GS-MESH: IMPLICIT SURFACE BRIDGING FOR COLLABORATIVE GAUSSIAN-MESH RECONSTRUCTION

Anonymous authors
Paper under double-blind review



Figure 1: **GS-MESH** bridges Gaussians and meshes through a Gaussian-native implicit surface, enabling detailed and coherent surface reconstruction while preserving high-quality novel-view synthesis.

ABSTRACT

3D Gaussian Splatting enables high-quality novel-view synthesis and real-time rendering, but its discrete primitives do not directly provide explicit and continuous surface representations. Existing methods either extract meshes through post-processing, preventing mesh feedback to Gaussians, or introduce additional geometric representations that require explicit coupling. To address these limitations, we propose **GS-Mesh**, a collaborative Gaussian-mesh reconstruction framework based on *implicit surface bridging*. Our key observation is that surface-aligned Gaussians naturally form attribute-rich oriented surface samples, whose positions, rotations, and opacities can directly induce a continuous implicit surface. Based on this observation, we instantiate Implicit Moving Least Squares (IMLS) on Gaussian primitives and combine it with differentiable surface extraction and mesh rendering, establishing a Gaussian-native implicit bridge for collaborative Gaussian-mesh optimization. We further introduce IMLS-guided densification to improve surface support in under-sampled regions and coarse-to-fine surface optimization to progressively recover coherent geometry and fine details. Experiments on standard surface reconstruction and novel-view synthesis benchmarks demonstrate accurate and detailed mesh reconstruction while retaining competitive rendering quality. The code will be made publicly available.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

3D Gaussian Splatting (Kerbl et al., 2023) enables high-quality novel-view synthesis and real-time rendering. However, many downstream applications, such as physical simulation and surface processing, require explicit meshes with continuous surfaces, which discrete Gaussian primitives do not directly provide. This motivates recovering accurate and detailed meshes from Gaussian representations while retaining high-quality rendering.

To this end, most existing methods follow a staged pipeline that first optimizes Gaussians and then extracts a mesh through post-processing, such as depth fusion, TSDF fusion, and Poisson reconstruction (Guédon & Lepetit, 2024; Yu et al., 2024b; Zhang et al., 2026a). Such decoupling prevents the mesh from constraining Gaussian optimization and may lose local geometric information during fusion or discretization, degrading thin structures and fine details. A few recent works (Yu et al., 2024a; Zhang et al., 2024; Guédon et al., 2025) attempt to address this limitation by incorporating surface geometry into Gaussian optimization. They jointly optimize Gaussians with additional geometric representations, such as an implicit field or explicit mesh, but require additional constraints to maintain their correspondence. This motivates a more direct question: *can the Gaussian representation itself induce a continuous surface that directly connects Gaussian rendering and mesh reconstruction?*

Classical point-set surface reconstruction provides a natural answer. Oriented points can define a continuous implicit surface by combining sample positions with local surface orientations (Alexa et al., 2001; Kolluri, 2008). Surface-aligned Gaussians can be viewed similarly as *attribute-rich oriented surface samples*: their centers locate local surface elements, rotations encode surface orientations, and opacities show their contributions. Based on this observation, we propose **GS-Mesh**, a collaborative Gaussian-mesh reconstruction framework based on *implicit surface bridging*. The native attributes of Gaussian primitives inherently define an implicit surface. We leverage this property to instantiate an Implicit Moving Least Squares (IMLS) field directly from the Gaussians, from which a mesh is extracted via differentiable Marching Cubes and rendered with differentiable rasterization. Since the implicit field is determined by the Gaussian parameters, Gaussian updates are naturally propagated to the implicit surface and the extracted mesh, while mesh-rendering supervision is back-propagated to the underlying Gaussians. In this way, the implicit surface serves as a bridge between Gaussian and mesh representations, enabling both rendering objectives to jointly refine the same geometry in a closed Gaussian–mesh optimization loop.

Making such a Gaussian-native surface bridge effective introduces two practical challenges. First, reliable implicit surface construction requires sufficient geometric support, whereas conventional Gaussian densification is primarily driven by image-space signals and may leave geometrically under-sampled regions insufficiently covered. We therefore introduce *IMLS-guided densification*, which uses the current implicit surface and local sampling density to generate additional surface-aligned Gaussians in sparsely represented regions. Second, a fixed surface resolution makes it difficult to balance global surface coherence and fine geometric details. We therefore adopt *coarse-to-fine surface optimization*, first establishing a smooth and globally coherent surface at low resolution and progressively increasing the resolution to recover finer geometric details.

Our contributions are summarized as follows:

- We propose GS-Mesh, which directly induces a Gaussian-native IMLS surface from Gaussian attributes and uses it as a differentiable bridge for collaborative Gaussian-mesh optimization, without introducing an independently parameterized implicit field.
- We introduce IMLS-guided densification to address insufficient geometric support in under-sampled regions by inserting new surface-aligned Gaussians according to the current implicit surface and local sampling density.
- We develop coarse-to-fine surface optimization to overcome the difficulty of balancing global surface coherence and fine geometric details, progressively increasing the IMLS grid resolution as optimization proceeds.

2 Related Work

Neural Implicit Surface Reconstruction. Neural implicit methods represent geometry as continuous fields and have demonstrated strong surface reconstruction quality. Early methods such as IDR (Yariv et al., 2020), UNISURF (Oechsle et al., 2021), VolSDF (Yariv et al., 2021), and NeuS (Wang et al., 2021) combine implicit surfaces with differentiable rendering for multi-view reconstruction. Subsequent methods further improve geometric fidelity through high-frequency representations and geometric priors, as exemplified by Neuralangelo (Li et al., 2023). Despite their accurate and continuous geometry, neural implicit methods typically require costly field optimization and rendering, motivating more efficient explicit representations for surface reconstruction.

Gaussian-based Surface Reconstruction. 3D Gaussian Splatting (Kerbl et al., 2023) provides an explicit scene representation with high-quality real-time rendering, but does not directly define a continuous surface. Recent methods adapt Gaussian primitives for surface reconstruction by encouraging surface alignment or improving geometric regularization (Guédon & Lepetit, 2024; Huang et al., 2024; Yu et al., 2024b; Zhang et al., 2026a). Their meshes, however, are typically obtained after Gaussian optimization through depth fusion, TSDF fusion, or other surface extraction procedures. Such post-processing separates mesh reconstruction from Gaussian optimization, preventing the extracted surface from providing direct geometric feedback.

Mesh-based Surface Reconstruction. Recent methods incorporate explicit meshes into differentiable reconstruction, enabling surface geometry to be directly optimized from multi-view images. MILo (Guédon et al., 2025) introduces mesh-in-the-loop optimization into Gaussian Splatting, using an auxiliary implicit representation for differentiable mesh extraction and geometric supervision. IMLS-Splatting (Yang et al., 2025) constructs differentiable surfaces from an optimizable point representation using IMLS, while Mesh Splatting (Zhang et al., 2026b) directly optimizes mesh geometry through differentiable splatting-based rendering. These methods demonstrate the benefit of bringing explicit surfaces into the optimization loop, but rely on auxiliary implicit variables, explicit mesh parameterization, or separate point representations to model geometry. In contrast, GS-Mesh directly interprets surface-aligned Gaussians as attribute-rich oriented samples and instantiates IMLS from their native attributes, providing a Gaussian-native implicit bridge to explicit meshes for collaborative Gaussian-mesh optimization.

