Gaussians Can Subtract: Neural Dissolve Fields for Hybrid Scene Rendering

Yiyan Wang[1,*] Haohan Liu[2,*] Tetsuo Shibuya[1]
[1]The University of Tokyo, Japan [2]Fuzhou University, China [*]Equal contribution
5770716834@utac.u-tokyo.ac.jp 852203208@fzu.edu.cn tshibuya@hgc.jp

Abstract

Hybrid Mesh–Gaussian rendering combines compact textured surfaces with flexible 3D Gaussian primitives, offering an efficient alternative to purely Gaussian scene representations for structured environments. Existing scene-reconstruction hybrids commonly follow an additive formulation, where Gaussian residuals compensate for errors left by the mesh. However, not every Mesh–Gaussian conflict is caused by missing content: residual geometry errors, occlusion mismatch, texture-projection artifacts, and view-dependent appearance can make the same mesh content consistent in one view but inconsistent in another. We therefore formulate hybrid rendering as a multi-view inconsistency problem and present Gaussolve, a subtract-and-refill framework that selectively dissolves inconsistent mesh content before residual Gaussians refill the revealed regions. We introduce a lightweight Neural Dissolve Field, parameterized by scalar spherical-harmonic Gaussians, to learn where and from which viewing directions mesh content should be dissolved. A continuous transmittance fusion rule progressively exposes residual Gaussians behind the dissolved mesh, preserving their reconstruction gradients and avoiding hard visibility discontinuities. Experiments on structured and semi-structured scenes demonstrate higher visual fidelity with fewer Gaussian primitives and faster rendering. Project page: https://ooosn.github.io/gaussians-can-subtract/

1. Introduction

3D Gaussian Splatting (3DGS) has become a strong representation for real-time novel-view synthesis due to explicit primitives and efficient differentiable rasterization [17]. However, practical environments contain broad regular structures, such as road surfaces with lane markings and large architectural surfaces [15]. Representing these large regular regions purely with Gaussian primitives can require a large primitive budget, increasing storage and rasterization cost [8, 15, 20]. Textured meshes provide a complementary representation: they compactly anchor broad regular surfaces, while Gaussians remain effective for thin structures, local residual geometry, and view-dependent appearance [6, 15, 17]. However, existing hybrid scene-reconstruction pipelines still inherit limitations in both how the mesh is constructed and how the two representations are combined.

First, we revisit how the textured mesh is constructed. Existing methods typically extract the mesh from a previously optimized NeRF or Gaussian representation and then bake the reconstructed appearance onto the mesh [6, 15], making the resulting texture inherit the constraints of such reconstruction. In particular, enforcing cleaner and more regular geometry can reduce the flexibility of these representations to fit the observed appearance, leading to a trade-off between geometric alignment and rendering quality [6, 14, 45]. Instead, we reconstruct the geometry independently and project the training images directly onto the mesh, thereby preserving the observed appearance when the geometry is reliable.

More importantly, we revisit how the mesh and Gaussians are combined. Even with such a textured mesh, the fundamental Mesh–Gaussian conflict remains. Conceptually, existing hybrids largely follow an additive formulation: the mesh accounts for the regular scene content, while Gaussians compensate for what the mesh fails to represent. But not every Mesh–Gaussian conflict is caused by missing content. Such conflicts may arise from residual geometry errors, occlusion mismatch, texture-projection artifacts, or view-dependent appearance, and importantly, the same mesh content can agree with one view while becoming inconsistent with another. We therefore reinterpret the remaining Mesh–Gaussian conflict as a multi-view inconsistency problem, which requires selectively suppressing inconsistent mesh content rather than only adding Gaussian residuals.

This reformulation exposes a fundamental mismatch between the problem and existing conflict-handling strategies. Existing mesh–Gaussian hybrids often resolve representation conflicts through hard visibility or assignment decisions. For example, prior methods remove inaccurate mesh regions to make room for Gaussians [15], enforce hard assignments between mesh and Gaussian representations [19], or discard Gaussians occluded by an opaque mesh surface [6, 19]. Although these designs differ in implementation, they reduce the conflict to fixed removal, hard assignment, or binary



Figure 1. **Gaussians Can Subtract.** Existing hybrids primarily compensate for mesh errors by adding Gaussian residuals. **Gaussolve** instead introduces a subtractive Neural Dissolve Field that selectively dissolves inconsistent mesh content and lets residual Gaussians refill the revealed regions. This explicit subtract-and-refill decomposition improves visual quality while retaining a compact hybrid representation. Compared with existing methods, Gaussolve achieves a more favorable quality–compactness trade-off by allowing meshes and Gaussians to cooperate rather than compete.

visibility, none of which captures inconsistencies that vary across viewpoints. In particular, hard visibility allows the mesh to completely suppress residual Gaussians behind it. When the mesh conflicts with the current observation, this suppression also blocks the reconstruction gradients required for the Gaussian branch to correct the inconsistency. Moreover, small depth changes can abruptly switch the mesh–Gaussian visibility state.

We propose Gaussolve, a hybrid Mesh–Gaussian framework that resolves these inconsistencies through a subtract-and-refill process (Fig. 1). Instead of treating the mesh as a fixed explanation and only adding Gaussian residuals around it, Gaussolve allows inconsistent mesh content to be selectively dissolved before residual Gaussians refill the revealed regions. To this end, we introduce a Neural Dissolve Field parameterized by scalar spherical-harmonic Gaussians, which learns where and from which viewing directions the mesh should be attenuated under the visual reconstruction objective. We further integrate mesh dissolution and Gaussian accumulation through continuous transmittance-based fusion, so that suppressing the mesh progressively exposes the residual Gaussians behind it rather than abruptly switching visibility. This preserves their reconstruction gradients and allows them to correct the inconsistencies exposed by mesh dissolution, while mesh regions that remain consistent with the current view are preserved.

Our contributions are summarized as follows:

- We reformulate hybrid Mesh–Gaussian rendering as a multi-view inconsistency problem, moving beyond the purely additive formulation by selectively subtracting inconsistent mesh content before Gaussian refill.
- We introduce a view-dependent Neural Dissolve Field that learns where and from which viewing directions mesh content should be dissolved, while preserving it in views where it remains consistent.
- We propose a continuous transmittance fusion rule that avoids hard visibility discontinuities, preserves reconstruction gradients for residual Gaussians, and enables them to refill the content revealed by mesh dissolution.
- We introduce a progressive optimization curriculum with loss-driven dual densification, further improving optimization stability and performance.

