DUALGS: UNIFIED GEOMETRY AND APPEARANCE MODELING WITH DUAL GAUSSIAN FIELDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in Gaussian Splatting have shown remarkable progress in neural scene representation and rendering. However, existing methods often face an inherent trade-off between geometric accuracy and photometric fidelity. While geometry-oriented approaches such as 2DGS focus on reconstructing thin and precise surfaces, they often suffer from limited rendering expressiveness. Conversely, appearance-oriented methods like 3DGS achieve photorealistic novel view synthesis but tend to produce blurred or inconsistent surfaces. In this work, we present DualGS, a unified framework that bridges this gap through a coarse-to-refine dual-branch design. Specifically, we introduce two specialized Gaussian fields, one dedicated to surface reconstruction and the other to appearance refinement, which are jointly optimized through edge-aware multi-view consistency constraints. The surface branch is guided by edge-aware photometric and feature consistency losses that leverage cross-view geometry cues, while the appearance branch enhances visual fidelity via rendering supervision. Furthermore, we introduce a cross-branch consistency mechanism, allowing the appearance field to provide auxiliary supervision for stabilizing surface optimization. Extensive experiments on standard benchmarks demonstrate that our method significantly improves reconstruction accuracy and rendering realism over state-of-the-art methods, achieving precise surfaces and high-fidelity appearance simultaneously. We will release our source code.

1 INTRODUCTION

Novel-view synthesis and surface reconstruction from multi-view images have seen remarkable progress in recent years, driven by advances in neural rendering and explicit scene representations. Among them, the representation of scenes as collections of oriented Gaussian primitives, popularly known as 3D Gaussian Splatting (3DGS) Kerbl et al. (2023), has emerged as a compelling paradigm. It explicitly models the scene by anisotropic 3D Gaussians, and leverages splatting-based rasterization Zwicker et al. (2001) to render novel views at high fidelity and efficiency. Nonetheless, despite its success in rendering quality, 3DGS exhibits limitations in precise surface geometry: the volumetric nature of the Gaussians and their unstructured placement often lead to geometry inconsistency, particularly around thin surfaces and edges. Recent work on 2DGS Huang et al. (2024) tackles this geometry shortcoming by collapsing the volume into oriented planar Gaussian disks aligned with surfaces. The planar representation enforces stronger surface alignment and normal consistency, enabling more accurate geometry recovery. However, this geometric focus comes at a cost: the expressiveness of 2D splats for view-dependent appearance and volumetric structures is reduced, which limits novel-view rendering quality and capacity to capture complex appearance phenomena.

Through our experiments in Sec. 4.1 and ablation studies, we can identify that a fundamental tension persists in Gaussian-based scene representations. Methods emphasizing photorealistic rendering, such as 3DGS, often achieve impressive view synthesis quality but at the cost of geometric precision. Conversely, approaches designed for accurate geometry recovery, such as 2DGS, tend to weaken rendering richness and lose volumetric expressiveness. When both objectives share a unified representation, introducing regularization or structural constraints to favor one inevitably hinders the other. This intrinsic conflict, *i.e.*, balancing geometric accuracy and appearance fidelity, has emerged as a central challenge in Gaussian Splatting-based reconstruction and rendering frameworks.

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

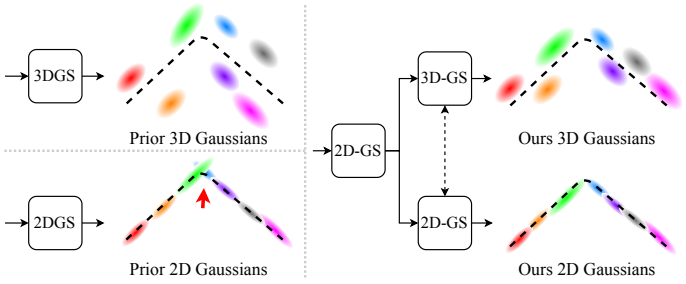

Figure 1: 3DGS favors appearance fidelity but suffers from geometric inconsistency, while 2DGS improves surface alignment at the cost of appearance and edge fidelity. Our cooperative 2D-3D design preserves both geometric and appearance fidelity.

In this work, as shown in Fig. 1, we seek to bridge this gap by designing a unified pipeline that can simultaneously deliver high-fidelity appearance and accurate surface reconstruction, with a particular emphasis on boundary and edge details, which are often overlooked yet crucial for downstream mesh-based rendering and novel-view realism. To reconcile these conflicting properties, we propose a two-stage dual-branch Gaussian splatting framework that initializes from 2D Gaussians and jointly optimizes both 2D and 3D Gaussians. Specifically, starting from a 2DGS-inspired initialization that provides a coarse surface alignment, we then refine the representation into two interacting branches: a surface branch, optimized to recover fine geometry (normals, depth, sharp boundaries) and an appearance branch, optimized for photorealistic novel-view synthesis. Critically, rather than treating these branches independently, we enforce cross-branch supervisory coupling so that the appearance branch guides the surface branch via image-gradient consistency and neighboring variation losses, and the surface branch ensures stable geometry while preserving rendering capacity. In this way, we exploit the complementary strengths of both 2D and 3D Gaussian representations: the geometric alignment of surface-aligned splats, and the rich view-dependent expressiveness of volumetric Gaussians. This collaborative scheme enables accurate surface alignment while maintaining high-fidelity appearance, particularly improving the representation of boundary regions. Extensive experiments on standard benchmarks demonstrate that our method achieves significantly improved surface accuracy while maintaining or improving novel-view synthesis quality compared to baselines. Our contributions are summarized as follows:

- We present a coarse-to-refine dual-branch Gaussian splatting framework that initializes with geometry-aware primitives and then refines them into interacting surface and appearance branches, achieving both accurate reconstruction and high-fidelity rendering.
- We introduce cross-branch supervision that transfers image-gradient consistency and variation cues from the appearance branch to the surface branch, enhancing boundary sharpness and cross-branch coherence.
- We demonstrate state-of-the-art results on several standard benchmarks showing improved boundary accuracy, geometric alignment, and rendering fidelity.

2 Related Work

Neural rendering techniques. NeRF Mildenhall et al. (2020) represents scenes as continuous volumetric densities with view-dependent colors, enabling high-fidelity novel view synthesis. However, it is slow due to extensive per-ray point sampling and the limited scalability of the global MLP. Recent methods Müller et al. (2022); Xu et al. (2023); Chen et al. (2022); Fridovich-Keil et al. (2022) address this by learning spatial features locally, significantly improving efficiency. Alternatively, rasterizing geometric primitives such as point clouds Aliev et al. (2020); Yifan et al. (2019), and voxel or sparse tensor grids Sun et al. (2022); Barron et al. (2022) provides computational flexibility but often suffers from discontinuities and outliers. Incorporating neural features with volumetric rendering Xu et al. (2022) can recover finer details, though at the cost of additional computation. 3DGS Kerbl et al. (2023) models scenes as sparse clouds of anisotropic Gaussians, achieving efficient rendering. Follow-ups enhance visual fidelity, robustness under sparse views Yang et al. (2024a); Niemeyer et al. (2024), and support dynamic scenes, relighting, and animation Luiten et al. (2023); Yang et al. (2024b); Gao et al. (2023); Ye et al. (2023). Geometry-aware variants Li et al. (2024a;b) and hybrid designs Fan et al. (2024); Lu et al. (2024); Zhou et al. (2025) further improve convergence and sparse-view performance, yet most 3DGS methods focus on appearance rather than explicit surface geometry.

Surface reconstruction from Gaussians. Reconstructing surfaces from 3DGS is challenging due to its unstructured nature and RGB-only supervision. Early approaches convert Gaussians to volu-