3 Method

Given calibrated multi-view images, GS-Mesh initializes Gaussian primitives from a sparse COLMAP reconstruction (Schonberger & Frahm, 2016) and first optimizes them through differentiable Gaussian rendering. We then instantiate a query-adaptive IMLS field directly from native Gaussian attributes and extract the mesh via differentiable Marching Cubes. Differentiable mesh rendering propagates image-space supervision back through the IMLS filed to the Gaussians, forming a closed Gaussian-mesh optimization loop. Meanwhile, IMLS-guided densification and coarse-to-fine surface optimization improve geometric support and progressively recover fine-scale details.

3.1 Scene Representation

We represent the scene with a set of surface-aligned Gaussian primitives $\mathcal{G} = \{G_i\}_{i=1}^{N}$, where each Gaussian is parameterized as:

$$G_i = (\boldsymbol{\mu}_i, \mathbf{s}_i, \mathbf{R}_i, \alpha_i, \mathbf{f}_i, \tilde{\mathbf{n}}_i). \tag{1}$$

Here, $\boldsymbol{\mu}_i \in \mathbb{R}^3$ denotes the Gaussian center, $\mathbf{s}_i \in \mathbb{R}^2$ its tangential scale, $\mathbf{R}_i \in SO(3)$ its rotation, α_i its opacity, and $\mathbf{f}_i$ its spherical harmonic coefficients. We additionally maintain an auxiliary orientation vector $\tilde{\mathbf{n}}_i$ for each Gaussian to resolve the sign ambiguity of its rotation-derived surfel normal. It is initialized as a random unit vector, normalized before use, and optimized through a differentiable rendered-normal consistency loss, as detailed in A.

3.2 Gaussian-Native Query-Adaptive IMLS

We construct the IMLS filed directly from the Gaussian representation by interpreting surface-aligned Gaussians as oriented samples.

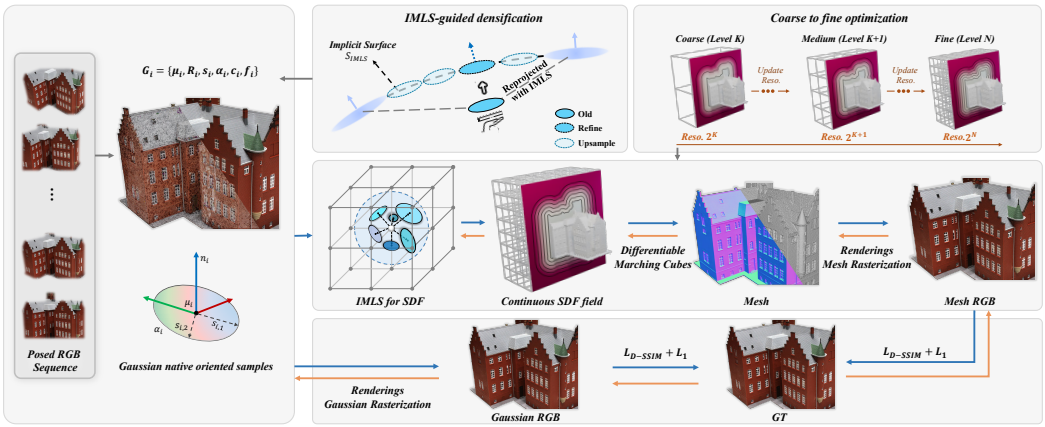

Figure 2: **Overview of GS-Mesh.** Starting from sparse Gaussian primitives, GS-Mesh directly aggregates their native attributes into a query-adaptive IMLS field. Differentiable Marching Cubes and mesh rasterization propagate image-space supervision through the mesh-IMLS field-Gaussian pathway, enabling collaborative optimization. IMLS-guided densification improves surface coverage, while coarse-to-fine optimization progressively recovers fine-scale details.

Gaussian attribute extraction. For each Gaussian G_i, we use its center $\boldsymbol{\mu}_i$ as a surface sample. Its unoriented normal is given by the axis orthogonal to the Gaussian tangent plane,

$$\mathbf{n}_i^{\text{raw}} = \mathbf{R}_i \mathbf{e}_3, \qquad \mathbf{e}_3 = (0,0,1)^\top. \tag{2}$$

Since this axis is sign-ambiguous, we orient it using the auxiliary normal $\tilde{\mathbf{n}}_i$:

$$\mathbf{n}_i = \operatorname{sgn}\left(\langle \mathbf{n}_i^{\text{raw}}, \tilde{\mathbf{n}}_i \rangle\right) \mathbf{n}_i^{\text{raw}}. \tag{3}$$

Here, $\operatorname{sgn}(\cdot)$ only selects the orientation. Auxiliary normals are optimized via rendered-normal alignment, while multi-view supervision and IMLS updates encourage local consistency.

For a camera centered at $\mathbf{o}$, we evaluate the spherical harmonic coefficients $\mathbf{f}_i$ along the viewing direction to obtain the view-dependent color $\mathbf{c}_i(\mathbf{o})$. The activated opacity $\bar{\alpha}_i$ is used as the confidence weight. This yields the oriented and colored sample set:

$$\mathcal{P}(\mathbf{o}) = \{(\boldsymbol{\mu}_i, \mathbf{n}_i, \bar{\alpha}_i, \mathbf{c}_i(\mathbf{o}))\}_{i=1}^{N}. \tag{4}$$

Gradients from the implicit-surface branch propagate to Gaussian positions, rotations, and appearance features, while opacity is detached and used only as a confidence weight.

Active voxel filter. Evaluating the IMLS field over a dense volumetric grid incurs cubic cost with respect to the grid resolution, although the target surface occupies only a small fraction of the volume. We therefore evaluate the field only at voxels supported by nearby Gaussians. After normalizing the scene to a unit cube, we organize Gaussian centers with a spatial hash grid and estimate the local sampling scale of each Gaussian as:

$$r_i = \frac{1}{K-1} \sum_{j \in \mathcal{N}_K(i) \setminus i} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2, \tag{5}$$

where $\mathcal{N}_K(i)$ denotes the K nearest Gaussian centers. A grid vertex $\mathbf{x}$ is marked active if it lies within the support of at least one Gaussian:

$$\mathcal{V}_{\text{act}} = \{\mathbf{x} \in \mathcal{V} \mid \exists i,\ \|\mathbf{x} - \boldsymbol{\mu}_i\|_2 \le \eta r_i\}, \tag{6}$$

where $\mathcal{V}$ is the full voxel grid and η controls the support radius. For each active voxel, we subsequently perform an exact local neighbor query before field evaluation. This restricts computation to surface-relevant regions and enables high-resolution field construction.

Query-adaptive IMLS. For each active query $\mathbf{x}$, we retrieve its K nearest Gaussians and define a query-adaptive bandwidth:

$$h(\mathbf{x}) = \eta \frac{1}{K} \sum_{i \in \mathcal{N}_K(\mathbf{x})} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2. \tag{7}$$

The contribution of Gaussian i is then

$$w_i(\mathbf{x}) = \bar{\alpha}_i \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2}{h(\mathbf{x})^2}\right). \tag{8}$$

Using these weights, we compute the IMLS field by aggregating local point-to-plane distances:

$$F(\mathbf{x}) = \frac{\sum_{i \in \mathcal{N}_K(\mathbf{x})} w_i(\mathbf{x}) \langle \mathbf{x} - \boldsymbol{\mu}_i, \mathbf{n}_i \rangle}{\sum_{i \in \mathcal{N}_K(\mathbf{x})} w_i(\mathbf{x})}. \tag{9}$$

The corresponding appearance field is interpolated using the same local weights:

$$C(\mathbf{x}) = \frac{\sum_{i \in \mathcal{N}_K(\mathbf{x})} w_i(\mathbf{x}) \mathbf{c}_i(\mathbf{o})}{\sum_{i \in \mathcal{N}_K(\mathbf{x})} w_i(\mathbf{x})}. \tag{10}$$

During back-propagation, the active-voxel mask and neighbor assignments are fixed, while gradients flow through the IMLS field and appearance interpolation to the Gaussian parameters.

