MonoGSDF: Exploring Monocular Geometric Cues for Gaussian Splatting-Guided Implicit Surface Reconstruction

Kunyi Li[1, 4]
kunyi.li@tum.de
Michael Niemeyer[2]
mniemeyer@google.com
Zeyu Chen[3]
chen-zy22@mails.tsinghua.edu.cn
Nassir Navab[1, 4]
nassir.navab@tum.de
Federico Tombari[1, 2]
tombari@in.tum.de

[1] Technical University of Munich
Munich, Germany
[2] Google
Zurich, Switzerland
[3] Tsinghua University
Beijing, China
[4] Munich Center for Machine Learning
Munich, Germany

arXiv:2411.16898v4 [cs.CV] 20 Nov 2025

Abstract

Accurate meshing from images remains a key challenge in 3D vision. While state-of-the-art 3D Gaussian Splatting (3DGS) methods excel at synthesizing photorealistic novel views through rasterization-based rendering, their reliance on sparse, explicit primitives severely limits their ability to recover watertight and topologically consistent 3D surfaces. We introduce MonoGSDF, a novel method that couples Gaussian-based primitives with a neural Signed Distance Field (SDF) for high-quality reconstruction with monocular RGB images as input. During training, the SDF guides Gaussians' spatial distribution, while at inference, Gaussians serve as priors to reconstruct surfaces, eliminating the need for memory-intensive Marching Cubes. To handle arbitrary-scale scenes, we propose a scaling strategy for robust generalization. A multi-resolution training scheme further refines details and monocular geometric cues from off-the-shelf estimators enhance reconstruction quality. Experiments on real-world datasets show MonoGSDF outperforms prior methods while maintaining efficiency.

1 Introduction

3D Gaussian Splatting (3DGS) [15] has emerged as a state-of-the-art method for high-quality novel view synthesis by leveraging the rasterization pipeline and representing 3D scenes using points characterized by Gaussian functions. However, 3DGS remains an explicit and discrete representation, posing challenges for accurate 3D surface reconstruction and meshing. This limitation is particularly important for applications such as geometry editing [6, 35, 42], 3D animation [29, 32, 48], and robotics [14, 49, 55]. The main challenge is that 3DGS models geometry as a discrete, unstructured point set. Because 3D Gaussians are optimized for

© 2025. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

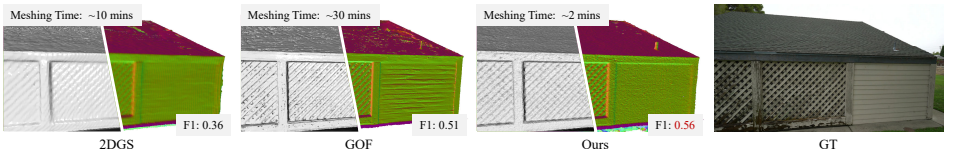


Figure 1: **MonoGSDF.** We show the reconstructed mesh. Compared to 2DGS [11] and GOF [47], ours achieves higher F1 scores (↑) and reconstructs smooth surfaces with fine details.

rendering quality instead of geometric consistency, post-processing methods such as Poisson surface reconstruction [13] struggle to recover accurate 3D surfaces.

These challenges have motivated recent investigations exploring how 3DGS can be used for high-quality surface reconstruction while maintaining its rendering speed and efficiency. Methods like 2DGS [11] address this by aligning 2D Gaussians to surfaces, although their depth map-based mesh extraction [6] struggles with thin structures and unbounded scenes due to resolution and resource limitations. Methods such as SuGaR [10] attempt to address these limitations by aligning 3D Gaussians with mesh faces. Nevertheless their reliance on Poisson reconstruction [13] still fundamentally limits reconstruction and meshing quality due to the inherent geometric inaccuracies of Gaussian representations. Although recent methods [23, 44, 54] have integrated SDF fields into Gaussian training for surface reconstruction, their incomplete exploration of the Gaussian-SDF relationship results in over-smoothed surfaces and limits their application within object-level reconstruction.

To bridge the gap between explicit Gaussian representations and implicit surface reconstruction while fully leveraging their complementary advantages, we present MonoGSDF, a unified framework that establishes a robust connection between 3D Gaussians and SDF-based surface modeling for high-fidelity reconstruction with only monocular RGB images as input. Our **contributions** are: 1. a Gaussian-based implicit surface reconstruction pipeline for monocular images, where the SDF guides Gaussians to better align near the surface during training and Gaussians act as priors for fast mesh extraction; 2. a simple yet efficient SDF-to-Opacity function that bridges implicit SDF with explicit Gaussians, coupled with a normalization strategy that maps unbounded 3D coordinates to a (0,1) range with minimal distortion; 3. a multi-resolution strategy based on wavelet transforms progressively refines details, while off-the-shelf geometric cues further enhance reconstruction quality. Experiments on real-world datasets demonstrate that our method delivers high-quality reconstructions and outperforms baselines by 13% in terms of Chamfer distance while preserving the view synthesis quality.

2 Related Works

2.1 View Synthesis and Gaussian Splatting

Neural Radiance Field (NeRF) [24] utilize multi-layer perceptions (MLP) as scene representation to predict geometry and view-dependent appearance. The MLP is optimized via a photometric loss through volume rendering. Subsequent methods have focused on optimizing NeRF's training and expressiveness using grid representations [27], improving rendering speed [31, 52] and scaling to unbounded scenes [2]. However, volume rendering typically requires substantial computational resources and extensive training durations. 3D Gaussian Splatting (3DGS) [15] has emerged as an efficient approach for real-time view synthesis through differentiable Gaussian functions. Subsequent works have enhanced rendering qual-

ity via anti-aliasing techniques [30, 39, 46] and improved rendering speed through Gaussian density control [41] and radiance field priors [21, 28]. Geometry-focused approaches like DNGaussian [18] address sparse view degradation, while GeoGaussian [19] preserves non-textured regions. Instantsplat [8] accelerates sparse view training using Dust3r [37] initialization, and Scaffold-GS [22] combines implicit-explicit representations. However, these methods primarily focus on appearance quality rather than underlying geometry, limiting their application to view synthesis.

2.2 Surface Reconstruction from Gaussians

Due to the discrete and unstructured nature of 3DGS, along with supervision being limited to RGB images during training, existing methods often struggle to accurately capture scene geometry, making surface reconstruction particularly challenging. SuGaR [10] addresses this by constructing a density field from Gaussians and extracting meshes via level-set searching. However, it is computationally expensive and struggles to reconstruct large, smooth surfaces like floors and walls. Other approaches [6, 9, 33, 38, 50] incorporate depth or normal estimators as priors to supervise Gaussians. 2DGS [11] and GSurfels [7] improves geometric alignment by flattening 3D Gaussians into 2D disks or surfels, enabling better surface representation. For surface reconstruction, they rely on either multi-view depth maps for TSDF fusion or Poisson reconstruction. However, TSDF fusion is constrained by fixed resolution and high memory usage for large scenes. To handle unbounded scenes, it requires contracting coordinates, introducing unnecessary distortions. Meanwhile, Poisson reconstruction often produces noisy surfaces due to inaccuracies in Gaussian geometry distribution. Instead of refining depth and normal maps, some methods [1, 23, 44, 54] integrate Signed Distance Field (SDF). 3DGSR [23] and GSDF [44] associate Gaussians with SDF via an additional branch that volume-renders depth and normal supervised by rendered Gaussian depth and normal. However, this approach is inefficient and fails to fully exploit the relationship between Gaussians and SDF. GS-Pull [54] and GSurf [1] improve upon this by leveraging SDF gradients to better align Gaussians with surfaces, but they are either restricted to object-level reconstruction or produce overly smooth results. Gaussian Opacity Fields (GOF) [47] approximates minimum accumulated alpha values from all views via ray tracing to construct an opacity field, followed by surface extraction using Marching Tetrahedra [17]. However, Gaussian distributions remain inconsistent and unstructured relative to scene surfaces. Our method leverages SDF to guide Gaussians to align closer to the surface during training. At inference, Gaussians serve as primitives to efficiently constrain zero-level set extraction, avoiding redundant free-space searches while achieving accurate reconstruction.

