



Article

Object-Aware Loss for Mask-Guided Mesh Reconstruction Based on 2D Gaussian Splatting

Sunghyun Woo, Jeongil Seo and HeeKyung Lee

Special Issue

Advances in Mathematical Methods for Image Processing and Computer Vision

Edited by
Prof. Dr. Hongwei Mo



Article

Object-Aware Loss for Mask-Guided Mesh Reconstruction Based on 2D Gaussian Splatting

Sunghyun Woo ¹, Jeongil Seo ^{1,*} and HeeKyung Lee ²¹ Department of Computer Engineering, Dong-A University, Busan 49315, Republic of Korea; 2242798@donga.ac.kr² Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea; lhk95@etri.re.kr

* Correspondence: jeongilseo@dau.ac.kr; Tel.: +82-51-200-7796

Abstract

Gaussian Splatting has emerged as a powerful representation for high-fidelity novel view synthesis. However, existing methods typically optimize the entire scene and require postprocessing to obtain object-level meshes. This process incurs unnecessary computational costs and can introduce background artifacts into the resulting object mesh. To address this problem, we propose an object-aware loss for mask-guided mesh reconstruction based on 2D Gaussian Splatting, which combines three complementary terms. First, the Masked L1 Loss restricts photometric supervision to the target-object region. Second, the Alpha Polarization Loss encourages high accumulated alpha values on the object and low values on the background. Third, the ObjectMark Polarization Loss supervises a learnable ObjectMark score, and Gaussians with scores at or below a predefined threshold are periodically pruned during optimization. By suppressing background-contributing Gaussians and focusing optimization on the target object, the proposed method enables efficient object-level reconstruction with fewer background artifacts. Among the compared methods, the proposed method achieves the highest surface reconstruction accuracy. It reduces the Chamfer Distance on the DTU dataset from 0.92 to 0.74, a 19.6% improvement, while producing cleaner object-level meshes. These results show that the proposed method enables direct and efficient object-level mesh reconstruction while reducing the need for postprocessing.

Keywords: 3D reconstruction; Gaussian Splatting; mesh extraction; novel view synthesis; surface reconstruction

MSC: 68U05



Academic Editor: Jonathan Blackledge

Received: 10 August 2026

Revised: 4 September 2026

Accepted: 4 September 2026

Published: 6 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Recent digital content has evolved beyond high-resolution visual quality toward immersive experiences such as augmented reality and virtual reality, where scenes can be observed from diverse viewpoints rather than from a fixed viewing direction. However, conventional visual media including standard video, panoramic imaging, and stereo-based capture provide only a limited set of recorded viewpoints. As a result, users can observe the scene only from captured views, and visual information from uncaptured viewpoints is not directly available. This limitation restricts free scene exploration and the synthesis of visual content from unseen viewpoints. To address this limitation, 3D reconstruction has attracted increasing attention. By reconstructing scenes and generating views beyond

the originally captured viewpoints, 3D reconstruction techniques enable users to observe scenes from desired viewpoints and support more flexible visual experiences.

With the growing interest in 3D reconstruction, Neural Radiance Fields (NeRF) [1] have emerged as a pioneering method for high-fidelity novel view synthesis (NVS). NeRF achieves photorealistic rendering through volumetric integration by representing a scene as a continuous radiance field. Specifically, NeRF uses a neural network to predict density and color from 3D positions and viewing directions, and synthesizes images by integrating these values along camera rays. However, NeRF-based methods incur high computational cost in both training and rendering due to dense ray sampling and volumetric integration. These limitations reduce their practicality in applications that require efficient optimization and fast rendering.

To overcome these limitations, 3D Gaussian Splatting (3DGS) [2] recently emerged as an efficient explicit representation for high-quality NVS. Instead of representing a scene with a neural implicit field, 3DGS models the scene using explicit Gaussians. Each Gaussian is parameterized by learnable attributes, including position, scale, rotation, color, and alpha. By representing a scene with explicit 3D Gaussians, 3DGS enables efficient rendering while achieving high visual quality and real-time performance. Owing to these advantages, Gaussian Splatting has recently been extended beyond NVS toward object-level mesh extraction for 3D asset reconstruction, editable object representation, object-level scene manipulation, and generation-ready 3D content creation [3,4]. Despite these advances, most reconstruction methods based on Gaussian Splatting are optimized using loss functions defined over the entire input image. This full-scene optimization loss is effective for complete scene rendering; however, it becomes inefficient when the target task is object-level mesh extraction. Existing Gaussian Splatting methods optimize the entire scene first and then obtain the target-object mesh through postprocessing. Since Gaussians are optimized over both the target object and background regions during training, many background Gaussians are retained along with the target object. Consequently, a larger number of Gaussians must be optimized, which increases training time and GPU memory usage. More importantly, background Gaussians can introduce unwanted geometry and artifacts during mesh extraction, making the resulting object mesh noisy and less suitable for direct use as a 3D asset.

To address these limitations, we propose an object-aware loss for mask-guided mesh reconstruction based on 2D Gaussian Splatting (2DGS) [5]. The object-aware loss uses object masks as priors, encouraging Gaussians to contribute primarily to object regions. The loss consists of three main components. First, the Masked L1 Loss restricts photometric supervision to the target-object region by excluding background pixels from the reconstruction error. Second, the Alpha Polarization Loss regularizes the accumulated alpha response according to the object mask. It encourages the accumulated alpha to approach one in object regions and zero in background regions. Through this loss, Gaussians contributing to the target object are encouraged to maintain high alpha, whereas background-contributing Gaussians are suppressed. Third, the ObjectMark Polarization Loss guides the learnable ObjectMark score of each Gaussian. The ObjectMark score represents the degree to which a Gaussian is associated with the target object. The accumulated ObjectMark response is optimized toward one in object regions and toward zero in background regions. Gaussians with low ObjectMark scores are regarded as background-related primitives and are removed during optimization and before mesh extraction. This optimization strategy reduces unnecessary background learning and limits the propagation of background geometry into the extracted mesh. Experimental results show that the proposed approach improves mesh reconstruction quality while reducing artifacts caused by background Gaussians. By focusing optimization on the target object, the approach also achieves lower computational

cost compared with existing mesh reconstruction methods. These results indicate that the approach can extract clean object-level meshes without relying on complex postprocessing.

The main contributions of this work are summarized as follows:

- We propose an object-aware loss for mask-guided mesh reconstruction based on 2DGS.
- We introduce a Masked L1 Loss that restricts photometric supervision to the target-object region by excluding background pixels from the photometric loss.
- We introduce Alpha Polarization Loss and ObjectMark Polarization Loss, which regularize the accumulated alpha response and learnable ObjectMark scores, respectively, enabling effective object-background separation.
- We introduce ObjectMark Pruning, which removes Gaussians with low ObjectMark scores during optimization, thereby reducing background Gaussians before mesh extraction.
- We show that the proposed loss improves object-level mesh quality and computational efficiency compared with existing mesh reconstruction methods, while reducing the need for complex postprocessing.

2. Related Work

2.1. Neural Radiance Fields

NeRF performs NVS from posed multi-view images by representing a scene as a continuous volumetric radiance field. This field is parameterized by a neural network that maps 3D coordinates and viewing directions to volume density and view-dependent color. The scene representation is optimized through differentiable volume rendering by minimizing a photometric reconstruction loss between rendered and ground-truth images. This formulation enables high-quality synthesis from unseen viewpoints, but generally requires dense sampling along camera rays and repeated network evaluations during both training and rendering. Subsequent methods have improved the efficiency of radiance field optimization by reducing network evaluations, introducing compact spatial representations, and employing explicit or hybrid scene structures [6–9]. Although these approaches substantially improve rendering efficiency, many NeRF-based pipelines still rely on per-scene optimization, volumetric integration, or additional geometric processing. Radiance fields have also been extended toward object-level tasks, for example by incorporating channel attention to improve 3D object detection [10], although such methods still operate on a field that has already been fitted to the entire scene. Moreover, these photometric losses are evaluated over the entire image and do not distinguish task-relevant object regions from background, which is inefficient when only object-level reconstruction is required.

2.2. 3D Gaussian Splatting

3DGS enables real-time NVS through an efficient explicit scene representation. Unlike neural implicit methods, 3DGS represents a scene using anisotropic Gaussians initialized from sparse structure-from-motion (SfM) [11] points, each carrying learnable position, scale, rotation, color, and alpha attributes. These primitives are projected into screen space and accumulated through differentiable alpha compositing, enabling efficient optimization and high-quality rendering without dense volumetric ray sampling. Several Gaussian Splatting methods have improved rendering quality, optimization stability, and computational efficiency through adaptive density control, regularization, anti-aliasing, more effective primitive management, and refined initialization from auxiliary sensing [12–16]. Gaussian primitives have also been paired with complementary representations in other capture settings, such as light field view reconstruction [17]. However, these improvements generally preserve the full-scene optimization loss. Therefore, even when the Gaussian

representation becomes more accurate or computationally efficient, it may still allocate model capacity to background regions that are unnecessary for object-level reconstruction.

2.3. Mesh Extraction

Mesh extraction is an important process for converting reconstructed 3D representations into explicit surface models that can be used in graphics, editing, simulation, and asset generation pipelines [18,19]. In NeRF-based reconstruction, meshes are commonly extracted by evaluating a learned density or signed distance field and applying an isosurface extraction algorithm such as Marching Cubes [20]. However, density-based NeRF representations are primarily optimized for view synthesis rather than explicit surface recovery, which can lead to noisy or ambiguous geometry. To improve surface reconstruction, several methods represent scene geometry with continuous surface formulations rather than density fields [21], among which signed distance fields (SDFs) have been the most widely adopted [22–25]. NeuS [26] represents surfaces as the zero-level set of an SDF and introduces a volume rendering formulation designed for multi-view surface reconstruction. VolSDF [27] also connects volume rendering with implicit surface modeling by converting a learnable SDF into volume density, enabling surface reconstruction from posed images. Neuralangelo [28] further improves large-scale dense reconstruction by combining neural SDFs with multi-resolution hash grids and numerical gradients. These methods improve geometric fidelity compared with density-only radiance fields, but extracting high-resolution meshes still requires dense spatial evaluation of implicit functions, which can be computationally expensive.

For Gaussian Splatting, mesh extraction poses a difficult challenge. Although 3DGS provides fast rendering through explicit Gaussians, these Gaussians are optimized mainly for image reconstruction rather than direct surface modeling. Therefore, the optimized Gaussians can be spatially unstructured, making direct mesh extraction difficult. Recent studies have addressed this limitation by incorporating additional geometric constraints, mesh-level optimization strategies, or object-aware priors into Gaussian representations [29–32]. Among Gaussian-based mesh reconstruction methods, SuGaR [33] regularizes Gaussians to align with scene surfaces and extracts meshes from the surface-aligned representation. GS2Mesh [34] renders stereo-aligned views from a trained 3DGS model, estimates depth using a pretrained stereo model, and fuses the resulting depth maps into a mesh. PGSR [35] improves geometric accuracy by introducing planar Gaussians, unbiased depth rendering, and multi-view consistency constraints. GS-2M [36] jointly reconstructs the surface mesh and decomposes surface material from the optimized Gaussians. 2DGS provides a particularly suitable foundation for surface reconstruction. Planar Gaussians produce more consistent depth and normal estimates than volumetric Gaussians and support mesh extraction through multi-view depth rendering and truncated signed distance function (TSDF) fusion [37]. These methods show that Gaussian representations can be converted into meshes, but most of them are designed for full-scene reconstruction. Recent object-centric methods attempt to address this problem by using semantic information, object masks, or target-aware reconstruction strategies. In particular, Object-Centric 2D Gaussian Splatting (OC-GS) [38] leverages object masks for targeted 2DGS optimization and constructs compact object-level Gaussian models through background removal and occlusion-aware pruning. However, existing object-level Gaussian reconstruction methods may still retain background Gaussians around the target object, especially when background removal depends on color similarity or occlusion-based criteria. In contrast, our method uses object masks to guide Gaussian optimization and reduce background-contributing Gaussians during training. This produces an object-aware Gaussian representation that is more suitable for clean object-level mesh extraction.

3. Preliminaries

We adopt 2DGS as the baseline representation because its planar Gaussians provide an explicit surface-oriented formulation suitable for geometrically accurate reconstruction and mesh extraction. The proposed method preserves the geometric representation, differentiable rasterization procedure, and geometry-aware regularization terms of 2DGS while introducing additional object-aware loss functions and pruning operations. Therefore, this section first reviews the 2DGS framework, including its Gaussian representation, differentiable rasterization process, and loss functions.

3.1. 2DGS Framework

2DGS represents a scene using a set of oriented planar Gaussian primitives embedded in three-dimensional space. Unlike the volumetric primitives employed by 3DGS, each 2D Gaussian forms an elliptical disk that locally approximates the underlying scene surface. Because a primitive spreads its density across a planar disk rather than through a volume, the direction of steepest density change is unambiguous and is taken directly as the primitive normal, which allows a 2D Gaussian to align with a thin surface. This surface-oriented representation provides an explicit tangent frame and enables geometrically consistent depth and normal estimation across different viewpoints. The state of a primitive consists of a disk center $\mathbf{p}_k \in \mathbb{R}^3$, a pair of unit tangent directions $\mathbf{t}_u$ and $\mathbf{t}_v$ that span the plane it lies in, and a per-axis extent s_u, s_v giving the Gaussian variance measured along each of those directions. The primitive normal is determined by $\mathbf{t}_w = \mathbf{t}_u \times \mathbf{t}_v$. Its orientation and planar scale are represented by $\mathbf{R} = [\mathbf{t}_u, \mathbf{t}_v, \mathbf{t}_w] \in \mathbb{R}^{3 \times 3}$ and $\mathbf{S} = \text{diag}(s_u, s_v, 0) \in \mathbb{R}^{3 \times 3}$, respectively. In particular, the zero scale along the normal direction constrains the Gaussian to a two-dimensional tangent plane. Using local tangent coordinates (u, v) , the geometry of the Gaussian disk is parameterized in homogeneous coordinates as

$$\begin{aligned} \mathbf{P}(u, v) &= \mathbf{p}_k + s_u \mathbf{t}_u u + s_v \mathbf{t}_v v = \mathbf{H} (u, v, 1)^\top, \\ \mathbf{H} &= \begin{bmatrix} s_u \mathbf{t}_u & s_v \mathbf{t}_v & \mathbf{0} & \mathbf{p}_k \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{RS} & \mathbf{p}_k \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}. \end{aligned} \quad (1)$$