3.3 Differentiable Mesh Extraction and Rendering

Given the IMLS implicit field F and appearance field C, we extract the zero-level surface using differentiable Marching Cubes (Lorensen & Cline, 1998), with vertex colors interpolated from C. We directly adopt the differentiable Marching Cubes implementation of IMLS-Splatting Yang et al. (2025). The resulting mesh is rendered using Nvdiffrast (Laine et al., 2020) to produce $\hat{\mathbf{I}}_m$. Image-space supervision then propagates along the Mesh $\rightarrow$ IMLS field $\rightarrow$ Gaussian pathway, providing mesh-based constraints on Gaussian geometry. Together with the Gaussian rendering branch, this forms a closed Gaussian-mesh collaborative optimization.

3.4 IMLS-guided densification

Standard Gaussian densification is primarily driven by image-space rendering gradients and does not explicitly account for geometric sampling. Consequently, regions with weak photometric gradients may remain under-sampled, providing insufficient support for implicit surface reconstruction. To address this issue, we leverage the current IMLS surface to identify geometrically sparse regions and introduce new Gaussians directly on the reconstructed surface.

We first project the existing Gaussian centers toward the locally reconstructed IMLS surface and update their normals using the corresponding field gradients. For each refined center, we measure local sparsity using the average neighbor distance r_i. To reduce sensitivity to isolated outliers, these distances are percentile-clipped and normalized into sparsity scores $\widehat{r}_i \in [0, 1]$. The number of new samples is determined by a monotonic mapping:

$$m_i = \lfloor m_{\min} + \sigma(\gamma(\widehat{r}_i - \tau))\,(m_{\max} - m_{\min}) \rfloor, \tag{11}$$

where $\sigma(\cdot)$ is the sigmoid function. Thus, densely sampled regions generate few or no additional primitives, whereas sparse regions receive more samples.

Let $\mathbf{t}_{i,1}$ and $\mathbf{t}_{i,2}$ be two orthogonal tangent directions derived from $\mathbf{n}_i$. Candidate points are sampled around the corresponding Gaussian as:

$$\mathbf{q}_{i,j} = \boldsymbol{\mu}_i + \delta_{i,j}^1 \mathbf{t}_{i,1} + \delta_{i,j}^2 \mathbf{t}_{i,2} + \delta_{i,j}^n \mathbf{n}_i, \tag{12}$$

where the tangential perturbations are proportional to r_i, while the normal perturbation has a substantially smaller variance. This anisotropic sampling favors surface coverage without introducing excessive off-surface points. Each candidate is subsequently projected onto the local IMLS surface:

$$\mathbf{q}_{i,j}^* = \mathbf{q}_{i,j} - \frac{F(\mathbf{q}_{i,j})}{\|\nabla F(\mathbf{q}_{i,j})\|_2^2} \nabla F(\mathbf{q}_{i,j}), \tag{13}$$

Robust reweighting suppresses inconsistent neighbors during projection. New Gaussian normals are derived from the field gradient, appearance and opacity are inherited from the parent, and scales are re-estimated from local spacing. A short Gaussian-only warm-up follows each densification step before collaborative Gaussian-mesh optimization resumes.

3.5 COARSE-TO-FINE OPTIMIZATION

A fixed grid resolution presents a trade-off between global surface coherence and fine geometric detail: a coarse grid suppresses high-frequency variations but limits geometric resolution, whereas a fine grid is more expensive and can be sensitive to unreliable geometry early in training. We therefore adopt a coarse-to-fine schedule, initializing the IMLS grid at 128^3 and increasing each spatial dimension by 64 at fixed intervals until reaching 512^3. The coarse stages first establish a rough and globally coherent surface, while subsequent refinements reduce discretization error and progressively recover fine-scale geometry.

3.6 TRAINING OBJECTIVES

Our overall objective combines Gaussian rendering, mesh rendering, surface regularization, and monocular normal supervision:

$$\mathcal{L} = \mathcal{L}_g + \mathcal{L}_{\text{mesh}} + \mathcal{L}_{\text{curv}} + \mathcal{L}_{\text{prior}}. \tag{14}$$

Gaussian rendering loss. Let $\hat{\mathbf{I}}_g$ and $\mathbf{I}$ denote the Gaussian-rendered and ground-truth images, respectively. We optimize:

$$\mathcal{L}_g = (1-\lambda)\mathcal{L}_1(\hat{\mathbf{I}}_g, \mathbf{I}) + \lambda\mathcal{L}_{\text{ssim}}(\hat{\mathbf{I}}_g, \mathbf{I}) + \lambda_N\mathcal{L}_N, \tag{15}$$

where $\mathcal{L}_N$ enforces Gaussian normal consistency and auxiliary-to-rotation normal alignment.

Mesh rendering loss. We apply the same photometric objective to the differentiably rendered mesh image $\hat{\mathbf{I}}_m$:

$$\mathcal{L}_{\text{mesh}} = (1-\lambda)\mathcal{L}_1(\hat{\mathbf{I}}_m, \mathbf{I}) + \lambda\mathcal{L}_{\text{ssim}}(\hat{\mathbf{I}}_m, \mathbf{I}). \tag{16}$$

This supervision propagates through the mesh-IMLS filed-Gaussian pathway, encouraging multi-view-consistent surface geometry.

Curvature loss. To encourage surface smoothness, we regularize the variation of the IMLS field normals near the zero level set. Let $\mathbf{N}(\mathbf{x})$ denote the normalized field gradient and $\Omega_b = \{\mathbf{x} \in \mathcal{V}_{\text{act}} \mid |F(\mathbf{x})| \le b\Delta\}$ the narrow-band samples. The curvature loss is defined as:

$$\mathcal{L}_{\text{curv}} = \frac{\sum_{\mathbf{x}\in\Omega_b} S(\mathbf{x}) \sum_{d=1}^{3} \rho(\mathbf{N}(\mathbf{x}+\Delta\mathbf{e}_d) - \mathbf{N}(\mathbf{x}))}{\sum_{\mathbf{x}\in\Omega_b} S(\mathbf{x})}, \tag{17}$$

where Δ is the voxel size, $S(\mathbf{x})$ is a detached confidence weight, and ρ is the Charbonnier penalty. This loss serves as a robust discrete curvature prior.

Normal prior loss. Following (Chen et al., 2024), we use monocular normal predictions to reduce the geometric ambiguity of photometric supervision. Let $\hat{\mathbf{N}}^g(\mathbf{p})$ and $\mathbf{N}^p(\mathbf{p})$ denote the rendered Gaussian and predicted normal at pixel $\mathbf{p}$. We combine an ℓ_1 penalty with cosine consistency:

$$\mathcal{L}_{\text{prior}} = \left\| \hat{\mathbf{N}}^g(\mathbf{p}) - \mathbf{N}^p(\mathbf{p}) \right\|_1 + 1 - \left\langle \hat{\mathbf{N}}^g(\mathbf{p}), \mathbf{N}^p(\mathbf{p}) \right\rangle. \tag{18}$$

4 EXPERIMENTS

4.1 DATASETS AND METRICS

We evaluate our method on three common datasets: (1) the DTU dataset (Jensen et al., 2014), which provides object-centric multi-view scans with accurate ground-truth geometry for quantitative evaluation; (2) the BlendedMVS dataset (Yao et al., 2020), which consists of in-the-wild captures with intricate shapes; and (3) the NeRF-synthetic dataset (Mildenhall et al., 2021), which contains eight synthetic object-level scenes with known ground-truth geometry and camera poses. Following existing work, we report the Chamfer distance for 15 scenes of the DTU dataset and 18 scenes from the low-res set of the BlendedMVS dataset, using official evaluation protocols. For the NeRF Synthetic dataset, we report the PSNR and SSIM metrics.