3 Method

3.1 Preliminaries

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [15] employs a set of 3D points to effectively render images from given viewpoints, each characterized by a Gaussian function with 3D mean $\mu_i \in \mathbb{R}^3$, covariance matrix $\Sigma_i \in \mathbb{R}^{3\times3}$, opacity value $\alpha_i \in \mathbb{R}$, RGB color values $\mathbf{c}_i \in \mathbb{R}^3$:

$$o_i(\mathbf{x}) = \alpha_i \times \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)\right), \quad \Sigma_i = R_i S_i S_i^T R_i^T. \tag{1}$$

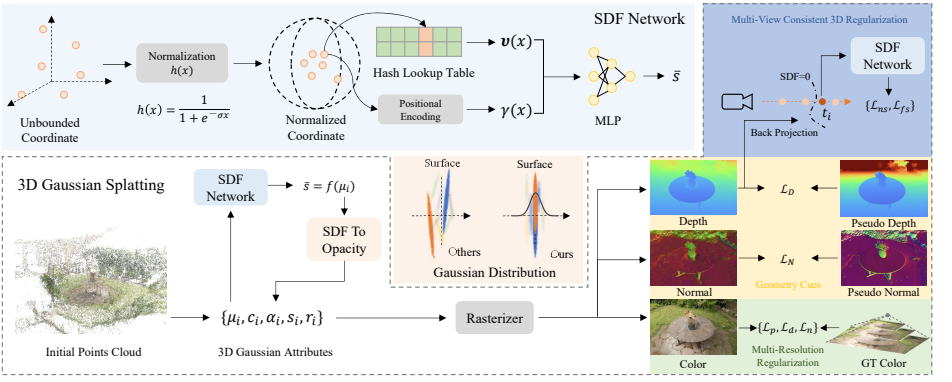


Figure 2: **Overview.** MonoGSDF synergizes SDF and Gaussian representations by converting queried SDF values into opacity, encouraging Gaussians to align with surfaces. It uses standard rasterization to render color, depth, and normals, enhanced by geometry cues and a multi-resolution strategy. SDF is jointly trained with 3D supervision from rendered depth.

Given a 3D position $\mathbf{x}$, $o_i(\mathbf{x})$ represents current opacity value contributed by the i-th Gaussian. To facilitate optimization, Σ_i is factorized into the product of a scaling matrix S_i, represented by scale factors $\mathbf{s}_i \in \mathbb{R}^3$, and a rotation matrix R_i encoded by a quaternion $\mathbf{r}_i \in \mathbb{R}^4$. 3D Gaussians are then projected onto a 2D image plane according to elliptical weighted average (EWA) [56] to render images for given views. Color $\bar{\mathbf{C}}(\mathbf{u})$, depth $\bar{D}(\mathbf{u})$, and normal $\bar{\mathbf{N}}(\mathbf{u})$ at pixel $\mathbf{u}$ is rendered by N projected and ordered Gaussians using point-based α-blending:

$$\{\bar{\mathbf{C}}, \bar{D}, \bar{\mathbf{N}}\}(\mathbf{u}) = \sum_{i \in N} T_i o_i \{\mathbf{c}_i, d_i, \mathbf{n}_i\}, \tag{2}$$

where $T_i = \prod_{j=1}^{i-1}(1 - o_j)$, depth d_i is the distance between camera center and the ray-Gaussian intersection plane, and Gaussian's normal $\mathbf{n}_i$ is approximated as the normal of the ray-Gaussian intersection plane.

Neural Signed Distance Field. While 3DGS excels in novel view synthesis, its discrete nature limits surface reconstruction. In contrast, SDF provide a continuous, watertight surface as the zero-level set. We use a hash-based neural representation and jointly train the SDF with 3DGS, enabling more precise and efficient surface extraction. To better encode the scene geometry, we choose One-blob $\gamma(\mathbf{x})$ [25] as positional embedding and a multi-resolution hash-based feature grid $\mathcal{V} = \{\mathcal{V}^l\}_{l=1}^{L}$ [27]. Feature $\mathcal{V}(h(\mathbf{x}))$ at any 3D point $\mathbf{x}$ are queried via trilinear interpolation. The geometry decoder f is an MLP which maps the 3D coordinate to an SDF value $\bar{s}$:

$$f(\mathbf{x}) = f(\gamma(h(\mathbf{x})), \mathcal{V}(h(\mathbf{x}))) \mapsto \bar{s}. \tag{3}$$

where h is a coordinate normalization. Therefore, for each 3D Gaussian located at μ_i, its corresponding SDF value can be written as $\bar{s}_i = f(\mu_i)$.

3.2 Gaussian-Guided Signed Distance Field

Connect Gaussians with SDF. As illustrated in Figure 2, our method begins by querying SDF values with each Gaussian 3D mean μ_i using Eq. 3, where zero values correspond to

surface points. While 3DGSR [23] introduced a bell-shaped function to map SDF value to Gaussian opacity, this approach assigns an opacity value $\alpha_i = 0.25$ to surface points ($\bar{s} = 0$), which contradicts the physical reality that surface points should exhibit maximum opacity ($\alpha_i = 1$). To address this limitation, we propose a Gaussian-shaped transformation that maintains both mathematical simplicity and computational efficiency while ensuring physically plausible opacity values $g(\bar{s})$:

$$g(\bar{s}_i) = \exp(-(\beta \times \bar{s}_i)^2) \mapsto \alpha_i, \tag{4}$$

where β is a hyperparameter. While previous 3DGS methods [11, 15, 22] optimize opacities through alpha blending without considering spatial distribution constraints, these approaches often result in limited geometric accuracy. Our method enforces a better distribution of Gaussians along the surface, leading to high geometric fidelity.