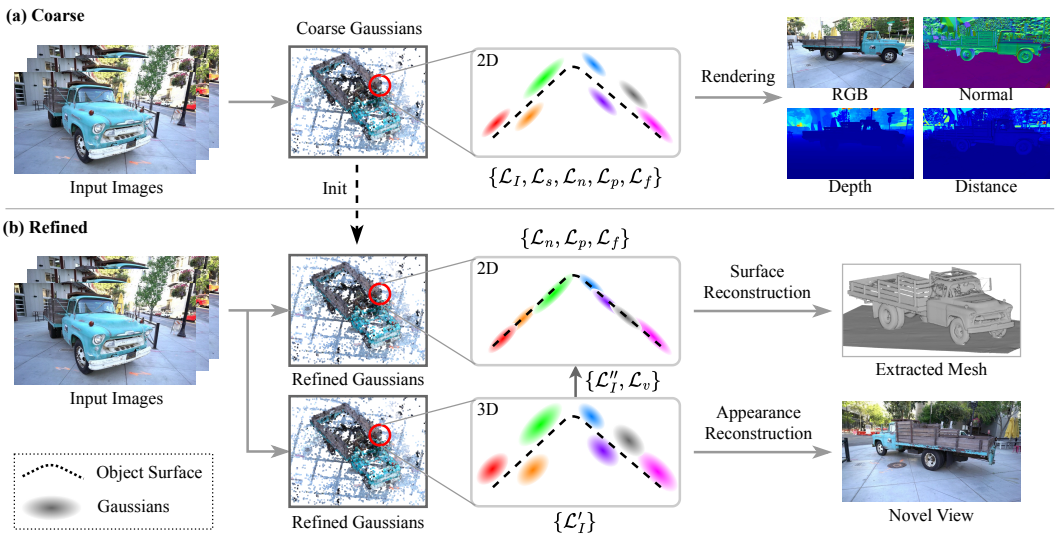

Figure 2: Overview of our framework. **(a) Coarse stage:** multi-view inputs initialize coarse 2D Gaussians which are jointly optimized with losses $\{\mathcal{L}_I, \mathcal{L}_s, \mathcal{L}_n, \mathcal{L}_p, \mathcal{L}_f\}$, producing initial geometry and preview renderings. **(b) Refined stage:** the coarse primitives are split into two specialized yet interacting branches. The *appearance* branch is trained solely with the color loss $\mathcal{L}_I'$ to maximize photometric fidelity. The *surface* branch is refined under geometry-oriented constraints $\{\mathcal{L}_n, \mathcal{L}_p, \mathcal{L}_f\}$, while receiving cross-branch supervision from the appearance renderings via $\{\mathcal{L}_I'', \mathcal{L}_v\}$.

metric densities or opacities such as SuGaR Guédon & Lepetit (2024) and GOF Yu et al. (2024b), or project them as 2D surfels for TSDF or Poisson fusion such as 2DGS Huang et al. (2024) and GSurfels Dai et al. (2024). Recent works integrate signed or unsigned distance fields to guide Gaussian placement. GSDF Yu et al. (2024a) and 3DGSR Lyu et al. (2024) jointly optimize SDFs with Gaussian parameters for smoother surfaces, while GS-Pull Zhang et al. (2024b) and GaussianUDF Li et al. (2025c) leverage gradients of the distance fields to pull Gaussians toward the zero-level set. Other methods Chen et al. (2024b); Wang et al. (2024); Huang et al. (2025); Han et al. (2025); Xu et al. (2025) use depth or normal priors derived from geometry foundation models Eftekhar et al. (2021); Bae & Davison (2024) to refine geometry. GausSurf Wang et al. (2024) integrate MVS guidance in texture-rich regions and normals in textureless areas. PGSR Chen et al. (2024a) further incorporates exposure compensation and planar regularization to handle photometric ambiguities.

Despite these advances, existing GS-based reconstruction methods still struggle with accurate geometry under challenging scenes. While flattening Gaussians and using surfel-based representations Guédon & Lepetit (2024); Huang et al. (2024) improve alignment, they often result in overly smooth geometry and indistinct boundaries. SDF-based approaches Zhang et al. (2024b); Lyu et al. (2024) require extra MLPs or grids, increasing complexity and limiting large-scale reconstruction. Moreover, reliance on external geometry priors or densification Wang et al. (2024); Dai et al. (2024); Xu et al. (2025) is constrained by initialization quality. Our method addresses these limitations by explicitly dividing geometry and appearance learning into dedicated branches, and placing a strong emphasis on edge and boundary detail via multi-view and gradient-based constraints. This yields a more balanced solution that simultaneously advances surface accuracy and view rendering fidelity.

3 METHOD

Preliminary. 3DGS Kerbl et al. (2023) explicitly represents a scene as a set of anisotropic 3D Gaussians that can be rendered from arbitrary viewpoints through a splatting-based rasterization process. Each Gaussian is parameterized by its mean position $\boldsymbol{\mu} \in \mathbb{R}^3$, covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$, color, and opacity. The covariance $\boldsymbol{\Sigma}$ can be factorized as: $\boldsymbol{\Sigma} = \boldsymbol{RSS}^\top \boldsymbol{R}^\top$, where $\boldsymbol{R}$ is a rotation matrix composed of three orthogonal eigenvectors, and $\boldsymbol{S}$ is a diagonal matrix encoding the scale along these directions. This decomposition corresponds to an ellipsoidal representation in 3D space: $\boldsymbol{R}$ defines the principal axes, while the diagonal entries of $\boldsymbol{S}$ determine the axis lengths. Given the smallest scale direction, we define the corresponding axis as the surface normal $\boldsymbol{n}$ of the Gaussian.

For a view with rotation $\boldsymbol{R}_c$ and center $\boldsymbol{T}_c$ in world coordinates, the distance from a local tangent plane to the view center is computed as: $\boldsymbol{d} = (\boldsymbol{R}_c^\top(\boldsymbol{\mu} - \boldsymbol{T}_c))^\top(\boldsymbol{R}_c^\top \boldsymbol{n})$. The depth $\boldsymbol{z}$ of a pixel $\boldsymbol{p}$ (in homogeneous coordinates) is then obtained by intersecting the viewing ray with this local plane: $\boldsymbol{z} = \boldsymbol{d}/(\boldsymbol{R}_c^\top \boldsymbol{n})^\top \boldsymbol{K}^{-1}\boldsymbol{p}$, where $\boldsymbol{K}$ denotes the intrinsic matrix of the camera. Per-pixel depth, normal, and distance maps can then be rendered via α-blending across all visible Gaussians. In our framework, we introduce a 2D Gaussian variant that aligns Gaussians more tightly along the underlying surface, facilitating consistent normal estimation and improved edge representation.

Overview. As shown in Fig. 2, our framework follows a coarse-to-refine design. In the coarse stage, the initial Gaussians are constructed and optimized with rendering and geometry losses for a stable representation. In the refined stage, the Gaussians are divided into an appearance branch for high-quality rendering and a surface branch for accurate geometry recovery. Rendered images from the appearance branch provide cross-branch supervision to the surface branch, enabling geometry-aware refinement without degrading rendering fidelity.

3.1 Learning Geometry for Surface Reconstruction

Scale regularization for surface alignment. The covariance matrix of each 3D Gaussian defines its ellipsoidal shape. To better align Gaussians with local surfaces, we flatten them along the direction of the smallest scale in $\boldsymbol{S} = \mathrm{Diag}(s_1, s_2, s_3)$, effectively converting the ellipsoid into a thin disk near the surface Chen et al. (2023); Huang et al. (2024); Zhang et al. (2024b). This is achieved by minimizing the smallest scale factor for each Gaussian:

$$\mathcal{L}_s = \left\| \min(s_1, s_2, s_3) \right\|_1 . \tag{1}$$

Color rendering. Following prior works Kerbl et al. (2023); Huang et al. (2024), we employ a color reconstruction objective that combines an ℓ_1 loss with a structural similarity term:

$$L_{\text{rgb}}(\tilde{\boldsymbol{I}}, \boldsymbol{I}) = (1-\beta)\left\|\tilde{\boldsymbol{I}} - \boldsymbol{I}\right\|_1 + \beta L_{\text{SSIM}}(\tilde{\boldsymbol{I}}, \boldsymbol{I}) , \tag{2}$$

where $\tilde{\boldsymbol{I}}$ and $\boldsymbol{I}$ denote the rendered and ground-truth images, respectively, and β balances the two terms. While this formulation effectively maintains global image fidelity, it often fails to capture sharp boundaries and fine textures, leading to over-smoothed reconstructions. To better preserve high-frequency details, we introduce an edge-aware rendering loss that incorporates gradient-level supervision:

$$\mathcal{L}_I = L_{\text{rgb}}(\tilde{\boldsymbol{I}}, \boldsymbol{I}) + \left\|\nabla\tilde{\boldsymbol{I}} - \nabla\boldsymbol{I}\right\|_1 , \tag{3}$$

where $\nabla\boldsymbol{I}$ denotes the normalized image gradient. By leveraging image gradients, this loss sharpens object boundaries and improves fine structural recovery.

• **Normal constraint.** To promote geometrically consistent surfaces, we constrain the rendered normals of Gaussian primitives to agree with those estimated from the depth map. While this alignment encourages local surface fidelity, it can become unreliable near object boundaries where normal directions are often ambiguous due to sparse neighborhood support. We therefore incorporate image gradient cues as indicators of structural discontinuities and use an edge-aware alignment loss:

$$\mathcal{L}_n^{\text{align}} = \xi \cdot \left\|\hat{\boldsymbol{N}} - \tilde{\boldsymbol{N}}\right\|_1 , \quad \xi = (1 - \nabla\boldsymbol{I})^2, \tag{4}$$

where $\hat{\boldsymbol{N}}$ is derived from local depth gradients Huang et al. (2024) and $\tilde{\boldsymbol{N}}$ denotes the rendered normal field. The weighting term ξ adaptively suppresses unreliable gradients near edges, ensuring valid and edge-aware normal supervision.

