Pano-GS: Perception-Aware Gaussian Optimization with Gradient Consistency and Multi-Criteria Densification for High-Quality Rendering

Yang Deng[1], Zhanke Wang[1], Jiahao Wu[1,2], Jie Liang[1,2], Jingui Ma[1], Yang Hu[1], Ronggang Wang[1,2]*

[1]Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Shenzhen Graduate School, Peking University,

[2] Peng Cheng Laboratory

{dengyang, zk_wang, wjh0616, liangjie}@stu.pku.edu.cn, majingui102@gmail.com, yanghu@stu.pku.edu.cn, rgwang@pkusz.edu.cn

Abstract

Reconstructing 3D scenes from multi-view image sequences remains a significant challenge in practical applications. While recent advances in 3D Gaussian Splatting have enabled high-quality rendering, existing methods rely heavily on pixel-level $\mathcal{L}_1$ loss, which misaligns with human perception, leading to a lack of high-frequency details and the emergence of artifacts. Additionally, the position gradient-based densification strategy often results in under-densified Gaussian primitives, thereby desgrading rendering quality. To address these challenges, we propose Pano-GS, a perception-aware Gaussian optimization framework. Specifically, we introduce a gradient consistency-constrained loss to capture high-frequency details, mitigating the inherent shortcomings of traditional $\mathcal{L}_1$ loss and enhancing reconstruction fidelity. In addition, we use a multi-criteria densification strategy to reduce the sole reliance on average position gradients. Extensive experiments demonstrate that Pano-GS achieves state-of-the-art performance, confirming its effectiveness and robust generalization across diverse real-world scenes.

Introduction

3D scene reconstruction has driven the development of various multimedia technologies including VR, AR, and metaverse. Given multiple 2D images captured from different viewpoints, 3D scene reconstruction aims to recover the geometric structure of a scene through meshes, points or voxels (Botsch et al. 2005; Yifan et al. 2019; Munkberg et al. 2022).

Novel view synthesis (NVS) is a foundational technique in computer graphics and scene reconstruction. It leverages multiple training images with known camera poses to model a scene, and enables the rendering of realistic images from novel viewpoints. Neural Radiance Fields (NeRF) (Mildenhall et al. 2021; Barron et al. 2021, 2022, 2023; Zhang et al. 2020) have dominated NVS through implicit volumetric representations. However, the reliance on MLPs and volume rendering results in prohibitively long training times (often tens of hours) and inefficient rendering, posing significant challenges for real-time applications. 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has emerged as an innovative solution for NVS. Through efficient differentiable rasterization, 3DGS achieves real-time rendering speeds exceeding 60 FPS. This rapid development has garnered significant attention and inspired a proliferation of works (Lu et al. 2024b; Ren et al. 2024; Zhang et al. 2024; Rota Bulò, Porzi, and Kontschieder 2024). Despite these advancements, 3DGS methods still face critical limitations in managing scenes with abundant textures.

Specifically, existing methods often model textured regions using a limited number of large-scale Gaussians (Rota Bulò, Porzi, and Kontschieder 2024), leading to blurring and a lack of fine details, as illustrated in Fig. 1. This issue arises from two primary limitations: (1) The pixel-wise $\mathcal{L}_1$ loss treats all pixel errors uniformly, neglecting textural significance (Wang et al. 2004). This inability to identify poorly reconstructed regions reduces gradient magnitudes, preventing parameter optimization and densification. As shown in Fig. 1, highlighted under-reconstructed regions exhibit smaller pixel errors than regions with low-frequency color differences. This occurs because areas with high-frequency textures, despite being under-reconstructed, still exhibit colors and intensities nearly identical to the ground-truth. Conversely, low-frequency regions produce more noticeable color variations. This discrepancy highlights the necessity for a more refined approach to identifying and addressing under-reconstructed regions. (2) The current densification strategy solely relies on average position gradients, disregarding the scale of Gaussians and the varying complexity of scene, such as long-distance versus short-range details. As a result, the densification process fails to align with the scene's actual requirements, leading to insufficient splits or clones in texture-rich areas (Lu et al. 2024a) and ultimately compromising reconstruction quality.

To address these limitations, we propose Pano-GS, a perception-aware Gaussian optimization framework. First, we introduce a novel gradient consistency-constrained loss to rectify the deficiencies of pixel-level $\mathcal{L}_1$ loss, aligning the gradient differences between the rendered and ground-truth images to restore high-frequency details. Second, we present a multi-criteria densification strategy that integrates maximum and average position gradients, pixel-level significance and Gaussian spatial coverage. This strategy re-

*Corresponding author.

Figure 1: Existing 3DGS-based methods predominantly employ photometric loss ($\mathcal{L}_1$), which only focuses on pixel-level discrepancies. Specifically, $\mathcal{L}_1$ is computed through pixel-wise subtraction between GT and rendered image, which fails to precisely locate under-reconstructed areas, leading to a lack of details and causing artifacts. However, with our proposed gradient consistency-constrained loss ($\mathcal{L}_{\text{GCL}}$), under-reconstructed regions can be effectively identified.

duces sole reliance on average position gradients, ensuring adaptive densification even where gradient signals are inadequate. Third, we devise a stochastic perturbation-driven exploration mechanism, injecting random perturbation into position parameters to explore more optimal spatial distributions of Gaussians.

Comprehensive experiments on Mip-NeRF 360 (Barron et al. 2022), Tanks&Temples (Knapitsch et al. 2017), and Deep Blending (Hedman et al. 2018) demonstrate that our method achieves state-of-the-art performance, confirming its effectiveness in reconstructing fine details and rendering novel views, establishing Pano-GS as a robust solution for intricate scene reconstruction.