Here, $\mathbf{H} \in \mathbb{R}^{4 \times 4}$ maps a point from the local tangent plane to world space while jointly encoding the center, orientation, and spatial extent of the primitive. Therefore, a local coordinate $\mathbf{u} = (u, v)$ specifies a position relative to the Gaussian center along the two tangent directions. The Gaussian response is evaluated directly in this normalized tangent coordinate system as

$$\mathcal{G}(\mathbf{u}) = \exp\left(-\frac{u^2 + v^2}{2}\right). \quad (2)$$

Although $\mathcal{G}(\mathbf{u})$ is isotropic in the local uv -coordinate system, the transformation $\mathbf{H}$ maps it to an oriented elliptical disk in world space. The center $\mathbf{p}_k$, scales (s_u, s_v) , and rotation represented by $(\mathbf{t}_u, \mathbf{t}_v)$ are learnable parameters. As in 3DGS, every primitive additionally carries an opacity α_k and view-dependent appearance $\mathbf{c}_k$ parameterized with spherical harmonics. Following 3DGS, 2DGS also inherits an adaptive density control strategy. It is invoked every 100 iterations while the iteration counter lies between 500 and 15,000, and it performs two operations. First, it clones and splits primitives whose accumulated positional gradient exceeds a threshold of 0.0002; because 2DGS does not rely on the gradient of the projected 2D center, the gradient of the 3D center is projected onto screen space and used as an approximation for this criterion. Second, it applies alpha-based pruning, which removes any primitive with opacity that falls below 0.05. The principal difference between 2DGS and 3DGS lies in how a primitive is evaluated at a pixel. The affine projection used in 3DGS is accurate only near the Gaussian center, and its approximation error grows with

distance from that center, which compromises perspective accuracy over the remainder of the disk. Instead, 2DGS explicitly computes the intersection between the pixel ray and the tangent plane of each Gaussian. Let $\mathbf{W} \in \mathbb{R}^{4 \times 4}$ denote the combined transformation from world space to screen space. A point on the Gaussian plane is projected according to $\mathbf{x} = (xz, yz, z, z)^\top = \mathbf{W}\mathbf{P}(u, v) = \mathbf{W}\mathbf{H}(u, v, 1, 1)^\top$, where $\mathbf{x}$ represents a homogeneous ray emitted from the camera passing through the pixel at image coordinate (x, y) and intersecting the disk at depth z . An alternative to the explicit intersection is to map the conic into screen space through the inverse $(\mathbf{W}\mathbf{H})^{-1}$, but this inversion becomes numerically unstable exactly when the disk degenerates toward a line in screen space, and discarding such ill-conditioned transformations by a fixed threshold destabilizes a differentiable optimization. Therefore, the ray is represented as the intersection of two homogeneous planes, $\mathbf{h}_x = (-1, 0, 0, x)^\top$ with normal $(-1, 0, 0)$ and offset x , and $\mathbf{h}_y = (0, -1, 0, y)^\top$ with normal $(0, -1, 0)$ and offset y . Because transforming a plane by a matrix $\mathbf{M}$ amounts to transforming its homogeneous parameters by the inverse transpose $\mathbf{M}^{-\top}$, applying $\mathbf{M} = (\mathbf{W}\mathbf{H})^{-1}$ to the two planes reduces to $\mathbf{h}_u = (\mathbf{W}\mathbf{H})^\top \mathbf{h}_x$ and $\mathbf{h}_v = (\mathbf{W}\mathbf{H})^\top \mathbf{h}_y$, so that the local plane parameters are obtained without ever forming the inverse of $\mathbf{W}\mathbf{H}$. Points of the disk are written $(u, v, 1, 1)$ in these local coordinates; the intersection must lie on both transformed planes, which gives the incidence condition $\mathbf{h}_u \cdot (u, v, 1, 1)^\top = \mathbf{h}_v \cdot (u, v, 1, 1)^\top = 0$. Solving it yields the ray-splat intersection in closed form,

$$u(\mathbf{x}) = \frac{\mathbf{h}_u^2 \mathbf{h}_v^4 - \mathbf{h}_u^4 \mathbf{h}_v^2}{\mathbf{h}_u^1 \mathbf{h}_v^2 - \mathbf{h}_v^2 \mathbf{h}_u^1}, \quad v(\mathbf{x}) = \frac{\mathbf{h}_u^4 \mathbf{h}_v^1 - \mathbf{h}_u^1 \mathbf{h}_v^4}{\mathbf{h}_u^1 \mathbf{h}_v^2 - \mathbf{h}_v^2 \mathbf{h}_u^1}, \quad (3)$$

where $\mathbf{h}_u^i$ and $\mathbf{h}_v^i$ denote the i -th parameter of the 4D homogeneous planes $\mathbf{h}_u$ and $\mathbf{h}_v$, respectively. The superscript is a parameter index and not an exponent, and $\mathbf{h}_u^3$ and $\mathbf{h}_v^3$ are always zero, which is why only the first, second, and fourth parameters appear in Equation (3). The resulting coordinate $\mathbf{u}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))$ is substituted into $\mathcal{G}$ to evaluate the primitive directly on its tangent plane, and substituting the same (u, v) back into the screen-space mapping recovers the intersection depth z used for sorting and for the geometric regularizers. This explicit intersection produces a perspective-correct Gaussian evaluation that remains consistent across viewpoints. When a Gaussian is observed from a highly oblique direction, its screen-space projection may degenerate toward a line segment and be missed entirely during rasterization. To prevent this, 2DGS applies an object-space low-pass filter that floors the response of the k -th primitive with a fixed screen-space Gaussian centered on the projected primitive center:

$$\hat{\mathcal{G}}(\mathbf{u}(\mathbf{x})) = \max \left\{ \mathcal{G}(\mathbf{u}(\mathbf{x})), \mathcal{G} \left(\frac{\mathbf{x} - \hat{\mathbf{p}}_k}{\sigma} \right) \right\} \quad (4)$$

where $\hat{\mathbf{p}}_k$ is the screen-space projection of the center $\mathbf{p}_k$ and σ is the radius of the low-pass response, which is fixed at $\sigma = \sqrt{2}/2$ so that a sufficient number of pixels is covered during rendering. We write the stabilized response as $\hat{\mathcal{G}}(\mathbf{u}(\mathbf{x}))$, following the original 2DGS formulation, in which the argument denotes the ray-splat intersection induced by the image coordinate $\mathbf{x}$. Equation (4) makes it clear that the second term of the maximum is evaluated at $\mathbf{x}$ itself rather than at the tangent coordinate, so that $\hat{\mathcal{G}}_k$ is ultimately a function of the pixel and not of $\mathbf{u}$ alone. This stabilization affects only the rasterization footprint, and does not change the underlying planar representation. For rendering, the Gaussian disks are assigned screen-space bounding boxes, grouped into tiles, and sorted according

to the depths of their centers. Their appearance is then accumulated from front to back using the same differentiable alpha-blending structure as 3DGS:

$$\mathbf{c}(\mathbf{x}) = \sum_{k=1}^K \mathbf{c}_k \alpha_k \hat{\mathcal{G}}_k(\mathbf{u}(\mathbf{x})) \prod_{j=1}^{k-1} (1 - \alpha_j \hat{\mathcal{G}}_j(\mathbf{u}(\mathbf{x}))) \quad (5)$$

where $\mathbf{x}$ is the image coordinate of the pixel, K is the total number of Gaussian disks associated with that pixel, k indexes those disks in depth order from the nearest to the farthest, α_k is the learnable opacity of the k -th primitive, and $\mathbf{c}_k$ denotes its view-dependent appearance. The product term represents the transmittance accumulated before the k -th primitive, and the traversal is terminated once the accumulated opacity saturates. Therefore, the quantity that is actually alpha-composited at a pixel is not α_k alone but the product $\alpha_k \hat{\mathcal{G}}_k(\mathbf{u}(\mathbf{x}))$, that is, the opacity weighted by the stabilized Gaussian response of that primitive at that pixel. Consequently, 2DGS retains the efficient differentiable rasterization of Gaussian Splatting while replacing volumetric primitives with perspective-correct and surface-aligned Gaussian disks.

3.2. 2DGS Loss

Although planar Gaussian disks provide a surface-oriented representation, photometric supervision alone does not guarantee that the optimized Gaussians form a geometrically consistent surface. Multiple semi-transparent Gaussian disks distributed at different depths along the same camera ray may produce similar rendered colors. Therefore, 2DGS combines an image reconstruction loss with depth distortion and normal consistency regularization. The photometric loss is defined using an L1 term and a differentiable structural dissimilarity term:

$$\mathcal{L}_c = (1 - \lambda_{\text{ssim}}) \mathcal{L}_1 + \lambda_{\text{ssim}} \mathcal{L}_{\text{D-SSIM}} \quad (6)$$

where $\mathcal{L}_1$ denotes the L1 reconstruction error between the rendered and ground-truth images, $\mathcal{L}_{\text{D-SSIM}}$ denotes the corresponding structural dissimilarity term inherited from 3DGS, and λ_{ssim} controls the contribution of the structural dissimilarity term. This loss supervises scene appearance but does not explicitly constrain the spatial distribution of Gaussian disks along each camera ray. Surface rendering terminates a ray at the first opaque interface it encounters, whereas alpha compositing imposes no cost on primitives dispersed in depth, so disks separated by a wide margin can still integrate to nearly identical color and depth. To concentrate the contributing Gaussians around a consistent surface, 2DGS minimizes the mutual distance between ray-splat intersections through the depth distortion loss

$$\mathcal{L}_d = \sum_{k,j} \omega_k \omega_j |z_k - z_j|, \quad (7)$$

where z_k and z_j are the depths of the k -th and j -th ray-splat intersections along the ray through a given pixel, and ω_k and ω_j are their corresponding blending weights. Here ω_k is the alpha-compositing weight of the k -th intersection, that is, the factor applied to $\mathbf{c}_k$ in Equation (5). This pairwise loss penalizes large depth differences between Gaussian disks contributing to the same pixel. Because the intersection depths z_k are themselves optimized rather than fixed sample positions, minimizing $\mathcal{L}_d$ concentrates the blending-weight distribution within a narrow depth interval by moving the primitives, thereby discouraging spatially dispersed disks from representing a single visible surface. The normal consistency loss further constrains the local orientations of the 2D Gaussian disks. For each pixel, 2DGS determines a representative surface intersection from the rendered depth distribution. In particular, the surface point is taken at the median intersection,

that is, the depth at which the accumulated opacity first reaches 0.5. Let $\mathbf{p}_s$ denote the 3D surface point obtained by back-projecting this depth. The surface normal estimated from the local variation of the rendered surface points is

$$\mathbf{N}(x, y) = \frac{\nabla_x \mathbf{p}_s \times \nabla_y \mathbf{p}_s}{\|\nabla_x \mathbf{p}_s \times \nabla_y \mathbf{p}_s\|_2}, \quad (8)$$

where $\nabla_x \mathbf{p}_s$ and $\nabla_y \mathbf{p}_s$ are spatial derivatives of the reconstructed surface points along the horizontal and vertical image directions. They are approximated using finite differences between neighboring back-projected depth points. The normal consistency loss is then formulated as

$$\mathcal{L}_n = \sum_k \omega_k (1 - \mathbf{n}_k^\top \mathbf{N}), \quad (9)$$

where k indexes the splats intersected along the ray, $\mathbf{n}_k$ is the camera-facing normal of the k -th Gaussian disk, and $\mathbf{N}$ is the normal estimated from the rendered depth surface. The dot product $\mathbf{n}_k^\top \mathbf{N}$ measures the directional agreement between the primitive normal and the reconstructed surface normal. Therefore, minimizing $\mathcal{L}_n$ aligns contributing Gaussian disks with the local surface orientation and improves consistency between the geometry encoded by Gaussian normals and the geometry obtained from rendered depth. The complete 2DGS training loss is expressed as

$$\mathcal{L}_{2\text{DGS}} = \mathcal{L}_c + \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n, \quad (10)$$

where λ_d and λ_n control the strengths of the depth distortion and normal consistency terms, respectively. In the original formulation these two weights are set to $\lambda_d = 1000$ for bounded scenes and $\lambda_d = 100$ for unbounded scenes, with $\lambda_n = 0.05$ in all cases. The photometric loss optimizes appearance, the depth distortion term concentrates contributing primitives around consistent surface depths, and the normal consistency term aligns their local orientations with the reconstructed surface. After optimization, 2DGS renders median depth maps from multiple training viewpoints and integrates them through TSDF fusion. The planar structure of the Gaussians and the geometry-aware regularization terms provide reliable depth and normal estimates for this fusion process. However, the 2DGS loss remains a full-scene optimization formulation. The photometric loss is evaluated over the complete image, and the depth distortion and normal consistency terms regularize all visible primitives without distinguishing the target object from the background. As a result, background Gaussian disks are optimized and preserved together with object-relevant disks even when only an object-level mesh is required. This increases the number of optimized primitives and may propagate unnecessary background geometry into the extracted mesh. Therefore, the proposed method retains 2DGS as its geometric baseline while extending its loss with mask-guided reconstruction and object-background Polarization Losses.

4. Method

4.1. Overview

We propose a mask-guided 2D Gaussian Splatting framework for object-level mesh reconstruction. As illustrated in Figure 1, given a set of posed multi-view images, binary masks corresponding to the target object are first obtained using Segment Anything Model 2 (SAM2) [39] and the Gaussians are initialized from a sparse point cloud reconstructed by SfM. The masks provide image-space priors that distinguish the target-object region from the surrounding background in each training view. The positions, rotations, scales, colors, and alphas of the Gaussians are then optimized using the differentiable rasterization framework of 2DGS. To formulate object-aware optimization, the proposed framework

modifies the loss function of 2DGS while retaining its geometric regularization terms. The photometric loss is replaced by a Masked L1 Loss that restricts appearance supervision to the target-object region. In addition, two polarization terms are introduced: the Alpha Polarization Loss regularizes the accumulated alpha response according to the object mask, whereas the ObjectMark Polarization Loss learns an explicit object-relevance attribute for each Gaussian. The resulting ObjectMark scores are subsequently used to remove background-related primitives through ObjectMark Pruning. The depth distortion loss and normal consistency loss inherited from 2DGS are retained because they constrain surface geometry rather than object–background separation. By jointly optimizing these terms, the framework preserves the geometric advantages of 2DGS while concentrating Gaussian optimization on the target object.

