high dynamic range imaging [12], and image cloning [27]. Poisson blending provides a foundation for gradient-domain fusion by solving the Poisson equation, which preserves source image gradients while adhering to the target image's boundary conditions. Gradient shop [3] and image melding [8] use the screened Poisson equation to enhance data fidelity. This can be accelerated using Fourier analysis [2], and convolution pyramids [11] incorporate multi-grid frameworks to allow high-resolution image composition. However, these methods require solving sparse linear systems, making them slow and challenging to handle complex boundaries, thus difficult to integrate into Gaussian Splatting. We instead adopt an efficient approach that directly blends the gradients of reprojected training views with the rendered images for seamless fusion under complex boundary conditions.

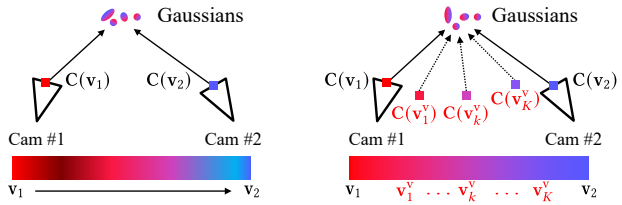
3. Background

3.1. 3D Gaussian Splatting

Gaussian Splatting optimizes explicit 3D Gaussian primitives for rapid view synthesis by minimizing a photometric loss between input views and their corresponding rendered images. Each Gaussian is parameterized by a scaling matrix $\mathbf{S}$, a rotation matrix $\mathbf{R}$, and a position $\mathbf{p}_k$:

$$G(\mathbf{p}) = \exp\left(-\tfrac{1}{2}(\mathbf{p}-\mathbf{p}_k)^\top \Sigma^{-1}(\mathbf{p}-\mathbf{p}_k)\right), \quad (1)$$

where the covariance matrix is $\Sigma = \mathbf{RSS}^\top\mathbf{R}^\top$. Each Gaussian also has opacity α and SH color components. Gaus-



(a) w/o virtual view regularization (b) w/ virtual view regularization

Figure 2. Gaussians' view-dependent color optimization from sparse views. $\mathbf{C}(\mathbf{v})$ denotes the observed color per view, with the color bar indicating transitions along interpolated view vectors. (a) Results overfitted to sparse inputs can produce abrupt color changes in intermediate views. (b) We address this by applying gradient-domain fusion at virtual views $\mathbf{C}(\mathbf{v}^{\mathrm{v}})$, which regularizes the view-dependent color functions.

sians are projected onto the image plane using EWA [43], and color is accumulated by alpha blending.

In real-world datasets, object colors vary across frames due to illumination, exposure, and view-dependent effects. With sparse and inconsistent inputs, we expect the colors in training images to project well onto the Gaussians' spherical harmonics, resulting in angularly smooth, view-dependent functions for seamless transitions. However, the rendered color in GS results from a complex blend of many Gaussians' view-dependent functions rather than a single one. Thus, when optimized only on a few training views, these functions often overfit to the observed directions, causing seams, floaters, and unnatural transitions in unseen views. To mitigate this, we regularize Gaussians at unobserved views by encouraging the rendered colors to reflect a smooth blending of adjacent input images. This results in spatially and perceptually coherent colors, improving transition quality across novel viewpoints (Figure 2).

3.2. Poisson Blending

We aim to seamlessly blend adjacent input images into novel views without visible boundaries and seams. To this end, we adopt the core principle of Poisson blending, which formulates a minimization problem enforcing the gradients $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ of the result image f within the overlap region Ω to match a reference gradient field $\mathbf{w}$, blended from the source and target images f^*. At the stitching boundary $\partial\Omega$, pixel values from the target image are fixed as a Dirichlet boundary condition. This can be written as:

$$\min_f \iint_\Omega \|\nabla f - \mathbf{w}\|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}. \tag{2}$$

Taking the first variation of Energy function (2) and applying the divergence theorem yields the Euler–Lagrange equation that reduces to the Poisson equation $\Delta f = \nabla \cdot \mathbf{w}$ over Ω, with $f|_{\partial\Omega} = f^*|_{\partial\Omega}$, where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator. Solving this Poisson equation produces a blended image that matches the reference gradient field within the overlap region, resulting in a large, sparse linear system that can be solved using direct or iterative methods.

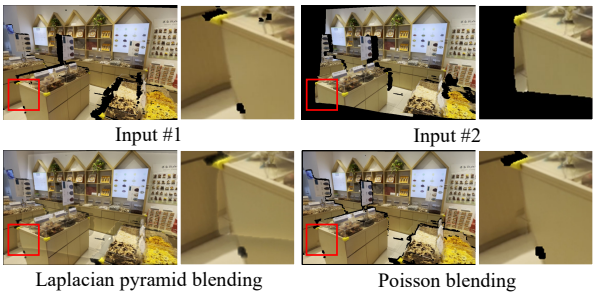


Figure 3. Image-domain fusion vs. gradient-domain fusion. Gradient-domain fusion, such as Poisson blending, yields more seamless and natural blending results compared to Laplacian pyramid blending, a representative image-domain fusion method.