In summary, the main contributions of our approach can be outlined as follows:

- We present a gradient consistency-constrained loss that overcomes the limitations of pixel-wise $\mathcal{L}_1$ loss, rectifying texture distortions by aligning gradients between the rendered and ground-truth images.
- We introduce a multi-criteria densification strategy that integrates maximum and average position gradients, pixel-level significance and Gaussian spatial coverage, thereby enhancing densification precision.
- Our method achieves state-of-the-art (SOTA) performance across diverse datasets, demonstrating significant improvements in rendering quality while maintaining real-time rendering capabilities.

Related Work

NeRF-based Novel View Synthesis

In recent years, novel view synthesis research has made substantial progress. NeRF (Mildenhall et al. 2021) has revolutionized 3D scene representation through neural implicit representation, employing multi-layer perceptrons (MLPs) and volume rendering to synthesize photorealistic views. Building on the foundation established by NeRF, subsequent studies have concentrated on various aspects, such as quality enhancement (Barron et al. 2021, 2023), dynamic scene synthesis (Pumarola et al. 2021; Fridovich-Keil et al. 2023; Park et al. 2021), training and rendering speed (Müller et al. 2022; Liu et al. 2020), and sparse inputs (Johari, Lepoittevin, and Fleuret 2022; Wang et al. 2023). Additionally, some approaches have explored alternative representations, such as feature grid-based (Fridovich-Keil et al. 2022; Chen et al. 2022; Liu et al. 2020) or feature point-based (Xu et al. 2022) to model the radiance field. Notably, Mip-NeRF 360 (Barron et al. 2022) achieved state-of-the-art rendering quality by tackling aliasing artifacts. Plenoxels (Fridovich-Keil et al. 2022) optimized a sparse voxel grid for rendering without MLPs. Instant-NGP (Müller et al. 2022) introduced hash grid encodings, which reduced training time from days to minutes. Despite these advancements, NeRF-based methods still face challenges in rendering and training speed, or in the management of intricate scenes.

3DGS-based Novel View Synthesis

Recently, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has emerged as a faster alternative. By employing 3D Gaussian primitives and point-based rasterization for real-time rendering, 3DGS exhibits significant advantages over NeRF-based approaches in both rendering speed and quality, which inspires a series of works (Liang et al. 2024; Lu et al. 2024b; Yu et al. 2024b; Ye et al. 2024; Zhang et al. 2024; Kheradmand et al. 2025; Rota Bulò, Porzi, and Kontschieder 2024; Kheradmand et al. 2024).

Meanwhile, extensive research has been devoted to enhancing 3DGS in various domains, including accelerating training speed (Fang and Wang 2024; Mallick et al. 2024; Lu et al. 2024a), introducing level-of-details (Ren et al. 2024;

Kerbl et al. 2024), improving surface reconstruction accuracy (Huang et al. 2024; Guédon and Lepetit 2024; Cheng et al. 2024; Yu, Sattler, and Geiger 2024; Yu et al. 2024a), dynamic scene modeling (Yang et al. 2024; Yan et al. 2024; Wu et al. 2025b, 2024, 2025a), 3D generation (Tang et al. 2023; Chung et al. 2023; Tang et al. 2024) and optimizing memory usage (Lee et al. 2024; Papantonakis et al. 2024).

Specifically in the area of quality improvement, Scaffold-GS (Lu et al. 2024b) introduced a hierarchical structure that aligns anchors with scene geometry, employing small MLPs to derive 3D Gaussian parameters through view direction, achieving state-of-the-art render quality. Mip-Splatting (Yu et al. 2024b) addressed artifacts by applying a 3D size filter and a 2D Mip anti-aliasing filter. Revised-GS (Rota Bulò, Porzi, and Kontschieder 2024) modified the densification strategy to densify Gaussians in high training error areas. Pixel-GS (Zhang et al. 2024) rescaled position gradients to mitigate artifacts near the camera.

Despite progress, most current 3DGS methods still fall short in complex scenes, particularly those with rich textures. To address this issue comprehensively, we propose a visual perception-aware optimization method: Pano-GS.

Preliminaries

3D Gaussian Splatting

3D Gaussian Splatting represents scenes as a collection of anisotropic 3D Gaussians. Each Gaussian primitive is parameterized by a center position μ, a covariance matrix Σ, an opacity value α, and spherical harmonics coefficients sh. Mathematically, a 3D Gaussian is expressed as:

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

where S is the scaling matrix and R is the rotation matrix.

For rendering, 3D Gaussian primitives are projected to 2D plane, and then the pixel color C is computed by α-blending:

$$C(\mathbf{x}') = \sum_i c_i \alpha_i' \prod_{j=1}^{i-1}(1-\alpha_j'), \quad \alpha_i' = \alpha_i G_i'(\mathbf{x}') \tag{2}$$

Here, c_i denotes the color of the i-th Gaussian, and $G_i'(x')$ represents the 2D projection of the i-th Gaussian at pixel x'.

The model is optimized by minimizing the combination of $\mathcal{L}_1$ and SSIM (Wang et al. 2004) losses between the rendered $\hat{I}$ and ground-truth image I:

$$\mathcal{L}_{\text{reconstruct}} = (1-\lambda)\mathcal{L}_1(I,\hat{I}) + \lambda\mathcal{L}_{ssim}(I,\hat{I}) \tag{3}$$

The adaptive density control mechanism prunes low opacity Gaussians and adds new Gaussians when the average magnitude of positional gradients exceeds a threshold.