Gaussian-Guided Normalization Our rendering pipeline follows the standard Gaussian rasterization process, generating color $\bar{\mathbf{C}}$, depth $\bar{D}$ and $\bar{\mathbf{N}}$ maps from Gaussian attributes, with the depth map subsequently used for SDF training. While 3DGSR [23] and GSDF [44] have adopted similar approaches, their grid-based representations necessitate predefined scene bounding boxes for volume initialization, restricting their applicability to object-level or small-scale scene reconstruction. Although MipNeRF [2] introduced a fixed coordinate contraction method for unbounded scenes, its mapping of infinite space to [-2, 2] conflicts with our hash-based feature field's requirement for (0, 1) input. Moreover, MipNeRF's linear transformation within a limited region of interest introduces substantial distortion in peripheral areas. To address these limitations and to enable scalable reconstruction of real-world unbounded scenes, our method introduces a Gaussian-guided normalization strategy. We first establish a bounding box based on the Gaussians, then apply a novel sigmoid-like normalization function that maps 3D point $\mathbf{x} \in \mathbb{R}^3$ to a fixed $(0,1)$ range:

$$h(\mathbf{x}) = 1/(1+\exp(-\sigma \mathbf{x})), \tag{5}$$

where $\sigma = 2/B$, $B \in \mathbb{R}^3$ is the bounding box size of initial Gaussians. This approach ensures the volume containing the primary Gaussians occupies the majority of the grid space. During Gaussian densification, while new Gaussians may extend beyond the initial bounding box, they remain within the normalized scope. Our sigmoid-like normalization maintains near-linear transformation for Gaussians within the primary bounding box while effectively contracting distant Gaussians, achieving minimal distortion.

3.3 Optimization

Geometry Cues. Precisely reconstructing real-world scenes from only monocular images remains a challenging problem. To further improve the reconsruction quality, we incorporate dense pseudo disparities $\hat{Z}$ from a depth estimator[1]. However, these suffer from scale ambiguity and view inconsistency. While DN Splatter [34] aligns depth via $\bar{D} = s\hat{D} + t$, it overlooks disparity shifts, causing depth distortion. We address this by refining pseudo depth as $\hat{D} = a/(s\hat{Z} + t) + b$ using rendered depth, then derive pseudo normals $\hat{\mathbf{N}} = \nabla\hat{D}/|\nabla\hat{D}|$ for stable Gaussian supervision.

$$\mathcal{L}_D = \sum |(a/(s \times \hat{Z} + t) + b - \bar{D}|, \quad \mathcal{L}_N = \sum (1 - \hat{\mathbf{N}} \cdot \bar{\mathbf{N}}). \tag{6}$$

[1] We use DepthAnything V2 [40].

where s,t and a,b represent learnable scale and shift parameters, and $\bar{\mathbf{N}}$ is rendered Gaussian normal. Therefore, the overall geometry regularization is $\mathcal{L}_{geo} = \lambda_D \mathcal{L}_D + \lambda_N \mathcal{L}_N$.

While the geometry cues offer valuable 2D supervision for improving depth and normal, they fail to constrain the 3D Gaussian distribution effectively. We address this limitation through the SDF, which enforces tied surface alignment of Gaussians.

Multi-View Consistent 3D Regularization. To train this hybrid model, enforcing all Gaussians to satisfy $\bar{s}_i = 0$ (leading to $\alpha_i = 1$) degrades rendering quality, as high-quality renderings require opacity variation, and sparse SDF supervision at Gaussian positions is insufficient. Dense supervision with monocular cues like MonoSDF [45] suffers from scale ambiguity and lacks multi-view consistency, while methods like 3DGSR [23] and GSDF [44] only offer 2D supervision through rendered Gaussian depth maps. Our approach leverages monocular cues and the 3D Gaussians' advantages, where the Gaussian-rendered depth $\bar{D}$ ensures multi-view consistency and provides reliable 3D SDF supervision via back-projection.

More specifically, we first sample M pixels $\{\mathbf{u} = \mathbf{u}_i \mid \mathbf{u}_i \in \mathbb{R}^2, \text{for } i = 1,2,...,M\}$ from the rendered Gaussian depth map $\bar{D}$. For each ray casts from a pixel, we sample $K = K_n + K_f$ points $\{t_k\}$ from the camera center to the surface, where K_n means near surface samples and K_f means free space samples. SDF values of each point can be queried as $\{\bar{s}_k\}$. For near surface samples within the truncation region tr, i.e. $S_{tr} = \{|\bar{D}(\mathbf{u}) - t_k| \leq tr\}$, we use the distance between the sampled point t_k and its surface $\bar{D}(\mathbf{u}_j)$ as an approximation of ground truth SDF value for supervision, and for points that are far from the surface $S_{fs} = \{\bar{D}(\mathbf{u}) - t_k > tr\}$, we apply a free-space supervision:

$$\mathcal{L}_{ns} = \sum_{\mathbf{u}_j \in \mathbf{u}} \sum_{t_k \in S_{tr}} ||\bar{s}_k - (\bar{D}(\mathbf{u}_j) - t_k)||_2, \quad \mathcal{L}_{fs} = \sum_{\mathbf{u}_j \in \mathbf{u}} \sum_{t_k \in S_{fs}} ||\bar{s}_k - 1||_2. \tag{7}$$

Therefore, the overall SDF regularization is $\mathcal{L}_{sdf} = \lambda_{ns} \mathcal{L}_{ns} + \lambda_{fs} \mathcal{L}_{fs}$.

Multi-Resolution Regularization. Unlike vanilla 3DGS [15] with L1/SSIM losses and opacity-based pruning, our method links opacity to spatial position, eliminating *Opacity Resets* but risking Gaussian redundancy. Instead of frequency-based strategies [51] that may blur structure, we introduce wavelet-based multi-resolution supervision: $\mathbf{C}_l = \mathcal{W}(\mathbf{C}, l)$, progressively refining detail while preserving structure. $\mathbf{C}$ is ground truth color images, $\mathcal{W}$ represents wavelet transform and l denotes the level of transformation. This stabilizes training and enables coarse-to-fine Gaussian distribution, preventing redundant Gaussians. The photometric loss is $\mathcal{L}_p = 0.8|\mathbf{C}_l - \bar{\mathbf{C}}| + 0.2 \cdot SSIM(\mathbf{C}_l, \bar{\mathbf{C}})$.

Objective Function. During training, we jointly optimize explicit Gaussians and implicit SDF. We apply distortion $\mathcal{L}_d$ and depth-normal $\mathcal{L}_n$ regularization as defined in GOF [47]. The overall objective function is defined as: $\mathcal{L} = \mathcal{L}_p + \mathcal{L}_{geo} + \mathcal{L}_{sdf} + \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n$.

Pruning. In our framework, opacity is tied to spatial position via SDF. We adopt a geometry-aware pruning strategy that removes Gaussians with $\bar{s}_i > tr$, replacing conventional opacity-based pruning with a surface-aware, physically grounded criterion.