2. Related Work

2.1. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [17] represents scenes using explicit 3D Gaussian primitives and achieves efficient optimization and rendering via differentiable rasterization, offering a strong quality–speed trade-off for novel-view synthesis. Subsequent works improve the Gaussian-only pipeline through densification, pruning, sorting, compression, or reduced memory footprints (e.g., StopThePop [26], Revising Densification [29], Taming 3DGS [23], LightGaussian [8], Mini-Splatting [9], Scaffold-GS [22], Octree-GS [27], and Reduced 3DGS [24]), and recent surveys summarize these branches [2].

2.2. Mesh Priors and Surface Reconstruction

SfM–MVS pipelines such as COLMAP [31, 32], as well as LiDAR and photogrammetry workflows, are widely used to produce point clouds, depth maps, and mesh priors for downstream rendering, editing, and interaction. Surface reconstruction from these inputs (e.g., Poisson reconstruction [16])

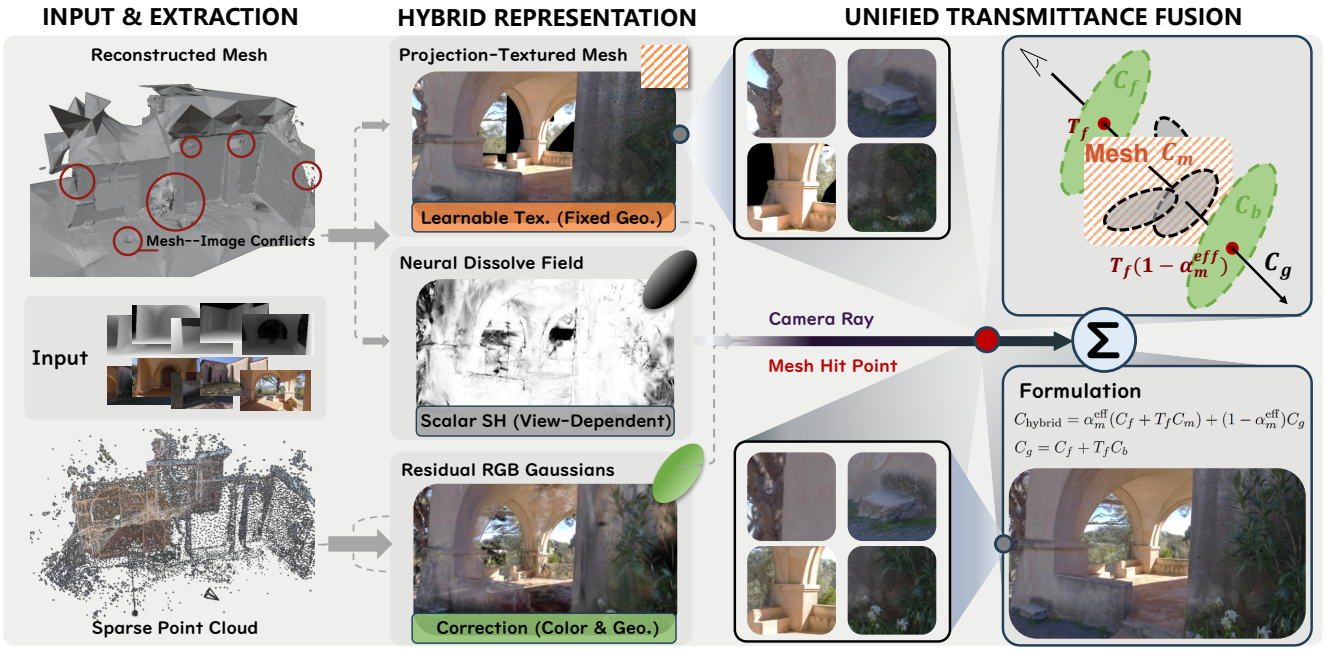


Figure 2. **Overview of Gaussolve.** The projection-textured mesh represents broad regular content, while the Neural Dissolve Field selectively suppresses mesh content that becomes inconsistent across views. Residual Gaussians refill the revealed regions, and all components are combined through continuous transmittance fusion.

remains standard engineering practice, but real-world capture conditions (noise, occlusions, non-uniform sampling) inevitably introduce local geometric errors and topological inconsistencies. As a result, reconstructed meshes often contain holes, self-intersections, or non-manifold connections, which mesh-repair methods have long targeted [1]. Such mesh priors can still disagree with image observations due to geometry errors, occlusion mismatch, or appearance inconsistencies. We therefore treat the mesh as a useful structural prior but not a perfect and strict occluder.

2.3. Hybrid Mesh–Gaussian Rendering

Hybrid representations typically employ a surface mesh to anchor large-scale structures and use Gaussians or surfels to capture thin geometry and appearance residuals. Such designs have been explored for general scene rendering (e.g., MeshGS [6], CityGo [21], HybridMeshGS [15], MeshSplats [35], OMeGa [4], UniMGS [40], and When Gaussian Meets Surfel [42]) and for controllable avatars and heads (e.g., SplattingAvatar [33], GaussianAvatars [25], HAHA [34], MeGA [37], HERA [3], Gaussian Pixel Codec Avatars [12], and relightable Gaussian avatars [30, 38, 43]). Related works such as SuGaR [10], MILo [11], and GaMeS [36] further demonstrate the benefits of explicit surfaces for geometric stability and mesh-in-the-loop optimization. However, many existing scene-reconstruction hybrids rely on two-pass rendering, hard depth-based splitting [21], or heuristically pruned hard occluders [15], which can suppress residual Gaussians, block their reconstruction gradients, or abruptly switch the mesh–Gaussian visibility state. Gaussolve instead formulates visibility fusion as a single continuous transmittance accumulation with a volumetric Neural Dissolve Field, enabling continuous fusion and selective suppression of inconsistent mesh content.