However, solving large linear systems is computationally expensive, and Poisson blending–being purely 2D–ignores 3D consistency and tends to suppress high-frequency details. Thus, using Poisson blending directly within the GS framework leads to splotchy and blurry results, as shown in Figure 10 (e). Instead of relying on Poisson blending, we draw inspiration from the ability of gradient-domain fusion (GF) to blend gradients and incorporate it into the GS framework. This achieves seamless novel view synthesis with smoother transitions across various viewpoints.

4. Method

4.1. Initialization of Gaussians

To achieve effective blending, precise depth reprojection is essential; thus, we begin by improving geometry initialization. Following prior multiview methods [5, 35], we use a point cloud from MVSFormer [5], but it is incomplete in single-observed regions and insufficient for stable initialization. To improve coverage and ensure accurate depth reprojection, we augment it with a dense two-view stereo point cloud from GMStereo [34] and combine both. Each image is paired with its nearest neighbor along the x-axis, resulting in $N-1$ stereo pairs from N images. For each pair $(\mathbf{C}_n, \mathbf{C}_m)$, we use rectified images to estimate forward and backward disparities $(\mathcal{D}^{\mathrm{f}}, \mathcal{D}^{\mathrm{b}})$ via the TVS network. Although TVS performs well in overlap regions, it struggles with occluded or single-observed areas. Thus, relying on the full depth map risks including unreliable points. To mitigate this, we hierarchically select high-confidence depth points to build a globally consistent point cloud (Figure 4).

Hierarchical point cloud reconstruction. We divide the forward and backward valid masks $(\mathcal{R}_{\mathrm{f}}, \mathcal{R}_{\mathrm{b}})$ of the stereo pair into three confidence-based subsets, denoted as $\mathcal{R}^{\mathrm{c}}$, $\mathcal{R}^{\mathrm{o}}$, and $\mathcal{R}^{\mathrm{s}}$. The first subset $\mathcal{R}^{\mathrm{c}}$ comprises the most confident depth, where pixel matching along the pair is consistent. The second subset $\mathcal{R}^{\mathrm{o}}$ represents the occluded regions that $\mathcal{R}^{\mathrm{c}}$ cannot cover. The third subset $\mathcal{R}^{\mathrm{s}}$ corresponds to globally single-observed regions, where depth is predicted without the use of stereo priors.

First, we identify reliable depth pixels based on forward-

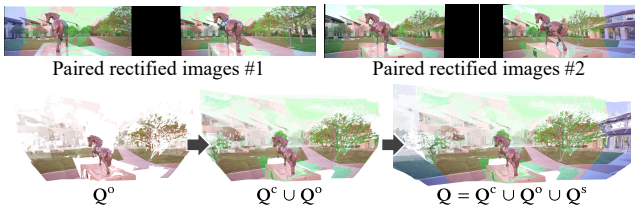


Figure 4. Hierarchical point cloud reconstruction. The first row shows paired rectified images with confidence-based segmentation: red ($\mathcal{R}^{c}$) for pairwise consistent areas, green ($\mathcal{R}^{o}$) for overlap regions with occlusion or mismatching, and blue ($\mathcal{R}^{s}$) for single-view areas. The second row shows the sequential point cloud reconstruction for initialization. The notation follows the main text.

backward consistency [22]. The forward disparity set is

$$\mathcal{R}_{f}^{c} = \left\{ \mathbf{u} \middle| \left\| \mathcal{D}^{f}(\mathbf{u}) + \mathcal{D}^{b}(\mathbf{u} + \mathcal{D}^{f}(\mathbf{u})) \right\|^{2} < \tau_{f} \right\}, \quad (3)$$

where $\tau_{f} = \alpha_1(\|\mathcal{D}^{f}(\mathbf{u})\|^2 + \|\mathcal{D}^{b}(\mathbf{u} + \mathcal{D}^{f}(\mathbf{u}))\|^2) + \alpha_2$, with α_1 and α_2 as constants. The backward mask is similarly defined. We use them to form a pairwise consistent point cloud $\mathbf{Q}^{c}$ by back-projecting the depth to world space.