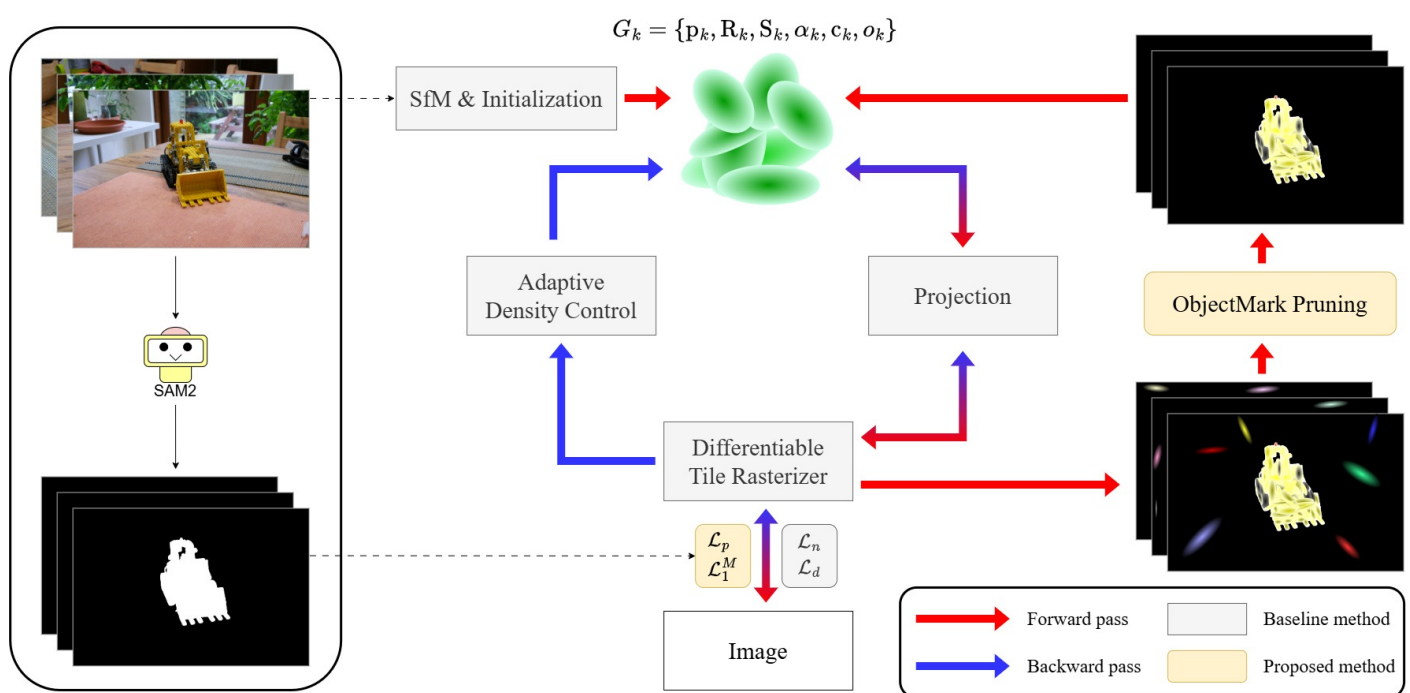


Figure 1. Overview of the proposed mask-guided 2D Gaussian Splatting framework. Red and blue arrows indicate the forward and backward optimization passes, respectively. Gray boxes represent components inherited from the 2DGS baseline, whereas yellow boxes denote the proposed mask-guided loss and the ObjectMark Pruning stage.

4.2. SAM2-Based Mask Generation

To guide Gaussian optimization toward the target object, our method requires object masks for the input views. We employ SAM2 [39] to obtain object masks for the posed multi-view images. SAM2 is a promptable visual segmentation foundation model that extends the original Segment Anything Model (SAM) [40] to both images and videos by introducing a streaming memory architecture. This architecture propagates object segments across frames through memory attention, and reduces to image segmentation when applied to a single frame. The model accepts sparse prompts such as points and bounding boxes as well as mask prompts, and exhibits strong zero-shot generalization across diverse object categories and visual domains. Given posed multi-view images $\mathbf{I}_v \in \mathbb{R}^{h \times w \times 3}$ of height h and width w , SAM2 produces a binary object mask $\mathbf{M}_v$ for the target object in each view:

$$\mathbf{M}_v \in \{0, 1\}^{h \times w} \quad (11)$$

where $M_v^i \in \{0, 1\}$ denotes the value of the object mask $\mathbf{M}_v$ at pixel i , with $M_v^i = 1$ for object pixels and $M_v^i = 0$ for background pixels. The target object is specified by a simple prompt on a reference view, such as a point or bounding box, and SAM2 propagates the segmentation to the remaining views through its memory mechanism, yielding consistent object masks across viewpoints without per-view re-annotation. These multi-view masks serve as object priors that separate object regions from background regions during training.

4.3. Masked L1 Loss

In 2DGS, the photometric loss is evaluated over the entire image. Therefore, the optimization process attempts to reproduce both the target object and the background. Such full-image supervision is suitable for scene-level reconstruction, but is inefficient for object-level mesh extraction because the background is not required in the final representation. To exclude background pixels from the appearance reconstruction loss, we introduce a Masked L1 Loss. The mask is applied to the ground-truth image so that its background pixels do not contribute to the loss; as such, they are not learned during training. The Masked L1 Loss is defined as

$$\mathcal{L}_1^M = \left\| \mathbf{M}_v \odot \hat{\mathbf{I}}_v - \mathbf{M}_v \odot \mathbf{I}_v \right\|_1, \quad (12)$$

where $\hat{\mathbf{I}}_v$ and $\mathbf{I}_v$ denote the rendered image and the corresponding ground-truth image for view v , respectively, and $\odot$ denotes element-wise multiplication. When $M_v^i = 0$, the corresponding background pixel makes no contribution to the photometric loss. When $M_v^i = 1$, the rendered color is optimized toward the ground-truth object color. In this way, the Masked L1 Loss prevents the background appearance from being optimized.

4.4. Alpha Polarization Loss

Although the Masked L1 Loss restricts color reconstruction to the target-object region, it operates at the image level and does not directly suppress every Gaussian associated with the background. To further guide the Gaussian representation toward the target object, we introduce the Alpha Polarization Loss. This loss regularizes the accumulated alpha rendered at each pixel according to the binary object mask. The Alpha Polarization Loss is defined over the entire rendered image as

$$\mathcal{L}_{pa} = \frac{1}{hw} \sum_{i=1}^{hw} \left[M_v^i (1 - A_i) + (1 - M_v^i) A_i \right], \quad (13)$$

where A_i is the accumulated alpha value rendered at pixel i , obtained from the Gaussians that contribute to it:

$$A_i = \sum_{k=1}^K \alpha_k \hat{\mathcal{G}}_k(\mathbf{u}(\mathbf{x}_i)) \prod_{j=1}^{k-1} \left(1 - \alpha_j \hat{\mathcal{G}}_j(\mathbf{u}(\mathbf{x}_i)) \right) \quad (14)$$

where $\mathbf{x}_i$ is the image coordinate of pixel i and where K , k , α_k , $\hat{\mathcal{G}}_k$, and the transmittance product are those of Equation (5). Equation (14) follows front-to-back alpha compositing, meaning that $A_i \in [0, 1]$ measures how strongly the Gaussian representation covers pixel i . The two bracketed terms in Equation (13) are mutually exclusive; minimizing the loss drives A_i toward one on object pixels and toward zero on background pixels, promoting foreground coverage while suppressing background opacity in the alpha that is used directly for rendering.

4.5. ObjectMark Polarization Loss

Although the Alpha Polarization Loss suppresses Gaussian contributions in background regions, alpha alone does not provide an explicit primitive-level criterion for object–background discrimination. Because it participates directly in visibility estimation and color compositing, it must simultaneously account for rendered appearance, occlusion, and opacity, and a low alpha value. As such, it does not necessarily indicate that a Gaussian belongs to the background. To provide each Gaussian with a dedicated object-relevance attribute, we introduce a learnable ObjectMark score $o_k \in [0, 1]$ for the k -th Gaussian, where a value close to one indicates strong association with the target object and a value close to zero indicates that the primitive is more likely to represent the background. All scores are initialized to 0.5, the neutral midpoint of the interval, which introduces no initial bias toward either category. A Gaussian created by adaptive density control inherits the score of its parent, so object-relevance information learned prior to densification is preserved. To allow supervision by the binary object masks, the per-Gaussian scores are accumulated at the image level and regularized by the ObjectMark Polarization Loss:

$$\mathcal{L}_{po} = \frac{1}{hw} \sum_{i=1}^{hw} \left[M_v^i (1 - O_i) + (1 - M_v^i) O_i \right] \quad (15)$$

where O_i is the accumulated ObjectMark value at pixel i :

$$O_i = \sum_{k=1}^K o_k \hat{\mathcal{G}}_k(\mathbf{u}(\mathbf{x}_i)) \prod_{j=1}^{k-1} \left(1 - o_j \hat{\mathcal{G}}_j(\mathbf{u}(\mathbf{x}_i)) \right). \quad (16)$$

Therefore, Equation (16) is the compositing operator of Equation (14) with the opacity α_k replaced by the ObjectMark score o_k . Each score is weighted by the footprint of its primitive at the pixel, and $O_i \in [0, 1]$ can be compared directly with the binary mask value M_v^i . As in Equation (13), minimizing $\mathcal{L}_{po}$ drives O_i toward one on object pixels and toward zero on background pixels. Because O_i is constructed from the learnable scores, the gradients propagate from image-space mask supervision back to the primitive-level attributes, so Gaussians that consistently contribute to object pixels across multiple views obtain high scores while those that mainly contribute to background pixels obtain low scores. The ObjectMark score represents only object relevance, allowing rendering and object–background separation to be optimized jointly without assigning both roles to alpha.

4.6. Overall Training Loss

The proposed mask-guided losses are combined with the depth distortion and normal consistency losses inherited from the 2DGS baseline. The two polarization terms introduced above act on alpha values and ObjectMark scores, respectively, but share the goal of separating object and background Gaussians. Thus, we combine them into a single Polarization Loss

$$\mathcal{L}_p = \lambda_{pa} \mathcal{L}_{pa} + \lambda_{po} \mathcal{L}_{po}, \quad (17)$$

where λ_{pa} and λ_{po} weight the Alpha Polarization Loss and the ObjectMark Polarization Loss, respectively, and are specified in Section 5.1. Using Equation (17), the complete training loss is defined as

$$\mathcal{L} = \mathcal{L}_1^M + \mathcal{L}_p + \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n, \quad (18)$$

where λ_d and λ_n follow the default settings of the 2DGS baseline. The four terms of Equation (18) correspond to the four losses shown in Figure 1. The proposed loss replaces the Photometric Loss $\mathcal{L}_c$ with the Masked L1 Loss. Accordingly, the D-SSIM term used

in the 2DGS photometric loss is not included during object-aware optimization. The three mask-guided terms act at different levels: $\mathcal{L}_1^M$ on pixel appearance, $\mathcal{L}_{pa}$ on rendered opacity, and $\mathcal{L}_{po}$ on the per-primitive object-relevance attribute. The depth distortion and normal consistency losses are inherited unchanged from 2DGS as geometric regularizers. The former concentrates ray-splat intersections around a consistent surface, and the latter aligns the Gaussian normals with the normals estimated from the rendered depth map. Together, these terms preserve the surface-oriented geometric regularization of 2DGS while preventing the optimization process from unnecessarily reconstructing the entire scene.

4.7. ObjectMark Pruning

The polarization losses reduce the accumulated alpha and ObjectMark responses in background regions. However, Gaussians with low ObjectMark scores are not automatically removed. To explicitly remove these primitives, we apply ObjectMark Pruning using the ObjectMark scores, as illustrated in Figure 2.

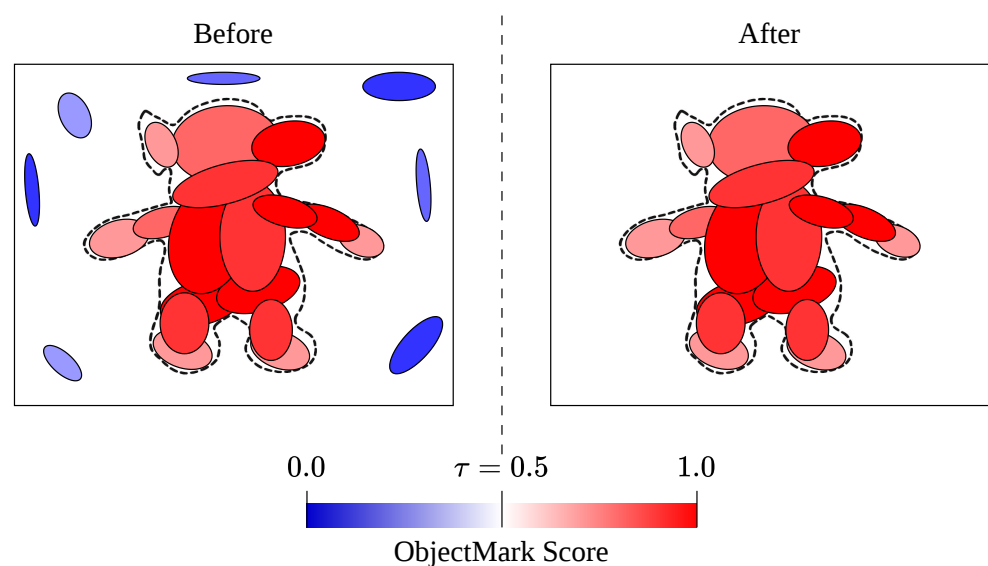


Figure 2. Illustration of ObjectMark Pruning. Gaussians are colored by their learnable ObjectMark score. The two panels show the Gaussian set before and after pruning at a predefined ObjectMark threshold τ .

Let $\mathcal{G}_{all}$ denote the current set of Gaussians. Given a pruning threshold τ , the set of primitives to be removed is defined as

$$\mathcal{G}_{remove} = \{G_k \in \mathcal{G}_{all} \mid o_k \leq \tau\}. \quad (19)$$

The pruning threshold is set to $\tau = 0.5$. Because every ObjectMark score is initialized to 0.5, a Gaussian is retained only when optimization provides sufficient evidence that its object relevance is greater than the neutral initialization. Conversely, a primitive for which the score remains at or below the midpoint is treated as background-related and hence removed. Unlike alpha-based pruning, ObjectMark Pruning does not rely solely on a primitive's rendering opacity. A Gaussian can have low alpha because of partial visibility or view-dependent coverage, and can overlap with other primitives even when it belongs to the target object. Instead, the ObjectMark score represents object relevance learned from multi-view mask supervision. In this way, it provides a criterion that is conceptually separated from the rendering contribution of the primitive. Applied repeatedly during optimization, this criterion progressively refines the Gaussian set. The process

removes background-related primitives while preserving Gaussians with learned scores that indicate an association with the target object. After training, we extract the mesh from the remaining object Gaussians via the TSDF fusion procedure inherited from 2DGS. Since most background primitives are removed before mesh extraction, the resulting mesh contains fewer background artifacts, reducing the need for additional postprocessing.

Figure 3 summarizes the proposed optimization loop. At each iteration, the rendered image and the corresponding input image are compared through the total loss, then the Gaussian attributes and ObjectMark scores are updated accordingly. Adaptive density control is applied on the schedule inherited from the 2DGS baseline; from iteration 2500 onward, ObjectMark Pruning is invoked at every 500 iterations to remove those primitives with scores remaining at or below τ . The loop repeats until the end of training, after which the mesh is extracted from the remaining object Gaussians by TSDF fusion.