Scaffold-GS

In Scaffold-GS, scenes are represented by anchor points. Each anchor comprises a position x_v, a context feature $\mathbf{f}_v$, a scale factor l_v, and learnable offsets O_i $(0 \le i < k)$. Each anchor point dynamically spawns k neural Gaussians with their position μ calculated as:

$$\{\mu_0, ..., \mu_{k-1}\} = x_v + \{\mathcal{O}_0, ...\mathcal{O}_{k-1}\} \cdot l_v. \tag{4}$$

The Gaussian attributes are decoded through lightweight MLPs from anchor features $\mathbf{f}_v$, relative viewing distance δ_{vc} and viewing direction $\vec{d_{cv}}$. The loss function combines $\mathcal{L}_1$ loss, SSIM loss and a volume regularization:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{ssim} + \lambda_{\text{vol}}\mathcal{L}_{\text{vol}} \tag{5}$$

It also incorporates a dynamic anchor densification strategy: anchors with low opacity are pruned, while high position-gradient anchors spawn new ones. This framework serves as the foundation for our method.

Method

Gradient Consistency Constrained Loss

Existing 3DGS-based methods encounter a fundamental dilemma between representing high-frequency details and satisfying optimization objectives (Rota Bulò, Porzi, and Kontschieder 2024). Theoretically, the model should generate numerous small Gaussians in texture-rich regions to capture fine details. However, in practice, the model may cover such areas with large-scale Gaussians, leading to underfitting of high-frequency content and causing blur artifacts.

Although some existing improvements (Rota Bulò, Porzi, and Kontschieder 2024; Zhang et al. 2024) have endeavored to tackle this issue, they overlook an inherent constraint of the primary optimization objective: the pixel-wise $\mathcal{L}_1$ loss. Specifically, the $\mathcal{L}_1$ loss, computed by pixel-wise subtraction between the rendered and ground-truth, calculates pixel errors independently, disregarding gradient correlations among neighboring pixels (e.g., edge continuity), thereby failing to capture structural information. In addition, it exhibits excessive sensitivity to low-frequency color variations while lacking effective constraints on high-frequency details. As shown in Fig. 1, under-reconstructed regions exhibit smaller pixel errors than low-frequency color differences; hence, the $\mathcal{L}_1$ loss fails to distinguish errors in visually crucial regions from those in less critical areas. Consequently, the optimization process neglects the prioritization of high-frequency details and leads to blurriness.

Based on this motivation, we introduce a gradient consistency constrained loss $\mathcal{L}_{\text{GCL}}$. This loss explicitly aligns the gradients of the rendered and ground-truth images, preserving edges and high-frequency textures. We downsample both the rendered image $\hat{I}$ and the ground-truth I to multiple scales and then use a pyramid structure to align their horizontal and vertical gradient fields at each scale as follows:

$$\mathcal{L}_{\text{GCL}} = \sum_{s=1}^{S} \lambda_s \cdot \Big(\|\nabla_x \hat{I}_s - \nabla_x I_s\| + \|\nabla_y \hat{I}_s - \nabla_y I_s\| \Big) \tag{6}$$

where s denotes the pyramid level, λ_s is the weight coefficient for each level, and ∇_x,∇_y represent the horizontal and vertical gradient operators.

By explicitly aligning multi-scale gradient distributions, the $\mathcal{L}_{\text{GCL}}$ guides the optimization process towards rectifying structural distortions, particularly in regions with high-frequency details. This loss item reduces excessive sensitivity to low-frequency color differences, preventing the model

from covering high-frequency areas with large-scale Gaussians. Instead, it helps capture essential texture information, leading to crisper and more detailed reconstructions.

Multi-criteria Densification Strategy

Existing Gaussian densification strategies rely heavily on average position gradients, which are prone to gradient vanishing (Lu et al. 2024a; Ye et al. 2024) and fail to reconstruct fine details. To address this, we introduce a multi-criteria densification approach that dynamically adjusts Gaussian density by explicitly modeling critical regions. It comprises three core modules: pixel significance estimation, Gaussian contribution scoring and multi-trigger densification.

Pixel Significance Estimation. We define an importance map $M \in \mathbb{R}^{H\times W}$ to quantify pixel-level significance. This map comprises two components: blur extent $E_{\text{blur}}(p)$ and texture intensity $E_{\text{texture}}(p)$, calculated as follows:

$$M(p) = a \cdot E_{\text{blur}}(p) + b \cdot E_{\text{texture}}(p) \tag{7}$$

Here, $E_{\text{texture}}(p)$ identifies regions with strong textures through edge detection, while $E_{\text{blur}}(p)$ assesses blur severity by computing gradient differences between the rendered and ground-truth images via a Scharr operator (Bradski 2000). The larger the gradient difference from ground-truth, the more blurred the region. Weights a and b balance the importance of these two components.

Gaussian Contribution Scoring. Next, we compute a contribution score S_i for each Gaussian G_i to quantify its relation to pixel significance. Since pixel rendering involves contributions from multiple Gaussians (Eq. 2), establishing this link is non-trivial. We therefore integrate two factors: the accumulated significance scores of the pixels covered by G_i and its actual contribution to the final rendering, yielding:

$$S_i = \sum_{p\in P_i} M(p)\cdot\omega_i, \quad \omega_i = \alpha_i' \prod_{j=1}^{i-1}(1-\alpha_j') \tag{8}$$

where P_i represents the set of pixels influenced by G_i when projected onto the 2D plane, α_i' is defined as in Eq. 2, and ω_i measures the Gaussian's contribution strength.