Normal smoothness. Although $\mathcal{L}_n^{\text{align}}$ constrains overall orientation, it may still yield unstable directions in smooth, textureless regions. To regularize these areas, we penalize spatial fluctuations of the rendered normal field:

$$\mathcal{L}_n^{\text{smooth}} = \left\|\zeta \cdot \nabla\tilde{\boldsymbol{N}}\right\|_1 , \quad \zeta = [\nabla\boldsymbol{I} < 0.1] , \tag{5}$$

which enforces local coherence and retains edge sharpness.

Neighbor consistency regularization. To further ensure geometric consistency, we add a neighboring variation term inspired by total variation regularization:

$$\mathcal{L}_n^{\text{anv}} = \sum_{k,(i,j)}^{\boldsymbol{I}} \xi_k \, \mathcal{R}\left(\left|\hat{\boldsymbol{N}}_k - \hat{\boldsymbol{N}}_{(i,j)}\right| - \eta^2\right)\left[\left|\tilde{\boldsymbol{N}}_k - \tilde{\boldsymbol{N}}_{(i,j)}\right| - \eta\right], \tag{6}$$

where k indexes neighboring pixels of $(i, j) \in \boldsymbol{I}$, and η distinguishes true surface edges from noise. $\mathcal{R}(\cdot)$ is the ReLU function, and $[\cdot]$ denotes the Iverson bracket.

Overall normal loss. Combining the above components, our total normal constraint is formulated as:

$$\mathcal{L}_n = \mathcal{L}_n^{\text{align}} + \mathcal{L}_n^{\text{smooth}} + \mathcal{L}_n^{\text{anv}} , \quad (7)$$

which promotes edge-aware, spatially coherent, and detail-preserving surface normals.

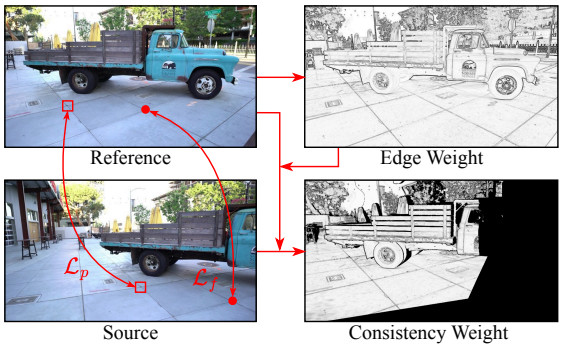

Figure 3: Overview of the multi-view geometry. The photometric loss $\mathcal{L}_p$ and feature loss $\mathcal{L}_f$ are computed between the reference and source views using homography-based warping. Edge and consistency weights are employed to emphasize boundary consistency while filtering out unreliable correspondences.

• **Multi-view geometry.** To enhance geometric precision, we introduce multi-view photometric and feature consistency constraints inspired by multi-view stereo techniques Schonberger & Frahm (2016); Campbell et al. (2008); Fu et al. (2022). These constraints refine surface geometry by projecting points across views and enforcing appearance coherence. Specifically, we employ a patch-based photometric loss and a point-based feature loss to exploit multi-view cues and disambiguate geometry near object boundaries, as shown in Fig. 3.

Photometric consistency. Let $\boldsymbol{I}_r$ and $\boldsymbol{I}_s$ denote the reference and source images, $\boldsymbol{R}_{rs}$ and $\boldsymbol{T}_{rs}$ denote the relative rotation and translation between the reference and source views. For a pixel $\boldsymbol{p}_r$ in the reference view, a local tangent plane is defined by normal $\boldsymbol{n}_r$ and distance $\boldsymbol{d}_r$. Using the homography matrix:

$$\boldsymbol{H}_{rs} = \boldsymbol{K}_s \left(\boldsymbol{R}_{rs} - \frac{\boldsymbol{T}_{rs} \boldsymbol{n}_r^\top}{\boldsymbol{d}_r} \right) \boldsymbol{K}_r^{-1} , \quad (8)$$

the pixel is projected to $\boldsymbol{p}_s^r = \boldsymbol{H}_{rs} \boldsymbol{p}_r$ in the source view. Based on local planarity, the reference patch $\boldsymbol{P}_r^{\boldsymbol{p}_r}$ is warped to the source patch $\boldsymbol{P}_s^{\boldsymbol{p}_s^r}$ via $\boldsymbol{H}_{rs}$, and photometric consistency is enforced by maximizing patch correlation:

$$\mathcal{L}_p = \frac{1}{\upsilon} \sum_{\boldsymbol{p}_r \in \boldsymbol{I}_r} \mathcal{V}_{rs}^{\boldsymbol{p}_r} \, \mathcal{O}_{rs}^{\boldsymbol{p}_r} \left(1 - \mathcal{H}(\boldsymbol{P}_r^{\boldsymbol{p}_r}, \boldsymbol{P}_s^{\boldsymbol{p}_s^r})\right), \quad (9)$$

where $\mathcal{H}(\cdot)$ is the normalized cross-correlation Yoo & Han (2009), υ is the number of valid pixels, $\mathcal{V}$ is the validity mask, and $\mathcal{O}$ is the edge-aware consistency weight.

Validity estimation. To handle occlusions and view-dependent visibility, we compute a binary mask indicating whether a reference pixel remains valid in a source image. Given a pixel $\boldsymbol{p}_r$ with rendered depth $\boldsymbol{z}_r$, its 3D position in the reference view and projection in the source view are:

$$\boldsymbol{x}_r = \boldsymbol{z}_r \boldsymbol{K}^{-1} \boldsymbol{p}_r , \quad \boldsymbol{p}_s' = \mathcal{T}(\boldsymbol{K} \boldsymbol{M}_s \boldsymbol{M}_r^{-1} \boldsymbol{x}_r), \quad (10)$$

where $\boldsymbol{M}$ denotes the camera extrinsic and $\mathcal{T}(\cdot)$ projects 3D coordinates to 2D pixels. A pixel is marked valid if its projection lies within the valid image bounds:

$$\mathcal{V}_{rs}^{\boldsymbol{p}_r} = \left\{ \boldsymbol{p}_r \mid (0,0) < \boldsymbol{p}_s' < (W, H) \right\}, \quad (11)$$

where (W, H) are the image dimensions.

Edge-aware consistency weight. To handle misaligned projections caused by depth inconsistencies Chen et al. (2024a), we introduce a consistency weight $\mathcal{O}$ to downweight unreliable correspondences. For a reference point $\boldsymbol{x}_r$ with source-view depth $\boldsymbol{z}_s$, the reprojection error is computed as:

$$\mathcal{P}(\boldsymbol{p}_r) = \|\boldsymbol{p}_r - \boldsymbol{p}_r'\|_2 , \quad \boldsymbol{p}_r' = \mathcal{T}(\boldsymbol{K} \boldsymbol{M}_r \boldsymbol{M}_s^{-1} \boldsymbol{x}_s), \quad \boldsymbol{x}_s = \mathcal{I}(\boldsymbol{M}_s \boldsymbol{M}_r^{-1} \boldsymbol{x}_r) \boldsymbol{z}_s , \quad (12)$$

where $\mathcal{I}(\cdot)$ denotes the depth-normalization of a 3D coordinate and $\boldsymbol{p}_r'$ is the reprojected pixel in the reference view. The edge-aware consistency weight is defined as:

$$\mathcal{O}_{rs}^{\boldsymbol{p}_r} = \begin{cases} \xi + \exp(-\mathcal{P}(\boldsymbol{p}_r)), & \text{if } \mathcal{P}(\boldsymbol{p}_r) < 1, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

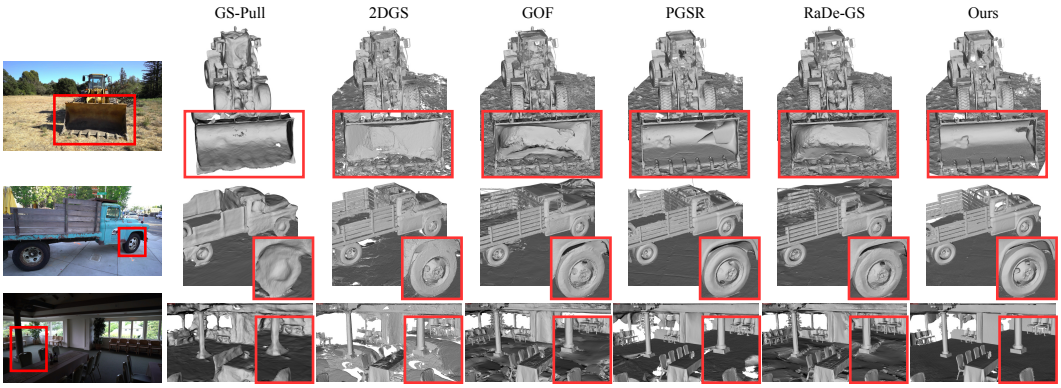

Figure 4: Surface reconstruction results on the TNT dataset. Our method handles varying illumination and weak textures, producing fine details and accurate planar regions.