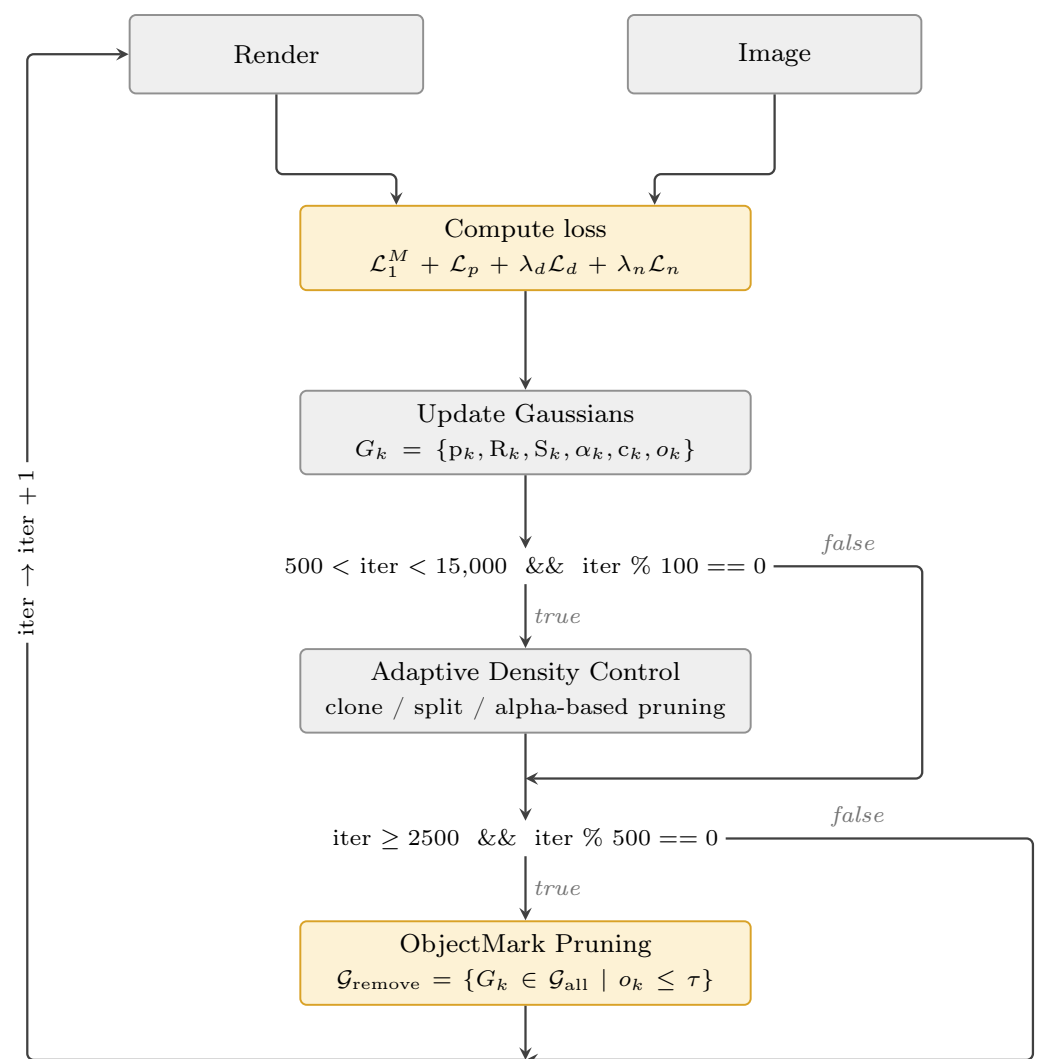


Figure 3. Overall flowchart of the proposed mask-guided optimization loop. Gray boxes denote components inherited from the 2DGS baseline, whereas yellow boxes denote the proposed mask-guided loss and the ObjectMark Pruning stage.

5. Experiments

5.1. Implementation Details

We conducted all experiments on a single NVIDIA RTX 4080 GPU. The spherical harmonics degree, learning rates, adaptive density control, and optimization settings for

Gaussian position, scale, rotation, alpha, and color all follow the default configuration of 2DGS, and we trained the model using the overall loss in Equation (18). The Polarization Loss weights are set to $\lambda_{pa} = 0.1$ and $\lambda_{po} = 1.0$, and the learning rate of the ObjectMark score is set to 0.01. We trained the model for 30,000 iterations in total. ObjectMark Pruning starts at 2500 iterations and is repeated every 500 iterations until the end of training, with the pruning threshold fixed at $\tau = 0.5$. For mask supervision, we used the ground-truth masks provided with the DTU [41] and BlendedMVS datasets [42]. For Mip-NeRF 360 [43], where ground-truth object masks are unavailable, we generated object masks using SAM2. We resized all masks to the training resolution using nearest-neighbor interpolation and used them as binary object masks during optimization. After optimization, we extracted the mesh using the same TSDF fusion pipeline as 2DGS. The extracted mesh was then rendered using nvdiffrast [44], a high-performance differentiable rasterizer, to obtain mesh renderings from evaluation viewpoints for quantitative mesh-level evaluation and qualitative visualization. For evaluation, we assessed both reconstruction quality and computational efficiency. To measure rendering fidelity, we report Peak Signal-to-Noise Ratio (PSNR). Before computing the error, we applied the object mask only to the ground-truth image.

The mean squared error (MSE) [45] is computed as

$$\text{MSE} = \frac{\|\hat{\mathbf{I}}_v - \mathbf{M}_v \odot \mathbf{I}_v\|_2^2}{h \cdot w \cdot 3}. \quad (20)$$

Based on the MSE, PSNR is defined as

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right), \quad (21)$$

where $\hat{\mathbf{I}}_v$ is the rendered mesh image for view v , with $\mathbf{I}_v$ and $\mathbf{M}_v$ as the corresponding ground-truth image and binary object mask and MAX_I as the maximum possible pixel intensity. This setting penalizes rendered artifacts outside the target-object region, since the masked ground-truth background is set to zero. For geometric accuracy, we report Chamfer Distance in millimeters on the DTU dataset only, where ground-truth geometry is available. For all datasets, we report mean Intersection over Union (mIoU), mean Accuracy (mACC), and PSNR to evaluate the object-region reconstruction quality and effectiveness of background removal. For computational efficiency, we report the training time, peak GPU memory usage, and peak number of Gaussians during optimization.

5.2. Datasets

We evaluated our method on three multi-view datasets: DTU, BlendedMVS, and Mip-NeRF 360. These datasets cover controlled object-centric captures, diverse multi-view reconstruction scenes, and challenging real-world environments with complex backgrounds. For all datasets, we used posed multi-view images and object masks for the target object.

5.2.1. DTU

DTU is a widely used multi-view stereo dataset for evaluating 3D reconstruction methods. It consists of object-centric indoor scenes captured under controlled laboratory conditions. Each scan is captured from 49 or 64 camera positions with an image resolution of 1600×1200 , and reference geometry is obtained using structured light scans. We evaluated on 15 scans using the preprocessed version provided by 2DGS.

5.2.2. BlendedMVS

BlendedMVS is a large-scale dataset designed for generalized multi-view stereo. It contains high-resolution multi-view images across diverse scenes, including architectures, sculptures, and small objects. The dataset provides rendered images with corresponding camera parameters and depth information. We used eight scenes at a resolution of 800×600 , and ground-truth object masks were available for all scenes. Because BlendedMVS includes diverse geometry and appearance variations, we used it to evaluate the robustness of our method under varied reconstruction conditions.

5.2.3. Mip-NeRF 360

Mip-NeRF 360 is a real-world NVS dataset designed for unbounded 360-degree scenes. It contains indoor and outdoor scenes with complex backgrounds and challenging view-dependent appearance. In this work, we used the bonsai and kitchen scenes, which are captured in an object-centric manner that makes them well suited for evaluating object-level reconstruction. The original images have a resolution of 3118×2078 and are downsampled by a factor of 2, yielding a training resolution of 1559×1039 . Since ground-truth masks are unavailable for this dataset, we generated object masks using SAM2. On this dataset, we evaluated whether our method can reduce background primitives and produce cleaner object-level reconstruction results under real-world complex background conditions.

5.3. DTU Results

We first evaluated our method on the DTU dataset, for which ground-truth surface geometry is available. The purpose of this experiment is to determine whether the proposed mask-guided loss improves geometric reconstruction while suppressing background-related Gaussians. We followed the evaluation protocol adopted in previous studies [24,26,28,46]. Table 1 combines two distinct sources of results. The rows marked with [†] are quoted from the OC-GS paper [38], and were not rerun on our hardware. They follow the same DTU Chamfer Distance evaluation protocol as our own runs, and are used only for direct comparison of geometric accuracy. The unmarked rows were evaluated by us under the same protocol and hardware environment using the respective public implementations.

Table 1. Quantitative mesh reconstruction results on the DTU dataset in terms of Chamfer Distance. Chamfer Distance is reported in millimeters, and lower values indicate better geometric accuracy. Red, orange, and yellow indicate the first-, second-, and third-best results in each column, respectively. [†] denotes values quoted from OC-GS [38]; these methods were not rerun on our hardware. All unmarked rows were run by us under the hardware, software, and evaluation protocol of Section 5.1. The upper group contains neural implicit methods, whereas the lower group contains Gaussian Splatting methods.

	scan24	scan37	scan40	scan55	scan63	scan65	scan69	scan83	scan97	scan105	scan106	scan110	scan114	scan118	scan122	Mean
NeRF [†]	1.83	2.39	1.79	0.66	1.79	1.44	1.50	1.20	1.96	1.27	1.44	2.61	1.04	1.13	0.99	1.54
NeuS [†]	0.83	0.98	0.56	0.37	1.13	0.59	0.60	1.45	0.95	0.78	0.52	1.43	0.36	0.45	0.45	0.77
3DGS [†]	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
SuGaR [†]	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
OC-GS [†]	0.47	0.82	0.30	0.43	0.93	0.96	0.86	1.26	1.03	0.72	0.74	1.25	0.47	0.75	0.55	0.77
GS2Mesh	0.60	0.94	0.68	0.50	1.29	1.15	0.78	1.50	1.25	0.77	0.58	0.85	0.42	0.42	0.58	0.82
GS-2M	0.90	1.06	1.44	1.38	1.76	2.14	1.17	1.73	1.20	1.44	1.32	1.02	1.04	0.99	1.30	1.33
PGSR	0.47	0.91	0.52	0.57	1.06	1.61	0.70	1.73	1.10	1.01	0.64	0.68	0.47	0.59	0.57	0.84
2DGS	0.62	1.07	0.72	0.42	1.32	1.11	0.90	1.47	1.23	0.78	0.71	1.39	0.48	0.99	0.59	0.92
Ours	0.50	0.93	0.52	0.40	0.97	0.68	0.70	1.11	0.97	0.68	0.70	1.14	0.40	0.90	0.44	0.74

Table 1 reports the Chamfer Distance for 15 DTU scans. The proposed method achieves the lowest mean Chamfer Distance of 0.74 among the evaluated methods. NeuS and OC-GS follow closely, both obtaining a mean distance of 0.77. This result shows that the proposed

method obtains low errors on a broad range of scans and achieves the best aggregate result. The proposed method falls outside the three best results on scans 106, 110, and 118, indicating that the effectiveness of object-aware optimization can depend on scene-specific geometry and mask coverage. The most direct comparison is with the 2DGS baseline, which obtains a mean error of 0.92 using surface-aligned Gaussian disks. Relative to this baseline, the mean Chamfer Distance falls from 0.92 to 0.74, corresponding to a 19.6% reduction. Among the neural implicit methods, NeRF records a mean Chamfer Distance of 1.54 and NeuS reduces the mean error to 0.77 through surface modeling based on the signed distance field. The proposed method reduces the mean Chamfer Distance of NeRF by 51.9% and that of NeuS by 3.9%. The margin over NeuS is 0.03 in absolute terms, so the advantage over the strongest neural implicit baseline is present but modest. Among the Gaussian Splatting methods, 3DGS obtains a mean distance of 1.96, which the proposed method reduces by 62.2%, reflecting the limitation of a representation primarily optimized for image synthesis. SuGaR, GS2Mesh, GS-2M, and PGSR achieve mean errors of 1.33, 0.82, 1.33, and 0.84, corresponding to reductions of 44.4%, 9.8%, 44.4%, and 11.9% in favor of the proposed method. Relative to the object-centric OC-GS at 0.77, the difference is again 0.03 in favor of the proposed method. This margin is too small to establish a meaningful ranking between the two, and both should be regarded as achieving comparable geometric accuracy on this dataset, which indicates that shaping the loss with object masks is competitive with a pipeline built specifically for object-level reconstruction.

Table 2 presents the object-level evaluation results. Our method achieves an mIoU of 0.9839, an mACC of 0.9942, and a PSNR of 25.13, obtaining the highest value for all three metrics. The 2DGS baseline achieves an mIoU of 0.9033, an mACC of 0.9624, and a PSNR of 20.91. Relative to this baseline, the proposed method improves mIoU by 8.9% and mACC by 3.3%, and raises PSNR by 4.22 dB. GS-2M provides the strongest competing result, obtaining an mIoU of 0.9614, an mACC of 0.9852, and a PSNR of 23.33. The corresponding margins are 2.3% in mIoU, 0.9% in mACC, and 1.80 dB in PSNR. PGSR records the lowest values among the compared methods, with an mIoU of 0.8075, an mACC of 0.9155, and a PSNR of 18.55, which the proposed method exceeds by 21.8% in mIoU and 6.58 dB in PSNR. GS2Mesh obtains an mIoU of 0.8796, an mACC of 0.9517, and a PSNR of 19.98, corresponding to gains of 11.9% in mIoU and 5.15 dB in PSNR. These results show that the reconstructed meshes preserve a larger proportion of the target object while producing fewer non-object renderings outside the reference mask. The gap is larger in PSNR than in mIoU, which is informative about where the compared methods differ: mIoU is already high for every method because the bulk of the object is recovered in all cases, whereas PSNR is computed against the masked ground truth and is driven by whatever is rendered outside the silhouette. The improvement is thus concentrated at the object boundary and in the surrounding empty space rather than on the object interior, which is precisely the region that determines whether an extracted mesh can be used without cleanup.

Table 2. Quantitative evaluation on the DTU dataset in terms of mIoU, mACC, and PSNR. All values are averaged over the 15 evaluated scans. Higher values indicate better performance. Bold indicates the best result.

Method	mIoU	mACC	PSNR (dB)
GS2Mesh	0.8796	0.9517	19.98
GS-2M	0.9614	0.9852	23.33
PGSR	0.8075	0.9155	18.55
2DGS	0.9033	0.9624	20.91
Ours	0.9839	0.9942	25.13

Figure 4 provides a detailed qualitative comparison with the 2DGS baseline. The highlighted regions show that 2DGS frequently retains background-associated fragments near the object boundaries that extend into surrounding regions. These artifacts are especially noticeable around thin structures, object bases, and boundary-adjacent background areas. In contrast, our method removes most of these residual structures and produces cleaner transitions between the reconstructed object and the surrounding background. The resulting meshes exhibit more compact silhouettes and fewer detached fragments while preserving the principal object geometry. This qualitative difference is consistent with the intended effect of the proposed object-aware optimization, which suppresses background-associated Gaussians during training rather than relying solely on post-reconstruction cleanup.