Multi-Trigger Densification. To address the limitations of densification based solely on average position gradient, we devise a multi-trigger mechanism that activates densification when either of the following conditions is met:

Gradient Trigger: The original densification strategy identifies growth candidates by averaging view-space positional gradients over M iterations. While effective in most scenarios, this approach overlooks Gaussians with substantial positional gradients in a few views. Therefore, we trigger densification when either the maximum or average position gradient exceeds its respective threshold:

$$\|\nabla_{\text{max}}\| > \tau_{\text{max}} \quad \text{or} \quad \|\nabla_{\text{avg}}\| > \tau_{\text{avg}} \tag{9}$$

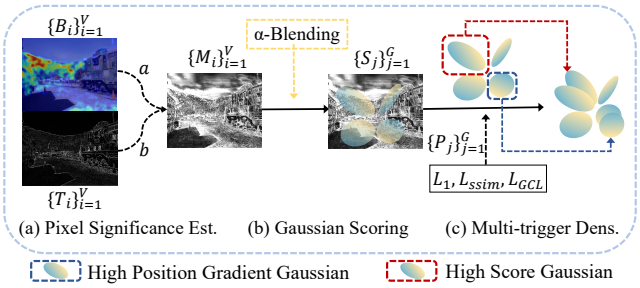


Figure 2: Overview of Multi-criteria Densification Strategy. (a) We establish an importance map $\{M_i\}_{i=1}^V$ to quantify pixel-level significance, calculated via blur extent $\{B_i\}_{i=1}^V$ and texture intensity $\{T_i\}_{i=1}^V$. (b) Through α-blending, we compute a contribution score $\{S_j\}_{j=1}^G$ for each Gaussian. (c) Densification is triggered when either the position gradient $\{P_j\}_{j=1}^G$ or the score $\{S_j\}_{j=1}^G$ exceeds the thresholds.

where

$$\|\nabla_{\text{max}}\| = \max_{k\in\mathcal{V}} \left\|\frac{\partial \mathcal{L}_k}{\partial \mu_{\text{2D}}}\right\|, \quad \|\nabla_{\text{avg}}\| = \frac{1}{|\mathcal{V}|}\sum_{k\in\mathcal{V}} \left\|\frac{\partial \mathcal{L}_k}{\partial \mu_{\text{2D}}}\right\| \tag{10}$$

Here, ∇_{max} denotes the maximum gradient, and ∇_{avg} signifies the average gradient. $\mathcal{V}$ represents the set of views, $\mathcal{L}_k$ is the loss for view k, and $\mu_{\mathbf{2D}}$ is the 2D position of the Gaussian. τ_{max} and τ_{avg} are predefined hyper-parameters.

Significance Score Trigger: The score S_i (computed by Eq. 8) indicates the necessity for densifying each Gaussian G_i. If G_i covers a large area dominated by high-importance pixels, it indicates under-reconstruction, and densification is triggered to increase local density. Conversely, densification is unnecessary if the covered area comprises mostly low-importance pixels or if the Gaussian covers limited areas. This strategy dynamically balances Gaussian distribution between under-reconstructed regions and well-reconstructed or low-frequency areas, enhancing detail fidelity and rendering quality. Specifically, densification is triggered when a Gaussian's score $S_i > \bar{S} + 2\sigma$, where $\bar{S}$ and σ represent the mean and standard deviation of scores across all Gaussians. For the algorithm pipeline, please refer to Fig. 2.

Stochastic Perturbation Exploration

During optimization, position parameters may stagnate when minor position adjustments have minimal impact on the rendered color, resulting in ineffective gradient signals from the loss function. To tackle this, we introduce stochastic perturbation into position parameters, compelling Gaussians to explore better positions and preventing them from getting trapped in local optima when gradients vanish. If a new position reduces the rendering error, the gradient signal would guide the parameters toward the improved optimum, gradually converging to the correct position.

Noteworthy, excessive noise can degrade rendering quality, while too little noise may fail to drive sufficient exploration. Since large-scale Gaussians exhibit greater translational robustness due to broader spatial coverage, their mag-

Method	Dataset Conference	Year	Mip-NeRF 360 PSNR↑	SSIM↑	LPIPS↓	Tanks&Temples PSNR↑	SSIM↑	LPIPS↓	Deep Blending PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF 360	CVPR	2022	27.67	0.793	0.234	22.22	0.759	0.257	29.40	0.901	0.245
3D-GS	SIGGRAPH	2023	27.43	0.814	0.217	23.71	0.845	0.178	29.46	0.900	0.247
Mip-Splatting	CVPR	2024	27.61	0.816	<u>0.215</u>	23.96	0.856	0.171	29.56	0.901	<u>0.243</u>
Scaffold-GS	CVPR	2024	27.59	0.810	0.232	24.05	0.853	0.173	<u>30.21</u>	**0.906**	0.254
Revised-GS	ECCV	2024	27.70	**0.823**	0.223	<u>24.10</u>	<u>0.857</u>	0.183	29.64	<u>0.905</u>	0.303
Pixel-GS	ECCV	2024	<u>27.71</u>	**0.823**	**0.194**	23.80	0.853	<u>0.151</u>	28.93	0.892	0.251
Pano-GS (Ours)	-	-	**28.08**	<u>0.818</u>	**0.194**	**24.56**	**0.862**	**0.136**	**30.49**	**0.906**	**0.229**

Table 1: Quantitative evaluation on the Mip-NeRF 360 (Barron et al. 2022), Tanks&Temples (Knapitsch et al. 2017), and Deep Blending (Hedman et al. 2018) datasets. **Bold** denotes the best results and <u>underline</u> the second best results.

nitudes of positional perturbations must significantly exceed those of small-scale Gaussians. Therefore, the noise term depends on covariances and scales:

$$\epsilon_\mu = \lambda_{\text{lr}} \cdot l_v \cdot \Sigma \cdot \eta \tag{11}$$

Here, $\eta \sim \mathcal{N}(0, I)$ denotes noise sampled from a standard normal distribution, and Σ is the covariance matrix. The product $\Sigma \cdot \eta$ generates anisotropic perturbation aligned with the spatial distribution of Gaussians. The scaling factor l_v (associated with the anchor point) regulates the scale of the spawned neural Gaussians (Lu et al. 2024b). By multiplying with l_v, the relationship between noise intensity and scale is effectively regulated. This mechanism guarantees adequate exploration for larger Gaussians, while mitigating the risk of noise-induced degradation in smaller primitives.