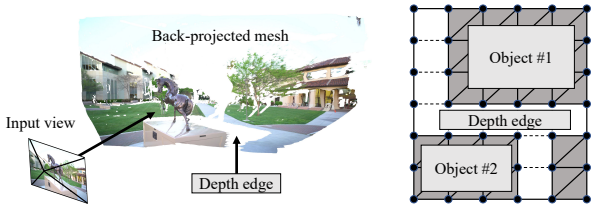
Next, we address the depth that is not represented by $\mathbf{Q}^{c}$ using stereo pair overlap regions. Overlap regions are defined as pixels in one stereo pair for which a match must fall within the rectified image bounds of the other pair: $\mathcal{O} = \{\mathbf{u} \in \mathcal{R}_{f} \mid \mathbf{v} = \mathbf{u} + \mathcal{D}^{f}(\mathbf{u}), \mathbf{v} \in \mathcal{R}_{b}\}$. The areas without projected $\mathbf{Q}^{c}$ within the overlap regions are designated as the second confidence subset $\mathcal{R}^{o} = \{\mathbf{u} \in \mathcal{O} \mid \mathbf{u} \notin \Pi(\mathbf{Q}^{c})\}$, where $\Pi()$ refers to the projection of $\mathbf{Q}^{c}$ to the rectified images. Using both the first and second confidence subsets, we reconstruct an overlap depth point cloud $\mathbf{Q}^{o}$, and project it onto the single-view coverage areas of each stereo pair.

We define single-view coverage $\mathcal{S} = \{\mathbf{u} \in \mathcal{R} \mid \mathbf{u} \notin \mathcal{O}\}$ as regions within the paired rectified image that have no overlap, where $\mathcal{R}$ refers to valid rectified regions. In these areas, we define the subset $\mathcal{R}^{s} = \{\mathbf{u} \in \mathcal{S} \mid \mathbf{u} \notin \Pi(\mathbf{Q}^{o})\}$, where the projected overlap points are absent over the single-view coverage, similar to the overlap region.

By hierarchically back-projecting pixels $\mathbf{u} \in \mathcal{R}^{c} \cup \mathcal{R}^{o} \cup \mathcal{R}^{s}$, we construct the point cloud $\mathbf{Q}$. We leverage three confidence subsets to effectively minimize multi-view inconsistency while constructing a dense point cloud from TVS depth pairs. This point cloud $\mathbf{Q}$ is used to initialize the Gaussians, providing a stable and accurate starting point.

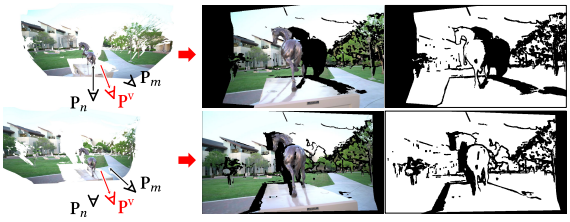
4.2. Virtual View Creation

We select camera pairs $(\mathbf{P}_n, \mathbf{P}_m)$ (as in Section 4.1) and render their depths $\hat{D}_n$ and $\hat{D}_m$ using GS depth rendering. We then construct meshes for each view by depth back-projection as shown in Figure 5. However, naïve back-projection connects all regions indiscriminately, resulting in large, incorrect triangles that fill occluded areas. To handle occlusions more accurately, we detect depth edges and exclude them from the triangulation process. Please refer to the supplementary document for details.



(a) Depth back-projection (b) Mesh triangulation

Figure 5. Mesh triangulation. To render a virtual view, we convert input images into vertex-colored meshes, where the long graph edges that correspond to depth edges are cut.



(a) Mesh generation (b) Rasterized mesh (c) Valid mesh mask

Figure 6. Virtual view creation. We generate meshes using the rendered depth from the nearest training image pair $(\mathbf{P}_n, \mathbf{P}_m)$ (a). We rasterize them onto the virtual view $\mathbf{P}^{v}$ to obtain the projected images $\hat{\mathbf{C}}_n^{v}$ and $\hat{\mathbf{C}}_m^{v}$ as shown in (b). Valid masks M_n^{v} and M_m^{v} in (c), indicating regions without occlusion, are created by comparing the rendered depth with the mesh depth buffer.

We create a virtual camera $\mathbf{P}^{v}$ by interpolating a given camera pair $(\mathbf{P}_n, \mathbf{P}_m)$, using spherical linear interpolation for rotations and linear interpolation for translations. Next, we rasterize the meshes into the virtual camera $\mathbf{P}^{v}$, generating images $\bar{\mathbf{C}}_n^{v}$ and $\bar{\mathbf{C}}_m^{v}$ (Figure 6). To enhance viewpoint diversity, we uniformly sample $K = 20$ virtual cameras $\{\mathbf{P}_l^{v}\}_{l=1}^{K}$ over the interpolation range from -0.5 to 1.5.

Validity masks M_n^{v} and M_m^{v} identify non-occluded regions by comparing the rendered depth $\hat{D}_m^{v}$ at the virtual viewpoint $\mathbf{P}^{v}$ with the mesh depth buffers $\bar{D}_n^{v}$ and $\bar{D}_m^{v}$. We compute the 0.2 quantile of the rendered depth values, denoted as $\hat{D}_{0.2}^{v}$, and define valid masks as:

$$M_k^{v} = \left\{ \mathbf{u} \mid \left\| \hat{D}^{v}(\mathbf{u}) - \bar{D}_k^{v}(\mathbf{u}) \right\| < \tau_d \cdot \hat{D}_{0.2}^{v} \right\}, \quad (4)$$

where $\tau_d = 0.1$, and $k \in \{n, m\}$. Gradient-domain fusion is performed using the rasterized mesh images and masks.