	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
NeRF [24]	1.90	1.60	1.85	0.58	2.28	1.27	1.47	1.67	2.05	1.07	0.88	2.53	1.06	1.15	0.96	1.49
VolSDF [43]	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86
NeuS [36]	1.00	1.37	0.93	0.43	1.10	0.65	0.57	1.48	1.09	0.83	0.52	1.20	0.35	0.49	0.54	0.84
N-angelo [20]	0.49	1.05	0.95	0.38	1.22	1.10	2.16	1.68	1.78	0.93	0.44	1.46	0.41	1.13	0.97	1.07
SuGaR [10]	1.47	1.33	1.13	0.61	2.25	1.71	1.15	1.63	1.62	1.07	0.79	2.45	0.98	0.88	0.79	1.33
2DGS [11]	0.48	0.91	0.39	0.39	1.01	0.83	0.81	1.36	1.27	0.76	0.70	1.40	0.40	0.76	0.52	0.80
GSurfel [7]	0.66	0.93	0.52	0.41	1.06	1.14	0.85	1.29	1.53	0.79	0.82	1.58	0.45	0.66	0.53	0.88
3DGSR [23]	0.68	0.84	0.70	0.39	1.16	0.87	0.77	1.48	1.26	0.87	0.69	0.80	0.42	0.64	0.60	0.81
GS-Pull [54]	0.51	0.56	0.46	0.39	0.82	0.67	0.85	1.37	1.25	0.73	0.54	1.39	0.35	0.88	0.42	0.75
GOF [47]	0.50	0.82	0.37	0.37	1.12	0.74	0.73	1.18	1.29	0.68	0.77	0.90	0.42	0.66	0.49	0.74
Ours	0.45	0.65	0.36	0.36	0.94	0.70	0.67	1.27	0.99	0.63	0.49	0.84	0.39	0.53	0.47	0.65

Table 1: **Quantitative Comparison on the DTU** [12]. We show the Chamfer distance. For N-angelo [20], we report the results from UniSDF [34] reproduction and we show the vanilla results in supp. mat..

	Implicit			Explicit			
	NeuS [36]	GeoNeus [9]	N-angelo [20]	SuGaR [10]	2DGS [11]	GOF [47]	Ours
Barn	0.29	0.33	0.70	0.14	0.36	0.51	0.56
Caterpillar	0.29	0.26	0.36	0.16	0.23	0.41	0.38
Courthouse	0.17	0.12	0.28	0.08	0.13	0.28	0.29
Ignatius	0.83	0.72	0.89	0.33	0.44	0.68	0.72
Meetingroom	0.24	0.20	0.32	0.15	0.16	0.28	0.25
Truck	0.45	0.45	0.48	0.26	0.26	0.59	0.62
Mean	0.38	0.35	0.50	0.19	0.30	0.46	0.47

Table 2: **Quantitative Results on the Tanks and Temples** [16].We show the F1-score.

4 Experiments

4.1 Experimental Settings

Datasets. We comprehensively evaluate our method on three public datasets: *DTU* dataset [12] which consists of indoor object scans; *Tanks and Temples* [16] which features six unbounded outdoor scenes; and *Mip-NeRF* 360 [2] for measuring novel view synthesis quality.

Baselines. We compare our proposal against several state-of-the-art methods. Among implicit methods, we compare against NeRF [24], VolSDF [43], NeuS [36], N-angelo [20], GeoNeus [9], Instant NGP [26] and MipNeRF 360 [2]. As for explicit Gaussian methods, we compare against 3DGS [15], SuGaR [10], Mip-Splatting [46], 2DGS [11], GSurfel [7], 3DGSR [23], GS-Pull [54] and GOF [47].

Metrics. We follow common practice and report surface accuracy as Chamfer Distance and F1-score on DTU and Tanks and Temples, respectively. We measure the visual fidelity of the synthesized novel views with PSNR, SSIM and LPIPS [53] on Mip-NeRF 360.

Implementation. We perform single GPU training (NVIDIA 3090) and use by default $\lambda_d = 100, \lambda_n = 0.05, \lambda_{ns} = 1000, \lambda_{fs} = 10, \lambda_{depth} = 0.05, \lambda_{normal} = 0.1$ and $\beta = 100$. We train 30000 iterations like other methods and do not require extra training for SDF. For more implementation details, please refer to supp. mat..

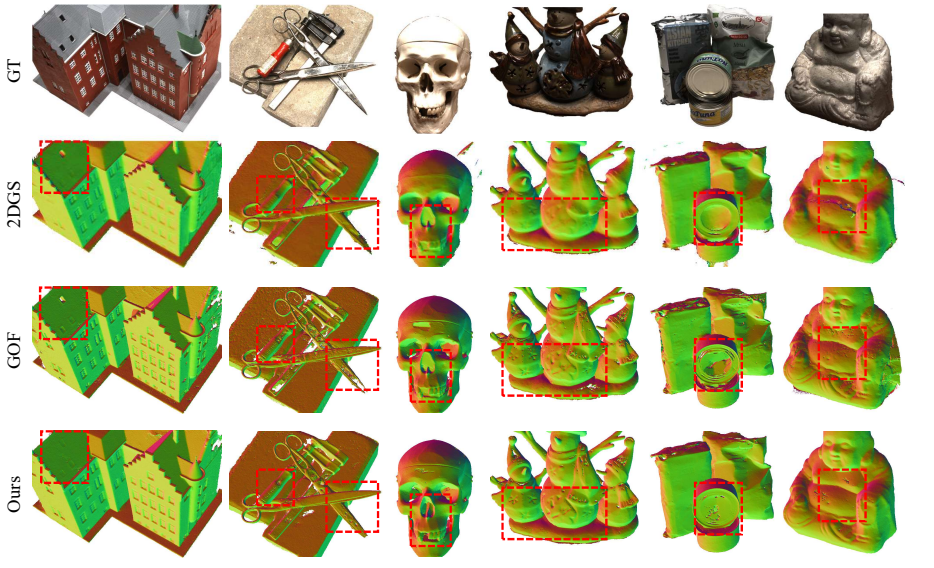


Figure 3: **Surface Reconstruction on the DTU [12].** We show normal maps from the reconstructed meshes.

Metric	NeRF [28]	InstantNGP [29]	MipNeRF360 [2]	3DGS [15]	SuGaR [11]	Mip-Splat [40]	2DGS [13]	GS-Pull [36]	GOF [47]	Ours
PSNR ↑	24.15	26.03	28.10	27.53	26.18	27.78	27.16	27.27	27.81	27.81
SSIM ↑	0.624	0.723	0.804	0.826	0.768	0.825	0.811	0.814	0.837	0.836
LPIPS ↓	0.443	0.294	0.232	0.212	0.291	0.220	0.244	0.230	0.193	0.189

Table 3: **Quantitative Results on Mip-NeRF 360 [2].** We evaluate the quality of rendered images against ground-truth images using PSNR, SSIM, and LPIPS.

4.2 Geometry Evaluation

We evaluate our method on the DTU dataset [12], achieving the lowest Chamfer Distance and outperforming all 3DGS-based and implicit methods in quality (Table 1). For N-angelo [20], we report results from [34] due to unverified original values. As shown in Figure 3, our method yields smoother, more complete surfaces—especially in sparse and flat regions—compared to noisy outputs from 2DGS [13] and GOF [47]. On the Tanks and Temples dataset [16], as shown in Table 2, our method matches the performance of leading implicit approaches [20] while cutting training time from 12+ to around 3 hours. We also slightly outperform GOF [47], with faster mesh extraction and similar training speed (Table 4). Figure 4 shows our superior handling of flat and transparent surfaces. Additional results on Mip-NeRF 360 [2], DTU [12], and Tanks and Temples [16] are in the supp. mat..

4.3 Novel View Synthesis

We further compare our method with state-of-the-art novel view synthesis (NVS) techniques on the Mip-NeRF 360 dataset [2]. The quantitative results presented in Table 3 demonstrate that our approach achieves remarkable rendering performance. Our method effectively aligns the Gaussians with the surface, leading to better surface reconstruction without compromising rendering quality.