Smaller reprojection errors yield higher weights, indicating reliable correspondences, while large errors from occluded or inconsistent regions, as well as boundary areas prone to occlusion, are suppressed (see Fig. 3).

Feature consistency. Photometric losses are commonly used to align views, but they may fail under illumination changes, noise, or low-texture regions, since the same 3D point may appear differently across images. To enhance robustness, we enforce cross-view consistency in a learned feature space. Concretely, we extract dense features $\boldsymbol{F} = f(\boldsymbol{I})$ from a pretrained encoder f Zhang et al. (2023), and denote the reference and source features as $\boldsymbol{F}_r$ and $\boldsymbol{F}_s$, respectively. Our feature consistency loss is defined as:

$$\mathcal{L}_f = \frac{1}{\upsilon} \sum_{\boldsymbol{p}_r \in \boldsymbol{I}_r} \mathcal{V}_{rs}^{\boldsymbol{p}_r} \, \mathcal{O}_{rs}^{\boldsymbol{p}_r} \left| 1 - \mathcal{C}(\boldsymbol{F}_r^{\boldsymbol{p}_r}, \boldsymbol{F}_s^{\boldsymbol{p}_s'}) \right|, \tag{14}$$

where $\mathcal{C}(\cdot)$ computes cosine similarity between feature vectors. By comparing features instead of raw colors, this loss provides a more semantically stable signal for aligning Gaussians across views, improving geometric reconstruction under challenging lighting and low-texture conditions.

• **Coarse loss.** The overall loss for training the coarse Gaussians is formulated as:

$$\mathcal{L}_{\text{Coarse}} = \mathcal{L}_I + \lambda_1 \mathcal{L}_s + (\lambda_2 \mathcal{L}_n + \lambda_3 \mathcal{L}_p + \lambda_4 \mathcal{L}_f), \tag{15}$$

where λ_1-λ_4 are weighting factors tuned on a validation set.

3.2 Learning Photometry for Appearance Reconstruction

To ensure high-fidelity novel view synthesis, we introduce an auxiliary branch based on 3DGS Kerbl et al. (2023), dedicated to appearance reconstruction. This branch is optimized using a standard color rendering loss:

$$\mathcal{L}_I' = L_{\text{rgb}}(\tilde{\boldsymbol{I}}', \boldsymbol{I}), \tag{16}$$

where $\tilde{\boldsymbol{I}}'$ is the rendered image from the appearance branch, and $L_{\text{rgb}}(\cdot)$ follows the definition in Eq. equation 2.

Although the surface branch primarily targets geometric alignment, we encourage its renderings to remain photometrically consistent with those from the appearance branch. To achieve this, the surface branch is jointly supervised by the ground-truth image and the detached output of the appearance branch:

$$L_{\text{rgb}}(\tilde{\boldsymbol{I}}, \tilde{\boldsymbol{I}}', \boldsymbol{I}) = \beta L_{\text{rgb}}(\tilde{\boldsymbol{I}}, \tilde{\boldsymbol{I}}') + (1-\beta) L_{\text{rgb}}(\tilde{\boldsymbol{I}}, \boldsymbol{I}). \tag{17}$$

This strategy enables the surface branch to leverage the photometrically accurate predictions of the appearance branch, improving convergence and stability during alternating optimization. To further preserve high-frequency details, we incorporate gradient-level supervision between the two renderings:

$$\mathcal{L}_I'' = L_{\text{rgb}}(\tilde{\boldsymbol{I}}, \tilde{\boldsymbol{I}}', \boldsymbol{I}) + \left\| \nabla \tilde{\boldsymbol{I}} - \nabla \tilde{\boldsymbol{I}}' \right\|_1 + \left\| \nabla \tilde{\boldsymbol{I}} - \nabla \boldsymbol{I} \right\|_1, \tag{18}$$

encouraging consistent edge sharpness across branches.

Table 1: Quantitative comparison on the DTU dataset using Chamfer Distance. The results are highlighted as 1st, 2nd and 3rd. The lower the better.

		24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**	Time
Implicit	NeRF Mildenhall et al. (2020)	1.90	1.60	1.85	0.58	2.28	1.27	1.47	1.67	2.05	1.07	0.88	2.53	1.06	1.15	0.96	1.49	>12h
	NeuS Wang et al. (2021)	1.00	1.37	0.93	0.43	1.10	0.65	0.57	1.48	1.09	0.83	0.52	1.20	0.35	0.49	0.54	0.84	>12h
	Neuralangelo Li et al. (2023)	0.37	0.72	0.35	0.35	0.87	0.54	0.53	1.29	0.97	0.73	0.47	0.74	0.32	0.41	0.43	0.61	>12h
Explicit	3DGS Kerbl et al. (2023)	2.14	1.53	2.08	1.68	3.49	2.21	1.43	2.07	2.22	1.75	1.79	2.55	1.53	1.52	1.50	1.96	5 m
	SuGaR Guédon & Lepetit (2024)	1.47	1.33	1.13	0.61	2.25	1.71	1.15	1.63	1.62	1.07	0.79	2.45	0.98	0.88	0.79	1.33	∼1 h
	MGSR Zhou et al. (2025)	1.29	2.46	1.60	1.03	1.30	1.12	0.99	1.26	1.21	0.77	0.88	1.45	0.81	1.14	0.87	1.21	15 m
	GSurfels Dai et al. (2024)	0.66	0.93	0.54	0.41	1.06	1.14	0.85	1.29	1.53	0.79	0.82	1.58	0.45	0.66	0.53	0.88	5 m
	GSDF Yu et al. (2024a)	1.01	0.96	0.45	0.37	1.36	0.76	0.76	1.59	1.30	0.83	0.62	1.24	0.42	0.52	0.54	0.85	>1.5h
	2DGS Huang et al. (2024)	0.48	0.91	0.39	0.39	1.01	0.83	0.81	1.36	1.27	0.76	0.70	1.40	0.40	0.76	0.52	0.80	7 m
	GS-Pull Zhang et al. (2024b)	0.51	0.56	0.46	0.39	0.82	0.67	0.85	1.37	1.25	0.73	0.54	1.39	0.35	0.88	0.42	0.75	6 m
	GOF Yu et al. (2024b)	0.50	0.82	0.37	0.37	1.12	0.74	0.73	1.18	1.29	0.68	0.77	0.90	0.42	0.66	0.49	0.74	32 m
	RaDe-GS Zhang et al. (2024a)	0.46	0.73	0.33	0.38	0.79	0.75	0.76	1.19	1.22	0.62	0.70	0.78	0.36	0.68	0.47	0.68	7 m
	MILo Guédon et al. (2025)	0.43	0.74	0.34	0.37	0.80	0.74	0.70	1.21	1.22	0.66	0.62	0.80	0.37	0.76	0.48	0.68	25 m
	GaussianUDF Li et al. (2025c)	0.62	0.67	0.43	0.42	0.83	0.86	0.72	1.20	1.03	0.68	0.61	0.63	0.43	0.56	0.52	0.68	>1 h
	PGSR Chen et al. (2024a)	0.34	0.58	0.29	0.29	0.78	0.58	0.54	1.01	0.73	0.51	0.49	0.69	0.31	0.37	0.38	0.53	15 m
	GausSurf* Wang et al. (2024)	0.35	0.55	0.34	0.34	0.77	0.58	0.51	1.10	0.69	0.60	0.43	0.49	0.32	0.40	0.37	0.52	-
	VA-GS Li et al. (2025b)	0.32	0.49	0.32	0.30	0.77	0.68	0.43	1.05	0.61	0.57	0.36	0.52	0.28	0.33	0.30	0.49	16 m
	GeoSVR Li et al. (2025a)	0.32	0.51	0.30	0.33	0.71	0.48	0.42	1.03	0.62	0.56	0.33	0.46	0.30	0.34	0.32	0.47	29 m
	Ours	0.32	0.44	0.26	0.30	0.72	0.62	0.41	0.95	0.58	0.52	0.37	0.45	0.27	0.33	0.31	0.46	10 m

Table 2: Quantitative comparison on the TNT dataset using F1-score. The higher the better.