4.3. Virtual View Regularization

We regularize the Gaussians via a screen-space loss, computed from the rasterized reference image pair $\bar{\mathbf{C}}^{v} = \{\bar{\mathbf{C}}_n^{v}, \bar{\mathbf{C}}_m^{v}\}$ and their corresponding masks $M^{v} = \{M_n^{v}, M_m^{v}\}$, which define the reference gradient fields.

Gradient-domain fusion. Gradient-domain fusion aims to minimize the difference between the gradients of the blended output and the reference images within overlapping

regions, satisfying given boundary conditions, as formulated in Energy function (2). Rather than explicitly solving the Poisson equation, we perform implicit GF during Gaussian optimization. A virtual view $\hat{\mathbf{C}}^{\mathrm{v}}$ rendered at the virtual camera $\mathbf{P}^{\mathrm{v}}$ using Gaussian Splatting is supervised by aligning its gradients to those of the two reference images $\bar{\mathbf{C}}^{\mathrm{v}}_n$ and $\bar{\mathbf{C}}^{\mathrm{v}}_m$ via an L1 loss:

$$\mathcal{L}_{\mathrm{gf}} = \sum M^{\mathrm{v}} \cdot \left\| \nabla \hat{\mathbf{C}}^{\mathrm{v}} - \nabla \bar{\mathbf{C}}^{\mathrm{v}} \right\|. \tag{5}$$

We supervise gradients not only in the overlapping valid regions $M^{\mathrm{v}}_n \cap M^{\mathrm{v}}_m$ but also across each reference image's entire valid mask. This naturally induces a soft constraint analogous to the Neumann boundary condition in Poisson blending, which aligns the results' image gradients with the reference images at the stitching boundary. Using an L1 loss, instead of the L2 loss in function (2), avoids bias toward a single reference and encourages soft blending of gradients from both images. By minimizing the gradient differences in screen-space, we achieve a similar effect to Poisson blending and force the smooth color variations captured in the input images to be preserved in the virtual views.

Occlusion handling. For regions invisible in both neighboring training pairs, we lack reliable cues–no overlapping pixels for gradient fields and no exclusive areas for boundary constraints. We instead use the surrounding context to smoothly complete the occluded areas. We define the union of valid masks as $M^{\mathrm{v}}_{nm} = M^{\mathrm{v}}_n \cup M^{\mathrm{v}}_m$ and apply L1 total variation (TV) regularization to the rendered view:

$$\mathcal{L}_{\mathrm{o}} = \sum (1 - M^{\mathrm{v}}_{nm}) \cdot \left\| \nabla \hat{\mathbf{C}}^{\mathrm{v}} \right\|. \tag{6}$$

This loss encourages piecewise smoothness in regions where geometry cannot be reconstructed, thereby improving the visual continuity of novel view results.

4.4. Optimization

We randomly sample 1/10 of the point cloud $\mathbf{Q}$ from Section 4.1 to initialize the Gaussian primitives, optimizing with color loss $\mathcal{L}_1$ and D-SSIM loss $\mathcal{L}_{\text{D-SSIM}}$ from 3DGS [15]. We also use depth loss from the initial point cloud, along with depth TV regularization. To address the depth inconsistency across rendered depth and initial stereo depth, we supervise rendered depth $\hat{D}$ using the projected multi-view depth map $D^{\mathbf{Q}}$ derived from the initial point cloud $\mathbf{Q}$:

$$\mathcal{L}_{\mathbf{Q}} = \sum M^{\mathbf{Q}} \cdot \left\| \hat{D} - D^{\mathbf{Q}} \right\|, \tag{7}$$

where $M^{\mathbf{Q}}$ is the valid mask for the projected depth map. Additionally, L2 depth TV regularization $\mathcal{L}_{\mathrm{dTV}} = \sum \|\nabla \hat{D}\|_2$ ensures consistent variations between depth values. Ultimately, the depth regularizer is defined as:

$$\mathcal{L}_{\mathrm{d}} = \lambda_{\mathbf{Q}} \mathcal{L}_{\mathbf{Q}} + \lambda_{\mathrm{dTV}} \mathcal{L}_{\mathrm{dTV}}. \tag{8}$$

The total loss function for optimization is defined as:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda \mathcal{L}_{\text{D-SSIM}} + \lambda_{\mathrm{gf}} \mathcal{L}_{\mathrm{gf}} + \lambda_{\mathrm{o}} \mathcal{L}_{\mathrm{o}} + \mathcal{L}_{\mathrm{d}}, \tag{9}$$

where $\lambda = 0.2$, $\lambda_{\mathrm{o}} = 0.003$, $\lambda_{\mathbf{Q}} = 0.01$ and $\lambda_{\mathrm{dTV}} = 0.04$. We vary λ_{gf} based on the interpolation ratio of the virtual views. Specifically, when the virtual view is interpolated between input views, we set $\lambda_{\mathrm{gf}} = 10$, while for extrapolated views, we use $\lambda_{\mathrm{gf}} = 1$. We construct meshes and rasterize images for reference gradients once at 3K iterations, when the Gaussian geometry has stabilized; further mesh updates bring negligible improvement and are thus skipped. The reference images are kept fixed throughout optimization to prevent contamination from rendered Gaussians.