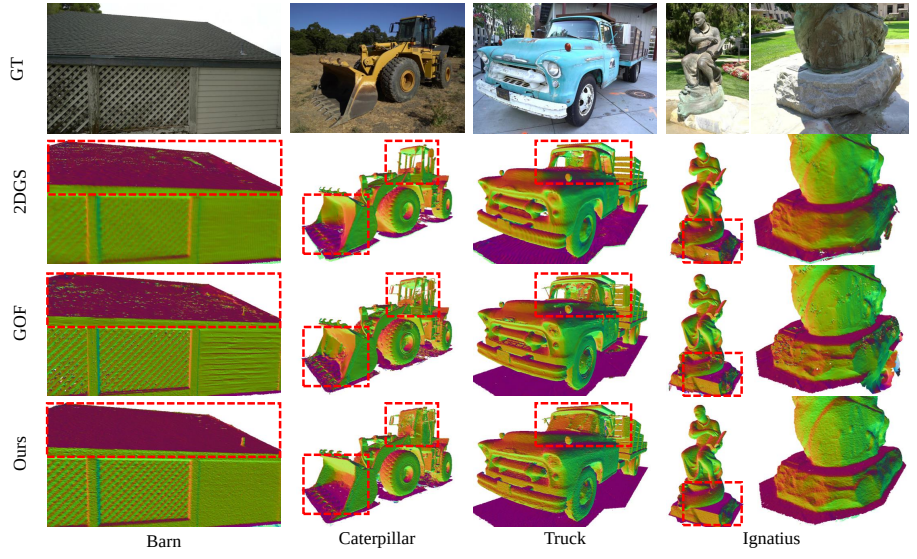


Figure 4: **Surface Reconstruction on the Tanks and Temples [16].** We show rendered normal maps from reconstructed meshes.

Run Time	2DGS	Ours (2D)	GOF	Ours
Training	17 min	39 min	2.8 h	3.1 h
Meshing	∼20 min	5 min	∼30 min	5 min

Table 4: **Runtime.** We report the training and meshing time. Compared to GOF [47], we avoid ray tracing for SDF evaluation, enabling fast meshing.

4.4 Runtime and Ablation Study

Runtime Analysis. Table 4 reports the runtime analysis on the Barn scene. Our method integrates seamlessly with different Gaussian rasterizers. "Ours (2D)" refers our method to using the 2DGS rasterizer. While SDF guidance enhances surface reconstruction, our approach achieves competitive training times and notably faster mesh extraction than baselines such as 2DGS and GOF, since our SDF network directly outputs values for any query point, whereas GOF relies on ray tracing and 2DGS requires TSDF fusion.

Ablations on Geometry Regularization. Table 5 presents ablation studies on the impact of different geometry regularization terms, showing similar performance gaps for pseudo normal and depth maps. Evaluating GOF with geometry cues only ("GOF+Geo") shows modest improvements (0.52). To highlight that geometry cues alone aren't the main factor, we increase λ_D and λ_N to 0.5, leading to performance degradation due to their adverse effects. Our complete pipeline achieves an 8% improvement, demonstrating the synergistic effect of our approach. Figure 5 illustrates the effectiveness of our geometry regularization, especially in improving surface consistency and reconstruction quality in transparent and flat regions.

Ablations on Normalization Function. MipNeRF 360's non-linear normalization [2] introduces distortions and conflicts with tinycudann's $(0, 1)$ input requirement. In contrast, our

Method	2DGS	Ours (2D)	$\lambda_D = 0$	$\lambda_N = 0$	$\lambda_D = 0, \lambda_N = 0$	w/ MipNeRF NF
F1 Score	0.36	0.40	0.52	0.51	0.50	0.52
Method	GOF	GOF+Geo	$\lambda_D = 0.5$	$\lambda_N = 0.5$	**Ours**	w/ 3DGSR SDF2O
F1 Score	0.51	0.52	0.45	0.47	**0.56**	0.24

Table 5: **Quantitative Ablation Study.** We report Barn scene F1-scores with and without different regularization.

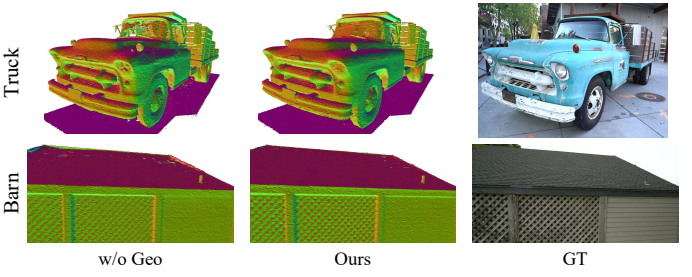


Figure 5: **Ablation Study on Geometry Regularization.** We show the reconstructed meshes with and without the geometry regularization.

near-linear normalization within the bounding box compresses outliers with minimal distortion. For fairness, we scale MipNeRF's normalization output to $(0, 1)$ and compare with ours in Table 5, where our method ("Ours") achieves superior performance.

Ablations on SDF to Opacity Function. We compare 3DGSR's SDF-to-Opacity function [23] ("w/ 3DGSR SDF2O") with our method in Table 5. Our approach achieves a higher F1-score, providing a simpler, more efficient solution with full opacity, while 3DGSR's opacity is capped at 0.25, limiting its effectiveness in unbounded outdoor scenes.

Ablations on SDF Fields. Table 5 shows that our Gaussian-based implicit surface reconstruction outperforms GOF, and when integrated with the 2DGS rasterizer ("Ours (2D)"), it consistently achieves higher F1-scores across settings.

5 Conclusion

We propose MonoGSDF, a Gaussian-based implicit surface reconstruction framework that bridges implicit and explicit representations via an SDF-to-opacity mapping. Gaussian-guided normalization stabilizes optimization in unbounded scenes, and multi-resolution training with geometric cues enables progressive and accurate reconstruction. Our method achieves SOTA performance, especially on flat and transparent surfaces where prior methods fail.

References

[1] Xu Baixin, Hu Jiangbei, Li Jiaze, and He Ying. Gsurf: 3d reconstruction via signed distance fields with direct gaussian supervision. *arXiv preprint arXiv:2411.15723*, 2024.

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

[3] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *arXiv preprint arXiv:2406.06521*, 2024.

[4] Hanlin Chen, Fangyin Wei, Chen Li, Tianxin Huang, Yunsong Wang, and Gim Hee Lee. Vcr-gaus: View consistent depth-normal regularizer for gaussian surface reconstruction. *arXiv preprint arXiv:2406.05774*, 2024.

[5] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21476–21485, 2024.

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

[7] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[8] 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 preprint arXiv:2403.20309*, 2, 2024.

[9] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 35:3403–3416, 2022.

[10] Antoine Guédon and Vincent Lepetit. 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*, pages 5354–5363, 2024.

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

[12] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

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

[14] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024.

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

[16] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[17] Jonas Kulhanek and Torsten Sattler. Tetra-nerf: Representing neural radiance fields using tetrahedra. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18458–18469, 2023.

[18] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20775–20785, 2024.

[19] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. Geogaussian: Geometry-aware gaussian splatting for scene rendering. In *European Conference on Computer Vision*, pages 441–457. Springer, 2025.

[20] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023.