3. Method

As shown in Fig. 2, given posed multi-view images and a textured mesh prior, Gaussolve constructs a hybrid representation with three branches: a textured mesh branch, residual RGB Gaussians, and a Neural Dissolve Field (NDF). The mesh branch provides compact structural and texture evidence for broad regular regions, while residual RGB Gaussians model content that cannot be consistently represented by the mesh, including fine details and view-dependent appearance. Instead of treating the mesh as a hard occluder, Gaussolve uses the NDF to selectively dissolve inconsistent mesh content and continuously exposes residual Gaussians through transmittance-based fusion.

3.1. Textured Mesh Prior

Gaussolve uses a textured mesh prior $\mathcal{M}$ as the mesh branch. We reconstruct the geometry with RealityCapture (RC) [7] and directly project the training images onto the mesh to

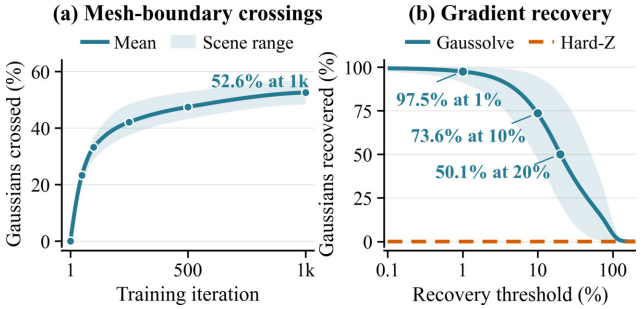


Figure 3. **Hard-Z brittleness and gradient recovery.** (a) Gaussians initially in front of the mesh frequently cross behind the Hard-Z visibility boundary during training. (b) For Gaussians whose reconstruction gradients are blocked by Hard-Z, Gaussolve recovers a substantial fraction of the ungated reference gradient. Curves and shading denote the four-scene mean and range.

obtain its texture. Compared with baking reconstructed appearance from NeRF or Gaussian representations [6, 15], direct projection better preserves observed high-frequency details, as shown in Fig. 7. To avoid test-view leakage, only training views are used for both MVS reconstruction and texturing. Unless otherwise stated, the default meshes are uniformly decimated to 100k faces. The resulting projected texture map defines the mesh color C_m used during fusion.

3.2. Continuous Transmittance Fusion

Hard depth splitting treats the mesh as a binary occluder, removing Gaussian contributions behind it. This cannot resolve multi-view mesh inconsistencies and also blocks corrective gradients to occluded residual Gaussians. Fig. 3 quantifies both the frequent visibility changes and the resulting gradient starvation. We therefore replace hard visibility with a continuous fusion rule.

For exposition, we describe the fusion rule using a ray-wise front/back decomposition and omit repeated pixel and view arguments. Let (C_f, T_f) denote the accumulated Gaussian color and transmittance before the mesh depth along a ray, and let C_b denote the accumulated Gaussian color behind the mesh depth, excluding the front transmittance. The full Gaussian rendering is

$$C_g = C_f + T_f C_b. \tag{1}$$

A direct soft fusion between the mesh color C_m and the Gaussian back segment is

$$C_{\text{hybrid}} = C_f + T_f \left(\alpha_m^{\text{eff}} C_m + (1 - \alpha_m^{\text{eff}}) C_b\right), \tag{2}$$

where $\alpha_m^{\text{eff}} \in [0, 1]$ is the effective mesh opacity. Rather than explicitly recovering C_b through division by T_f, we substitute $T_f C_b = C_g - C_f$ and obtain the division-free form

$$C_{\text{hybrid}} = \alpha_m^{\text{eff}} (C_f + T_f C_m) + (1 - \alpha_m^{\text{eff}}) C_g. \tag{3}$$

Thus $\alpha_m^{\text{eff}} = 1$ preserves the mesh evidence, while $\alpha_m^{\text{eff}} = 0$ falls back to Gaussian rendering. In practice, the rasterized mesh depth is used only to compute the front/back Gaussian partition; dissolving the mesh changes only the final blending weight, not the mesh geometry.

3.3. Neural Dissolve Field

Eq. (3) requires estimating the effective mesh opacity $\alpha_m^{\text{eff}}(\mathbf{u})$. A natural alternative is to learn this opacity directly on the mesh surface, either as per-vertex alpha or as a texture-map alpha. However, such surface-alpha fields are limited in two ways.

First, surface alpha is view-invariant, while mesh inconsistencies vary with viewpoint. In geometry and occlusion conflicts, a false protrusion, hole, or inaccurate boundary may block the correct visual content only from certain viewpoints. Similarly, the textured mesh branch has fixed appearance and cannot selectively hand off highlights and reflection-like residuals to residual RGB Gaussians from specific directions.

Second, surface alpha is limited by its parameterization. Per-vertex alpha is tied to mesh tessellation, while texture-map alpha requires indirect view-space-to-texture-space optimization and can produce fragmented masks. We therefore model these inconsistencies with a volumetric, view-dependent Neural Dissolve Field.

The NDF is represented by dissolve Gaussians $\mathcal{D}$. Each dissolve Gaussian stores standard Gaussian geometry and opacity parameters, but predicts a scalar dissolve value instead of RGB color. For viewing direction $\mathbf{v}$, the i-th dissolve Gaussian predicts

$$c_{d,i}(\mathbf{v}) = \sigma(\text{SH}_{d,i}(\mathbf{v})) \in [0, 1], \tag{4}$$

where $\text{SH}_{d,i}$ denotes a scalar spherical-harmonic function attached to the i-th dissolve Gaussian.

We rasterize dissolve Gaussians using the same Gaussian rasterizer. Let $\hat{\alpha}_{d,i}(\mathbf{u})$ denote the effective per-pixel opacity of the i-th dissolve Gaussian at pixel $\mathbf{u}$, sorted in front-to-back order. The dissolve value is composited as

$$D(\mathbf{u}) = \sum_i T_{d,i}(\mathbf{u})\, \hat{\alpha}_{d,i}(\mathbf{u})\, c_{d,i}(\mathbf{v}), \tag{5}$$

with

$$T_{d,i}(\mathbf{u}) = \prod_{j<i} \left(1 - \hat{\alpha}_{d,j}(\mathbf{u})\right). \tag{6}$$

Although written as $D(\mathbf{u})$, the dissolve value is view-dependent through $c_{d,i}(\mathbf{v})$. The effective mesh opacity used in Eq. (3) is

$$\alpha_m^{\text{eff}}(\mathbf{u}) = 1 - D(\mathbf{u}). \tag{7}$$

Thus, $D(\mathbf{u}) = 0$ preserves the mesh contribution, while $D(\mathbf{u}) = 1$ fully dissolves it. The dissolve branch does not emit radiance. It only attenuates mesh evidence in the final fusion, allowing residual RGB Gaussians to refill the revealed content.