The noise intensity follows a two-stage annealing strategy: warm-up ($t < T_{\text{start}}$) and exploration ($T_{\text{start}} < t < T_{\text{end}}$). During warm-up, perturbations are disabled, allowing the model to converge swiftly without noise interference. Subsequently, during the exploration stage, the noise intensity is initially strong to facilitate thorough exploration of the parameter space, followed by a gradual reduction to ensure stable convergence. Specifically, the learning rate λ_{lr} is set to zero during the warm-up stage to facilitate noise-free learning. In the exploration stage, a cosine annealing strategy (Loshchilov and Hutter 2016) is employed to adjust λ_{lr}:

$$\lambda_{\text{lr}} = \lambda_{\text{final}} + 0.5(\lambda_{\text{init}} - \lambda_{\text{final}}) \left(1 + \cos\left(\pi \cdot \frac{t - T_{\text{start}}}{T_{\text{end}} - T_{\text{start}}}\right)\right) \tag{12}$$

Here, λ_{init} and λ_{final} represent the initial and final learning rates, t is the current iteration, T_{start} is the start iteration of the exploration stage and T_{end} indicates the end iteration.

Implementation Details

Loss function. Our loss function comprises three components: pixel-wise $\mathcal{L}_1$ loss, SSIM term $\mathcal{L}_{ssim}$ and the gradient consistency-constrained loss $\mathcal{L}_{\text{GCL}}$. We set both λ_1 and λ_2 to 0.2. The total loss is formulated as:

$$\mathcal{L} = (1 - \lambda_1)\mathcal{L}_1 + \lambda_1 \mathcal{L}_{ssim} + \lambda_2 \mathcal{L}_{\text{GCL}} \tag{13}$$

Training Details. Our method is implemented in PyTorch (Paszke 2019). We use the Adam optimizer and retain the implementations from Scaffold-GS.

For the gradient consistency-constrained loss, we get multi-scale images by downsampling with average pooling. The pyramid level S is set to 5, and per-level weight λ_s is set to 0.5 (Eq. 6). The weights a and b are set to 50 and 25 (Eq. 7), with the maximum gradient threshold ∇_{max} = 0.0016 and the average gradient threshold ∇_{avg} = 0.0002 (Eq. 9). Multi-criteria densification is introduced at 7,000 iterations, adhering to the coarse-to-fine training paradigm. The perturbation strength λ_{init} is initialized at 0.2 and gradually decreased to λ_{final} = 0.002 (Eq. 12). Stochastic perturbation starts after 3k iterations to avoid disrupting early convergence. The model is trained for 30k iterations. More hyperparameter settings are provided in the appendix. All experiments are conducted on an NVIDIA RTX 3090 GPU.

Experiment

Experimental Setup

Datasets. Following previous works (Lu et al. 2024b; Zhang et al. 2024), our method is evaluated on three public datasets, including 9 scenes from Mip-NeRF 360 (Barron et al. 2022), 2 scenes from Tanks&Temples (Knapitsch et al. 2017), and 2 scenes from Deep Blending (Hedman et al. 2018). To maintain the same experimental settings, Mip-NeRF 360 scenes are downsampled to 1600 pixel width, while other datasets remain at original resolutions.

Baselines. We compare our method with state-of-the-art novel view synthesis approaches, including Mip-NeRF 360 (Barron et al. 2022), 3D-GS (Kerbl et al. 2023), Mip-Splatting (Yu et al. 2024b), Scaffold-GS (Lu et al. 2024b), as well as the recent Revised-GS (Rota Bulò, Porzi, and Kontschieder 2024) and Pixel-GS (Zhang et al. 2024). To ensure a fair comparison, we adopt the official implementations for all baselines, and retrain Scaffold-GS and Pixel-GS to maintain consistent experimental conditions.

Results and Comparisons

Quantitative results. Quantitative results are presented in Tab. 1. Our method consistently outperforms baselines across all three metrics, demonstrating its state-of-the-art performance. This improvement stems from our novel solution to the local underfitting problem inherent in existing 3DGS-based methods. Moreover, its consistent performance on diverse datasets, ranging from texture-rich Mip-NeRF