		Barn	Caterpillar	Courthouse	Ignatius	Meetingroom	Truck	**Mean**	Time
Implicit	Geo-Neus Fu et al. (2022)	0.33	0.26	0.12	0.72	0.20	0.45	0.35	>12h
	NeuS Wang et al. (2021)	0.29	0.29	0.17	0.83	0.24	0.45	0.38	>12h
	Neuralangelo Li et al. (2023)	0.70	0.36	0.28	0.89	0.32	0.48	0.50	>12h
Explicit	3DGS Kerbl et al. (2023)	0.13	0.08	0.09	0.04	0.01	0.19	0.09	8 m
	DN-Splatter Turkulainen et al. (2025)	0.15	0.11	0.07	0.18	0.01	0.20	0.12	20 m
	SuGaR Guédon & Lepetit (2024)	0.14	0.16	0.08	0.33	0.15	0.26	0.19	∼2 h
	GSurfels Dai et al. (2024)	0.24	0.22	0.07	0.39	0.12	0.24	0.21	5 m
	2DGS Huang et al. (2024)	0.41	0.23	0.16	0.51	0.17	0.45	0.32	8 m
	RaDe-GS Zhang et al. (2024a)	0.43	0.32	0.21	0.69	0.25	0.51	0.40	9 m
	GS-Pull Zhang et al. (2024b)	0.60	0.37	0.16	0.71	0.22	0.52	0.43	18 m
	GOF Yu et al. (2024b)	0.51	0.41	0.28	0.68	0.28	0.59	0.46	40 m
	PGSR Chen et al. (2024a)	0.66	0.44	0.20	0.81	0.33	0.66	0.52	26 m
	VA-GS Li et al. (2025b)	0.71	0.45	0.21	0.82	0.40	0.64	0.54	22 m
	GeoSVR Li et al. (2025a)	0.68	0.49	0.34	0.83	0.37	0.66	0.56	38 m
	Ours	0.71	0.48	0.26	0.83	0.40	0.66	0.56	21 m

In addition to photometric coherence, we also regularize the geometric variation of the rendered depth $\tilde{\boldsymbol{D}}$ and normal $\tilde{\boldsymbol{N}}$ maps from the surface branch, guided by the appearance branch. Following Eq. equation 6, the associated neighboring variation loss is defined as:

$$\mathcal{L}_v = \mathcal{L}_v^{\text{anv}}(\tilde{\boldsymbol{N}}, \tilde{\boldsymbol{I}}') + \mathcal{L}_v^{\text{anv}}(\tilde{\boldsymbol{D}}, \tilde{\boldsymbol{I}}'). \tag{19}$$

• **Refined loss.** Starting from the coarse Gaussians, both the surface and appearance branches are jointly refined. The total refinement objective combines appearance and geometric constraints:

$$\mathcal{L}_{\text{Refined}} = \mathcal{L}_I' + \lambda_5 \mathcal{L}_I'' + \lambda_6 \mathcal{L}_v + (\lambda_2 \mathcal{L}_n + \lambda_3 \mathcal{L}_p + \lambda_4 \mathcal{L}_f), \tag{20}$$

where λ_5 and λ_6 are weights determined through validation. This joint optimization allows the surface and appearance branches to mutually enhance each other, improving both geometric fidelity and rendering realism.

4 Experiments

Implementation details. Our training pipeline generally follows 3DGS Kerbl et al. (2023). The threshold in $\mathcal{L}_n^{\text{anv}}$ is set to $\eta = 0.01$. For surface reconstruction, the model is trained for 20k iterations, with the first 15k used for coarse geometric initialization to stabilize Gaussian primitives, followed by joint refinement of geometry and appearance. For novel view synthesis, an additional 10k iterations are performed to optimize the appearance branch. All experiments are conducted using an NVIDIA RTX 4090 GPU.

Evaluation protocols. We evaluate our approach on the DTU Jensen et al. (2014), Tanks and Temples (TNT) Knapitsch et al. (2017), Mip-NeRF 360 Barron et al. (2022), and Deep Blending Hedman et al. (2018) datasets, which cover diverse indoor and outdoor scenes with complex

geometry and lighting conditions. Camera poses and sparse point clouds are reconstructed using COLMAP Schönberger & Frahm (2016), and input images are downsampled for efficiency. For surface reconstruction, we follow standard practice Huang et al. (2024); Yu et al. (2024b); Chen et al. (2024a); Zhang et al. (2024a), using 15 DTU scenes and 6 TNT scenes. Depth maps are rendered from all training views, and meshes are extracted via TSDF fusion Curless & Levoy (1996). We report Chamfer Distance (CD) on DTU and F1-score on TNT for quantitative evaluation. For novel view synthesis, following 3DGS Kerbl et al. (2023), one out of every eight images is reserved for testing, while the remaining are used for training. We evaluate image quality using PSNR, SSIM, and LPIPS metrics Kerbl et al. (2023); Huang et al. (2024); Zhang et al. (2024b); Chen et al. (2024a).

4.1 Performance Evaluation

Results on DTU. We evaluate our method on the DTU dataset Jensen et al. (2014) following the standard evaluation protocol, where reconstructions are clipped with the provided masks and metrics are computed on the foreground regions. As shown in Table 1, our method achieves the lowest mean Chamfer Distance, outperforming all implicit and explicit baselines across most scenes. Compared with implicit representations such as NeuS Wang et al. (2021) and Neuralangelo Li et al. (2023), our method attains notably higher accuracy with over tenfold faster training speed. Among explicit methods, our approach delivers finer geometric details and cleaner boundaries than recent Gaussian-based techniques Zhang et al. (2024a); Huang et al. (2024); Zhang et al. (2024b), while maintaining competitive efficiency. In contrast to implicit models that typically reconstruct only the foreground, our method produces complete and coherent meshes suitable for downstream rendering and editing applications. The view rendering results on the training set are summarized in Table 3. Our method achieves the best perceptual quality in terms of SSIM. Owing to the simplicity of single-object scenes in DTU, 3DGS Kerbl et al. (2023) achieves higher PSNR, highlighting its strength in handling less complex geometry.

Table 3: Quantitative comparison on DTU and TNT.

	DTU			TNT		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS Kerbl et al. (2023)	37.31	0.952	0.146	27.40	0.882	0.161
2DGS Huang et al. (2024)	34.45	0.938	0.169	24.64	0.840	0.214
GSurfels Dai et al. (2024)	27.04	0.873	0.319	20.10	0.722	0.352
GS-Pull Zhang et al. (2024b)	30.01	0.899	0.226	20.56	0.778	0.311
GOF Yu et al. (2024b)	35.34	0.948	0.131	23.86	0.868	0.166
RaDe-GS Zhang et al. (2024a)	35.37	0.954	0.123	24.33	0.869	0.170
PGSR Chen et al. (2024a)	33.82	0.945	0.148	22.93	0.806	0.204
Ours	36.22	0.956	0.138	27.12	0.889	0.133

Results on TNT. We further evaluate our method on the TNT dataset Knapitsch et al. (2017), which contains large-scale outdoor scenes with complex geometry and illumination variations. As shown in Table 2, our method achieves the highest average F1-score, outperforming all implicit and explicit baselines. Compared with implicit approaches, our method provides more complete and accurate surface reconstructions while being much faster. Among explicit Gaussian-based representations, it surpasses PGSR Zhang et al. (2024a) and RaDe-GS Zhang et al. (2024a) with improved structural consistency, especially in challenging scenes such as Caterpillar and Meetingroom. GS-Pull Zhang et al. (2024b) focuses solely on foreground reconstruction and often produces oversmoothed surfaces. While several Gaussian-based methods converge faster, our approach attains a favorable balance between efficiency and fidelity. A visual comparison of the reconstructed surfaces is provided in Fig. 4. For view synthesis on TNT (see Table 3), our method achieves the best SSIM and LPIPS scores, indicating stable appearance rendering under complex setups.

Table 4: Quantitative comparison on Mip-NeRF 360.

	Outdoor scenes			Indoor scenes		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF Mildenhall et al. (2020)	21.46	0.458	0.515	26.84	0.790	0.370
DeepBlendingHedman et al. (2018)	21.54	0.524	0.364	26.40	0.844	0.261
Instant NGP Müller et al. (2022)	22.90	0.566	0.371	29.15	0.880	0.216
MERF Reiser et al. (2023)	23.19	0.616	0.343	27.80	0.855	0.271
BakedSDF Yariv et al. (2023)	22.47	0.585	0.349	27.06	0.836	0.258
Mip-NeRF 360Barron et al. (2022)	24.47	0.691	0.283	31.72	0.917	0.180
3DGS Kerbl et al. (2023)	24.64	0.731	0.234	30.41	0.920	0.189
SuGaR Guédon & Lepetit (2024)	22.93	0.629	0.356	29.43	0.906	0.225
2DGS Huang et al. (2024)	24.34	0.717	0.246	30.40	0.916	0.195
VCR-GauS Chen et al. (2024b)	24.31	0.707	0.280	30.53	0.921	0.184
GS-Pull Zhang et al. (2024b)	23.76	0.703	0.278	30.78	0.925	0.182
GOF Yu et al. (2024b)	24.82	0.750	0.202	30.79	0.924	0.184
RaDe-GS Zhang et al. (2024a)	24.81	0.750	0.207	30.64	0.921	0.189
PGSR Chen et al. (2024a)	24.45	0.730	0.224	30.41	0.930	0.161
GausSurf Wang et al. (2024)	25.09	0.753	0.212	30.05	0.920	0.183
VA-GS Li et al. (2025b)	25.00	0.760	0.191	30.63	0.933	0.153
GeoSVR Li et al. (2025a)	24.83	0.738	0.218	30.46	0.921	0.172
paper Ours	25.15	0.763	0.185	30.93	0.934	0.148

Results on Mip-NeRF 360. We further evaluate our framework on the Mip-NeRF 360 dataset Barron et al. (2022) for novel view synthesis across large-scale indoor and outdoor scenes. As summa-

rized in Table 4, our method delivers superior results on almost all evaluation metrics, demonstrating improved fidelity and generalization to unseen viewpoints over both neural rendering and Gaussian-based baselines. These results indicate that the coarse Gaussian initialization and the separate appearance branch indeed benefit the quality of view synthesis. Although Mip-NeRF Barron et al. (2022) achieves slightly higher PSNR on certain indoor scenes, its structural consistency (SSIM) and perceptual quality (LPIPS) remain lower. Among Gaussian-based variants, 2DGS Huang et al. (2024), SuGaR Guédon & Lepetit (2024), PGSR Chen et al. (2024a), and GS-Pull Zhang et al. (2024b) show inferior performance, suggesting that constraining Gaussians to 2D or planar forms compromises their appearance reconstruction capability. In contrast, our method preserves the full 3D expressiveness of Gaussian primitives by a separate branch, achieving both geometrically accurate surfaces and photorealistic rendering results.