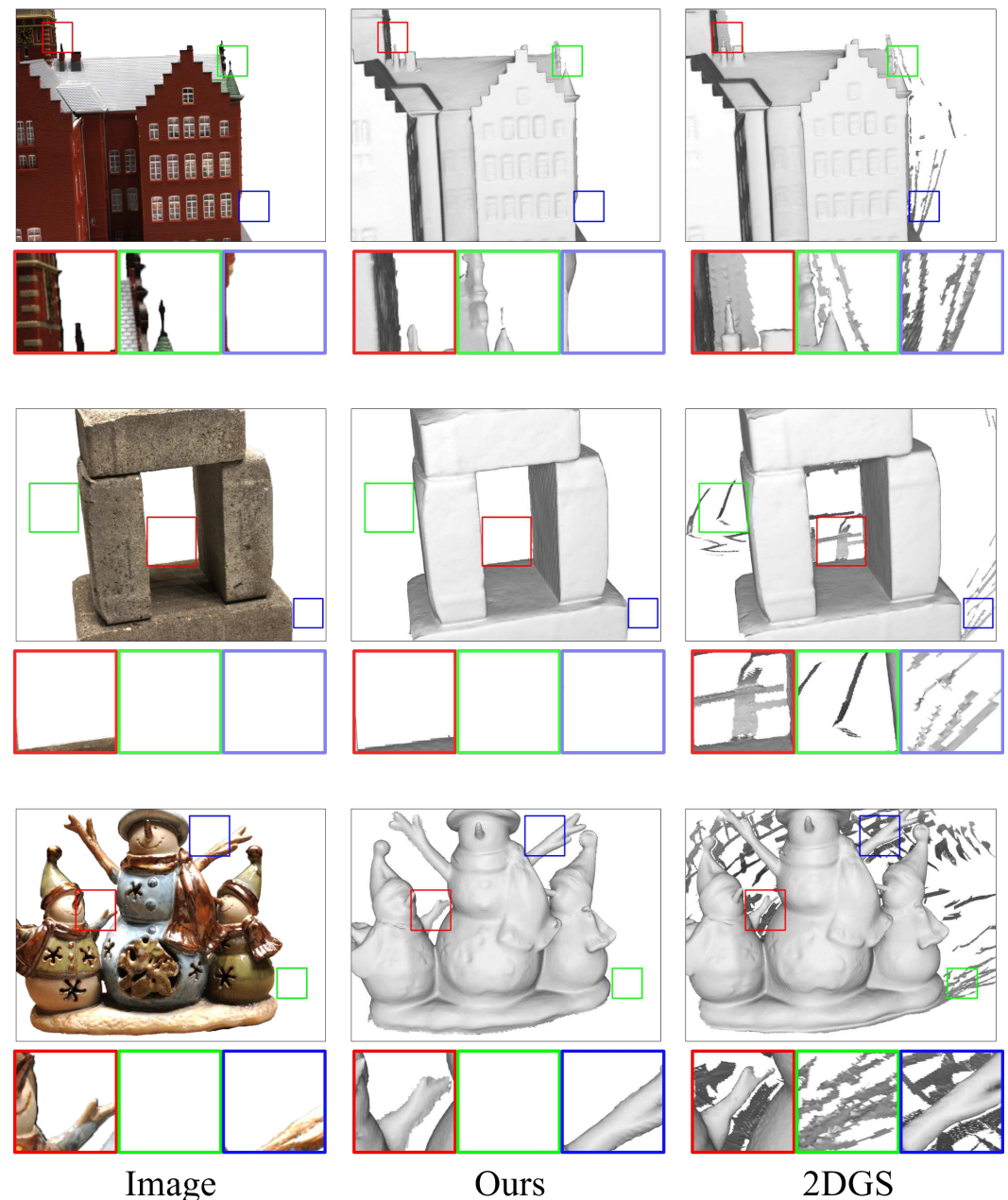


Figure 4. Detailed mesh comparison between the proposed method and 2DGS on the DTU dataset. The columns show the reference image, the proposed method, and 2DGS, evaluated from the same viewpoints. Colored boxes identify regions containing boundary distortions, noisy fragments, or residual background geometry.

Table 3 reports the computational cost of each method. Our method requires 18.03 min of training, 1.12 GB of peak GPU memory, and 0.19 M peak Gaussians, obtaining the lowest value for all three measurements. Compared with 2DGS, which requires 28.40 min, 1.72 GB, and 0.28 M Gaussians, the proposed method reduces training time by 36.5%, peak GPU memory by 34.9%, and peak Gaussian count by 32.1%. GS2Mesh and PGSR require similar training times of 20.35 and 20.15 min, so the reductions in optimization time are more modest at 11.4% and 10.5%. However, our method reduces the Gaussian count by 64.8% and 40.6%, respectively, and peak GPU memory usage by 83.2% and 50.9%, respectively. GS-2M has the highest computational cost, requiring 63.27 min and reaching 0.65 M Gaussians, which the proposed method reduces by 71.5% and 70.8%, respectively. Figure 5 shows that our method maintains the smallest number of Gaussians throughout training compared with the other methods. The curve also shows that the reduction is not obtained by a late cull. The proposed method separates from the baselines early and stays below them for the whole schedule, so the saving is a lower sustained primitive count rather than a smaller final model. This is what makes the memory and time reductions in Table 3 possible, since both are governed by the primitive count carried through optimization rather than by the count at the end. The reductions in peak GPU memory and Gaussian count hold up against every compared method, ranging from 34.9% to 83.2% and from 32.1% to 70.8%, respectively, which indicates that the savings come from maintaining a smaller primitive set throughout optimization. The reduction in Gaussian count supports the intended function of the mask-guided losses and ObjectMark Pruning, since Gaussians that contribute primarily to the background are suppressed and removed instead of being retained throughout full-scene optimization.

Table 3. Computational efficiency on the DTU dataset. Training time, peak GPU memory consumption, and peak Gaussian count are reported as averages over the 15 evaluated scans. Lower values indicate greater computational efficiency and a more compact representation. Bold indicates the best result.

Method	Time (min)	GPU (GB)	Gaussians (M)
GS2Mesh	20.35	6.65	0.54
GS-2M	63.27	3.74	0.65
PGSR	20.15	2.28	0.32
2DGS	28.40	1.72	0.28
Ours	18.03	1.12	0.19

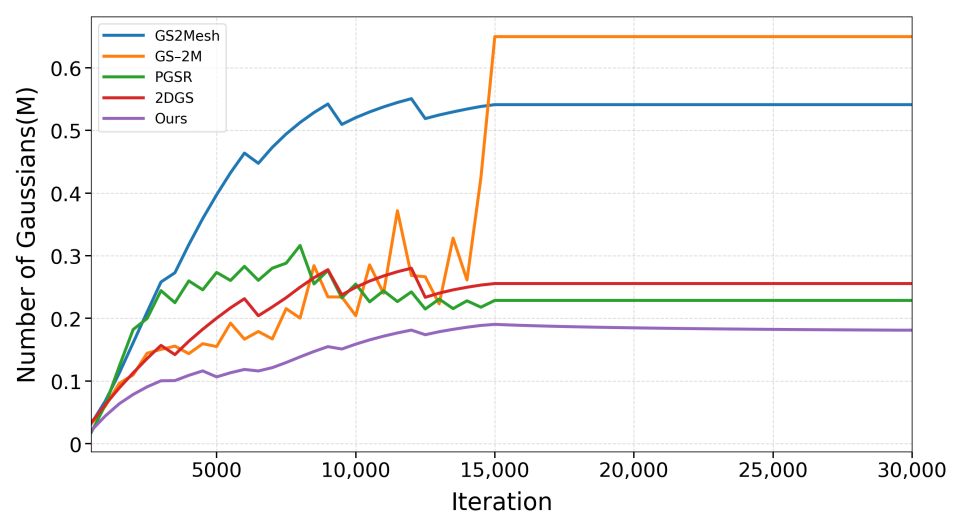


Figure 5. Number of Gaussians over the training iterations on the DTU dataset. The curves show the growth and stabilization of the Gaussian sets for GS2Mesh, GS-2M, PGSR, 2DGS, and the proposed method. The vertical axis is given in millions (M) of Gaussians.

Figure 6 presents qualitative mesh rendering results across multiple DTU scans. The differences are most apparent around object silhouettes and in regions where the reconstructed surface meets the background. 2DGS frequently retains elongated or isolated fragments extending outward from the target object, and similar residual geometry remains in several PGSR reconstructions, particularly around the lateral and lower boundaries. These artifacts are consistently visible across objects with different shapes and surface appearances, indicating that full-scene reconstruction can preserve background-associated geometry even when the principal object surface is recovered. GS-2M and GS2Mesh generally produce cleaner and more complete object structures than 2DGS and PGSR, but residual non-object geometry is still visible in several scenes, especially near the object base or along outer contours. In contrast, our method produces compact meshes with silhouettes that more closely follow the corresponding ground-truth objects, while also preserving characteristic structures such as protruding parts and thin boundary regions. Across the displayed scans, surrounding fragments are largely suppressed without introducing substantial loss of the principal object geometry. This qualitative behavior is consistent with the intended role of the mask-guided losses and ObjectMark Pruning, which restrict optimization to the target region and remove primitives with low object relevance.

We examined how the ObjectMark scores evolve under mask-guided supervision on DTU and how the periodic pruning stages behave throughout training. For this analysis, we recorded the ObjectMark score distribution and the number of removed primitives at every pruning stage and averaged the results over the 15 evaluated scans. Figure 7 shows the evolution of the ObjectMark score distribution over the full training schedule. As shown in Figure 7, all primitives are initialized at the neutral value of 0.5. Mask-guided supervision moves the score distribution away from this midpoint within the first few hundred iterations, initially toward intermediate intervals and subsequently toward the two ends of the score range. Before pruning begins, most primitives already occupy the highest-score interval, while a distinct low-score population accumulates in the lowest interval. Thus, rather than being separated by their initialization, the two populations are separated by the loss before any primitive is removed. This observation motivates delaying ObjectMark Pruning until iteration 2500 rather than applying it from the beginning. By this point, the scores have moved away from their neutral initialization, allowing pruning decisions to reflect accumulated mask supervision rather than uninformative initial scores. The pruning threshold is accordingly set at the initialization value, so that primitives whose scores have not been driven toward the object-relevant side of the score range are eligible for removal. After the first pruning stage at iteration 2500, the fraction of high-score primitives increases further. The subsequent sawtooth pattern reflects the alternation between densification, which introduces new primitives with intermediate ObjectMark scores, and pruning, which removes primitives that fail to acquire sufficient object relevance. Toward the end of training, nearly all surviving Gaussians lie in the highest-score interval.

Figure 8 further shows the score intervals occupied by the primitives at the time of removal. The first pruning stage at iteration 2500 accounts for the largest number of removals, with most deleted primitives belonging to the lowest-score interval. The number of removed primitives then decreases sharply and becomes negligible well before the end of training. In the later stages the removals are distributed more evenly across the sub-threshold intervals, with the 0.1–0.2 interval contributing the most and the interval immediately below the threshold contributing the fewest at nearly every stage. Across the whole schedule, the intervals well below the threshold account for the majority of removals, which indicates that ObjectMark Pruning acts mainly on primitives whose learned scores provide clear evidence of low object relevance rather than on primitives lying close to the decision boundary. The rapid decay in removal counts further shows that the Gaussian set

stabilizes well before mesh extraction, consistent with the plateau in the primitive count shown in Figure 5.

Overall, the DTU experiments show that the proposed loss improves object-region consistency and computational efficiency while maintaining competitive geometric accuracy.



Figure 6. Qualitative mesh rendering results on the DTU dataset. Ground truth (GT), the proposed method (Ours), 2DGS, PGSR, GS-2M, and GS2Mesh are presented under corresponding viewpoints.

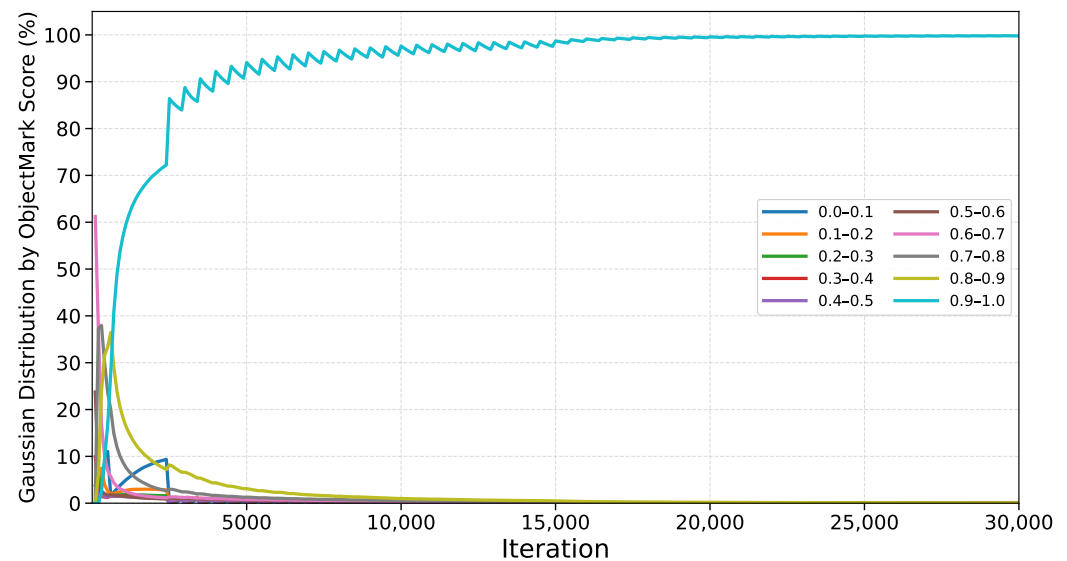


Figure 7. Distribution of Gaussians across ObjectMark score intervals during training on the DTU dataset, averaged over the 15 evaluated scans. Each curve represents the percentage of the current Gaussian set with ObjectMark scores within the corresponding interval.

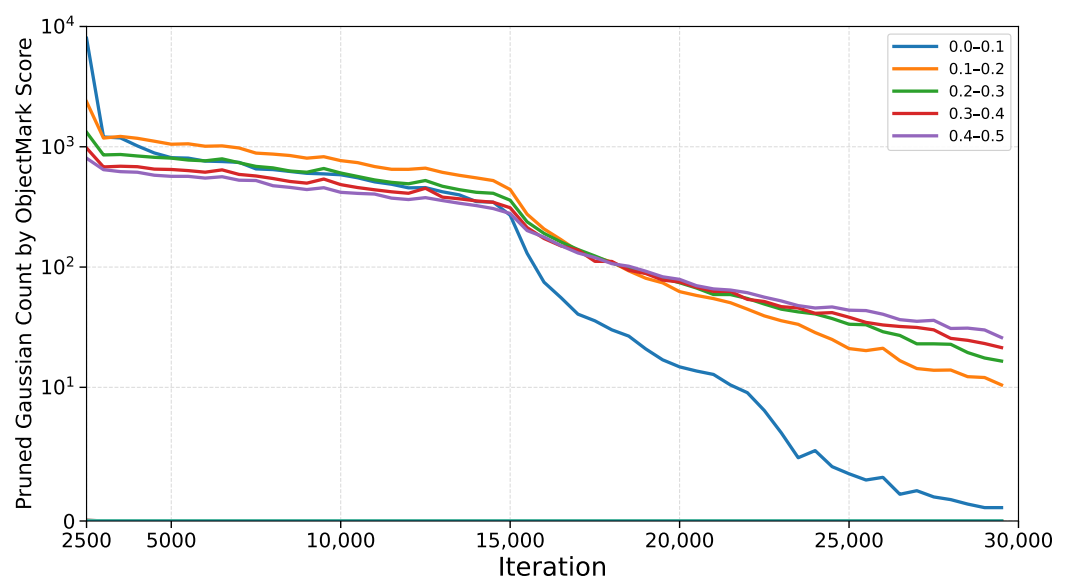


Figure 8. Number of Gaussians removed at each ObjectMark Pruning stage on the DTU dataset, averaged over the 15 evaluated scans and categorized by the five score intervals below the threshold $\tau = 0.5$. The horizontal axis starts at iteration 2500, when ObjectMark Pruning is first applied.