5. Experiments

Datasets and metrics. We use two real-world datasets with diverse indoor and outdoor scenes that exhibit view-dependent color: Tanks and Temples (8 scenes) [16] and DL3DV-10K (11 scenes) [21]. All images are downsampled to 1/4 resolution, with 17 sparsely selected images per scene having sufficient overlap. We use the first, middle, and last images for training, and the rest for evaluation. We measure standard metrics such as PSNR and SSIM [33], and perceptual metrics including LPIPS [40] and DISTS [9] to better capture perceptual artifacts like seams.

Comparison. We compare our method against a range of view synthesis techniques for sparse input, including NeRF-based methods (FreeNeRF [36], FlipNeRF [30], and SparseNeRF [32]), 3DGS-based methods using monocular depth priors (DNGaussian [20] and CoherentGS [25]), as well as approaches using MVS neural networks (MVPGS [35] and SCGaussian [26]). In addition, we evaluate against MVSplat [7] and TranSplat [38], a generalizable model designed for view synthesis from sparse input. We also evaluate alternative regularizers replacing our GF loss to test their effectiveness in improving plausibility of view-dependent functions, using both screen-space virtual views and world-space strategies that directly target Gaussians.

Implementation details. We use an AMD Ryzen 9 7950X processor and one NVIDIA RTX 4090 GPU. Gaussians are optimized for 5K iterations and densified every 100 iterations from the 500–4,000th iteration. We use Open3D [41] for mesh rendering, and training takes five minutes per scene; refer to the supplementary material for more details.

5.1. View Synthesis Comparisons

Quantitative evaluation. NeRF-based methods interpolate unobserved or ambiguous information smoothly, reporting lower metrics (Table 1). Generalizable models such as MVSplat prioritize generalization capabilities but sacrifice quality. GS-based methods are sensitive to initialization: DNGaussian and SCGaussian suffer from sparse

Table 1. Quantitative comparison on Tanks and Temples and DL3DV-10K datasets. Our method achieves competitive results across all metrics and particularly excels in perceptual metrics such as LPIPS and DISTS.

Method	Tanks and Temples				DL3DV-10K				Average			
	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
FreeNeRF [36]	20.50	0.6504	0.3927	0.1882	22.16	0.7094	0.3640	0.1917	21.46	0.6846	0.3761	0.1902
FlipNeRF [30]	21.30	0.6825	0.3550	0.1758	21.76	0.7129	0.3470	0.1804	21.57	0.7001	0.3504	0.1785
SparseNeRF [32]	21.67	0.6682	0.3869	0.2029	21.90	0.6870	0.3919	0.2293	21.80	0.6791	0.3898	0.2182
MVSplat [7]	14.88	0.3558	0.5136	0.2198	14.40	0.3724	0.5253	0.2326	14.60	0.3654	0.5204	0.2272
TranSplat [38]	15.09	0.3766	0.5063	0.2226	14.48	0.3859	0.5239	0.2390	14.74	0.3820	0.5165	0.2321
DNGaussian [20]	20.19	0.6404	0.4225	0.1830	19.82	0.6476	0.3963	0.1653	19.98	0.6446	0.4073	0.1728
CoherentGS [25]	19.18	0.6941	0.2829	0.1272	20.74	0.7237	0.2707	0.0973	20.08	0.7112	0.2758	0.1099
SCGaussian [26]	17.07	0.4827	0.5260	0.2462	19.16	0.6114	0.4359	0.2003	18.28	0.5572	0.4738	0.2196
MVPGS [35]	23.12	0.8086	0.1984	0.0804	24.28	0.8231	0.2033	0.0767	23.79	0.8170	0.2012	0.0783
Ours	23.92	0.8207	0.1698	0.0627	24.33	0.8223	0.1820	0.0548	24.16	0.8216	0.1769	0.0581

Figure 7. Qualitative comparisons on the Tanks and Temples dataset. Novel views are reconstructed using three input images. Our method better handles color variations across views, enabling perceptually plausible view synthesis with better visual quality.

Figure 8. Qualitative comparisons on the DL3DV-10K dataset. Novel views are reconstructed using three input images. Our method demonstrates robust performance even in scenes with large depth ranges or complex geometries, maintaining fine details.

or random point clouds, while CoherentGS struggles with unstable monocular depth. MVPGS enhances robustness with MVS-based initialization but exhibits color overfitting, resulting in visible seams and a lower perceptual score. Our method uses TVS depth priors and produces smooth view-dependent colors, achieving higher perceptual metrics (LPIPS, DISTS) with competitive PSNR and SSIM values.