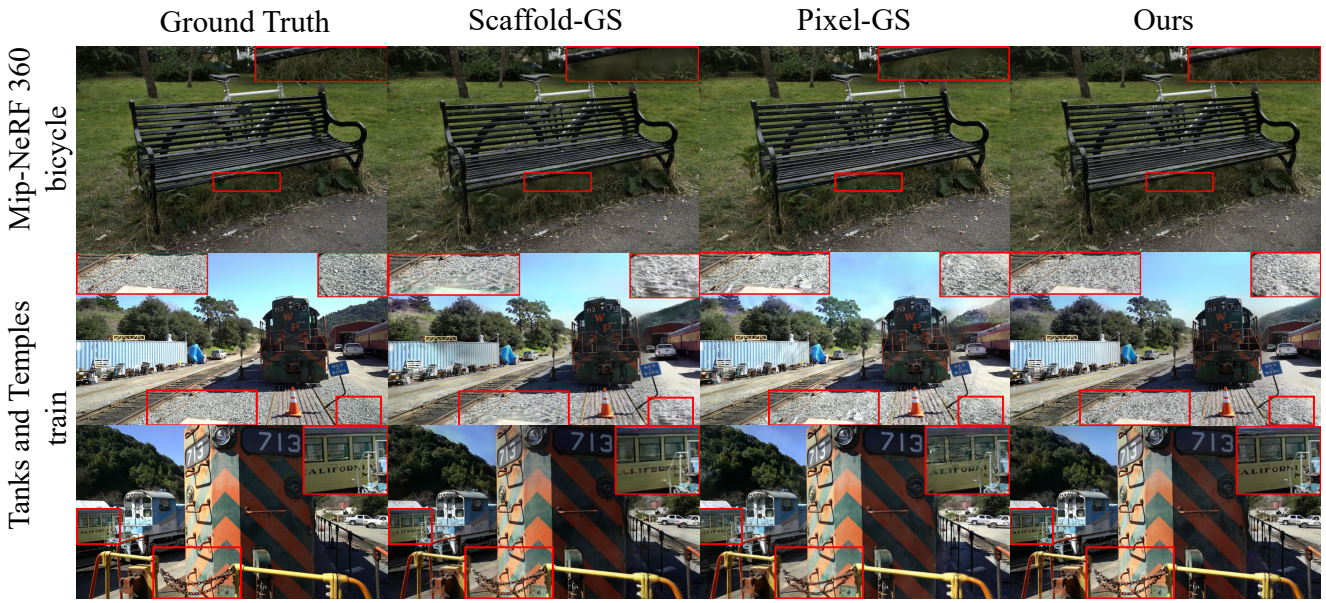


Figure 3: Qualitative comparison of our Pano-GS with state-of-the-art methods including Scaffold-GS (Lu et al. 2024b) and Pixel-GS (Zhang et al. 2024). The red-highlighted regions indicate where our method outperforms others. Compared to baselines, our approach captures high-frequency details, achieving high-quality reconstruction even in challenging areas.

Dataset	Mip-NeRF 360		Tanks&Temples		Deep Blending	
Method	Size	Time	Size	Time	Size	Time
Scaffold-GS	180 MB	40 m	79 MB	19 m	54 MB	25 m
Pixel-GS	1.24 GB	66 m	1.01 GB	41 m	1.08 GB	40 m
Pano-GS	420 MB	70 m	107 MB	41 m	148 MB	39 m

Table 2: Comparison of model size and training time between our method **Pano-GS** and state-of-the-art methods.

360 scenes to weakly textured surfaces in Deep Blending, underscores its generalizability, establishing a new benchmark for real-time, high-fidelity NVS.

Qualitative results. As shown in the magnified regions in Fig. 3, our method significantly reduces blurring and artifacts in challenging regions; by expanding points in these regions, our method enables more detailed reconstruction.

Furthermore, we compare Gaussian point distributions across Scaffold-GS, Pixel-GS, and our Pano-GS in Fig. 4. Scaffold-GS struggles with noticeable blurriness in regions where initial points are sparse. Pixel-GS enhances reconstruction to some extent, but still exhibits blurring and artifacts in challenging areas, as highlighted in Fig. 4; moreover, its new points tend to concentrate in already dense regions, leading to redundancy and only marginal gains. In contrast, Pano-GS distributes points more uniformly across the scene, filling gaps and outperforming Pixel-GS across all datasets.

Efficiency Comparison. Tab. 2 quantifies model size and training efficiency. Compared to Pixel-GS, our Pano-GS strikes a balance between rendering quality and efficiency, delivering high-fidelity novel views with less storage while maintaining the same training time.

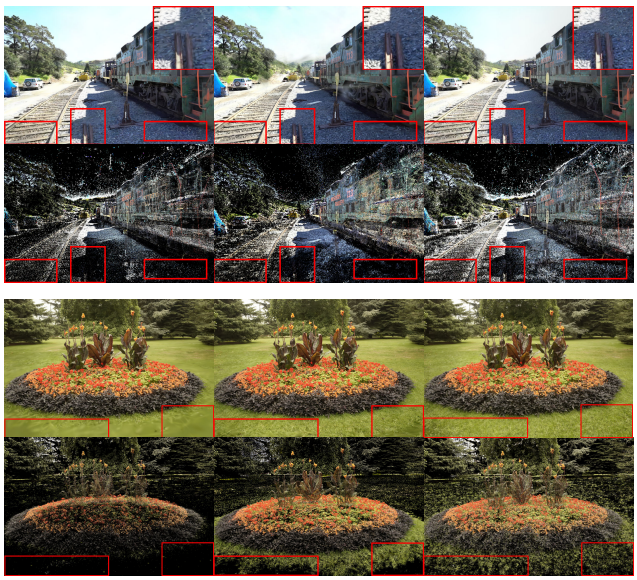
Components			Tanks&Temples		
GCL	MDS	SPE	PSNR ↑	SSIM ↑	LPIPS ↓
			24.05	0.853	0.173
✔			24.45	0.860	0.151
	✔		24.23	0.858	0.146
		✔	24.20	0.856	0.158
✔	✔		24.50	0.861	0.137
✔		✔	24.45	0.861	0.150
	✔	✔	24.39	0.859	0.146
✔	✔	✔	**24.56**	**0.862**	**0.136**

Table 3: **Ablation on the Tanks&Temples.** Our baseline model is Scaffold-GS (Lu et al. 2024b). GCL: Gradient Consistency-constrained Loss; MDS: Multi-criteria Densification Strategy; SPE: Stochastic Perturbation Exploration.