5.4. BlendedMVS Results

We next evaluated the proposed method on BlendedMVS to examine its behavior across objects with diverse geometry, texture, and appearance.

As shown in Table 4, the proposed method obtains an mIoU of 0.9904, an mACC of 0.9950, and a PSNR of 23.54, achieving the highest value for every metric. Compared with the 2DGS baseline, which obtains 0.9340, 0.9635, and 19.50, this corresponds to improvements of 6.0% in mIoU and 3.3% in mACC, together with a 4.04 dB gain in PSNR. GS-2M is the strongest competitor, obtaining an mIoU of 0.9896, an mACC of 0.9948, and a PSNR of 23.27, so the remaining margins are only 0.1% in mIoU, 0.02% in mACC, and 0.27 dB in PSNR. These margins are too small to establish a meaningful ranking between the two methods, and both should be regarded as providing comparably accurate object-level reconstruction on this dataset. GS2Mesh and PGSR obtain PSNR values of 17.14 and

19.13, which the proposed method exceeds by 6.40 dB and 4.41 dB, with a corresponding 7.9% mIoU improvement over GS2Mesh. Their lower PSNR values are accompanied by lower object-region overlap, particularly for GS2Mesh, suggesting that residual non-object surfaces and inaccurate object boundaries affect both mask-based consistency and rendered-image fidelity. Because the proposed method and GS-2M are statistically indistinguishable on this dataset, the relevant comparison here is not accuracy but cost; the same object-level quality is reached with roughly a quarter of the primitives and less than half the training time reported in Table 5.

Table 4. Quantitative evaluation on the BlendedMVS dataset in terms of mIoU, mACC, and PSNR. All values are averaged over the eight evaluated scenes. Higher values indicate better performance. Bold indicates the best result.

Method	mIoU	mACC	PSNR (dB)
GS2Mesh	0.9183	0.9454	17.14
GS-2M	0.9896	0.9948	23.27
PGSR	0.9297	0.9516	19.13
2DGS	0.9340	0.9635	19.50
Ours	0.9904	0.9950	23.54

Figure 9 compares the reconstructed meshes with their ground-truth counterparts. Our method preserves the overall object shape and major surface structures across the three scenes while producing little geometry outside the intended object regions. The reconstructed meshes closely follow the ground-truth silhouettes despite substantial differences in object scale and geometry, including elongated structures, fine facial features, and smooth curved surfaces. Although some high-frequency surface details are smoothed during reconstruction, the principal geometric characteristics of each object are retained.

The consistency of these results across visually and geometrically different scenes suggests that the mask-guided optimization remains effective under varying object shapes and surface appearances. This observation is consistent with the high aggregate object-region scores reported in Table 4.

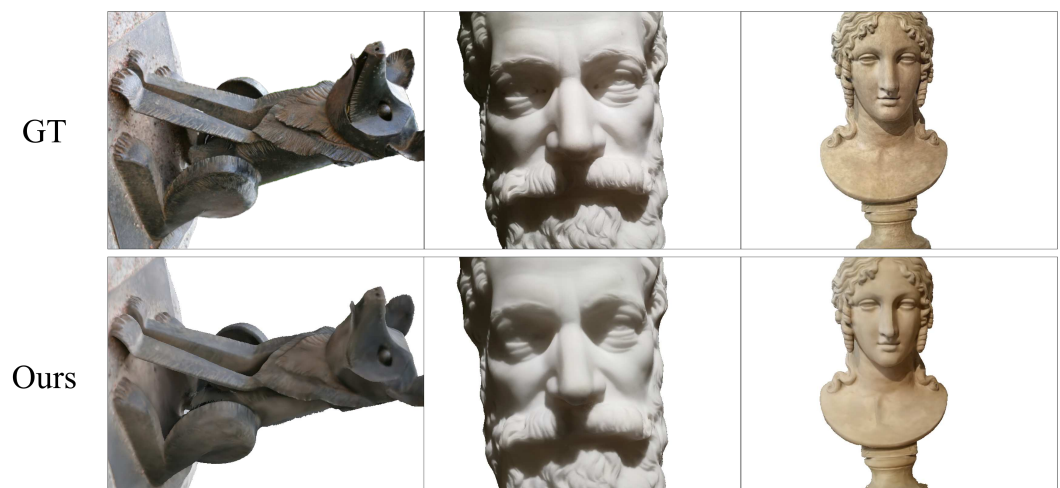


Figure 9. Qualitative mesh rendering results on the BlendedMVS dataset. The top row, labeled GT, shows the ground-truth reference images, and the bottom row shows the corresponding meshes reconstructed by the proposed method, rendered from the same viewpoints across three scenes.

Figure 10 provides a more detailed comparison with GS-2M, which is the strongest quantitative competitor on BlendedMVS. Although GS-2M achieves similar aggregate metrics, the highlighted regions reveal local holes and surface discontinuities that are not

fully reflected by the global scores. In contrast, our method produces more continuous surfaces in these regions while preserving the surrounding object geometry.

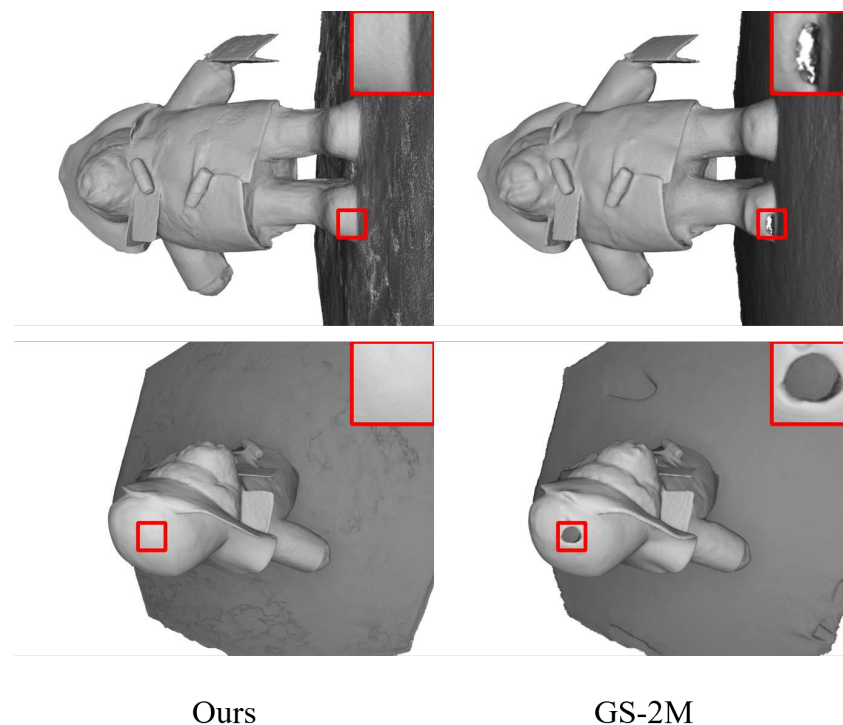


Figure 10. Qualitative comparison between the proposed method and GS-2M on the BlendedMVS dataset. Highlighted regions identify holes, incomplete local surfaces, and spurious geometry.

Table 5. Computational efficiency on the BlendedMVS dataset. Training time, peak GPU memory consumption, and peak Gaussian count are reported, averaged over the eight evaluated scenes. Lower values indicate greater computational efficiency and a more compact representation. Bold indicates the best result.

Method	Time (min)	GPU (GB)	Gaussians (M)
GS2Mesh	8.27	2.20	0.55
GS-2M	19.60	2.18	0.44
PGSR	18.08	2.38	0.24
2DGS	8.76	0.71	0.35
Ours	7.64	0.43	0.11

Table 5 reports the computational efficiency results. Our method requires 7.64 min, 0.43 GB of peak GPU memory, and 0.11 M peak Gaussians, the lowest values among the evaluated methods. Compared with the 2DGS baseline at 8.76 min, 0.71 GB, and 0.35 M Gaussians, this corresponds to reductions of 12.8% in training time, 39.4% in peak GPU memory, and 68.6% in peak Gaussian count. GS2Mesh has a similar training time of 8.27 min, giving a smaller 7.6% reduction, but its 0.55 M Gaussians and 2.20 GB of GPU memory are reduced by 80.0% and 80.5%. GS-2M and PGSR require 19.60 and 18.08 min, which the proposed method reduces by 61.0% and 57.7%. Relative to the 2DGS baseline, the reduction in Gaussian count is more than five times the reduction in training time. The method achieves object-level reconstruction quality comparable to or better than the strongest baseline while using a smaller Gaussian representation.

5.5. Mip-NeRF 360 Results

We finally evaluated the proposed method on the bonsai and kitchen scenes of Mip-NeRF 360. These scenes contain cluttered backgrounds, complex appearance, and wide camera trajectories, making them suitable for assessing object-level reconstruction in real-world environments.

Figure 11 compares object-level reconstruction results in scenes containing substantial background geometry. 2DGS retains numerous fragments around the target objects, while GS2Mesh preserves even larger non-object structures, particularly near the object base and surrounding background regions. PGSR and GS-2M achieve cleaner object–background separation, but local leakage and boundary irregularities remain visible in the rendered masks. In contrast, our method produces a more compact reconstruction with substantially fewer surrounding fragments, and the corresponding masks more closely follow the target-object silhouettes while preserving the principal object structures.

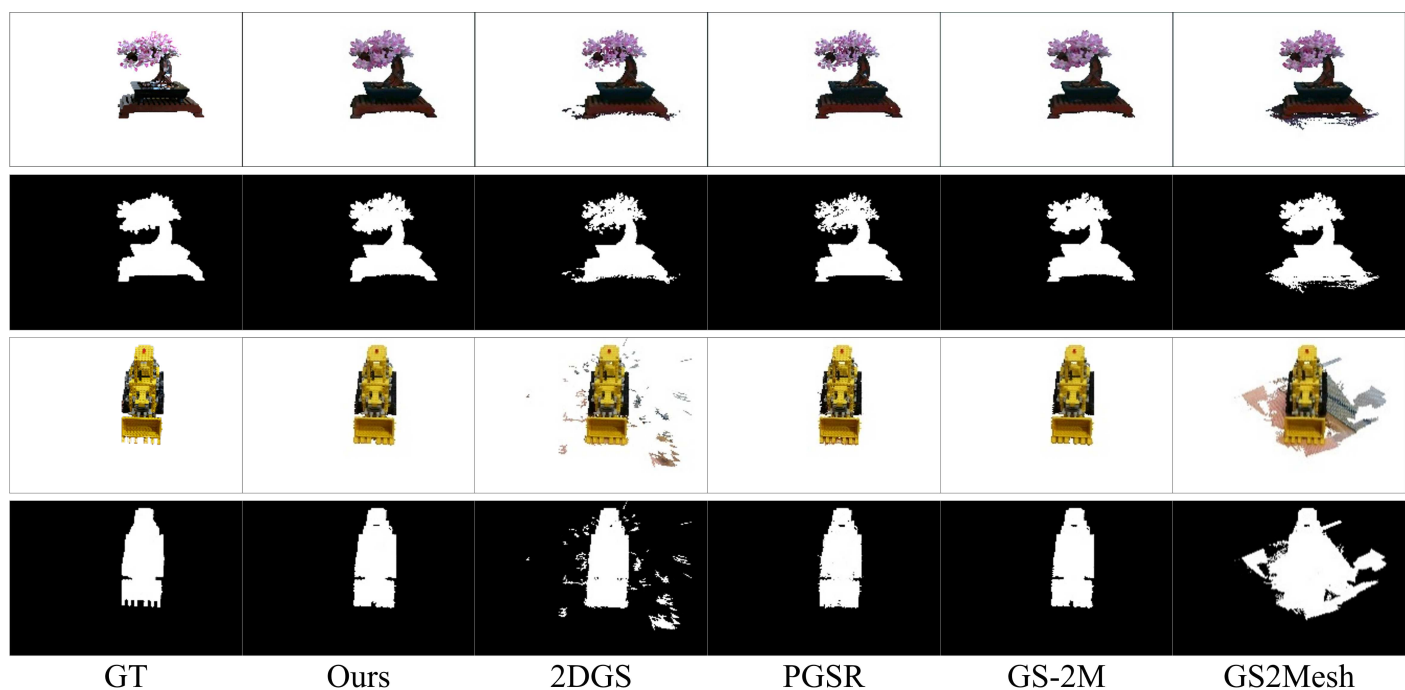


Figure 11. Qualitative mesh rendering results on the Mip-NeRF 360 dataset. GT, the proposed method, 2DGS, PGSR, GS-2M, and GS2Mesh are compared. The first two rows show the bonsai scene and the last two rows show the kitchen scene; within each pair, the upper row gives the rendered object appearance and the lower row the corresponding mask.

Unlike the object-centric datasets, these scenes contain substantial background content relative to the target objects, providing a more challenging setting for evaluating object–background separation. The results show that the proposed object-aware optimization suppresses background-associated geometry during training, reducing the need for a separate post-reconstruction cleanup stage to obtain the target-object mesh.

Table 6 presents the quantitative object-region results. In both scenes, our method obtains the highest mIoU and mACC. For bonsai, the proposed method achieves an mIoU of 0.9646 and an mACC of 0.9953, improving on the 2DGS baseline values of 0.8814 and 0.9847 by 9.4% and 1.1%, respectively. The strongest competing mIoU is 0.9350 from GS-2M, which the proposed method exceeds by 3.2%. The improvement in mIoU is consistent with a reduction in residual structures outside the target-object mask. For kitchen, the proposed method achieves an mIoU of 0.9511 and an mACC of 0.9943, corresponding to improvements of 18.2% and 2.5% over the 2DGS values of 0.8044 and 0.9696. GS-2M obtains

a closely comparable mIoU of 0.9484, leaving a margin of only 0.3%, whereas GS2Mesh reaches 0.5309 and is improved on by 79.2%. The relative gain over 2DGS is roughly twice as large on kitchen as on bonsai, reflecting the greater amount of clutter and background geometry retained by the full-scene baseline in that scene.

Table 6. Quantitative evaluation on the Mip-NeRF 360 dataset. We report mIoU and mACC for the bonsai and kitchen scenes. Higher values indicate better performance. Bold indicates the best result.