[21] Tianqi Liu, Guangcong Wang, Shoukang Hu, Liao Shen, Xinyi Ye, Yuhang Zang, Zhiguo Cao, Wei Li, and Ziwei Liu. Mvsgaussian: Fast generalizable gaussian splatting reconstruction from multi-view stereo. In *European Conference on Computer Vision*, pages 37–53. Springer, 2025.

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

[23] Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting. *arXiv preprint arXiv:2404.00409*, 2024.

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

[25] Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (ToG)*, 38(5):1–19, 2019.

[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4): 102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

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

[28] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. Radsplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps. *arXiv preprint arXiv:2403.13806*, 2024.

[29] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5020–5030, 2024.

[30] Xiaowei Song, Jv Zheng, Shiran Yuan, Huan-ang Gao, Jingwei Zhao, Xiang He, Weihao Gu, and Hao Zhao. Sa-gs: Scale-adaptive gaussian splatting for training-free anti-aliasing. *arXiv preprint arXiv:2403.19615*, 2024.

[31] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5459–5469, 2022.

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

[33] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv preprint arXiv:2403.17822*, 2024.

[34] Fangjinhua Wang, Marie-Julie Rakotosaona, Michael Niemeyer, Richard Szeliski, Marc Pollefeys, and Federico Tombari. Unisdf: Unifying neural representations for high-fidelity 3d reconstruction of complex scenes with reflections. *arXiv preprint arXiv:2312.13285*, 2023.

[35] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20902–20911, 2024.

[36] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[37] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[38] Yaniv Wolf, Amit Bracha, and Ron Kimmel. Surface reconstruction from gaussian splatting via novel stereo views. *arXiv preprint arXiv:2404.01810*, 2024.

[39] Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-scale 3d gaussian splatting for anti-aliased rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20923–20931, 2024.

[40] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024.

[41] Runyi Yang, Zhenxin Zhu, Zhou Jiang, Baijun Ye, Xiaoxue Chen, Yifei Zhang, Yuantao Chen, Jian Zhao, and Hao Zhao. Spectrally pruned gaussian fields with neural compensation. *arXiv preprint arXiv:2405.00676*, 2024.

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

[43] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.

[44] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. Gsdf: 3dgs meets sdf for improved rendering and reconstruction. *arXiv preprint arXiv:2403.16964*, 2024.

[45] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022.

[46] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024.

[47] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024.

[48] Ye Yuan, Xueting Li, Yangyi Huang, Shalini De Mello, Koki Nagano, Jan Kautz, and Umar Iqbal. Gavatar: Animatable 3d gaussian avatars with implicit mesh learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 896–905, 2024.

[49] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023.

[50] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. Rade-gs: Rasterizing depth in gaussian splatting. *arXiv preprint arXiv:2406.01467*, 2024.

[51] Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric Xing. Fregs: 3d gaussian splatting with progressive frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21424–21433, 2024.

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

[53] 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.

[54] Wenyuan Zhang, Yu-Shen Liu, and Zhizhong Han. Neural signed distance function inference through splatting 3d gaussians pulled on zero-level set. *Advances in Neural Information Processing Systems*, 37:101856–101879, 2025.

[55] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21634–21643, 2024.

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

A Implementation Details

A.1 Hyperparameters

In the following, we report implementation details and hyperparameters used for our method.
Gaussian Splatting. For hyperparameters used in the Gaussian rasterization, we follow previsous works [11, 15, 47]. We train our Gaussian model and SDF network jointly with 30000 iterations. We set the initial learning rate for Gaussians' position as 0.00016 and the final initial learning rate as 0.0000016. And we set learning rates for scales and rotation as 0.005, 0.001, respectively. We start the Gaussian densification after 500 iterations and until 15000 iterations. We densify Gaussians every 100 iterations and the gradient threshold for densification is 0.0002. We start the distortion loss $\mathcal{L}_d$ and the depth-normal loss $\mathcal{L}_n$ after 3000 iterations and 7000 iterations, respectively. And we set the wavelet level as 3 and gradually increase the resolution until 10000 iteration. After that, we use the full resolution for training.
SDF Network. We use two-layer MLPs and the hidden dimension is 32. We initialize β from Eq. 4 as 100. We start train our SDF from 5000 iterations and until 30000 iterations. The learning rate for our SDF network is 0.002. For each iteration, we sample $M = 10000$ pixels and $K_n = 11, K_f = 64$ points.
Geometry Regularization. We set the learning rate for s,t,a,b in Eq. 6 as 0.01. And for Tanks and Temples, we set $\lambda_D = 0.05, \lambda_N = 0.1$, for DTU and Mip-NeRF 360, we set $\lambda_D = 0.01, \lambda_N = 0.01$.

B Distortion and Depth-Normal Loss

We apply a distortion loss [11] and depth-normal loss [47] as discussed in Section 3.2. The distortion loss concentrates the weight distribution along the rays by minimizing the distance between the ray-splat intersections:

$$\mathcal{L}_d = \sum_{i,j} \omega_i \omega_j |z_i - z_j|, \tag{8}$$

where $\omega_i = o_i(x)\prod_{j=1}^{i-1}(1-o_j(x))$ is the blending weight of the $i-$th intersection and z_i is the depth of the intersection points. The depth-normal loss is defined as:

$$\mathcal{L}_n = \sum (1 - \bar{\mathbf{N}} \cdot \nabla(\bar{D})), \tag{9}$$

where $\nabla(\bar{D})$ is the gradient of rendered Gaussian Depth.

C Mesh Extraction.

Previous Marching Cubes-based methods [23, 44] require dense querying across both occupied and free space, resulting in inefficiencies and limited resolution—especially in unbounded scenes with coordinate distortions. In contrast, we adopt Marching Tetrahedra [17] and leverage spatial cues from 3D Gaussians to restrict the search to regions near the surface, significantly reducing unnecessary computation while improving surface alignment and reconstruction efficiency. Unlike GOF [47], which performs costly ray tracing to query opacity values for each point, our method directly outputs SDF values, accelerating meshing from 30 minutes to just 5 minutes.

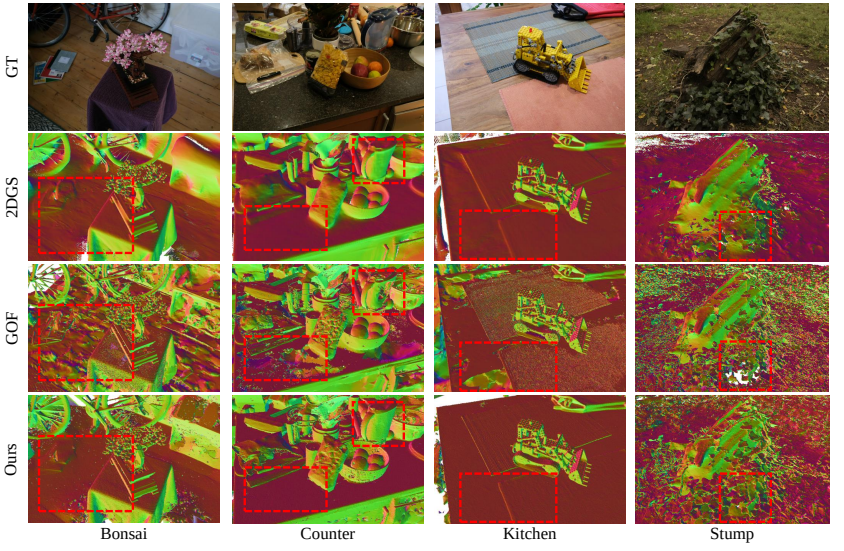


Figure 6: **Surface Reconstruction on the Mip-NeRF 360 Dataset [2].** We show the rendered normal maps from reconstructed meshes.