Table 1: **Quantitative comparison of mesh reconstruction on the DTU dataset.** We report Chamfer distance (mm, ↓) for 15 test scenes. Colors indicate 1st, 2nd, and 3rd place per scene.

		24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
implicit	NeRF (Mildenhall et al., 2021)	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	> 12h
	VolSDF (Yariv et al., 2021)	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	> 12h
	NeuS (Wang et al., 2021)	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	> 12h
	Neuralangelo (Li et al., 2023)	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	> 12h
Gaussian	3DGS (Kerbl et al., 2023)	2.14	1.53	2.08	1.68	3.49	2.21	1.43	2.07	2.22	1.75	1.79	2.55	1.53	1.52	1.50	1.96	5.2m
	SuGaR (Guédon & Lepetit, 2024)	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	52m
	GaussianSurfels (Dai et al., 2024)	0.66	0.93	0.54	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	6.7m
	2DGS (Huang et al., 2024)	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	8.9m
	GOF (Yu et al., 2024b)	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	55m
	RaDe-GS (Zhang et al., 2026a)	0.46	0.73	0.33	0.38	0.79	0.75	0.76	1.19	1.22	0.62	0.70	0.78	0.36	0.68	0.47	0.68	8.3m
mesh	MILo (Guédon et al., 2025)	0.43	0.74	0.34	0.37	0.80	0.74	0.70	1.21	1.22	0.66	0.62	0.80	0.37	0.76	0.48	0.68	25m
	Mesh Splatting (Zhang et al., 2026b)	0.46	0.73	0.49	0.43	0.77	0.82	0.65	1.03	0.95	0.52	0.58	0.59	0.37	0.44	0.42	0.62	23m
	IMLS-Splatting (Yang et al., 2025)	0.52	1.02	0.37	0.32	0.86	0.50	0.48	1.15	0.76	0.59	0.37	0.67	0.33	0.33	0.34	0.57	11m
	Ours	0.40	0.52	0.29	0.37	0.92	0.62	0.51	1.07	0.83	0.61	0.45	0.71	0.40	0.43	0.44	0.57	45m

Table 2: **Quantitative comparison of mesh reconstruction on the BlendedMVS dataset.** We report Chamfer distance (mm, ↓) for 18 test scenes. Colors indicate 1st, 2nd, and 3rd place per scene.

Method	Basketball	Bear	Bread	Camera	Clock	Cow	Dog	Doll	Dragon	Durian	Fountain	Gundam	House	Jade	Man	Monster	Sculpture	Stone	Mean
NeuS	2.96	3.00	2.85	2.61	2.75	2.04	2.75	2.17	2.95	3.14	3.03	1.62	3.23	4.25	2.29	1.92	2.10	2.51	2.68
GaussianSurfels	1.59	1.52	1.41	1.75	5.08	3.21	2.97	2.32	2.93	4.01	2.65	0.97	1.76	3.49	2.23	1.36	3.07	2.02	2.46
SuGaR	8.00	9.73	7.65	7.77	9.21	8.69	9.27	8.75	9.76	8.04	9.05	7.32	7.28	10.7	9.29	8.20	8.98	9.19	8.72
IMLS-Splatting	2.48	1.86	2.69	3.61	2.96	2.80	2.85	2.32	2.39	3.35	2.83	1.78	3.01	5.10	2.61	1.99	2.04	2.85	2.75
Mesh Splatting	1.26	1.16	0.75	1.75	1.99	1.30	2.08	2.02	1.57	2.36	2.37	0.90	1.94	3.27	1.52	1.30	1.64	1.64	1.71
Ours	1.75	1.88	1.02	1.73	2.28	2.04	2.12	2.74	1.91	2.53	2.69	1.18	1.76	3.13	2.18	1.90	2.44	1.67	2.05

4.2 Comparison of mesh reconstruction

Quantitative comparison. Table 1 and Table 2 report the Chamfer distances on the DTU and BlendedMVS datasets, respectively. On DTU, our method achieves the lowest mean error of 0.57 mm, on par with IMLS-Splatting, and ranks among the top three on 12 of the 15 scenes. On BlendedMVS, it achieves the second-lowest mean error of 2.05 mm, outperforming all implicit and Gaussian-based baselines as well as IMLS-Splatting. These results demonstrate strong reconstruction accuracy across both datasets, further supported by the qualitative comparisons below.

Qualitative comparison. As shown in Fig. 3 and Fig. 4, our method reconstructs clean and smooth surfaces while preserving fine geometric structures, such as the window frames in Scan 24, the recessed patterns in Scan 69, and the feather details in Scan 106. Gaussian-based methods such as 2DGS and RaDe-GS decouple optimization from mesh extraction, resulting in incomplete structures and missing details. IMLS-Splatting produces floating artifacts due to its independently optimized point representation. Although MILo incorporates explicit mesh optimization, its auxiliary SDF variables are coupled with the Gaussian representation through indirect consistency constraints, limiting the recovery of complex structures and fine details. In contrast, our method propagates mesh-based supervision directly back to the Gaussians and incorporates a normal prior, yielding more coherent surfaces with finer geometric details.

4.3 Novel-view synthesis

We evaluate novel-view synthesis on the NeRF Synthetic dataset using the native Gaussian renderer. As shown in Table 3, our method achieves 32.10 dB PSNR and 0.97 SSIM, outperforming MILo and PGSR while remaining competitive with 2DGS. These results show that the IMLS-derived mesh supervision improves geometric reconstruction while retaining high-quality Gaussian rendering. Combined with the improved reconstruction accuracy demonstrated above, our method achieves a favorable balance between geometric quality and novel-view synthesis quality.

Table 3: **Quantitative comparison on NeRF Synthetic.**

Method	PSNR↑	SSIM↑
2DGS	33.06	0.97
PGSR	30.44	0.96
MILo	31.85	0.97
Ours	32.10	0.97

4.4 Ablation Study

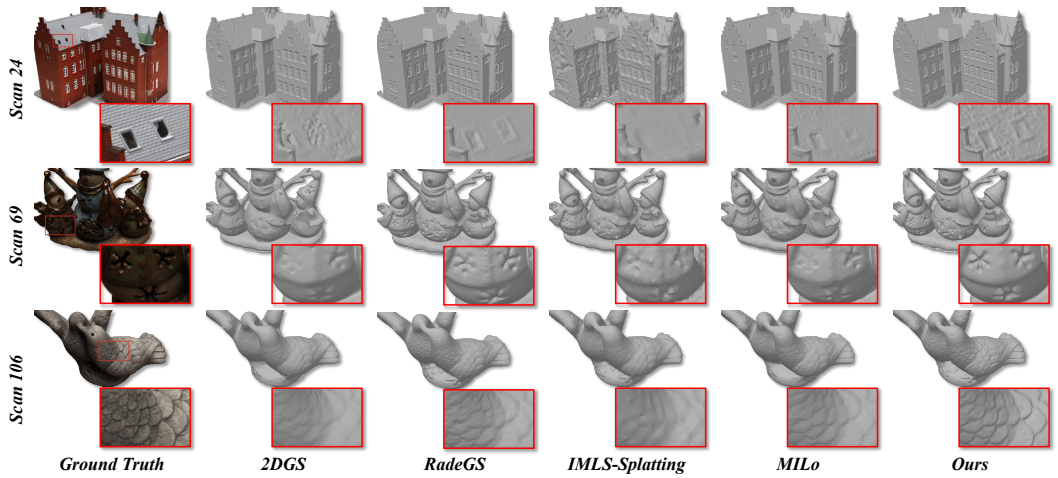

Figure 3: **Qualitative comparison on the DTU dataset.** We compare mesh reconstructions on three challenging scenes. 2DGS oversmooths details, RaDe-GS and MILo blur sharp structures, and IMLS-Splatting introduces surface artifacts. Our method preserves sharp edges and high-frequency details while maintaining smooth meshes.