Results on Deep Blending. We additionally validate our framework on the Deep Blending dataset Hedman et al. (2018), which features real-world indoor scenes with complex illumination, reflections, and imperfect camera calibrations. As shown in Table 5, our method delivers the best rendering performance among recent Gaussian-based methods, with consistent improvements across almost all metrics. Compared with 3DGS Kerbl et al. (2023), 2DGS Huang et al. (2024) and GOF Yu et al. (2024b), our approach better preserves high-frequency textures and geometric consistency across views, leading to more stable and realistic renderings. While appearance-only models often struggle under varying lighting, our geometry-aligned optimization effectively maintains view coherence and avoids artifacts around reflective and shadowed regions. These results further confirm the robustness and generalization ability of our representation in challenging real-world conditions.

Table 5: Quantitative comparison on Deep Blending.

	Dr Johnson			**Playroom**		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS Kerbl et al. (2023)	28.94	0.896	0.248	29.93	0.901	0.244
SuGaR Guédon & Lepetit (2024)	28.71	0.889	0.273	30.12	0.898	0.261
2DGS Huang et al. (2024)	28.89	0.898	0.259	29.88	0.901	0.259
GS-Pull Zhang et al. (2024b)	25.69	0.830	0.387	25.89	0.838	0.375
GOF Yu et al. (2024b)	27.85	0.893	0.257	30.16	0.904	0.242
RaDe-GS Zhang et al. (2024a)	27.83	0.896	0.257	30.04	0.905	0.243
PGSR Chen et al. (2024a)	28.61	0.891	0.251	29.92	0.903	0.243
VA-GS Li et al. (2025b)	29.10	0.900	0.241	30.22	0.905	0.241
Ours	29.06	0.904	0.239	30.32	0.906	0.239

Computational cost. Note that our method does not directly maintain two independent 2DGS and 3DGS models. Both branches are initialized from the same coarse Gaussian set and share equal Gaussian counts. As shown in Table 6, each branch uses about 111k Gaussians (222k total), comparable to other geometry-aware methods such as 2DGS. Unlike vanilla 3DGS, which spreads a large number of Gaussians (393k) throughout the entire volumetric space for rendering, our geometry-driven constraints concentrate Gaussians near the surface, reducing redundancy and yielding compact models (26 MB per branch). The additional memory and training cost mainly arise from loss computation rather than Gaussian duplication. Overall, our current computational cost are comparable to PGSR while achieving better performance; notably, with only half the number of Gaussians, our method can still reach performance comparable to PGSR (see Table 1 and ablation studies).

Table 6: Computational cost, Gaussian count, and memory usage on the DTU dataset.

	Count	GPU Mem.	Model Size	Time
3DGS Kerbl et al. (2023)	392,978	1.11 GB	92.95 MB	5 min
2DGS Huang et al. (2024)	211,820	0.68 GB	47.67 MB	7 min
PGSR Chen et al. (2024a)	252,273	2.14 GB	55.29 MB	15 min
Ours	222,332	2.10 GB	52.58 MB	10 min

5 Conclusion

In this paper, we presented a dual-branch Gaussian framework that effectively bridges the gap between accurate surface reconstruction and high-fidelity rendering. Unlike previous single-representation methods, our approach introduces a geometry-oriented surface branch and an appearance-oriented rendering branch, linked through cross-branch supervision. This design enables geometry learning to exploit photometric cues without degrading rendering quality, achieving a balanced improvement in both geometric accuracy and visual fidelity. With scale regularization, normal alignment, and multi-view geometric constraints, our method achieves coherent and detailed surfaces under challenging conditions. Extensive experiments demonstrate state-of-the-art results in both reconstruction accuracy and visual realism. We believe this framework provides new insights into balancing geometry and appearance learning in Gaussian-based scene representations.

AI USE STATEMENT

In this work, we used generative AI tools solely for language-related assistance, including English grammar correction, improving the readability of the manuscript, and translation between Chinese and English. Generative AI tools were not used to generate research ideas, develop theoretical models, formulate mathematical claims, design or evaluate methodologies or experiments, implement methods, interpret experimental results, or produce scientific content. All AI-assisted text was carefully reviewed and revised by the authors. The authors take full responsibility for the final content of this work, including all claims, analyses, and conclusions.

REFERENCES

Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *European conference on computer vision*, pp. 696–712. Springer, 2020.

Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9535–9545, 2024.

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

Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I 10*, pp. 766–779. Springer, 2008.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pp. 333–350. Springer, 2022.

Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. PGSR: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2024a.

Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3D gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023.

Hanlin Chen, Fangyin Wei, Chen Li, Tianxin Huang, Yunsong Wang, and Gim Hee Lee. VCR-GauS: View consistent depth-normal regularizer for gaussian surface reconstruction. *Advances in Neural Information Processing Systems*, 37:139725–139750, 2024b.

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

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

Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10786–10796, 2021.

Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2024.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5501–5510, 2022.

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

Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3D gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv preprint arXiv:2311.16043*, 2023.

Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5354–5363, 2024.

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

Liang Han, Xu Zhang, Haichuan Song, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. SparseRecon: Neural implicit surface reconstruction from sparse views with feature and depth consistencies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28514–28524, 2025.

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

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 2024.

Han Huang, Yulun Wu, Chao Deng, Ge Gao, Ming Gu, and Yu-Shen Liu. FatesGS: Fast and accurate sparse-view surface reconstruction using gaussian splatting with depth-feature consistency. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 406–413, 2014.

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

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

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3D gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20775–20785, 2024a.

Jiahe Li, Jiawei Zhang, Youmin Zhang, Xiao Bai, Jin Zheng, Xiaohan Yu, and Lin Gu. GeoSVR: Taming sparse voxels for geometrically accurate surface reconstruction. *Advances in Neural Information Processing Systems*, 38:108809–108837, 2025a.

Qing Li, Huifang Feng, Xun Gong, and Yu-Shen Liu. Va-gs: Enhancing the geometric representation of gaussian splatting via view alignment. In *Advances in Neural Information Processing Systems*, volume 38, pp. 173867–173886. Curran Associates, Inc., 2025b.

Shujuan Li, Yu-Shen Liu, and Zhizhong Han. GaussianUDF: Inferring unsigned distance functions through 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025c.

Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. GeoGaussian: Geometry-aware gaussian splatting for scene rendering. In *European Conference on Computer Vision*, pp. 441–457. Springer, 2024b.

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

Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. SparseNeuS: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*, pp. 210–227. Springer, 2022.

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

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

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

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421, 2020.

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

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

Christian Reiser, Rick Szeliski, Dor Verbin, Pratul Srinivasan, Ben Mildenhall, Andreas Geiger, Jon Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5459–5469, 2022.

Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2421–2431. IEEE, 2025.

Jiepeng Wang, Yuan Liu, Peng Wang, Cheng Lin, Junhui Hou, Xin Li, Taku Komura, and Wenping Wang. GausSurf: Geometry-guided 3D gaussian splatting for surface reconstruction. *arXiv preprint arXiv:2411.19454*, 2024.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 16453–16463, 2025.

Linning Xu, Yuanbo Xiangli, Sida Peng, Xingang Pan, Nanxuan Zhao, Christian Theobalt, Bo Dai, and Dahua Lin. Grid-guided neural radiance fields for large urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8296–8306, 2023.

Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-NeRF: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5438–5448, 2022.

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

Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2024b.

Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. BakedSDF: Meshing neural sdfs for real-time view synthesis. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–9, 2023.

Keyang Ye, Tianjia Shao, and Kun Zhou. Animatable 3D gaussians for high-fidelity synthesis of human motions. *arXiv preprint arXiv:2311.13404*, 2023.

Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019.

Jae-Chern Yoo and Tae Hee Han. Fast normalized cross-correlation. *Circuits, Systems and Signal Processing*, 28:819–843, 2009.