D Additional Experimental Results

D.1 Results on DTU and Mip-NeRF 360

In Table 6, we present the results reported in the Neuralangelo publication [20] on the DTU dataset. It is important to note that the quantitative results from the Neuralangelo paper have been flagged by other studies and discussions on the official GitHub repository as non-reproducible[2].

To provide a more comprehensive evaluation, we include additional qualitative comparisons on the DTU dataset [12] in Figure 14 and further demonstrate results on the Mip-NeRF 360 dataset [2] in Figure 6 and 7. Our method excels in handling reflective surfaces and produces reconstructions that are not only smoother but also exhibit finer details, highlighting its capability for more accurate and visually appealing scene representations.

E More Ablation

E.1 Ablation on Different Rasterizers

In Figure 8, we present reconstruction results using different Gaussian rasterizers, showcasing the versatility and adaptability of our method. These results demonstrate that our approach can be seamlessly integrated into existing Gaussian methods, no matter it's 2D Gaussians [11] or 3D Gaussians [47], enhancing their performance without requiring significant modifications. Furthermore, our method consistently delivers high-quality reconstructions with fine details, highlighting its effectiveness in capturing intricate scene structures such as transparent and reflection areas.

[2]Refer to this GitHub issue for details.

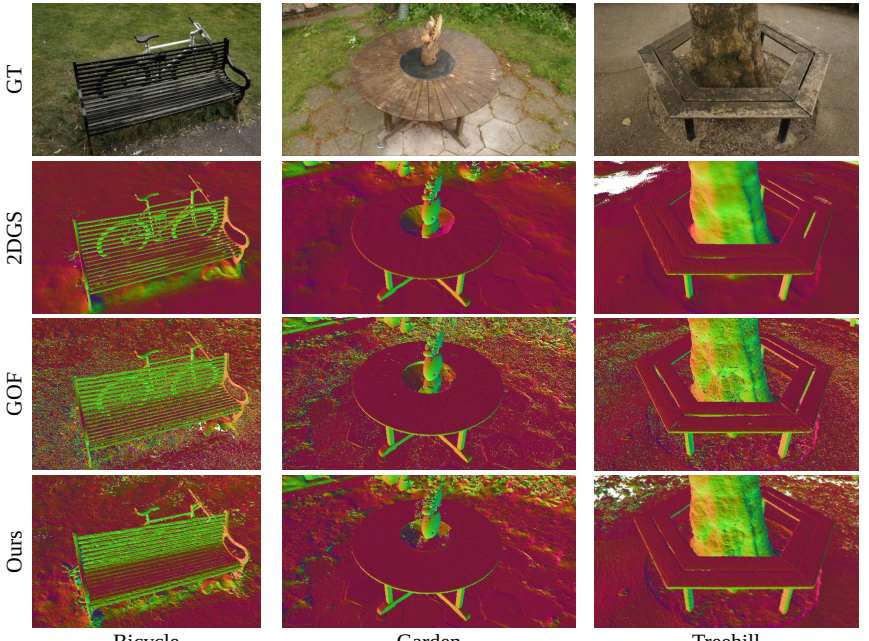


Figure 7: **Additional Results on Mip-NeRF 360 [2].**

E.2 Ablations on Geometry Cues.

Our analysis reveals significant artifacts in Depth Anything v2's estimated depth maps due to inherent scale uncertainty, despite global alignment attempts. As shown in Figure 9, we compare our alignment method "Ours" with DN Splatter's approach [33] "DN Align", demonstrating our method's superior ability to reduce depth errors between rendered and aligned pseudo depth. Interestingly, despite the inaccuracies in DN Splatter's aligned pseudo depth, the reconstruction F1 score remains comparable to our method. This finding suggests that while these limitations underscore the unreliability of pseudo depth maps for precise reconstruction, necessitating our model's strong ability to compensate for misalignments, the primary improvement stems from our novel Gaussian-SDF collaboration pipeline rather than geometric cues alone.

E.3 Ablations on Pruning Strategy.

Our method links the opacity value of each Gaussian to it's SDF value with our SDF-to-Opacity function. While this maintains computational equivalence to conventional opacity-based pruning, it provides a physically-grounded criterion.

E.4 Ablations on Multi-Resolution Regularization.

Our Multi-Resolution training strategy employs a coarse-to-fine approach, initially utilizing fewer Gaussians at lower resolutions to establish robust scene geometry before progressively

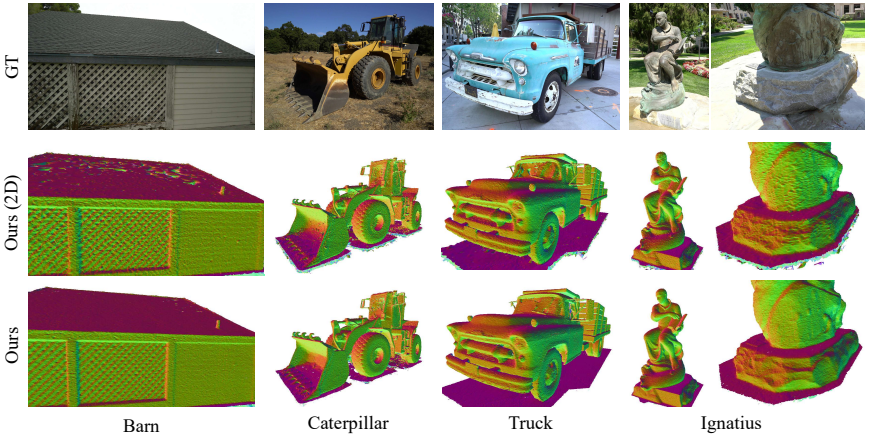


Figure 8: **Mesh Extraction with Different Gaussian Rasterizer.**

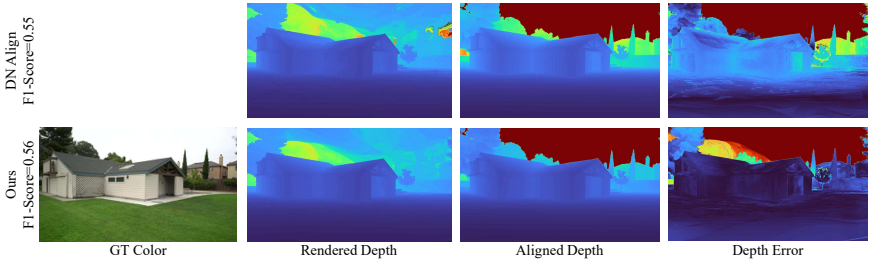


Figure 9: **Ablations on Geometry Cues.** We show the ground truth color image, rendered depth, aligned pseudo depth and depth error (in log) on Barn. DN Align represents our pipeline with DN Splatter's [33] alignment.

refining details. Figure 11 quantitatively illustrates the controlled growth of Gaussian populations throughout training. While the quantitative results remain the same since the evaluation only focus on foreground objects, the qualitative results shows significant improvement. Figure 10 demonstrates the effectiveness of our approach, showing that multi-resolution regularization enables reliable reconstruction of objects with limited views, whereas standard training struggles in such scenarios.