Method	Bonsai		Kitchen	
	mIoU	mACC	mIoU	mACC
GS2Mesh	0.7501	0.9611	0.5309	0.8867
GS-2M	0.9350	0.9920	0.9484	0.9929
PGSR	0.9181	0.9903	0.9321	0.9913
2DGS	0.8814	0.9847	0.8044	0.9696
Ours	0.9646	0.9953	0.9511	0.9943

Table 7 jointly compares optimization time and rendered-mesh fidelity. On bonsai, the proposed method requires 10.62 min and achieves a PSNR of 23.69, making it both the fastest and the highest-PSNR method in this scene. The 2DGS baseline requires 29.75 min for a PSNR of 19.95; the proposed method therefore reduces training time by 64.3% while gaining 3.74 dB in PSNR. GS-2M obtains a comparable PSNR of 23.58, only 0.11 dB below the proposed method but requires 25.77 min, so the proposed method reaches the same fidelity 58.8% faster. On kitchen, the proposed method requires 12.18 min and achieves a PSNR of 25.30, reducing the 31.73 min of 2DGS by 61.6% while raising PSNR by 6.07 dB. GS-2M again provides the second-highest PSNR at 24.86, 0.44 dB lower, with a training time of 26.20 min that the proposed method reduces by 53.5%. The pattern across both scenes is that the accuracy margin over GS-2M is small, whereas the reduction in optimization time is consistently above 50%. This separation is larger here than on the object-centric datasets, which is consistent with the amount of background each scene contains: the more of the captured volume that is irrelevant to the target object, the more optimization the full-scene baselines spend on geometry that mesh extraction will discard. The combination of shorter optimization time and higher PSNR is consistent with the proposed loss reducing unnecessary background reconstruction. Overall, the Mip-NeRF 360 results show that object-aware optimization remains effective in cluttered real-world scenes. The proposed method provides the highest object-region metrics and rendered-mesh PSNR while requiring the shortest training time in both evaluated scenes.

Table 7. Comparison of training time and mesh-rendering PSNR on the Mip-NeRF 360 dataset. Lower training time and higher PSNR indicate better performance. Bold indicates the best result.

Method	Bonsai		Kitchen	
	Time (min)	PSNR (dB)	Time (min)	PSNR (dB)
GS2Mesh	45.32	16.39	53.43	15.89
GS-2M	25.77	23.58	26.20	24.86
PGSR	41.21	22.53	42.92	23.47
2DGS	29.75	19.95	31.73	19.23
Ours	10.62	23.69	12.18	25.30

5.6. Ablation Study

The ablation study evaluates the contributions of the three mask-guided losses and the two pruning criteria by replacing or removing each in turn from the complete formulation. The ObjectMark Polarization Loss and ObjectMark Pruning are reported as a single ablation entry, denoted $-(\mathcal{L}_{po} + o_k \text{ Pruning})$, because the two are not independently separable in this formulation. Since o_k is used only during ObjectMark Pruning, retaining $\mathcal{L}_{po}$ without the pruning stage leaves the learned scores unused and reproduces that entry exactly. The converse configuration, retaining the pruning stage while removing $\mathcal{L}_{po}$, is degenerate; with no gradient reaching o_k , every score remains at its initialization, which coincides with the threshold, so every primitive satisfies Equation (19) and the entire Gaussian set is deleted at iteration 2500.

As shown in Table 8, the complete formulation achieves the highest performance, with an mIoU of 0.9839, an mACC of 0.9942, and a PSNR of 25.13. The corresponding qualitative result in Figure 12 contains the most clearly separated object boundary and the fewest visible background fragments. The $\mathcal{L}_1^M \rightarrow \mathcal{L}_c$ variant replaces the Masked L1 Loss with the 2DGS photometric loss, restoring the $\mathcal{L}_{D-SSIM}$ term, and yields an mIoU of 0.9827, an mACC of 0.9932, and a PSNR of 24.34 dB. When the Masked L1 Loss is replaced only with the L1 loss, the $\mathcal{L}_1^M \rightarrow \mathcal{L}_1$ variant achieves an mIoU of 0.9802, an mACC of 0.9923, and a PSNR of 24.92 dB. Compared with $\mathcal{L}_1^M \rightarrow \mathcal{L}_c$, removing the $\mathcal{L}_{D-SSIM}$ term increases PSNR by 0.58 dB but decreases mIoU by 0.0025 and mACC by 0.0009. Nevertheless, both full-image variants remain inferior to the complete formulation across all three metrics, supporting the effectiveness of restricting photometric supervision to the target-object region.

Two additional variants examine the respective roles of the two pruning mechanisms. The $-\alpha_k$ Pruning variant disables the inherited alpha-based pruning while retaining ObjectMark Pruning, and achieves an mIoU of 0.9827, an mACC of 0.9914, and a PSNR of 24.43 dB. Conversely, the $-(\mathcal{L}_{po} + o_k \text{ Pruning})$ variant removes both terms while retaining the inherited alpha-based pruning, yielding an mIoU of 0.9836, an mACC of 0.9941, and a PSNR of 24.93 dB. Both variants perform below the complete formulation across all three metrics, indicating that the two pruning mechanisms provide complementary contributions. The larger degradation observed when alpha-based pruning is removed is consistent with their distinct roles: the inherited pruning eliminates Gaussians with low opacity, whereas ObjectMark Pruning removes Gaussians with low object relevance learned through mask-guided supervision.

Table 8. Ablation study on the DTU dataset. The variants either replace the Masked L1 Loss with the 2DGS photometric loss ($\mathcal{L}_1^M \rightarrow \mathcal{L}_c$) or unmasked L1 loss ($\mathcal{L}_1^M \rightarrow \mathcal{L}_1$), or remove the inherited alpha-based pruning ($-\alpha_k$ Pruning), the ObjectMark Polarization Loss with its pruning stage ($-(\mathcal{L}_{po} + o_k \text{ Pruning})$), or the Alpha Polarization Loss ($-\mathcal{L}_{pa}$). The baseline is 2DGS, and all values are averaged over the 15 evaluated scans. Higher is better, and the best results are shown in bold.

Variant	mIoU	mACC	PSNR (dB)
Full	0.9839	0.9942	25.13
$\mathcal{L}_1^M \rightarrow \mathcal{L}_c$	0.9827	0.9932	24.34
$\mathcal{L}_1^M \rightarrow \mathcal{L}_1$	0.9802	0.9923	24.92
$-\alpha_k$ Pruning	0.9827	0.9914	24.43
$-(\mathcal{L}_{po} + o_k \text{ Pruning})$	0.9836	0.9941	24.93
$-\mathcal{L}_{pa}$	0.9028	0.9585	23.25
Baseline	0.9033	0.9624	20.91

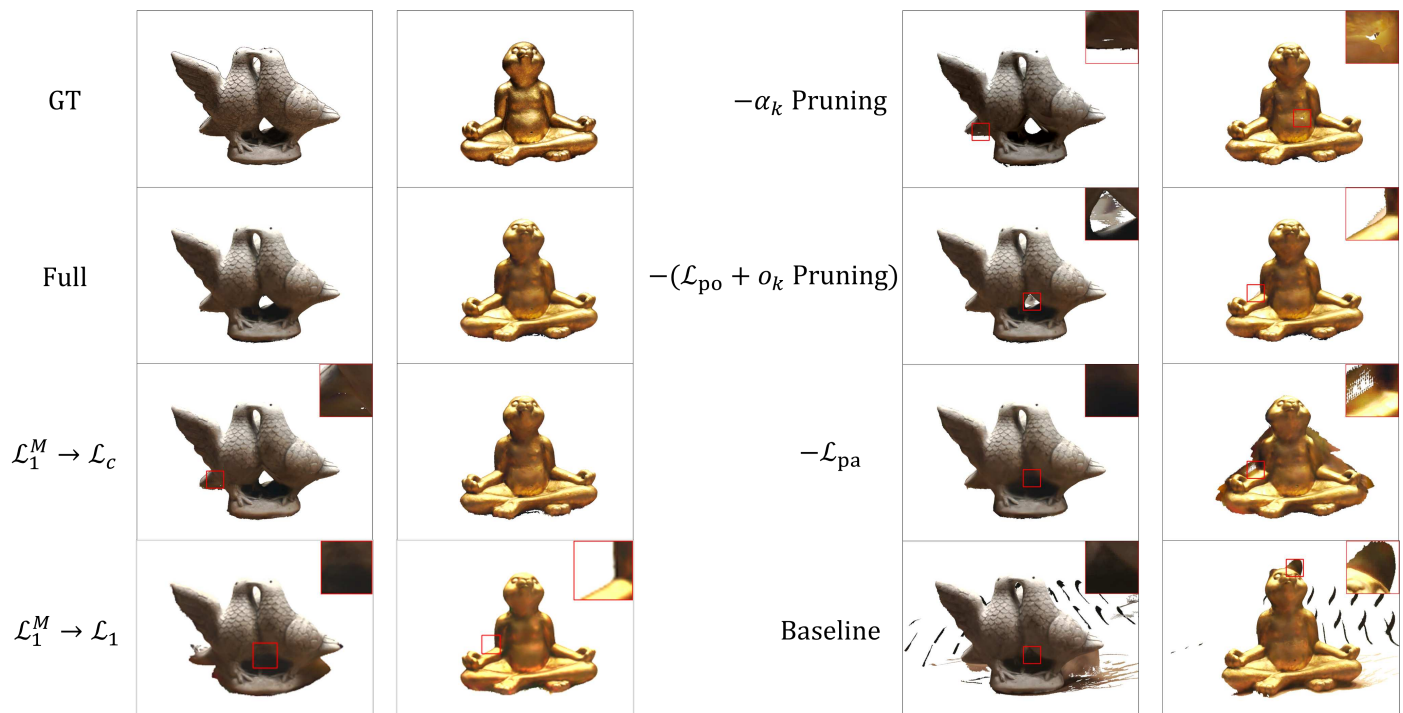


Figure 12. Qualitative ablation comparison on the DTU dataset. The panels show the ground truth, the complete method, the five ablation configurations reported in Table 8, and the 2DGS baseline. The enlarged regions show boundary errors, residual background geometry, and local surface artifacts.

Removing the Alpha Polarization Loss produces a substantial reduction in all metrics. The resulting variant obtains an mIoU of 0.9028, an mACC of 0.9585, and a PSNR of 23.25. Its mask-based metrics fall back to approximately the level of the 2DGS baseline, which achieves an mIoU of 0.9033 and an mACC of 0.9624, although the variant's PSNR of 23.25 remains 2.34 dB above the baseline value of 20.91. This result shows that alpha-level polarization is central to the proposed formulation. Without $\mathcal{L}_{pa}$, the accumulated opacity is not explicitly encouraged to approach one in object regions and zero in background regions. Consequently, the remaining ObjectMark-based supervision and pruning cannot fully compensate for the absence of opacity separation, and the object-background overlap returns to baseline level even though rendering fidelity is partly retained. Figure 12 highlights this failure. Without $\mathcal{L}_{pa}$, the reconstruction moves back toward the baseline, with boundary errors and residual background surfaces reappearing around the object, whereas the two photometric variants and the two pruning variants remain closer to the complete method. Because the 2DGS baseline optimizes a photometric loss over the entire image, it neither restricts appearance supervision to the target object nor forces background Gaussians to have low accumulated opacity or low object relevance. Therefore, it retains background-induced geometry and irregular object boundaries, and the complete method improves on it by the margins reported in Table 2. The comparison among the variants confirms the complementary roles assigned to the loss components in the proposed formulation.

6. Discussion

The experimental results show that the proposed method is effective for object-level mesh reconstruction. Our method improves geometric reconstruction, object-region consistency, and mesh rendering fidelity while reducing training time, GPU memory consumption, and the number of Gaussian primitives. These results indicate that suppressing background-related primitives contributes not only to computational efficiency but also to the geometric quality of the extracted object mesh. The Masked L1 Loss restricts pho-

tometric supervision to the target-object region. The Alpha Polarization Loss aligns the accumulated opacity with the binary object mask, thereby suppressing background opacity while preserving object regions. The ObjectMark Polarization Loss introduces a separate primitive-level relevance attribute, allowing object-background discrimination to be learned independently of the alpha parameter used for visibility and color compositing. The ObjectMark Pruning then removes background-related Gaussians with low ObjectMark scores. Removing the Alpha Polarization Loss produces the largest performance degradation, showing that alpha-level separation is central to stable object-focused optimization. The Masked L1 Loss improves object-region consistency by limiting appearance reconstruction to the object mask, whereas ObjectMark Pruning provides an additional refinement by removing low-relevance primitives that remain after optimization. The results show that the proposed components perform distinct and complementary functions.

Nevertheless, the proposed method has several limitations. Its performance depends on the accuracy and multi-view consistency of the input object masks. Mask errors may exclude valid object regions or include portions of the background, simultaneously affecting the masked photometric loss, accumulated alpha response, and ObjectMark supervision.

To examine this dependence directly, we retrained the proposed method on the BlendedMVS bust scene shown in Figure 9 using deliberately perturbed object masks, dilating and eroding the ground-truth masks at two strengths each while keeping every other setting fixed. Figure 13 compares the resulting meshes with the ground truth and with the mesh obtained from the unperturbed masks. Dilated masks label part of the background as object, so background material is no longer suppressed and survives as fringes and thin residual surfaces attached to the silhouette and to the base of the bust. Eroded masks exclude true foreground pixels near the object boundary, causing silhouette geometry to be lost and resulting in missing portions of the hair along with holes in the remaining hair region. Accordingly, mask dilation and erosion expose complementary failure modes by degrading the reconstruction in opposite directions: dilation introduces background artifacts by including regions outside the true object boundary, whereas erosion damages the reconstructed object by excluding valid foreground regions. Further evaluation is required for dynamic scenes, inaccurate camera poses, severe occlusion, and objects containing thin or reflective structures.

Overall, the experiments support the central assumption of this work: object-level mesh reconstruction benefits from an optimization loss that distinguishes object-relevant Gaussian primitives from background-related primitives. By introducing this distinction into both the loss and the pruning procedure, the proposed method produces a more compact representation and cleaner object-level geometry.