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

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

Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. RaDe-GS: Rasterizing depth in gaussian splatting. *arXiv preprint arXiv:2406.01467*, 2024a.

Jingyang Zhang, Shiwei Li, Zixin Luo, Tian Fang, and Yao Yao. Vis-mvsnet: Visibility-aware multi-view stereo network. *International Journal of Computer Vision*, 131(1):199–214, 2023.

Wenyuan Zhang, Yu-Shen Liu, and Zhizhong Han. Neural signed distance function inference through splatting 3D gaussians pulled on zero-level set. *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b.

Qingyuan Zhou, Yuehu Gong, Weidong Yang, Jiaze Li, Yeqi Luo, Baixin Xu, Shuhao Li, Ben Fei, and Ying He. MGSR: 2D/3D mutual-boosted gaussian splatting for high-fidelity surface reconstruction under various light conditions. *International Conference on Computer Vision*, 2025.

Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. EWA volume splatting. In *Proceedings Visualization*, pp. 29–538. IEEE, 2001.

A Baseline Methods

We benchmark our method against representative approaches for both surface reconstruction and novel view synthesis. For Gaussian-based representations, we include 3DGS Kerbl et al. (2023), 2DGS Huang et al. (2024), GOF Yu et al. (2024b), RaDe-GS Zhang et al. (2024a), and PGSR Chen et al. (2024a), which cover the major design choices spanning volumetric rendering, surfel-style

geometry modeling, and geometry-appearance coupling. We further compare with implicit SDF-based reconstructions, such as GS-Pull Zhang et al. (2024b), NeuS Wang et al. (2021) and Neuralangelo Li et al. (2023), which reconstruct surfaces through signed distance fields and render them via volumetric ray marching. For quantitative evaluation, we primarily rely on the metrics reported in the original publications. For qualitative comparisons, visual results are reproduced using the official implementations to ensure consistent evaluation across scenes and datasets. All results on the DTU Jensen et al. (2014), TNT Knapitsch et al. (2017), Mip-NeRF 360 Barron et al. (2022) and Deep Blending Hedman et al. (2018) datasets are obtained from our own runs with the corresponding official codebases. The only exception is 3DGS Kerbl et al. (2023), where we use the publicly released pre-trained models for the Mip-NeRF 360 and Deep Blending datasets provided by the original authors.

B Ablation Studies

We conduct extensive ablation experiments on the DTU dataset to validate the contribution of each component in our framework. The evaluation considers surface reconstruction and view synthesis for both the surface (2D) and appearance (3D) branches. Results are reported in Table 7.

(1) *Scale regularization.* Removing the scale constraint $\mathcal{L}_s$ causes a degradation in both geometric and rendering quality, indicating that $\mathcal{L}_s$ effectively stabilizes Gaussian sizes and prevents over-expansion during optimization, which is crucial for precise surface localization.

(2) *Normal constraint.* Removing the entire normal constraint $\mathcal{L}_n$ leads to degraded surface accuracy, confirming its essential role in guiding Gaussians toward true geometry. When only the smoothness and neighbor variation terms ($\mathcal{L}_n^{\text{smooth}}$, $\mathcal{L}_n^{\text{anv}}$) are removed, performance remains comparable, indicating that the alignment term $\mathcal{L}_n^{\text{align}}$ is the dominant contributor to geometric precision.

(3) *Multi-view geometry.* Omitting the photometric consistency loss $\mathcal{L}_p$ or the feature consistency loss $\mathcal{L}_f$ results in worse reconstruction, reflecting the necessity of multi-view geometry constraints to align Gaussians across different viewpoints and suppress view-dependent artifacts.

(4) *Cross-branch supervision.* We examine the effect of the guidance from the appearance to surface branch. Removing the edge-aware color supervision $\mathcal{L}_I''$ or the neighbor variation loss $\mathcal{L}_v$ individually causes a slight drop in geometric performance but a larger decline in rendering quality, whereas removing both simultaneously results in a significant degradation in both surface and rendering metrics. This confirms that the coupled guidance between branches facilitates geometry-aware color reconstruction and improves surface alignment.

(5) *Edge-aware item.* Removing the edge-aware item from $\mathcal{L}_I$ and $\mathcal{L}_I''$ worsens both surface and rendering metrics, suggesting that incorporating image gradient cues helps preserve fine structural details and sharp boundaries in both geometry and appearance.

(6) *2D/3D branches.* The ablations show that collapsing the two branches degrades both appearance and geometry. (a) Using only the 2DGS-based surface branch (without the 3DGS-based appearance branch) severely reduces rendering quality, demonstrating that the dedicated appearance branch is essential for preserving fine textures and photometric realism while allowing the surface branch to focus purely on geometry. (b) Meanwhile, removing the 2D (surface) branch significantly worsens geometric accuracy, confirming the necessity of our design. The results also show that our method can achieve performance comparable to PGSR using only a single branch.

(7) *Stage-level analysis.* We include an ablation comparing coarse-only and refined-only variants. The coarse stage offers a stable initialization, while the refined stage sharpens boundaries and improves surface alignment and cross-view consistency, leading to consistent gains in both CD and rendering quality.

Overall, the complete model achieves the best balance between geometric fidelity and photometric quality, validating the complementary design of the two branches and the importance of each proposed component. Geometric constraints ($\mathcal{L}_s, \mathcal{L}_n, \mathcal{L}_p, \mathcal{L}_f$) primarily benefit the surface branch by improving structural accuracy and smoothness, while photometric and cross-branch constraints ($\mathcal{L}_I', \mathcal{L}_I'', \mathcal{L}_v$) enhance the appearance branch, leading to more consistent rendering and sharper visual details.

Table 7: Ablations on the DTU dataset. The reconstructed surface from the 2D branch is evaluated using CD, while view synthesis from the 2D and 3D branches is assessed by rendering metrics.

	2D Branch				3D Branch		
	CD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
w/o $\mathcal{L}_s$	0.48	28.31	0.858	0.210	35.43	0.954	0.145
w/o $\mathcal{L}_n$	0.50	**28.60**	**0.861**	**0.201**	35.54	0.954	0.145
w/o $\mathcal{L}_n^{\text{smooth}}$, $\mathcal{L}_n^{\text{anv}}$	0.47	28.49	0.858	0.208	35.51	0.954	0.145
w/o $\mathcal{L}_p$	0.50	28.54	0.854	0.222	35.57	0.951	0.158
w/o $\mathcal{L}_f$	0.49	28.42	0.857	0.209	35.42	0.954	0.145
w/o $\mathcal{L}_I''$	0.47	28.27	0.856	0.214	35.47	0.954	0.144
w/o $\mathcal{L}_v$	**0.46**	28.40	0.858	0.210	35.39	0.954	0.145
w/o $\mathcal{L}_I''$, $\mathcal{L}_v$	0.48	28.27	0.855	0.214	35.46	0.954	0.145
w/o edge item	0.49	28.40	0.858	0.210	35.46	0.954	0.146
w/o 3D branch	0.49	28.24	0.855	0.214	-	-	-
w/o 2D branch	0.53	-	-	-	35.44	0.954	0.145
w/ coarse-only	0.49	27.41	0.846	0.223	-	-	-
w/ refined-only	0.53	28.38	0.853	0.220	35.42	0.953	0.152
Ours	**0.46**	28.40	0.858	0.209	**35.59**	**0.955**	**0.142**

Some components have limited gains in ablations. (1) The proposed modules are task-specific. Geometry-oriented components mainly improve the 2D branch (CD), while appearance-oriented components primarily benefit the 3D branch (rendering quality). Therefore, removing a component may cause only minor changes in the metrics of the other task, resulting in relatively small overall gains. (2) Baselines already perform well on globally smooth regions that dominate the metrics. Our improvements mainly focus on hard cases (boundaries, thin structures, cross-view inconsistencies), which occupy fewer pixels and are underrepresented in global metrics. (3) The proposed components are designed to work synergistically within the coarse-to-fine dual-branch framework rather than independently. Removing a single component yields minor metric changes, while removing coupled supervision causes clear degradation, indicating that gains stem from coordinated supervision. (4) Branch-level (2D/3D) and stage-level (coarse/fine) ablations yield large metric changes, confirming the effectiveness of our framework design.

C More Evaluation Results

The novelty of our method is the explicit separation of geometry and appearance to resolve the conflict in single-branch Gaussian methods, where geometric regularization and photometric rendering compete: strengthening one often degrades the other. Our dual-branch design assigns appearance to an unconstrained 3DGS-style branch and geometry to a 2DGS-style branch, with one-way supervision from appearance to surface. This asymmetric supervision lets geometry leverage stable photometric cues without affecting rendering, which cannot be achieved in a single branch, and consistently improves performance.