Qualitative evaluation. Figures 7 and 8 present qualitative comparisons on the Tanks and Temples dataset and the DL3DV-10K dataset, respectively. NeRF-based approaches

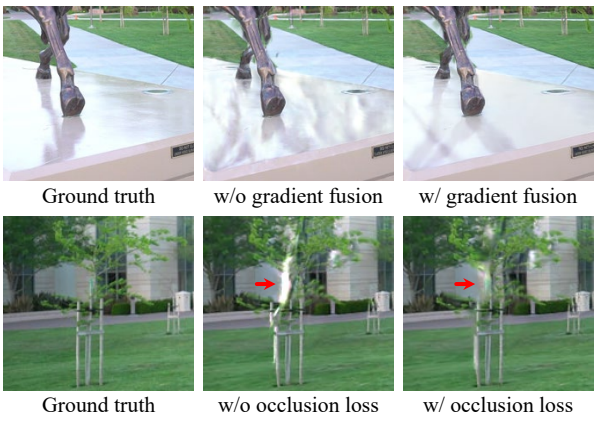


Figure 9. Qualitative ablation of gradient-domain fusion and occlusion loss. Our GF loss smooths view-dependent functions and enhances perceptual plausibility in novel views, while the occlusion loss encourages smooth inpainting of occluded regions, both contributing to visually more coherent synthesis.

like SparseNeRF, relying on MLPs, often overly smooth unobserved views, resulting in a noticeable reduction in overall image quality. CoherentGS often produces distorted geometries due to a shortage of multi-view constraints. Moreover, its direct use of pixel colors frequently results in seam artifacts, because of the overfitting to the regions with significant color differences (third and fourth rows of Figure 7 and third row of Figure 8). SCGaussian, which relies on matched features for initialization, effectively captures overall image trends but struggles to preserve fine details. MVPGS, while effective in many scenarios, exhibits artifacts and blurred colors in regions with high color variance due to its limited handling of view-dependent colors. Our approach of transferring the gradient of the Poisson-blended image to the gradient of the virtual view more successfully preserves fine details with smoother color reproduction.

5.2. Ablation Studies

We initialize 3D Gaussians from dense point clouds generated by TVS, enabling stable view synthesis under sparse inputs. To reduce color overfitting to training views–often causing unnatural color shifts during view transitions–we use GF loss at virtual views to smooth the view-dependent functions. We further introduce an occlusion loss that promotes smooth completion of globally occluded regions by referencing nearby visible areas. Table 2 and Figure 9 show the ablation results for each component. Our initialization enhances multi-view consistency, resulting in stronger overall metrics. GF loss improves perceptual quality, as reflected in LPIPS and DISTS, by promoting smoother and more coherent view-dependent color at the test view. Occlusion loss yields marginal quantitative gains but better visual plausibility in occluded areas (Figure 9, second row).

Table 2. Ablation study of initialization, gradient-domain fusion loss, and occlusion loss on the Tanks and Temples dataset.

Our init.	Grad. fusion	Occl.	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
–	–	–	20.28	0.7416	0.2592	0.0967
✓	–	–	23.51	0.8124	0.1771	0.0708
✓	✓	–	23.75	0.8193	0.1702	0.0641
✓	✓	✓	23.92	0.8207	0.1698	0.0627

Table 3. Comparisons of gradient-domain fusion loss on the Tanks and Temples dataset.

Gradient-domain fusion comparisons	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
(a) Sparse point cloud init.	23.95	0.8176	0.1907	0.0734
(b) k-NN color consistency	23.80	0.8116	0.1932	0.0809
(c) Color TV on virtual views	21.79	0.7396	0.2667	0.1668
(d) L1 loss on virtual views	23.72	0.8105	0.1992	0.0929
(e) Poisson-blended virtual views.	23.75	0.8089	0.2019	0.0976
(f) Gradient-domain fusion (ours)	23.92	0.8207	0.1698	0.0627

5.3. Gradient-domain Fusion Loss Comparisons

To ensure smooth view transitions, we explore alternatives to our virtual-view-based GF, replacing only that loss while keeping others unchanged. Table 3 and Figure 10 show results; implementation details are in the supplementary material. (a) and (b) operate in world-space by manipulating Gaussians directly, while (c)—(f) apply screen-space regularization with virtual views, as in ours.

(a) Samples 1% of the point cloud to reduce Gaussians and smooth view-dependent functions, and (b) applies L2 regularization via k-NN (50 neighbors) to align Gaussian colors; both yield good metrics but degrade unseen-view quality with occlusion artifacts and abrupt shifts. (c) L1 total variation on virtual views smooths seams but removes fine details. (d) Replaces GF with L1 photometric supervision on virtual views, worsening color overfitting. (e) Poisson blends two virtual reference images to produce pseudo-GT; while partially reducing seams, blending inconsistencies cause uneven smoothing, and blurriness propagates to test views. (f) Ours preserves sharpness and achieves seamless synthesis, showing the effectiveness of GF loss.