E.5 Ablations on Different Meshing Methods.

Figure 12 illustrates the limitations of depth fusion-based mesh extraction in unbounded scenes, revealing significant distortion caused by resolution constraints and MipNeRF's [2] normalization artifacts. Comparative results in Figure 13 demonstrate our method's advantages across different meshing techniques, showing that our Gaussian-guided surface reconstruction achieves both superior efficiency and enhanced detail preservation compared to existing approaches.

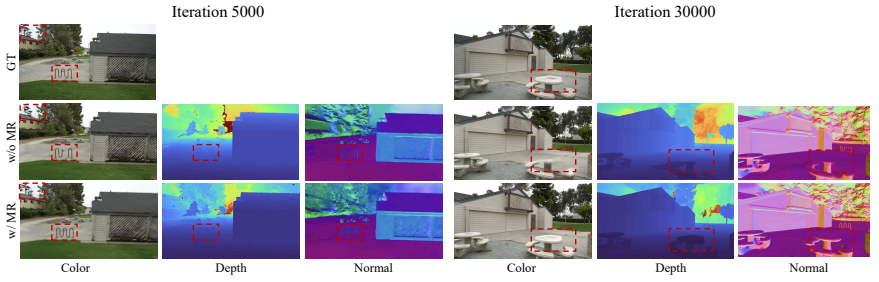


Figure 10: **Ablation Study on Multi-Resolution Regularization.** We show a comparison of ground truth color images alongside our rendered color, depth, and normal outputs at various training iterations on Barn. The results demonstrate that our multi-resolution regularization enables more efficient and complete scene geometry reconstruction, achieving superior convergence compared to standard training approaches.

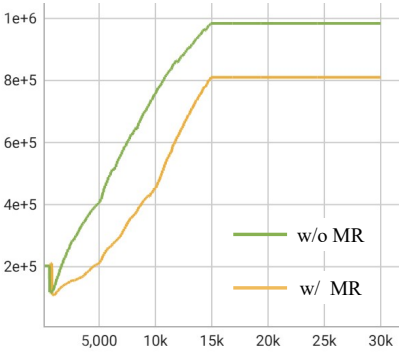
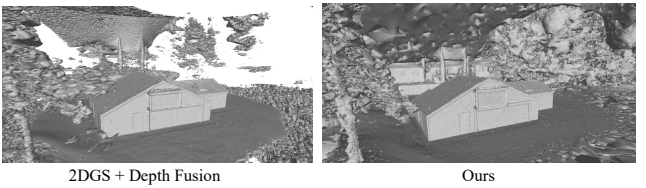


Figure 11: **Total Gaussian Population Growth on Barn (Training Iterations vs. Gaussian Count).** Our proposed multi-resolution regularization achieves comparable reconstruction and rendering quality with significantly fewer Gaussians than standard training approaches, demonstrating improved computational efficiency without compromising output quality.

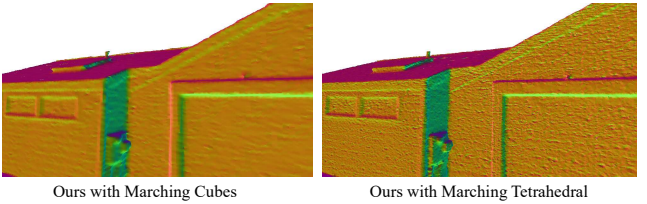
F Limitation Discussion

The pseudo depth maps exhibit high uncertainty in the far distance regions and inconsistency across multiple views, which limits our method's ability to reconstruct distant background areas, such as the sky. However, since foreground objects are the primary focus in reconstruction tasks, our method demonstrates superior performance in these regions. To address this limitation, we plan to introduce an uncertainty-based weighting mechanism for the geometry regularization.



2DGS + Depth Fusion Ours

Figure 12: **Comparison on Depth Fusion and Ours on Barn.**



Ours with Marching Cubes Ours with Marching Tetrahedral

Figure 13: **Ablation Study on Different Meshing Methods.** While Marching Cubes struggles with resolution limitations in unbounded scene reconstruction, resulting in low-quality meshes with missing details, our method consistently produces high-quality meshes with enhanced geometric fidelity and preserved fine structures.

	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
N-angelo [20]	0.49	1.05	0.95	0.38	1.22	1.10	2.16	1.68	1.78	0.93	0.44	1.46	0.41	1.13	0.97	1.07
N-angelo* [20]	0.37	0.72	0.35	0.35	0.87	0.54	0.53	1.29	0.97	0.73	0.47	0.74	0.32	0.41	0.43	0.61
Ours	0.45	0.65	0.36	0.36	0.94	0.70	0.67	1.27	0.99	0.63	0.49	0.84	0.39	0.53	0.47	0.65

Table 6: **Quantitative Comparison on the DTU Dataset [12].** We show the Chamfer distance. For Neuralangelo [20], we report the results from UniSDF [34] reproduction as N-angelo, and the results from Neuralangelo publication as N-anglo*, which is not reproducible.

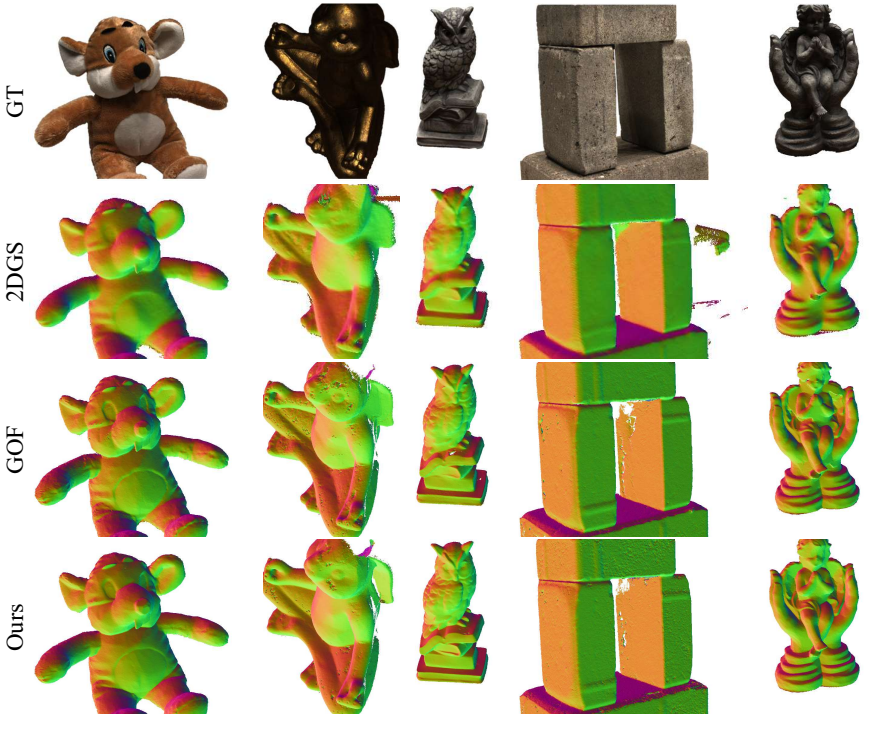


Figure 14: **Additional Results on DTU [12].**