Ablation Study

To assess the effectiveness of the gradient consistency-constrained loss (GCL), multi-criteria densification strategy (MDS), and stochastic perturbation exploration (SPE), we conducted ablation studies on Tanks&Temples datasets.

Validity of GCL. As shown in Tab. 3, the GCL delivers substantial improvements across PSNR, SSIM, and LPIPS, demonstrating its pivotal role. Fig. 5 further illustrates that the GCL markedly improves rendering quality, especially in regions with fine details. This improvement stems from



(a) Scaffold-GS (b) Pixel-GS (c) Ours

Figure 4: Rendered images and corresponding point distributions. Scaffold-GS leaves numerous voids and blurriness. Pixel-GS improves fidelity yet still suffers from uneven point distribution. In contrast, our method effectively adds points where needed, yielding sharper reconstructions.

the GCL's heightened sensitivity to pixel gradients, enabling more precise alignment of rendered images with the high-frequency details of ground-truth.

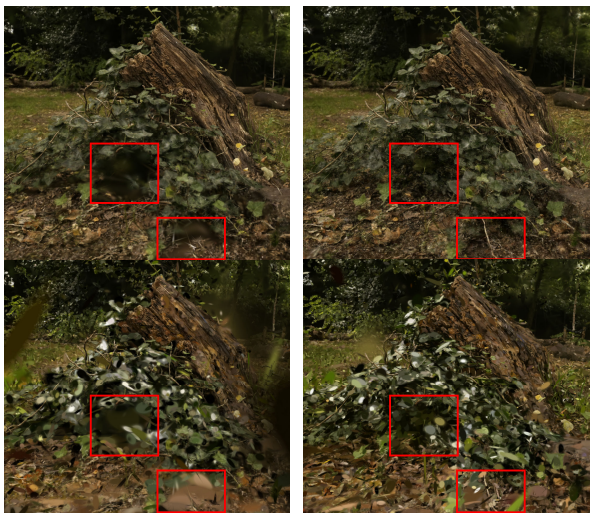
Consideration of MDS. The MDS optimizes reconstruction by dynamically adjusting Gaussian density in under-reconstructed regions. Fig. 6 compares the Gaussian scales in under-reconstructed regions of the original densification strategy with our MDS. The highlighted regions show that MDS effectively eliminates the artifacts caused by potentially oversized Gaussians in the original densification strategy. By generating smaller Gaussians where needed, MDS enhances visual quality by filling gaps and further refines rendering outcomes in intricate scenes.

Effectiveness of SPE. SPE enhances performance in weak gradient regions by introducing random perturbations, helping the model escape local optima, which provides consistent improvements across all metrics, as Tab. 3 shown.

Limitation

Our method enhances detail and visual quality while keeping real-time rendering, demonstrating its effectiveness in solving under-fitting issues. However, our method may increase the number of Gaussians and raise some storage costs. Additionally, our model depends on accurate camera poses and lacks geometric support. These limitations will be prioritized in our future work.

Figure 5: Qualitative results of ablation study. Each component's impact is assessed by removing it from the complete model, with corresponding metrics displayed on the images.



(a) Scaffold-GS (b) Ours

Figure 6: Comparison of Gaussians with two densification strategies. The original strategy of Scaffold-GS (left) results in under-reconstructed regions with oversized Gaussians. In contrast, our MDS (right) effectively fills these gaps via split small-scale Gaussians, enhancing rendering quality.

Conclusion

In this work, we introduce Pano-GS, a perception-aware Gaussian optimization framework. Unlike existing 3DGS-based methods which rely on the pixel-wise $\mathcal{L}_1$ loss and adopt a densification strategy based solely on average position gradients, our approach addresses these limitations through three key innovations: a gradient consistency-constrained loss to capture high-frequency details, a multi-criteria densification strategy that adaptively controls Gaussian density, and a stochastic perturbation mechanism to explore optimal spatial configurations. Extensive experiments demonstrate that Pano-GS achieves state-of-the-art rendering quality while maintaining real-time rendering capabilities, establishing it as a robust solution for complex novel view synthesis tasks.

Acknowledgements

This work is financially supported by Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (Grant No. 2024B1212010006), this work is also financially supported for Outstanding Talents Training Fund in Shenzhen, Shenzhen Science and Technology Program (Grant No. SYSPG20241211173440004 and Grant No. RCJC20200714114435057), and Ant Group.

References

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 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.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 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.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2023. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19697–19705.

Botsch, M.; Hornung, A.; Zwicker, M.; and Kobbelt, L. 2005. High-quality surface splatting on today's GPUs. In *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005.*, 17–141. IEEE.

Bradski, G. 2000. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11): 120–123.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, 333–350. Springer.

Cheng, K.; Long, X.; Yang, K.; Yao, Y.; Yin, W.; Ma, Y.; Wang, W.; and Chen, X. 2024. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first International Conference on Machine Learning*.

Chung, J.; Lee, S.; Nam, H.; Lee, J.; and Lee, K. M. 2023. Luciddreamer: Domain-free generation of 3d gaussian splatting affenes. *arXiv preprint arXiv:2311.13384*.

Fang, G.; and Wang, B. 2024. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, 165–181. Springer.

Fridovich-Keil, S.; Meanti, G.; Warburg, F. R.; Recht, B.; and Kanazawa, A. 2023. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12479–12488.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5501–5510.

Guédon, A.; and Lepetit, V. 2024. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5354–5363.