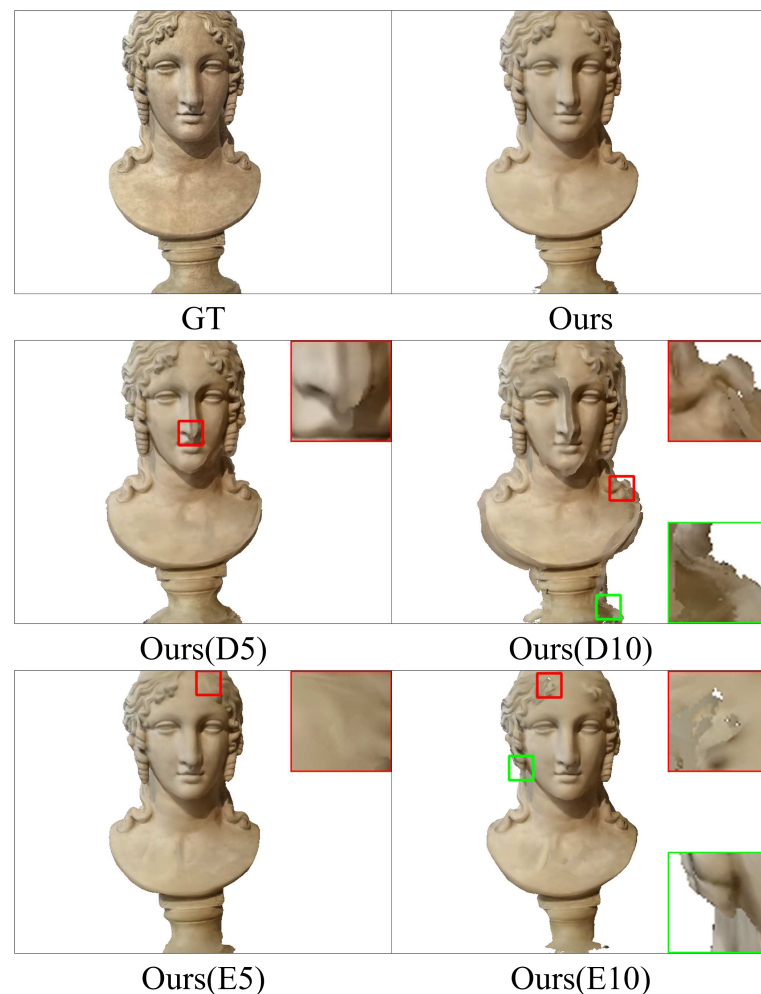


Figure 13. Effect of perturbing the ground-truth object masks on the reconstructed mesh. D and E denote dilated and eroded masks, respectively, and the trailing number is the size of the square structuring element. D5 and E5 use a 5×5 kernel, while D10 and E10 use a 10×10 kernel. Colored boxes identify regions containing boundary distortions, missing object mesh geometry, or residual background geometry.

7. Conclusions

This study has presented a mask-guided object-level mesh reconstruction method based on 2D Gaussian Splatting. The proposed loss combines a Masked L1 Loss, an Alpha Polarization Loss, and an ObjectMark Polarization Loss, which respectively restrict appearance supervision to the target object, enforce alpha-based object–background separation, and learn primitive-level object relevance, after which ObjectMark Pruning removes those primitives that score low. Across the DTU, BlendedMVS, and Mip-NeRF 360 datasets, the proposed method obtains the highest object-region accuracy among the compared methods, lowering the mean Chamfer Distance on DTU from 0.92 to 0.74 and reducing training time relative to the 2DGS baseline by 36.5%. By suppressing background Gaussians during optimization and concentrating the reconstruction process on the target object, the method enables efficient extraction of clean target-object meshes without complex postprocessing. Despite these encouraging results, there is further space for improvement. Future work will investigate uncertainty-aware mask supervision and the joint optimization of object segmentation and Gaussian reconstruction, so that reconstruction quality is no longer bounded by that of a fixed input mask. Extending the evaluation to more challenging capture conditions such as dynamic scenes, inaccurate camera poses, and severe occlusions also remains open.

Author Contributions: Conceptualization, S.W. and J.S.; methodology, S.W. and J.S.; software, S.W.; validation, S.W., J.S., and H.L.; formal analysis, S.W.; investigation, S.W.; resources, J.S.; data curation, S.W.; writing—original draft preparation, S.W.; writing—review and editing, S.W., J.S., and H.L.; visualization, S.W.; supervision, J.S.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Education (MOE) under the Glocal University 30 Project at Dong-A University (2026-STR-S1-06) and by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [2710109867, Development of User-Customized Freeform Future Display]. The authors acknowledge that the research was funded in part by ETRI, as required when utilizing the contents of the associated report.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This study uses three publicly available datasets: DTU [41], Blended-MVS [42], and Mip-NeRF 360 [43].

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2DGS	Two-Dimensional Gaussian Splatting
3DGS	Three-Dimensional Gaussian Splatting
GT	Ground Truth
mACC	Mean Accuracy
mIoU	Mean Intersection over Union
MSE	Mean Squared Error
NeRF	Neural Radiance Fields
NVS	Novel View Synthesis
OC-GS	Object-Centric Two-Dimensional Gaussian Splatting
PSNR	Peak Signal-to-Noise Ratio
SAM	Segment Anything Model
SAM2	Segment Anything Model 2
SDF	Signed Distance Field
SfM	Structure-from-Motion
TSDF	Truncated Signed Distance Function

References

1. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Commun. ACM* **2022**, *65*, 99–106. [\[CrossRef\]](#)
2. Kerbl, B.; Kopanas, G.; Leimkühler, T.; Drettakis, G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* **2023**, *42*, 139. [\[CrossRef\]](#)
3. Chen, G.; Wang, W. A Survey on 3D Gaussian Splatting. *ACM Comput. Surv.* **2026**, *58*, 313. [\[CrossRef\]](#)
4. Fei, B.; Xu, J.; Zhang, R.; Zhou, Q.; Yang, W.; He, Y. 3D Gaussian Splatting as a New Era: A Survey. *IEEE Trans. Vis. Comput. Graph.* **2025**, *31*, 4429–4449. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; Gao, S. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *Proceedings of the ACM SIGGRAPH 2024 Conference Papers, Denver, CO, USA, 28 July–1 August 2024*; ACM: New York, NY, USA, 2024; pp. 1–11. [\[CrossRef\]](#)
6. Müller, T.; Evans, A.; Schied, C.; Keller, A. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* **2022**, *41*, 102. [\[CrossRef\]](#)
7. Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; Su, H. TensorRF: Tensorial Radiance Fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2022; pp. 333–350. [\[CrossRef\]](#)

8. Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; Kanazawa, A. Plenoxels: Radiance Fields Without Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022*; IEEE: New York, NY, USA, 2022; pp. 5491–5500. [\[CrossRef\]](#)
9. Gu, X.; Chang, Y.; Xia, J.; Yu, Y.; Tian, Z.; Chen, J. OptiNeRF: A Spatially Optimized Neural Rendering Framework for Complex Scene Reconstruction. *Mathematics* **2026**, *14*, 842. [\[CrossRef\]](#)
10. evTwoZhu, M.; Gong, Y.; Gu, D.; Tian, C. Improving 3D Object Detection in Neural Radiance Fields With Channel Attention. *CAAI Trans. Intell. Technol.* **2025**, *10*, 1446–1458. [\[CrossRef\]](#)
11. Schönberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*; IEEE: New York, NY, USA, 2016; pp. 4104–4113. [\[CrossRef\]](#)
12. Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; Dai, B. Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024*; IEEE: New York, NY, USA, 2024; pp. 20654–20664. [\[CrossRef\]](#)
13. Yu, Z.; Chen, A.; Huang, B.; Sattler, T.; Geiger, A. Mip-Splatting: Alias-Free 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024*; IEEE: New York, NY, USA, 2024; pp. 19447–19456. [\[CrossRef\]](#)
14. Ye, Z.; Li, W.; Liu, S.; Qiao, P.; Dou, Y. AbsGS: Recovering Fine Details in 3D Gaussian Splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM), Melbourne, VIC, Australia; ACM: New York, NY, USA, 2024*; pp. 1053–1061. [\[CrossRef\]](#)
15. Zhang, F.; Wang, Y.; Yi, B.; Ma, J. Adaptive Deletion of Gaussian Ellipsoids in 3D Gaussian Splatting. *Mathematics* **2026**, *14*, 1197. [\[CrossRef\]](#)
16. Bi, L.; Tan, M.; Zhao, Z.; Li, J.; Wang, X. Fast 3D Gaussian Reconstruction for Open-Pit Mine Teleoperated Excavation via Monocular-LiDAR Fusion. *Mathematics* **2026**, *14*, 1191. [\[CrossRef\]](#)
17. evTwoCong, R.; Cui, Z.; Wang, Y.; Ke, W.; Lyv, W.; Zhang, Y.; Sheng, H. Combining Gaussian and Pixel Representation for Light Field View Reconstruction. *IEEE Trans. Image Process.* **2026**, *35*, 4177–4191. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Zhang, B.; Yu, X.; Cai, W.; Zhou, X.; Wang, B.; Zhang, T. A Structure-Aware Triangular Mesh Simplification Based on Graph Neural Network (GNN)-Guided Quadric Error Metrics (QEM). *Mathematics* **2026**, *14*, 1610. [\[CrossRef\]](#)
19. Zhu, X.; Zhao, Y.; You, L. Neuronal Mesh Reconstruction from Image Stacks Using Implicit Neural Representations. *Mathematics* **2025**, *13*, 1276. [\[CrossRef\]](#)
20. Lorensen, W.E.; Cline, H.E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Seminal Graphics: Pioneering Efforts that Shaped the Field*; Association for Computing Machinery: New York, NY, USA, 1998; pp. 347–353. [\[CrossRef\]](#)
21. Zhu, Z.; Iglesias, A.; Zhou, L.; You, L.; Zhang, J. PDE-Based 3D Surface Reconstruction from Multi-View 2D Images. *Mathematics* **2022**, *10*, 542. [\[CrossRef\]](#)
22. Park, J.J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019*; IEEE: New York, NY, USA, 2019; pp. 165–174. [\[CrossRef\]](#)
23. Yariv, L.; Kasten, Y.; Moran, D.; Galun, M.; Atzmon, M.; Basri, R.; Lipman, Y. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 2492–2502.
24. Oechsle, M.; Peng, S.; Geiger, A. UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021*; IEEE: New York, NY, USA, 2021; pp. 5569–5579. [\[CrossRef\]](#)
25. Niemeyer, M.; Mescheder, L.; Oechsle, M.; Geiger, A. Differentiable Volumetric Rendering: Learning Implicit 3D Representations Without 3D Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020*; IEEE: New York, NY, USA, 2020; pp. 3501–3512. [\[CrossRef\]](#)
26. Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; Wang, W. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-View Reconstruction. In *Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*; Curran Associates, Inc.: Red Hook, NY, USA, 2021; pp. 27171–27183.
27. Yariv, L.; Gu, J.; Kasten, Y.; Lipman, Y. Volume Rendering of Neural Implicit Surfaces. In *Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*; Curran Associates, Inc.: Red Hook, NY, USA, 2021; pp. 4805–4815.
28. Li, Z.; Müller, T.; Evans, A.; Taylor, R.H.; Unberath, M.; Liu, M.Y.; Lin, C.H. Neuralangelo: High-Fidelity Neural Surface Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023*; IEEE: New York, NY, USA, 2023; pp. 8456–8465. [\[CrossRef\]](#)
29. Yu, Z.; Sattler, T.; Geiger, A. Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes. *ACM Trans. Graph.* **2024**, *43*, 271. [\[CrossRef\]](#)
30. Guédon, A.; Gomez, D.; Maruani, N.; Gong, B.; Drettakis, G.; Ovsjanikov, M. MILo: Mesh-In-the-Loop Gaussian Splatting for Detailed and Efficient Surface Reconstruction. *ACM Trans. Graph.* **2025**, *44*, 223. [\[CrossRef\]](#)

31. Wang, L.; Zhou, F.; Yu, B.; Cao, P.; Yin, J. OMEGAS: Object Mesh Extraction From Large Scenes Guided by Gaussian Segmentation. *IEEE Trans. Circuits Syst. Video Technol.* **2026**, *36*, 792–802. [\[CrossRef\]](#)
32. Chen, H.; Li, C.; Wang, Y.; Lee, G.H. NeuSG: Neural Implicit Surface Reconstruction with 3D Gaussian Splatting Guidance. *arXiv* **2023**, arXiv:2312.00846. [\[CrossRef\]](#)
33. Guédon, A.; Lepetit, V. 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), Seattle, WA, USA, 16–22 June 2024*; IEEE: New York, NY, USA, 2024; pp. 5354–5363. [\[CrossRef\]](#)
34. Wolf, Y.; Bracha, A.; Kimmel, R. GS2Mesh: Surface Reconstruction from Gaussian Splatting via Novel Stereo Views. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2024; pp. 207–224. [\[CrossRef\]](#)
35. Chen, D.; Li, H.; Ye, W.; Wang, Y.; Xie, W.; Zhai, S.; Wang, N.; Liu, H.; Bao, H.; Zhang, G. PGSR: Planar-Based Gaussian Splatting for Efficient and High-Fidelity Surface Reconstruction. *IEEE Trans. Vis. Comput. Graph.* **2025**, *31*, 6100–6111. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Nguyen, D.M.; Avenhaus, M.; Lindemeier, T. GS-2M: Material-Aware Gaussian Splatting for High-Fidelity Mesh Reconstruction. *Comput. Graph. Forum* **2026**, e70347. [\[CrossRef\]](#)
37. Curless, B.; Levoy, M. A Volumetric Method for Building Complex Models from Range Images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*; ACM: New York, NY, USA, 1996; pp. 303–312. [\[CrossRef\]](#)
38. Rogge, M.; Stricker, D. Object-Centric 2D Gaussian Splatting: Background Removal and Occlusion-Aware Pruning for Compact Object Models. In *Proceedings of the 14th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, Porto, Portugal, 23–25 February 2025; pp. 519–530. [\[CrossRef\]](#)
39. Ravi, N.; Gabeur, V.; Hu, Y.T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. SAM 2: Segment Anything in Images and Videos. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, Singapore, 24–28 April 2025.
40. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023*; IEEE: New York, NY, USA, 2023; pp. 3992–4003. [\[CrossRef\]](#)
41. Jensen, R.; Dahl, A.; Vogiatzis, G.; Tola, E.; Aanaes, H. Large Scale Multi-View Stereopsis Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014*; IEEE: New York, NY, USA, 2014; pp. 406–413. [\[CrossRef\]](#)
42. Yao, Y.; Luo, Z.; Li, S.; Zhang, J.; Ren, Y.; Zhou, L.; Fang, T.; Quan, L. BlendedMVS: A Large-Scale Dataset for Generalized Multi-View Stereo Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020*; IEEE: New York, NY, USA, 2020; pp. 1787–1796. [\[CrossRef\]](#)
43. Barron, J.T.; Mildenhall, B.; Verbin, D.; Srinivasan, P.P.; Hedman, P. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022*; IEEE: New York, NY, USA, 2022; pp. 5460–5469. [\[CrossRef\]](#)
44. Laine, S.; Hellsten, J.; Karras, T.; Seol, Y.; Lehtinen, J.; Aila, T. Modular Primitives for High-Performance Differentiable Rendering. *ACM Trans. Graph.* **2020**, *39*, 194. [\[CrossRef\]](#)
45. Wang, Z.; Bovik, A.C. Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measures. *IEEE Signal Process. Mag.* **2009**, *26*, 98–117. [\[CrossRef\]](#)
46. Darmon, F.; Bascle, B.; Devaux, J.C.; Monasse, P.; Aubry, M. Improving Neural Implicit Surfaces Geometry with Patch Warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022*; IEEE: New York, NY, USA, 2022; pp. 6250–6259. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.