5.4. User Study

To assess the perceptually plausible color transitions (**Q1**) and overall quality (**Q2**), we conduct two user studies: (**A**) Comparison among FreeNeRF, CoherentGS, MVPGS, and ours, and (**B**) Comparison of alternatives replacing only the GF in Section 5.3 (a), (b), (d), (e), including ours. Each study involves 18 participants using 8-second, 20 fps novel-view videos from *Tanks and Temples* on a 46-inch FHD monitor. In each trial, participants view two videos side by side and answer the following questions using a two-alternative forced-choice (2AFC) paradigm. We evaluate all method–scene combinations in random order with left/right positions also randomized to avoid biased evaluation. Refer to the supplemental document for more details.

We report pairwise win rates against competitors in Table 4. High win rates with sufficient trials lead to extremely small paired t-test p-values ($p < 0.0001$), omitted in the ta-

Ground truth (a) Sparse init. (b) k-NN consist. (c) Color TV (d) L1 virtual view (e) Poisson-blended (f) Ours

Figure 10. Qualitative comparisons of gradient-domain fusion loss with alternative strategies. Our GF loss produces smooth and perceptually non-objectionable color transitions without sacrificing fine details.

Table 4. Pairwise win rates of our method against baselines and gradient fusion alternatives. The p-values are omitted since all comparisons yield extremely low values ($p < 0.0001$).

(A) Baselines	**Q1**. Win (%)	**Q2**. Win (%)
vs. FreeNeRF [36]	98.6	100.0
vs. CoherentGS [25]	88.2	91.0
vs. MVPGS [35]	87.5	89.6
(B) Gradient-domain fusion alternatives	**Q1**. Win (%)	**Q2**. Win (%)
vs. (a) Sparse point cloud init.	88.9	88.9
vs. (b) k-NN color consistency	92.4	87.5
vs. (d) L1 loss on virtual views	90.3	91.0
vs. (e) Poisson-blended virtual views	89.6	92.4

ble, indicating that our superiority is highly unlikely due to chance. Our method outperforms both prior approaches and GF alternatives. Results for natural color transitions (**Q1**) and overall quality (**Q2**) show similar trends, yet method-specific differences appear. For example, the k-NN-based 3D regularizer (b) has little effect on color plausibility (**Q1**) but improves thin-structure coherence, boosting the overall quality (**Q2**). This result shows that our approach excels in natural color transitions and can be further enhanced by complementary regularization.

While t-tests confirm pairwise differences, they are less intuitive for overall ranking. Thus, we apply a Bayesian Bradley-Terry model [4] with MCMC to estimate skill scores and credible intervals (Figure 11). Our method ranks highest with nearly non-overlapping intervals, showing clear superiority over GF alternatives and other baselines. These results indicate that seamless color transitions strongly correlate with human preference, even when differences in PSNR and SSIM are small, highlighting the perceptual benefits of our method.

6. Discussion and Conclusion

We have presented a view synthesis method that enables seamless transitions while preserving fine details from sparse inputs. Our approach initializes robust Gaussian primitives using two-view stereo with confidence-based aggregation and integrates gradient-domain fusion into Gaussian Splatting via virtual view regularization with depth back-projection. We compare against existing sparse view synthesis techniques and alternatives replacing gradient-domain fusion. User studies demonstrate that seamless color transitions strongly correlate with overall quality, with our method being clearly preferred in both t-tests and Bradley–Terry analysis. Ultimately, we show that gradient-domain fusion facilitates effective and seamless view synthesis, even in 3D reconstruction scenarios.

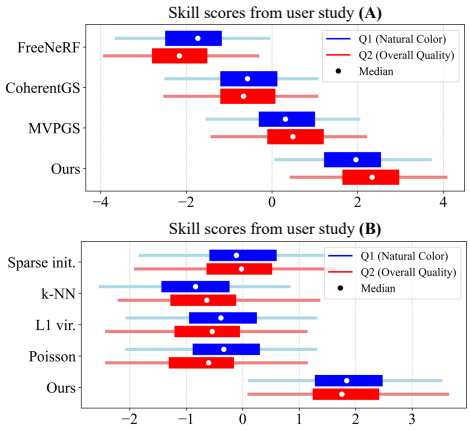


Figure 11. Skill scores from user studies **(A)** and **(B)**. The Bayesian Bradley–Terry model estimates median skill scores (circles) with 50% (thick bars) and 95% (thin bars) highest-density credible intervals. Higher scores indicate greater user preference

While our method improves perceptual quality and visual plausibility, it depends on accurate Gaussian Splatting depth; errors can degrade blending. Future work may incorporate geometry enhancements from methods such as 2DGS [14] or SuGaR [13], and address globally unseen regions with pretrained inpainting models. Moreover, smooth color transitions do not always reflect true appearance: highly specular materials exhibit sharp BRDF lobes that are difficult to recover from sparse views. Addressing this remains a promising direction for future work.