Hedman, P.; Philip, J.; Price, T.; Frahm, J.-M.; Drettakis, G.; and Brostow, G. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6): 1–15.

Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; and Gao, S. 2024. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, 1–11.

Johari, M. M.; Lepoittevin, Y.; and Fleuret, F. 2022. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18365–18375.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Kerbl, B.; Meuleman, A.; Kopanas, G.; Wimmer, M.; Lanvin, A.; and Drettakis, G. 2024. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4): 1–15.

Kheradmand, S.; Rebain, D.; Sharma, G.; Sun, W.; Tseng, Y.-C.; Isack, H.; Kar, A.; Tagliasacchi, A.; and Yi, K. M. 2024. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37: 80965–80986.

Kheradmand, S.; Rebain, D.; Sharma, G.; Sun, W.; Tseng, Y.-C.; Isack, H.; Kar, A.; Tagliasacchi, A.; and Yi, K. M. 2025. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37: 80965–80986.

Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4): 1–13.

Lee, J. C.; Rho, D.; Sun, X.; Ko, J. H.; and Park, E. 2024. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21719–21728.

Liang, Z.; Zhang, Q.; Hu, W.; Zhu, L.; Feng, Y.; and Jia, K. 2024. Analytic-splatting: Anti-aliased 3d gaussian splatting via analytic integration. In *European conference on computer vision*, 281–297. Springer.

Liu, L.; Gu, J.; Zaw Lin, K.; Chua, T.-S.; and Theobalt, C. 2020. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33: 15651–15663.

Loshchilov, I.; and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Lu, T.; Dhiman, A.; Srinath, R.; Arslan, E.; Xing, A.; Xiangli, Y.; Babu, R. V.; and Sridhar, S. 2024a. Turbo-GS: Accelerating 3D Gaussian Fitting for High-Quality Radiance Fields. *arXiv preprint arXiv:2412.13547*.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024b. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 20654–20664.

Mallick, S. S.; Goel, R.; Kerbl, B.; Steinberger, M.; Carrasco, F. V.; and De La Torre, F. 2024. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, 1–11.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15.

Munkberg, J.; Hasselgren, J.; Shen, T.; Gao, J.; Chen, W.; Evans, A.; Müller, T.; and Fidler, S. 2022. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8280–8290.

Papantonakis, P.; Kopanas, G.; Kerbl, B.; Lanvin, A.; and Drettakis, G. 2024. Reducing the memory footprint of 3d gaussian splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7(1): 1–17.

Park, K.; Sinha, U.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Seitz, S. M.; and Martin-Brualla, R. 2021. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5865–5874.

Paszke, A. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10318–10327.

Ren, K.; Jiang, L.; Lu, T.; Yu, M.; Xu, L.; Ni, Z.; and Dai, B. 2024. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*.

Rota Bulò, S.; Porzi, L.; and Kontschieder, P. 2024. Revising densification in gaussian splatting. In *European Conference on Computer Vision*, 347–362. Springer.

Tang, J.; Chen, Z.; Chen, X.; Wang, T.; Zeng, G.; and Liu, Z. 2024. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, 1–18. Springer.

Tang, J.; Ren, J.; Zhou, H.; Liu, Z.; and Zeng, G. 2023. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*.

Wang, G.; Chen, Z.; Loy, C. C.; and Liu, Z. 2023. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9065–9076.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 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.

Wu, J.; Peng, R.; Jiao, J.; Yang, J.; Tang, L.; Xiong, K.; Liang, J.; Yan, J.; Liu, R.; and Wang, R. 2025a. LocalDyGS: Multi-view Global Dynamic Scene Modeling via Adaptive Local Implicit Feature Decoupling. *arXiv preprint arXiv:2507.02363*.

Wu, J.; Peng, R.; Wang, Z.; Xiao, L.; Tang, L.; Yan, J.; Xiong, K.; and Wang, R. 2025b. Swift4D: Adaptive divide-and-conquer Gaussian Splatting for compact and efficient reconstruction of dynamic scene. *arXiv preprint arXiv:2503.12307*.

Xu, Q.; Xu, Z.; Philip, J.; Bi, S.; Shu, Z.; Sunkavalli, K.; and Neumann, U. 2022. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5438–5448.

Yan, J.; Peng, R.; Tang, L.; and Wang, R. 2024. 4D Gaussian Splatting with Scale-aware Residual Field and Adaptive Optimization for Real-time rendering of temporally complex dynamic scenes. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 7871–7880.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2024. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 20331–20341.

Ye, Z.; Li, W.; Liu, S.; Qiao, P.; and Dou, Y. 2024. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 1053–1061.

Yifan, W.; Serena, F.; Wu, S.; Öztireli, C.; and Sorkine-Hornung, O. 2019. Differentiable surface splatting for point-based geometry processing. *ACM Transactions On Graphics (TOG)*, 38(6): 1–14.

Yu, M.; Lu, T.; Xu, L.; Jiang, L.; Xiangli, Y.; and Dai, B. 2024a. Gsdf: 3dgs meets sdf for improved rendering and reconstruction. *arXiv preprint arXiv:2403.16964*.

Yu, Z.; Chen, A.; Huang, B.; Sattler, T.; and Geiger, A. 2024b. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19447–19456.

Yu, Z.; Sattler, T.; and Geiger, A. 2024. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics (TOG)*, 43(6): 1–13.

Zhang, K.; Riegler, G.; Snavely, N.; and Koltun, V. 2020. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*.

Zhang, Z.; Hu, W.; Lao, Y.; He, T.; and Zhao, H. 2024. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *European Conference on Computer Vision*, 326–342. Springer.