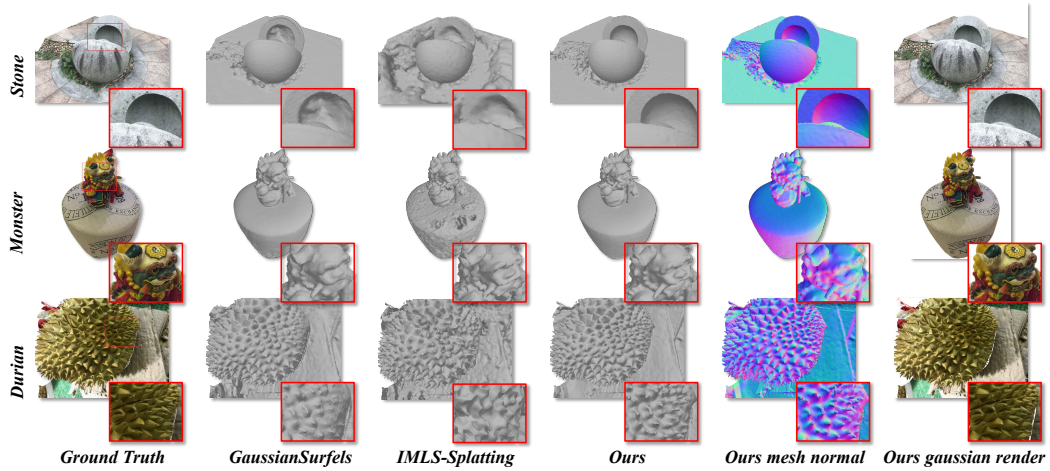

Figure 4: **Qualitative comparison on the BlendedMVS dataset.** GaussianSurfels oversmooths local geometry, while IMLS-Splatting produces noisy surfaces. Our method recovers high-frequency details with smooth meshes, consistent normals, and high-quality Gaussian rendering.

We conduct cumulative ablations on the DTU dataset, with each component added to the preceding configuration. Quantitative and qualitative results are reported in Table 4 and Fig. 5, respectively. Starting from the base model, the monocular normal prior $\mathcal{L}_{\text{prior}}$ reduces the Chamfer distance from 0.80 mm to 0.69 mm by providing surface-orientation cues and mitigating the geometric ambiguity of photometric supervision. Adding $\mathcal{L}_{\text{mesh}}$ further reduces the error to 0.62 mm. This surface-based supervision complements the permissive alpha-blended Gaussian rendering objective with stronger visibility and silhouette constraints, thereby promoting more coherent geometry. IMLS-guided densification improves geometric support in under-sampled regions, further reducing the error to 0.57 mm, while $\mathcal{L}_{\text{curv}}$ maintains the same Chamfer distance at the reported precision.

Table 4: **Ablation studies on the DTU dataset.**

Configuration	CD
Base	0.80
$+\,\mathcal{L}_{\text{prior}}$	0.69
$+\,\mathcal{L}_{\text{mesh}}$	0.62
+ IMLS densification	0.57
$+\,\mathcal{L}_{\text{curv}}$ (Full)	0.57

The representative example in Fig. 5 further illustrates the role of each component. The base model exhibits a geometric depression in a region with strong highlights, reflecting the inherent geometry-

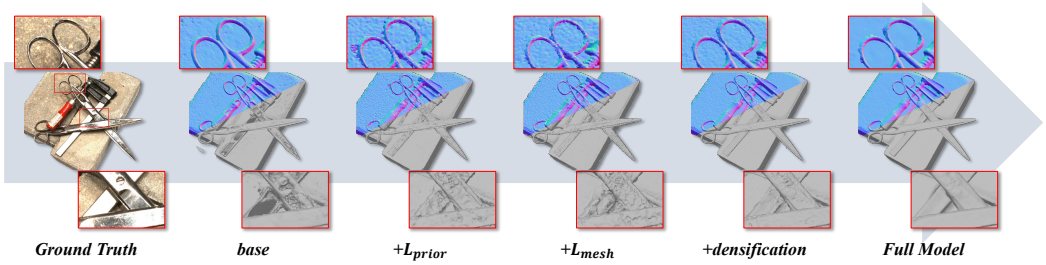

Figure 5: **Ablation study on the DTU dataset.** Normal prior improves surface orientation and reduces geometric ambiguity, while mesh supervision further enhances visibility consistency and geometric coherence. IMLS-guided densification recovers more accurate geometric structures, and curvature regularization further smooths the reconstructed mesh.

appearance ambiguity of Gaussian rendering, together with noticeable surface roughness. The normal prior corrects this depression by providing explicit surface-orientation guidance. However, monocular normal predictions are less reliable near thin structures and object boundaries, causing the added prior to introduce spurious floating geometry in these regions. Such artifacts may be weakly penalized by alpha-blended Gaussian rendering but produce incorrect visibility and silhouettes when rendered as a surface. Consequently, $\mathcal{L}_{\text{mesh}}$ suppresses the newly introduced floating geometry through surface-based rendering supervision. IMLS-guided densification then provides denser and more uniform surface support, reducing local roughness and further improving geometric accuracy. Finally, $\mathcal{L}_{\text{curv}}$ suppresses residual high-frequency normal variations within the narrow band, producing the smoothest visual result.

5 Limitation and Future Work

As analyzed in Sec. B.1, IMLS grid resolution is closely related to mesh quality when the underlying point distribution provides sufficient geometric support. For large-scale unbounded scenes, however, even 1024^3 grids may provide insufficient spatial resolution, while further increasing the resolution incurs prohibitive memory costs. We therefore explore a MILo-style adaptive sampling strategy (Guédon et al., 2025), which enables Gaussian-native mesh extraction on unbounded scenes with reasonable quality. The resulting mesh reconstructions are shown in Fig. 7, with implementation details provided in the supplementary material. Nevertheless, this strategy generates up to 8M candidate queries, many of which do not contribute to the final surface, substantially slowing training to approximately eight hours per Tanks and Temples scene (Knapitsch et al., 2017) in our setup. Efficient surface-aware query allocation for jointly achieving high-quality Gaussian rendering and mesh reconstruction in unbounded scenes remains an important direction for future work.

6 Conclusion

We presented **GS-Mesh**, a collaborative Gaussian-mesh reconstruction framework that bridges Gaussian primitives and explicit meshes through a Gaussian-native implicit surface. By interpreting surface-aligned Gaussians as oriented samples, GS-Mesh directly instantiates IMLS from their native attributes, enabling differentiable Gaussian-surface-mesh optimization without an independently parameterized implicit field. Together with IMLS-guided densification and coarse-to-fine surface optimization, our approach recovers accurate and coherent surfaces while retaining competitive rendering quality. We hope this work encourages further exploration of Gaussian-native continuous geometry, particularly for efficient reconstruction of large-scale and unbounded scenes.

AI USE STATEMENT

Large language models were used solely to aid and polish the writing, including improving language quality and readability and ensuring that the method is described clearly. The authors reviewed all AI-assisted revisions and take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies Gaussian Mesh reconstruction using publicly available datasets and pretrained models, without collecting new personal data or involving human subjects. We assert that this research has been carried out in accordance with the code of ethics.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility and verification, we provide implementation and evaluation details in this paper and will make the associated source code publicly available upon acceptance.

REFERENCES

Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T Silva. Point set surfaces. In *Proceedings Visualization, 2001. VIS'01.*, pp. 21–29. IEEE, 2001.