Acknowledgements Min H. Kim acknowledges the Samsung Research Funding & Incubation Center (SRFC-IT2402-02), the Korea NRF grant (RS-2024-00357548), the MSIT/IITP of Korea (RS-2022-00155620, RS-2024-00398830, RS-2024-00436680, and 2017-0-00072), and Microsoft Research Asia. James Tompkin acknowledges NSF CAREER 2144956.

References

[1] Pravin Bhat, C. Lawrence Zitnick, Noah Snavely, Aseem Agarwala, Maneesh Agrawala, Brian Curless, Michael Cohen, and Sing Bing Kang. Using photographs to enhance videos of a static scene. In *Eurographics Symp. on Rendering 2007*, pages 327–338, 2007.

[2] Pravin Bhat, Brian Curless, Michael Cohen, and C Lawrence Zitnick. Fourier analysis of the 2d screened poisson equation for gradient domain problems. In *Proc. ECCV 2008*, pages 114–128. Springer, 2008.

[3] Pravin Bhat, C Lawrence Zitnick, Michael Cohen, and Brian Curless. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics (TOG)*, 29(2):1–14, 2010.

[4] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[5] Chenjie Cao, Xinlin Ren, and Yanwei Fu. Mvsformer: Multi-view stereo by learning robust image features and temperature-based depth. *Transactions of Machine Learning Research*, 2023.

[6] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proc. ACM SIGGRAPH '93*, page 279–288, 1993.

[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 *Proc. ECCV 2025*, pages 370–386, 2025.

[8] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics (TOG)*, 31(4):1–10, 2012.

[9] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.

[10] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv e-prints*, pages arXiv–2403, 2024.

[11] Zeev Farbman, Raanan Fattal, and Dani Lischinski. Convolution pyramids. *ACM Transactions on Graphics (TOG)*, 30 (6):175, 2011.

[12] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. *ACM Transactions on Graphics (TOG)*, 21(3):249–256, 2002.

[13] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proc. IEEE/CVF CVPR 2024*, pages 5354–5363, 2024.

[14] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *Proc. ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023.

[16] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. In *ACM Transactions on Graphics (TOG)*, page 78. ACM, 2017.

[17] Johannes Kopf, Michael F Cohen, and Richard Szeliski. First-person hyper-lapse videos. *ACM Transactions on Graphics (TOG)*, 33(4):1–10, 2014.

[18] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. Seamless image stitching in the gradient domain. In *Proc. ECCV 2004*, pages 377–389, 2004.

[19] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proc. ACM SIGGRAPH '96*, page 31–42, 1996.

[20] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proc. IEEE/CVF CVPR 2024*, pages 20775–20785, 2024.

[21] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proc. IEEE/CVF CVPR 2024*, pages 22160–22169, 2024.

[22] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proc. the AAAI conference on artificial intelligence*, 2018.

[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. In *Proc. ECCV 2020*, pages 405–421. Springer, 2020.

[24] 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 *Proc. IEEE/CVF CVPR 2022*, pages 5480–5490, 2022.

[25] Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. Coherentgs: Sparse novel view synthesis with coherent 3d gaussians. In *Proc. ECCV 2025*, pages 19–37. Springer, 2025.

[26] Rui Peng, Wangze Xu, Luyang Tang, Liwei Liao, Jianbo Jiao, and Ronggang Wang. Structure consistent gaussian splatting with matching prior for few-shot novel view synthesis. In *Proc. NeurIPS 2024*, 2024.

[27] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, 2003.

[28] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proc. ECCV 2016*, pages 501–518. Springer, 2016.

[29] Steven M Seitz and Charles R Dyer. View morphing. In *Proc. ACM SIGGRAPH '96*, pages 21–30, 1996.

[30] Seunghyeon Seo, Yeonjin Chang, and Nojun Kwak. Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In *Proc. IEEE/CVF CVPR 2023*, pages 22883–22893, 2023.

[31] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proc. IEEE/CVF CVPR 2023*, pages 1599–1610, 2023.

[32] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proc. IEEE/CVF CVPR 2023*, pages 9065–9076, 2023.

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

[34] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[35] Wangze Xu, Huachen Gao, Shihe Shen, Rui Peng, Jianbo Jiao, and Ronggang Wang. Mvpgs: Excavating multi-view priors for gaussian splatting from sparse input views. In *Proc. ECCV 2025*, pages 203–220. Springer, 2025.

[36] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proc. IEEE/CVF CVPR 2023*, pages 8254–8263, 2023.

[37] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proc. IEEE/CVF CVPR 2024*, pages 10371–10381, 2024.

[38] Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In *Proc. AAAI 2025*, pages 9869–9877, 2025.

[39] Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. Cor-gs: sparse-view 3d gaussian splatting via co-regularization. In *Proc. ECCV 2025*, pages 335–352. Springer, 2025.

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

[41] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[42] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *Proc. ECCV 2025*, pages 145–163. Springer, 2025.

[43] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.