Table 1. **Main comparison on structured and semi-structured scenes.** The target regime contains broad regular structures where a fixed-budget textured mesh can replace redundant Gaussian primitives. **Bold** indicates the best performance, and cell colors highlight the top , second-best , and third-best results .

Method	Mesh	Hybrid	ML-Hypersim-AI (Structured)					Deep Blending (Semi-Structured)				
			PSNR↑	SSIM↑	LPIPS↓	ΔGS↑	Aux.%	PSNR↑	SSIM↑	LPIPS↓	ΔGS↑	Aux.%
3DGS	×	×	34.39	0.959	0.155	0%	–	28.50	0.877	0.272	0%	–
Reduced 3DGS	×	×	34.04	0.957	0.153	62%	–	28.43	0.875	0.277	51%	–
3DGS-MCMC	×	×	35.82	0.960	0.140	-1%	–	28.56	0.875	0.268	-12%	–
SuGaR	✓	×	32.29	0.950	0.149	33%	–	27.58	0.858	0.306	19%	–
PGSR	✓	×	32.95	0.958	0.147	61%	–	28.44	0.863	0.302	28%	–
MeshSplatting	✓	×	28.47	0.903	0.236	–	–	24.71	0.809	0.359	–	–
3D-GES	×	✓	34.87	0.961	0.130	63%	13%	28.42	0.875	0.262	41%	17%
Hard-Z Proxy	✓	✓	34.31	0.958	0.119	27%	–	28.32	0.872	0.234	39%	–
Ours (3DGS)	✓	✓	35.35	**0.962**	**0.108**	30%	7%	28.75	0.876	0.228	47%	8%
Ours (Reduced 3DGS)	✓	✓	35.29	0.959	0.113	**67%**	8%	28.67	0.877	0.232	**65%**	13%
Ours (3DGS-MCMC)	✓	✓	**36.89**	0.961	0.110	-6%	4%	**28.93**	**0.880**	**0.227**	-3%	4%

Table 2. **Comparison with MeshGS [6] and Hybrid Mesh-Gaussian [15] on the Deep Blending dataset.** For the baselines, we report the **3DGS-based** results directly from the respective papers.

Method	Mesh	Hybrid	DrJohnson				Playroom				Average			
			PSNR↑	SSIM↑	LPIPS↓	ΔGS↑	PSNR↑	SSIM↑	LPIPS↓	ΔGS↑	PSNR↑	SSIM↑	LPIPS↓	ΔGS↑
MeshGS	✓	✓	28.80	**0.900**	0.240	–	30.50	0.900	0.250	–	29.65	**0.900**	0.245	∼30%
Hybrid Mesh-Gaussian	✓	✓	29.01	0.888	0.262	43%	30.11	**0.901**	0.255	30%	29.56	0.895	0.259	36%
Ours	✓	✓	**29.69**	0.894	**0.240**	**58%**	**30.93**	0.893	**0.223**	**61%**	**30.31**	0.894	**0.232**	**59.5%**

3.4. Optimization Curriculum and Loss-Driven Dual Densification

We progressively introduce the three components instead of optimizing the full hybrid representation from the beginning. 1) We first optimize the residual Gaussian branch alone using the standard 3DGS objective, which provides a stable initialization for the subsequent hybrid optimization. 2) We then introduce the projection-textured mesh and NDF and enable continuous transmittance fusion. During hybrid optimization, we freeze the mesh texture and employ loss-driven dual densification: First, we sample high-error pixels from the current fused rendering. These pixels are back-projected into 3D, with mesh depth serving as a coarse geometric anchor, to add candidates to both the dissolve branch and the residual RGB branch. The former increases the capacity to dissolve inconsistent mesh content, while the latter increases the capacity to refill the revealed regions. All inserted candidates remain fully trainable and are not constrained to the mesh surface. 3) Finally, after Gaussian densification terminates, we continue optimizing the Gaussian representation for a short stabilization period before unfreezing the mesh texture and jointly fine-tuning all components.

The main Gaussian branch follows the same global densification window as standard 3DGS. For plug-in variants, each Gaussian backbone retains its update strategy. Dissolve Gaussians are initialized according to multi-view mesh-texel errors. Additional details of the initialization and dual densification are provided in the supplementary material.

4. Experiments

4.1. Setup

Datasets and metrics. We evaluate on ML-Hypersim-AI [28], Deep Blending [13]. These datasets cover structured indoor scenes, semi-structured indoor/outdoor scenes, and foliage-heavy unstructured scenes. We report PSNR and SSIM [39] for distortion and structural fidelity, LPIPS [44] for perceptual similarity, Gaussian reduction $\Delta GS = (N_{\text{3DGS}} - N_{\text{method}})/N_{\text{3DGS}}$, and Aux.% as the fraction of auxiliary Gaussians within total Gaussians. Stress tests on unstructured scenes are provided in the supplementary material.

Baselines. We compare against Gaussian-only, surface-oriented, and hybrid baselines. For Gaussian-only methods, we include 3DGS [17], 3DGS-MCMC [18], and Reduced 3DGS [24]. For surface-oriented reconstruction, we compare with SuGaR [10], PGSR [5], and MeshSplatting [14]. We further include 3D-GES [42] as a compact hybrid baseline. For a controlled comparison with hard depth-based fusion, we instantiate a Hard-Z Proxy with the same textured mesh prior as Gaussolve. Since official implementations of representative Mesh–Gaussian hybrid scene-reconstruction meth-