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. *Advances in Neural Information Processing Systems*, 37:139725–139750, 2024.

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*, pp. 1–11, 2024.

Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5354–5363. IEEE, 2024.

Antoine Guédon, Diego Gomez, Nissim Maruani, Bingchen Gong, George Drettakis, and Maks Ovsjanikov. Milo: Mesh-in-the-loop gaussian splatting for detailed and efficient surface reconstruction. *ACM Transactions on Graphics (TOG)*, 44(6):1–15, 2025.

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*, pp. 1–11, 2024.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 406–413, 2014.

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

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.

Ravikrishna Kolluri. Provably good moving least squares. *ACM Transactions on Algorithms (TALG)*, 4(2):1–25, 2008.

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (ToG)*, 39(6):1–14, 2020.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8456–8465. IEEE, 2023.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.

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.

Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5569–5579. IEEE, 2021.

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

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.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026.

Kaizhi Yang, Liu Dai, Isabella Liu, Xiaoshuai Zhang, Xiaoyan Sun, Xuejin Chen, Zexiang Xu, and Hao Su. Imls-splatting: Efficient mesh reconstruction from multi-view images via point representation. *ACM Transactions on Graphics (TOG)*, 44(4):1–11, 2025.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1787–1796. IEEE, 2020.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020.

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.

Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. Gsdf: 3dgs meets sdf for improved neural rendering and reconstruction. *Advances in Neural Information Processing Systems*, 37:129507–129530, 2024a.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics (ToG)*, 43(6):1–13, 2024b.

Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiao-Xiao Long, and Ping Tan. Radegs: Rasterizing depth in gaussian splatting. *ACM Transactions on Graphics*, 45(2):1–14, 2026a.

Ruiqi Zhang, Jiacheng Wu, and Jie Chen. Mesh splatting for end-to-end multiview surface reconstruction. *arXiv preprint arXiv:2601.21400*, 2026b.

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, 2024.

A Implementation Details

Training setup. We initialize the Gaussian representation from the sparse point cloud produced by COLMAP and optimize each scene for 30k iterations using Adam. The spherical harmonic degree is set to 3. Adaptive densification is performed every 100 iterations from iteration 500 to 15k, with opacity reset every 3k iterations. The DSSIM weights in both $\mathcal{L}_g$ and $\mathcal{L}_{\text{mesh}}$ are set to 0.2. We use weights of 0.05 and 0.1 for Gaussian normal consistency and auxiliary-normal alignment, respectively. The monocular normal prior is introduced after 7k iterations with a weight of 0.05. We use MoGe-2 (Wang et al., 2026) for monocular normal prediction. The mesh-rendering loss is activated at iteration 15k with a weight of 0.2. All experiments were conducted on a single NVIDIA GeForce RTX 4090D GPU.

Auxiliary normal optimization. Each auxiliary normal $\tilde{\mathbf{n}}_i$ is initialized as a random unit vector and normalized before use. Since the sign selection in Eq. equation 3 is discrete, no gradient is propagated through $\text{sgn}(\cdot)$. Instead, we rasterize the auxiliary normals and align the resulting normal map $\widehat{\mathbf{N}}_{\text{aux}}$ with the detached camera-facing normal map $\widehat{\mathbf{N}}_{\text{rot}}$ derived from the Gaussian rotations:

$$\mathcal{L}_{\text{ori}} = \frac{1}{|\Omega|} \sum_{p \in \Omega} \left(1 - \left\langle \widehat{\mathbf{N}}_{\text{aux}}(p), \text{sg}\left[\widehat{\mathbf{N}}_{\text{rot}}(p)\right] \right\rangle\right), \tag{19}$$

where Ω denotes the visible pixels and $\text{sg}[\cdot]$ is the stop-gradient operator. Together with Gaussian normal consistency, multi-view supervision, and normalized IMLS-gradient updates during densification, this loss encourages locally consistent orientations.

IMLS field construction. The reconstruction domain is defined as a cubic bounding box covering the Gaussian centers with a 10% margin and is normalized to the unit cube. We organize the Gaussians using a 32^3 spatial hash grid and evaluate the IMLS field only at active voxels supported by nearby Gaussians. For each query, we retrieve its $K = 12$ nearest Gaussians and set the spatial bandwidth adaptively to 1.5 times the local neighbor distance.

Coarse-to-fine mesh optimization. Mesh optimization begins at a grid resolution of 128^3. Starting from iteration 16k, the resolution is increased by 64 every 2k iterations until reaching 512^3. The narrow-band curvature regularization is activated at iteration 15k and linearly warmed up over 1k iterations. We use a band width of two voxels, a Charbonnier parameter of ϵ, and sample at most N_{curv} active voxels at each regularization step. The curvature loss weight is set to 0.05, and we sample at most $65{,}536$ active voxels at each regularization step.

IMLS-guided densification. IMLS-guided densification is performed every 2k iterations from iteration 17k to 23k. Local sampling density is estimated using the mean distance to the 16 nearest neighbors and normalized after clipping at the 10th and 90th percentiles. Each Gaussian generates up to ten candidates, which are perturbed primarily along its tangent plane and subsequently projected onto the current IMLS surface. The new Gaussians inherit the appearance and opacity of their parents, while their scales are initialized according to the local point spacing. After each densification step, we optimize only the Gaussian branch for 500 iterations before resuming collaborative Gaussian–mesh optimization.

B Additional Experiments

B.1 Relationship between IMLS Grid Resolution and Mesh Quality

We reconstruct the same scene from an identical Gaussian point cloud using IMLS grids at resolutions of 256, 320, 384, 448, and 512 to analyze the relationship between grid resolution and mesh quality. Increasing the resolution up to 448 progressively improves surface quality, recovering thin track structures and finer bucket geometry, as highlighted by the red and blue insets in Fig 6. The marginal improvement from 448 to 512 indicates that mesh quality is also constrained by the underlying point distribution. This result shows that increasing the grid resolution alone is insufficient for continuous quality improvement. We therefore progressively increase the IMLS grid resolution in a coarse-to-fine manner while applying IMLS-guided densification to geometrically under-sampled regions, enabling the higher-resolution grid to recover finer surface details.

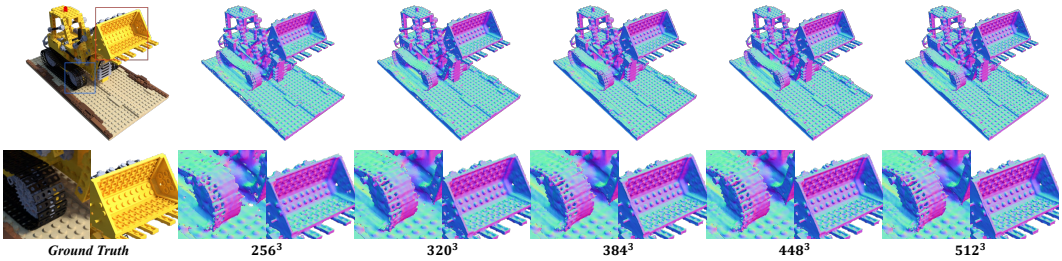

Figure 6: **Comparison of mesh reconstruction quality at different grid resolutions.** The red and blue insets highlight the effects of resolution on fine geometric details.

B.2 Ablation on Active-Voxel Filtering

We evaluate active-voxel filtering on the DTU dataset at a grid resolution of 384^3. Compared with dense evaluation, our filter reduces the number of queried voxels from 56.623M to 0.794M, corresponding to only 1.403% of the full grid. Consequently, the IMLS field computation time decreases from 7966.98 ms to 22.97 ms, including 14.36 ms for active-voxel selection. The Marching Cubes time is also reduced from 13.95 ms to 5.12 ms, while the reconstruction accuracy remains unchanged with a CD of 0.57. These results demonstrate that active-voxel filtering eliminates most irrelevant voxel queries and enables efficient high-resolution surface extraction without compromising reconstruction quality.