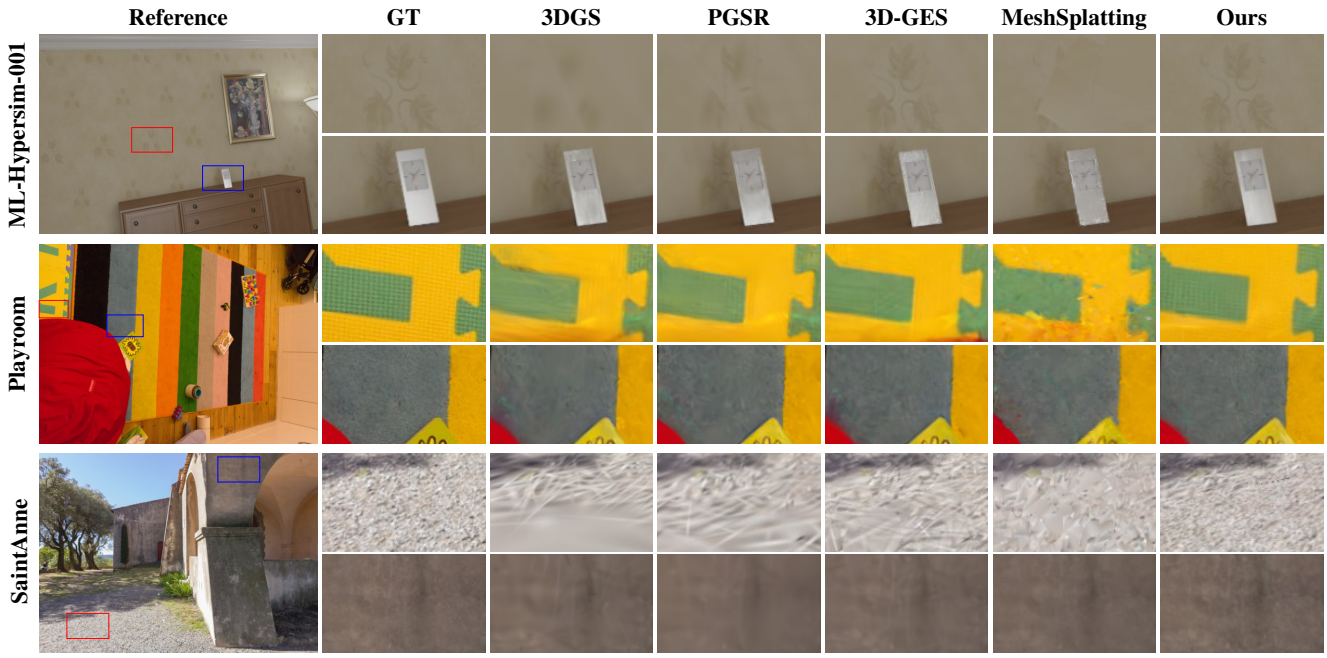


Figure 4. **Qualitative comparisons on novel-view synthesis.** Compared with 3DGS, PGSR, 3D-GES, and MeshSplatting, Gaussolve better preserves weakly textured surfaces. The red and blue boxes indicate the corresponding enlarged regions.

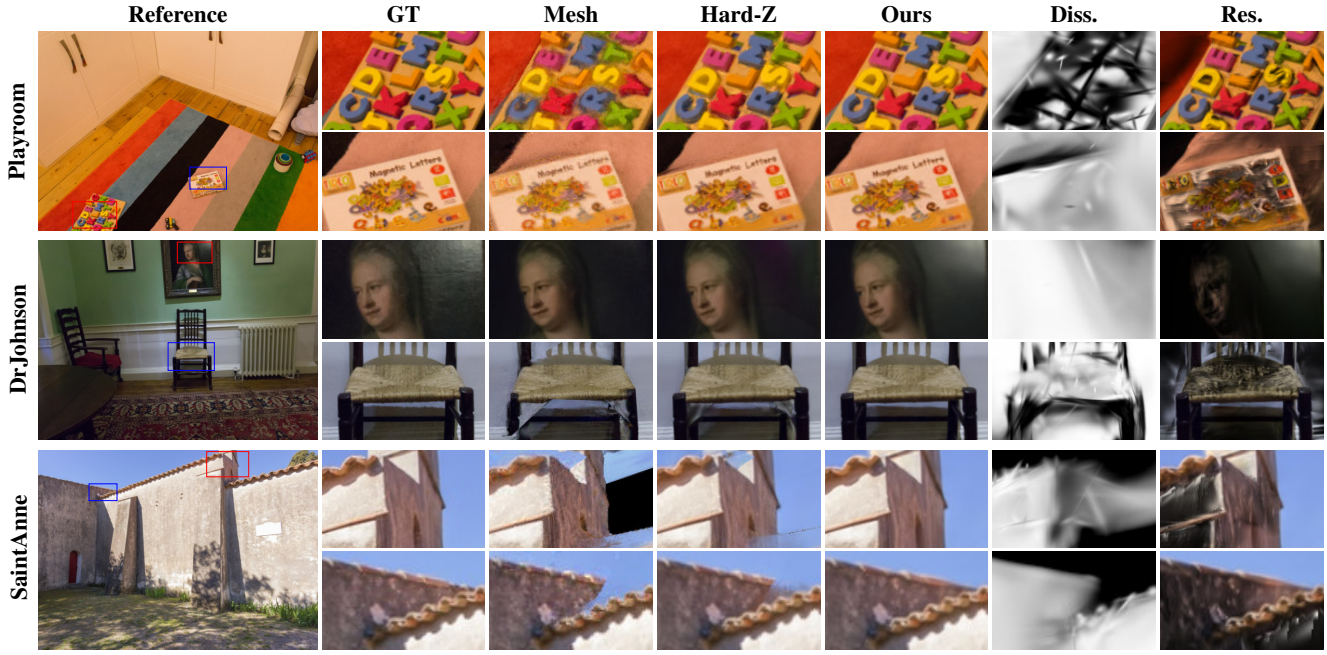


Figure 5. **Visualization of the subtract-and-refill mechanism.** Compared with the projection-textured mesh and the Hard-Z hybrid baseline, Gaussolve selectively dissolves inconsistent mesh content (Diss.) and lets residual Gaussians refill the revealed regions (Res.), producing cleaner boundaries and more accurate appearance. The red and blue boxes indicate the corresponding enlarged regions.

ods such as MeshGS [6] and Hybrid Mesh-Gaussian [15] are unavailable, we report the published Deep Blending results in Tab. 2. To evaluate its compatibility with different Gaussian backbones, we further plug Gaussolve into 3DGS-MCMC [18] and Reduced 3DGS [24]. Detailed implementations are provided in the supplementary material.