Table 5: **Effect of active-voxel filtering at a grid resolution of 384^3.** All timings are reported in milliseconds. Active selection is included in the IMLS field time.

Method	Query ratio (%)	Queried voxels (M)	Active filter	IMLS field	Marching Cubes	Rasterization	CD ↓
Dense voxels	100.000	56.623	–	7966.98	13.95	4.56	0.57
Active voxels	**1.403**	0.794	14.36	**22.97**	**5.12**	**4.17**	0.57

B.3 Why IMLS?

We compare TSDF- and IMLS-based mesh extraction using the same optimized Gaussian representation, with identical spatial bounds, volumetric resolution, Marching Cubes implementation, and evaluation protocol. TSDF reconstructs the implicit field by rendering and fusing multi-view depth maps, whereas IMLS directly constructs the field from the Gaussian geometry. As shown in Table 6, IMLS achieves lower Chamfer distance on 14 of the 18 scenes and reduces the mean error from 3.15 mm to 2.05 mm, corresponding to a 34.9% improvement. The gains are particularly pronounced on challenging scenes such as Clock and Jade, suggesting that directly extracting the implicit surface from the Gaussian representation can avoid errors introduced by intermediate depth rendering and multi-view fusion, thereby better preserving the underlying geometry.

Table 6: **Comparison of TSDF- and IMLS-based mesh extraction on the BlendedMVS dataset.** Both methods extract meshes from the same optimized Gaussian representation. We report the Chamfer distance (mm, ↓) on 18 test scenes. The better result for each scene is shown in **bold**.

Extraction	Basketball	Bear	Bread	Camera	Clock	Cow	Dog	Doll	Dragon	Durian	Fountain	Gundam	House	Jade	Man	Monster	Sculpture	Stone	Mean
TSDF	2.33	2.05	1.10	1.95	7.68	4.15	2.68	**1.83**	3.30	2.79	**2.44**	3.71	2.36	9.33	3.04	**1.64**	**1.73**	2.50	3.15
IMLS	**1.75**	**1.88**	**1.02**	**1.73**	**2.28**	**2.04**	**2.12**	2.74	**1.91**	**2.53**	2.69	**1.18**	**1.76**	**3.13**	**2.18**	1.90	2.44	**1.67**	**2.05**

B.4 Coarse to Fine Optimization

We compare the proposed coarse-to-fine schedule with a fixed 512^3 grid under identical settings on DTU. Coarse-to-fine optimization improves or matches all scenes, reducing the mean Chamfer distance from 0.59 to 0.57 while shortening training time from 57 to 45 minutes. This demonstrates that progressive resolution refinement improves both reconstruction accuracy and computational efficiency.

Table 7: **Comparison between fixed-resolution and coarse-to-fine optimization on DTU.** We report the Chamfer distance (mm, ↓) for each scene, the mean Chamfer distance, and the total training time.

Training schedule	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean ↓	Time ↓
Fixed 512^3	0.41	0.53	0.31	0.39	0.92	0.65	0.53	1.07	0.84	0.61	0.46	0.76	0.41	0.45	0.44	0.59	57 min
Coarse-to-fine	0.40	0.52	0.29	0.37	0.92	0.62	0.51	1.07	0.83	0.61	0.45	0.71	0.40	0.43	0.44	**0.57**	45 min

C Time and Memory Analysis

Table 8 reports the average runtime and peak GPU memory of a single forward pass on the DTU dataset at different grid resolutions. Our Gaussian-native query-adaptive IMLS evaluates the implicit field only at active voxels, avoiding the overhead of processing the entire dense grid. As the resolution increases, more active voxels and mesh faces are introduced to capture finer geometry, leading to higher runtime and memory consumption. Even at 512^3, IMLS field evaluation takes only 34.53 ms, with peak GPU memory below 5 GiB, demonstrating efficient high-resolution Gaussian-to-mesh conversion with moderate memory consumption.

Table 8: **Runtime and peak GPU memory at different grid resolutions.** All results are measured on the DTU dataset. Runtime is reported for a single forward pass.

Grid resolution	256^3	320^3	384^3	448^3	512^3
GPU memory (GiB)	0.713	1.266	2.081	3.219	4.725
Active voxels (M)	0.235	0.459	0.794	1.263	1.887
IMLS field (ms)	17.61	19.60	22.97	26.93	34.53
Marching Cubes (ms)	1.77	3.30	5.12	8.49	12.37
Rasterization (ms)	1.82	2.95	4.17	5.42	6.64
Face number (M)	0.498	0.793	1.157	1.590	2.094

D Extension of Our Method to Unbounded Scenes

Our main formulation targets object-centric scenes, for which the Gaussian-native IMLS field is evaluated on a regular $R \times R \times R$ grid, followed by Marching Cubes to extract the zero level set. This representation provides a simple and efficient discretization when the scene is spatially bounded. However, directly extending it to large-scale unbounded scenes is impractical. In particular, a uniform grid must simultaneously cover a substantially larger spatial extent and maintain sufficient resolution around fine geometric structures. Consequently, even a 1024^3 grid may become too coarse for accurate surface reconstruction, while further increasing the resolution leads to prohibitive memory and computational costs.

To alleviate this limitation, we adopt a MILo-style (Guédon et al., 2025) adaptive spatial discretization and replace the dense Cartesian grid with a Delaunay tetrahedralization constructed from Gaussian-induced query points. Importantly, this modification only changes the spatial locations at which our implicit field is queried; the underlying surface representation remains the same Gaussian-native IMLS formulation introduced in the main paper. We therefore do not introduce an additional learnable SDF or an independent geometric representation.

Following the Gaussian pivot construction used in MILo, we generate nine query points from each Gaussian: its center and eight corner points defined in its local anisotropic coordinate frame. Specifically, for a Gaussian with center $\boldsymbol{\mu}_k$, rotation matrix $\mathbf{R}_k$, and scale vector $\mathbf{s}_k$, its query points are given by

$$\mathbf{p}_{k,i} = \boldsymbol{\mu}_k + \mathbf{R}_k \left(\mathbf{s}_k \odot \mathbf{b}_i\right), \qquad i = 0, \ldots, 8, \tag{20}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{b}_0 = \mathbf{0}$ corresponds to the Gaussian center, and $\{\mathbf{b}_i\}_{i=1}^{8}$ denote the eight corners of a unit box in the local Gaussian coordinate system. In contrast to sampling only Gaussian centers, these additional points provide samples on both sides of the locally represented surface and account for the anisotropic extent and orientation of each Gaussian.

We then compute a 3D Delaunay tetrahedralization over the resulting query points. Compared with a uniform Cartesian grid, this discretization naturally allocates a larger number of tetrahedra to regions

densely supported by Gaussians while using larger tetrahedra in sparsely populated regions. It therefore avoids uniformly discretizing the large empty volume encountered in unbounded scenes. Since recomputing the Delaunay connectivity is relatively expensive and the Gaussian geometry changes progressively during optimization, we update the tetrahedralization every 500 training iterations and reuse its connectivity between consecutive updates.

For every Delaunay vertex $\mathbf{p}_{k,i}$, we evaluate its signed distance using exactly the same Gaussian-native IMLS formulation as in the bounded-scene setting. Given the Gaussian centers, normals, and detached opacity-based confidence weights, our query-adaptive IMLS directly produces

$$f_{k,i} = f_{\mathrm{IMLS}}(\mathbf{p}_{k,i}), \tag{21}$$

without introducing additional per-vertex SDF parameters. This distinction is important: the Delaunay structure serves only as an adaptive discretization of the continuous IMLS field, whereas the implicit surface itself is still induced entirely by the current Gaussian geometry. As a result, the extension preserves the same Gaussian-to-surface coupling used throughout our method.

Given the Delaunay tetrahedra and the IMLS values at their vertices, we replace Marching Cubes with differentiable Marching Tetrahedra. For each tetrahedron whose vertices contain SDF values of opposite signs, the zero crossing along an intersected edge $(\mathbf{p}_a, \mathbf{p}_b)$ is obtained by linear interpolation,

$$\mathbf{v} = \frac{f_a \mathbf{p}_b - f_b \mathbf{p}_a}{f_a - f_b}, \tag{22}$$

where f_a and f_b are the corresponding Gaussian-native IMLS values. These intersection points are assembled according to the Marching Tetrahedra cases to form the extracted triangle mesh. With the tetrahedral connectivity fixed between two Delaunay updates, the extracted vertex locations remain differentiable with respect to both the query-point coordinates and their IMLS values, allowing mesh-based supervision to propagate back to the underlying Gaussian representation.

Apart from the adaptive discretization and the use of Marching Tetrahedra, all remaining components follow the bounded-scene formulation described in the main paper.

This adaptive construction makes Gaussian-native mesh extraction applicable to large-scale scenes without allocating a dense volumetric grid over the entire scene extent. However, it is not yet computationally optimal. On the Tanks and Temples dataset, the Gaussian-induced tetrahedralization can produce up to approximately 8 million candidate IMLS queries, although only a fraction of them eventually participate in tetrahedra intersected by the zero level set. Evaluating these unnecessary candidates substantially increases the optimization cost, resulting in a training time of approximately eight hours per scene in our current implementation. We therefore regard efficient surface-aware query allocation (i.e., identifying and evaluating only samples that are likely to contribute to the final zero level set) as an important direction for extending collaborative Gaussian-mesh reconstruction to large-scale unbounded scenes.

E More Visualization Results

This section presents additional qualitative visualization results, including full reconstruction results and further comparisons on the DTU dataset (Figs. 8 and 9), as well as novel view synthesis comparisons on the NeRF-Synthetic dataset (Fig. 10).



Figure 7: **Examples of mesh reconstructions on the Tanks and Temples dataset.**



Figure 8: **Full visualization results on the DTU dataset.**

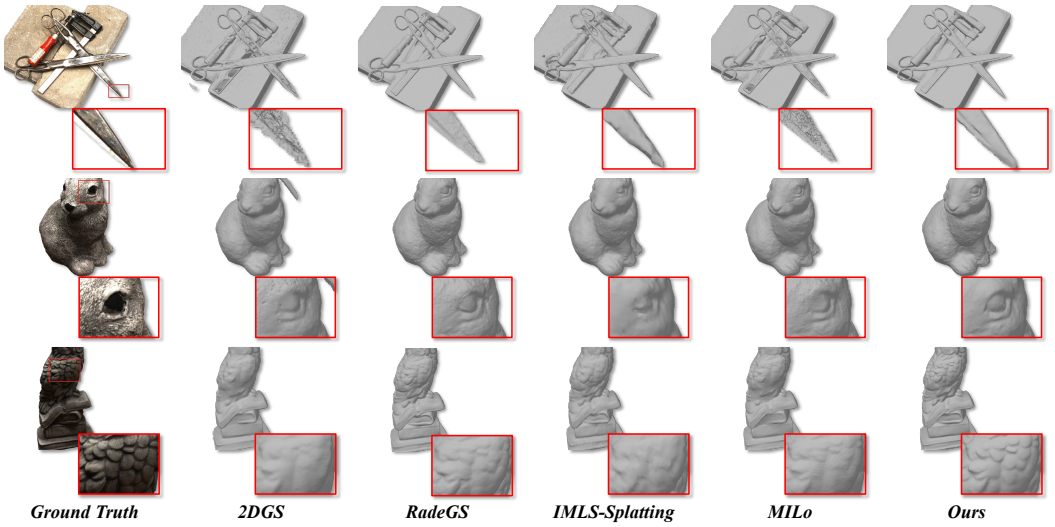

Figure 9: **More qualitative comparisons on the DTU dataset.**

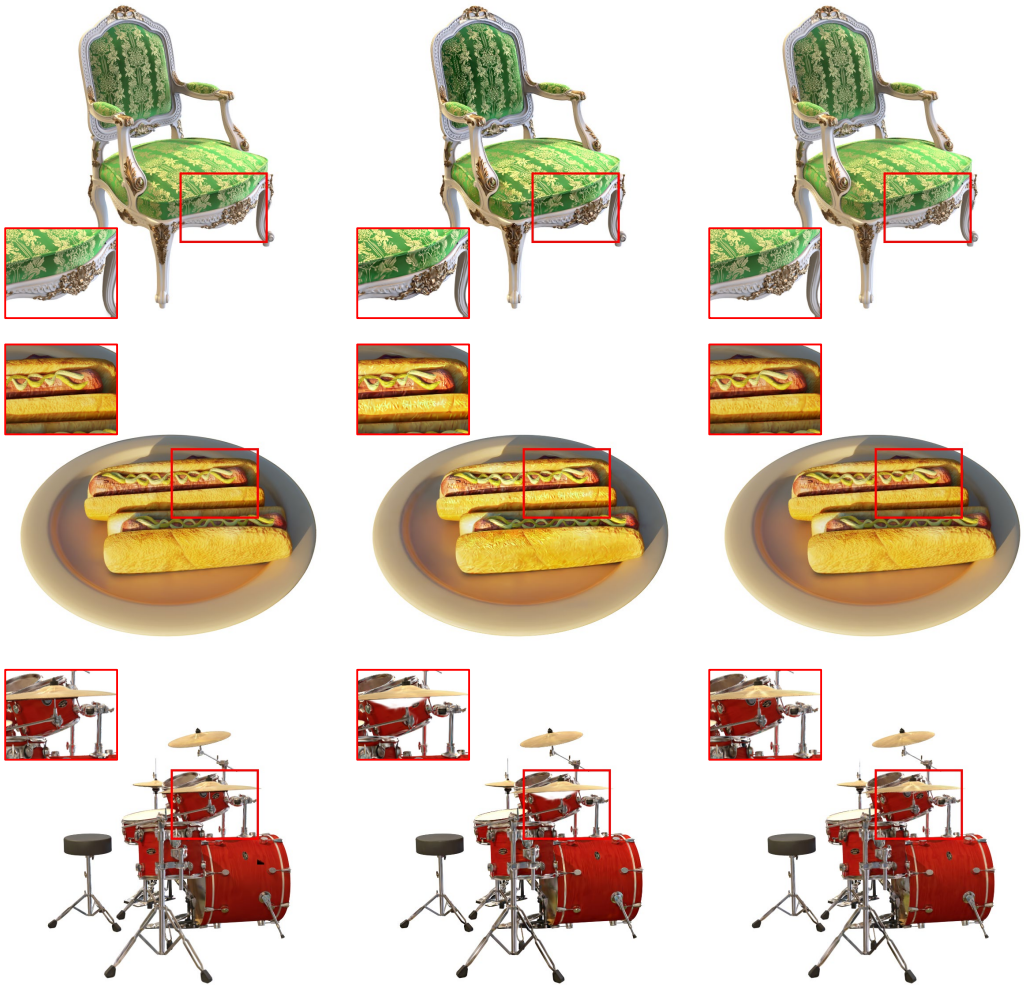

Figure 10: **Qualitative comparison of novel view synthesis on the NeRF-Synthetic dataset.**