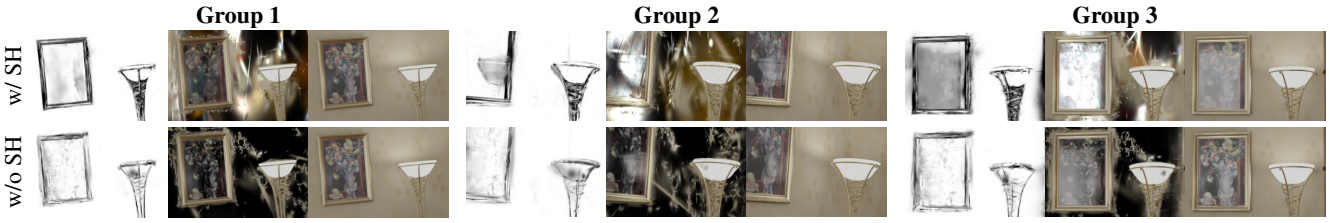


Figure 6. **Directional dissolve for view-dependent appearance.** A view-invariant dissolve field can suppress some static geometry errors, but it cannot provide a directional handoff for view-specific evidence. The scalar-SH dissolve field attenuates mesh evidence only from conflict-prone viewing directions, allowing residual RGB Gaussians to recover the corresponding view-dependent details.

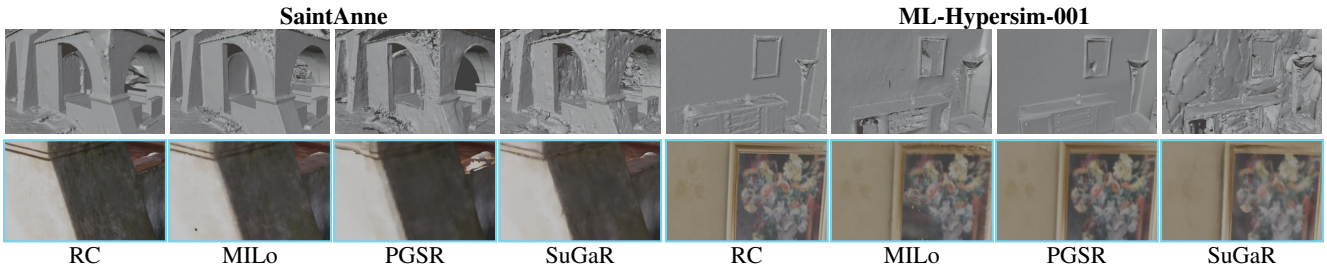


Figure 7. **Textured mesh priors and direct image projection.** We compare RC, MILo, PGSR, and SuGaR mesh priors on SaintAnne and ML-Hypersim-001. The top row visualizes the reconstructed mesh geometry, and the bottom row shows enlarged textured renderings. Direct projection preserves sharper local appearance than appearance baked from reconstructed representations.

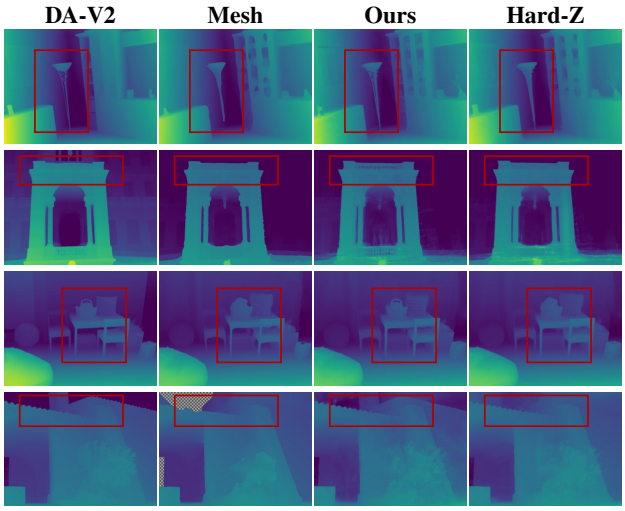


Figure 8. **Rendered-depth comparison.** Aligned Depth Anything V2 (DA-V2) reference [41], mesh depth, Gaussolve, and Hard-Z. Red boxes mark representative conflict regions.

All experiments are run on an NVIDIA RTX 5000 Ada GPU.

4.2. Quantitative and Qualitative Comparisons

Table 1 summarizes the results on structured and semi-structured scenes. Across all three Gaussian backbones, integrating Gaussolve consistently improves reconstruction quality, showing that the proposed fusion is not tied to a specific Gaussian optimization strategy. The 3DGS-MCMC variant reaches the highest PSNR on both datasets, while the Reduced 3DGS variant provides a stronger quality–compactness trade-off with substantially fewer Gaussians. The MCMC variants show slightly negative ΔGS because 3DGS-MCMC removes the opacity reset and opacity-based pruning used by vanilla 3DGS. We retain its original update strategy as our goal here is to evaluate whether the fusion framework improves the backbone without changing its native optimization behavior. Compared with the Hard-Z Proxy under the same textured-mesh setting, Gaussolve consistently improves reconstruction quality, confirming the benefit of continuous subtract-and-refill fusion over hard visibility. Table 2 further compares against MeshGS and Hybrid Mesh-Gaussian on the two commonly reported Deep Blending scenes. Gaussolve achieves higher average PSNR and substantially lower LPIPS while reducing the Gaussian population by 59.5%, compared with approximately 30% for MeshGS and 36% for Hybrid Mesh-Gaussian.

4.3. Subtract-and-Refill Analysis

Gaussolve separates complementary roles between the textured mesh and the dissolve-and-refill branches. Fig. 5 visualizes this learned division of labor: the mesh branch explains regular structures, the dissolve field concentrates on local mesh–image conflicts, and residual RGB Gaussians allocate capacity to the revealed regions and remaining fine details. This decomposition supports the subtract-and-refill interpretation: Gaussolve selectively attenuates inconsistent

Table 3. **Efficiency breakdown.** Average rendering efficiency, storage, and memory on the Deep Blending dataset.

	3DGS			Reduced 3DGS		
Component	Storage	FPS	Mem.	Storage	FPS	Mem.
Mesh Only	0.069G	1003	0.98G	0.069G	988	0.98G
Dissolve GS Only	0.016G	1110	0.09G	0.016G	1111	0.09G
Residual GS Only	0.251G	265	0.77G	0.134G	463	0.49G
GS Baseline	0.655G	99	**1.64G**	**0.071G**	**311**	**0.48G**
Ours	**0.336G**	168	1.70G	0.219G	236	1.42G
Ours (Parallel)	**0.336G**	**174**	1.71G	0.219G	241	1.43G

Table 4. **Ablation study on ML-Hypersim-001.** Ablations of representation components, training design, and mesh geometry/appearance.

Variant	Faces	PSNR↑	SSIM↑	LPIPS↓
A. Component Ablation (RC)				
Mesh Only	–	27.97	0.894	0.249
Residual GS Only	–	34.25	0.960	0.153
Mesh + Residual GS (Hard-Z)	–	33.51	0.932	0.197
Mesh + NDF	–	29.76	0.896	0.241
Full	–	**35.21**	**0.963**	**0.106**
B. Training-Design Ablation (RC)				
w/o Scalar SH	–	34.64	0.954	0.116
w/o Dual Densification	–	34.80	0.957	0.115
w/o Progressive Initialization	–	35.02	0.959	0.109
C. Geometry and Appearance Ablation (PGSR)				
PGSR + Projection	10K	35.52	0.963	0.112
PGSR + Projection	50K	35.56	0.964	**0.111**
PGSR + Projection	100K	**35.61**	0.964	0.112
PGSR + Projection	1M	35.57	**0.965**	0.112
PGSR + Baked Color	100K	35.42	0.961	0.140

Table 5. **Depth consistency with the aligned DA-V2 reference.** We report the standard AbsRel, RMSE, and δ_1 metrics used by Depth Anything V2 [41], together with Boundary F1 to measure depth-discontinuity alignment. Boundary F1 uses a shared Sobel threshold and a two-pixel matching tolerance.

Method	AbsRel↓	RMSE↓	δ_1 ↑	Boundary F1↑
Mesh	0.0787	0.0909	0.9725	0.5274
Hard-Z	0.0868	0.0845	0.9736	0.6720
Ours	**0.0680**	**0.0785**	**0.9751**	**0.6814**

mesh content and refills the exposed regions, rather than simply adding another Gaussian layer on top of the mesh.

To further examine the subtract-and-refill behavior, Fig. 8 compares the aligned DA-V2 reference with the mesh depth and rendered depths from Hard-Z and Gaussolve. Hard-Z largely follows the mesh depth, whereas Gaussolve suppresses inconsistent mesh evidence and allows residual Gaussians to explain the revealed regions. Table 5 further shows that Gaussolve achieves the best depth consistency and boundary accuracy across all four metrics.

Fig. 6 isolates the view-dependent part of the dissolve field. The View-invariant variant, without scalar SH, cannot follow direction-specific appearance residuals. With scalar SH, it can attenuate mesh content only from conflicting viewing directions, allowing residual RGB Gaussians to recover highlights and reflection-like details when needed.

Table 6. **Training overhead.** Training time and peak GPU memory for Gaussolve with different Gaussian backbones.

Method	3DGS		Reduced 3DGS		3DGS-MCMC		Mesh
	Time	Mem.	Time	Mem.	Time	Mem.	Mem.
Baseline	32.5min	3.67G	24.5min	1.51G	36.0min	4.11G	–
Ours	32.9min	4.91G	34.1min	4.78G	53.1min	6.95G	3.95G

4.4. Ablations

Table 4 summarizes the ablations on ML-Hypersim-001. The component and training-design ablations use the RC mesh. The full model achieves 35.21 PSNR and 0.106 LPIPS, outperforming the residual-GS-only and Hard-Z variants. Removing scalar SH, dual densification, or progressive initialization also degrades reconstruction quality, validating the proposed training design. For the geometry and appearance ablation, we use the PGSR mesh. Performance remains stable from 10K to 1M faces, while at around 100K faces, direct projection reduces LPIPS from 0.140 to 0.112 compared with baked appearance while maintaining similar PSNR and SSIM. Furthermore, the consistent performance across various resolutions and appearance constructions indicates that Gaussolve does not depend on a narrowly tuned mesh configuration.

4.5. Rendering and Training Efficiency

Table 3 reports the average efficiency. With the 3DGS backbone, Ours (Parallel) improves rendering speed from 99 to 174 FPS while reducing storage from 0.655G to 0.336G. Replacing the Gaussian backbone with Reduced 3DGS further increases our rendering speed to 241 FPS and reduces storage to 0.219G. Although the hybrid remains slower than standalone Reduced 3DGS due to the additional mesh and NDF rendering overhead, it achieves substantially better reconstruction quality. Introducing the mesh also incurs additional training cost (Table 6), but the overhead remains within a manageable range, as the mesh itself has a nearly constant memory footprint independent of the Gaussian population.

5. Conclusion

We presented Gaussolve, a hybrid Mesh–Gaussian renderer that reformulates representation conflicts as multi-view inconsistencies. Gaussolve selectively dissolves inconsistent mesh content with a Neural Dissolve Field and lets residual Gaussians refill the revealed regions. Experiments on structured and semi-structured scenes demonstrate improved visual quality across different Gaussian backbones, together with substantial Gaussian reduction and real-time rendering.

References

[1] Marco Attene. A lightweight approach to repairing digitized polygon meshes. *The Visual Computer*, 26(11):1393–1406, 2010. 3

[2] Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, and Jiebo Luo. 3d gaussian splatting: Survey, technologies, challenges, and opportunities. *arXiv preprint arXiv:2407.17418*, 2024. 2

[3] Hongrui Cai, Yuting Xiao, Xuan Wang, Jiafei Li, Yudong Guo, Yanbo Fan, Shenghua Gao, and Juyong Zhang. Hera: Hybrid explicit representation for ultra-realistic head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 260–270, 2025. 3

[4] Yuhang Cao, Haojun Yan, and Danya Yao. Omega: Joint optimization of explicit meshes and gaussian splats for robust scene-level surface reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4386–4395, 2026. 3

[5] 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. 5

[6] Jaehoon Choi, Yonghan Lee, Hyungtae Lee, Heesung Kwon, and Dinesh Manocha. Meshgs: Adaptive mesh-aligned gaussian splatting for high-quality rendering. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 3310–3326, 2024. 1, 3, 4, 5, 6

[7] Epic Games. RealityCapture. https://www.capturingreality.com/, 2024. 3

[8] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. In *Advances in Neural Information Processing Systems*, pages 140138–140158, 2024. 1, 2

[9] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *Computer Vision – ECCV 2024*, pages 165–181, 2024. 2

[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 (CVPR)*, pages 5354–5363, 2024. 3, 5

[11] 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*, 44(6):1–15, 2025. 3

[12] Divam Gupta, Anuj Pahuja, Nemanja Bartolovic, Tomas Simon, Forrest Iandola, and Giljoo Nam. Gaussian pixel codec avatars: A hybrid representation for efficient rendering. *CoRR*, abs/2512.15711, 2025. 3

[13] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics*, 37(6):257:1–257:15, 2018. 5

[14] Jan Held, Sanghyun Son, Renaud Vandeghen, Daniel Rebain, Matheus Gadelha, Yi Zhou, Anthony Cioppa, Ming C. Lin, Marc Van Droogenbroeck, and Andrea Tagliasacchi. Meshsplatting: Differentiable rendering with opaque meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7320–7329, 2026. 1, 5

[15] Binxiao Huang, Zhihao Li, Shiyong Liu, Xiao Tang, Jiajun Tang, Jiaqi Lin, Yuxin Cheng, Zhenyu Chen, Xiaofei Wu, and Ngai Wong. Hybrid mesh-gaussian representation for efficient indoor scene reconstruction. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)*, pages 1143–1151, 2025. 1, 3, 4, 5, 6

[16] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, pages 61–70, 2006. 2

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. 1, 2, 5

[18] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. In *Advances in Neural Information Processing Systems*, pages 80965–80986, 2024. 5, 6

[19] Jiyeop Kim and Jongwoo Lim. Integrating meshes and 3D gaussians for indoor scene reconstruction with SAM mask guidance. *arXiv preprint arXiv:2407.16173*, 2024. 1

[20] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21719–21728, 2024. 1

[21] Weihang Liu, Yuhui Zhong, Yuke Li, Xi Chen, Jiadi Cui, Honglong Zhang, Lan Xu, Xin Lou, Yujiao Shi, Jingyi Yu, and Yingliang Zhang. Citygo: Lightweight urban modeling and rendering with proxy buildings and residual gaussians. In *SIGGRAPH Asia 2025 Conference Papers*, 2025. 3

[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 (CVPR)*, 2024. 2

[23] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Francisco Vicente Carrasco, Markus Steinberger, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. 2

[24] Panagiotis Papantonakis, Georgios Kopanas, Bernhard Kerbl, Alexandre Lanvin, and George Drettakis. Reducing the memory footprint of 3d gaussian splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 7 (1):1–17, 2024. 2, 5, 6

[25] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20299–20309, 2024. 3

[26] Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. Stopthepop: Sorted gaussian splatting for view-consistent real-time rendering. *ACM Transactions on Graphics*, 43(4):1–17, 2024. 2

[27] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. 2

[28] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10912–10922, 2021. 5

[29] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in gaussian splatting. In *Computer Vision – ECCV 2024*, pages 347–362, 2024. 2

[30] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. *arXiv preprint arXiv:2312.03704*, 2023. 3

[31] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 2

[32] Johannes Lutz Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision – ECCV 2016*, pages 501–518. Springer, 2016. 2

[33] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1606–1616, 2024. 3

[34] David Svitov, Pietro Morerio, Lourdes Agapito, and Alessio Del Bue. Haha: Highly articulated gaussian human avatars with textured mesh prior. In *Computer Vision – ACCV 2024*, pages 105–122, 2024. 3

[35] Rafał Tobiasz, Grzegorz Wilczyński, Marcin Mazur, Sławomir Tadeja, and Przemysław Spurek. Meshsplats: Mesh-based rendering with gaussian splatting initialization. *arXiv preprint arXiv:2502.07754*, 2025. 3

[36] Joanna Waczyńska, Piotr Borycki, Sławomir Tadeja, Jacek Tabor, and Przemysław Spurek. GaMeS: Mesh-based adapting and modification of gaussian splatting. *arXiv preprint arXiv:2402.01459*, 2024. 3

[37] Cong Wang, Di Kang, Heyi Sun, Shenhan Qian, Zixuan Wang, Linchao Bao, and Song-Hai Zhang. Mega: Hybrid mesh-gaussian head avatar for high-fidelity rendering and head editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26274–26284, 2025. 3

[38] Shaofei Wang, Tomas Simon, Igor Santesteban, Timur Bagautdinov, Junxuan Li, Vasu Agrawal, Fabian Prada, Shoou-I Yu, Pace Nalbone, Matt Gramlich, Roman Lubachersky, Chenglei Wu, Javier Romero, Jason Saragih, Michael Zollhoefer, Andreas Geiger, Siyu Tang, and Shunsuke Saito. Relightable full-body gaussian codec avatars. *arXiv preprint arXiv:2501.14726*, 2025. 3

[39] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[40] Zeyu Xiao, Mingyang Sun, Yimin Cong, Lintao Wang, Dongliang Kou, Zhenyi Wu, Dingkang Yang, Peng Zhai, Zeyu Wang, and Lihua Zhang. Unimgs: Unifying mesh and 3d gaussian splatting with single-pass rasterization and proxy-based deformation. *arXiv preprint arXiv:2601.19233*, 2026. 3

[41] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *Advances in Neural Information Processing Systems*, 2024. 7, 8

[42] Keyang Ye, Tianjia Shao, and Kun Zhou. When gaussian meets surfel: Ultra-fast high-fidelity radiance field rendering. *ACM Transactions on Graphics*, 44(4), 2025. 3, 5

[43] Youyi Zhan, Tianjia Shao, He Wang, Yin Yang, and Kun Zhou. Interactive rendering of relightable and animatable gaussian avatars. *arXiv preprint arXiv:2407.10707*, 2024. 3

[44] 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 (CVPR)*, pages 586–595, 2018. 5

[45] Qingyuan Zhou, Yuehu Gong, Weidong Yang, Jiaze Li, Yeqi Luo, Baixin Xu, Shuhao Li, Ben Fei, and Ying He. MGSR: 2d/3d mutual-boosted gaussian splatting for high-fidelity surface reconstruction under various light conditions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27295–27304, 2025. 1