C.1 Surface reconstruction

Figs. 5, 6, and 7 present qualitative comparisons across the Deep Blending, DTU, and TNT datasets, respectively. On the Deep Blending dataset Hedman et al. (2018) (Fig. 5), our method produces clean and complete surfaces even under challenging lighting conditions. Competing methods such as 2DGS Huang et al. (2024), GOF Yu et al. (2024b), PGSR Chen et al. (2024a), and RaDe-GS Zhang et al. (2024a) frequently suffer from noisy surfaces, distorted walls, and missing structures around object boundaries. These artifacts are especially visible in low-light regions and cluttered indoor scenes. In contrast, our reconstructions maintain smooth yet detailed geometry, capturing planar structures, furniture contours, and high-frequency textures with fewer distortions.

On the DTU dataset Jensen et al. (2014) (Fig. 6), where objects exhibit simpler shapes but often contain reflective or glossy materials, our approach consistently preserves fine-scale geometric features. While other baseline methods tend to oversmooth surfaces or lose sharp edges (see arrows in Fig. 6), our reconstructions maintain crisp boundaries and faithfully recover curved regions and thin

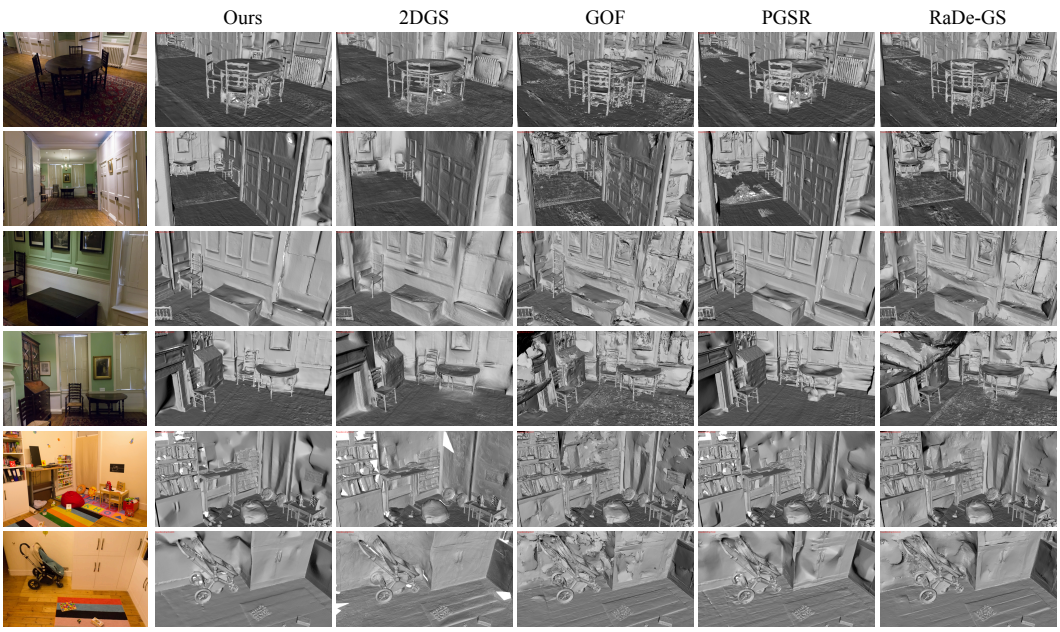

Figure 5: Qualitative comparison of surface reconstruction on the Deep Blending dataset. Our method produces clean, complete, and geometrically stable surfaces under challenging lighting and cluttered indoor layouts, while baseline methods exhibit noise, boundary distortions, and missing structures.

Table 8: Evaluation of sparse view reconstruction on the DTU dataset.

3DGS Kerbl et al. (2023)	2DGS Huang et al. (2024)	PGSR Chen et al. (2024a)	Ours
3.54	1.69	3.83	**1.45**

structures. This demonstrates the advantage of our geometry-appearance decoupling in handling complex reflectance.

For large-scale indoor and outdoor scenes from the TNT dataset Knapitsch et al. (2017) (Fig. 7), our method yields more continuous and watertight meshes. It retains high-frequency structures such as beams, railings, and small holes that are frequently broken or collapsed in baseline results. Methods like 2DGS Huang et al. (2024) and GOF Yu et al. (2024b) often produce non-manifold regions or fragmented geometry, while GS-Pull Zhang et al. (2024b) tends to over-smooth due to its SDF extraction. In contrast, our approach avoids these issues and reconstructs sharp edges and global structures more reliably.

Overall, the above visual comparisons show that our dual-branch framework substantially improves geometric stability, boundary fidelity, and robustness across diverse scene types.

C.1.1 Sparse view reconstruction

To evaluate performance under sparse views, we follow the protocol of SparseNeuS Long et al. (2022) and test on the DTU dataset using only 3 views in the large-overlap setting. As shown in Table 8, our method achieves the lowest Chamfer Distance, outperforming both Gaussian-based and geometry-aware baselines. These results indicate that our geometry-focused branch and cross-branch supervision remain effective even under sparse-view conditions.

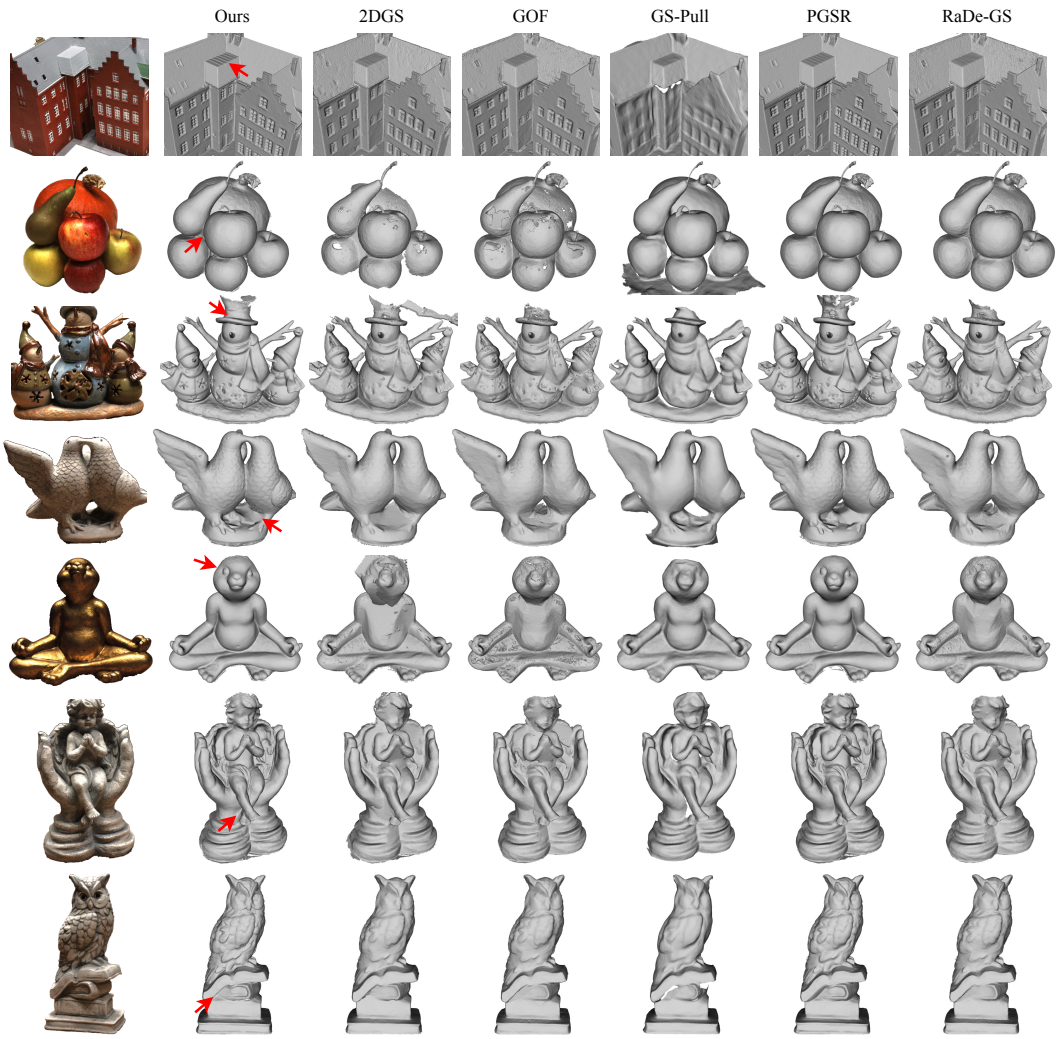

Figure 6: Qualitative comparison of surface reconstruction on the DTU dataset. Our approach excels at recovering fine-grained geometry and handling reflective or glossy materials, preserving sharp contours and intricate details that competing methods fail to reconstruct reliably.

C.2 Appearance Reconstruction

C.2.1 Results on Mip-NeRF 360

To thoroughly assess novel view synthesis performance of our method, we evaluate per-scene PSNR, SSIM, and LPIPS on the Mip-NeRF 360 dataset Barron et al. (2022), as shown in Table 9. Across nearly all scenes, our method attains the best PSNR, SSIM, and LPIPS scores, and achieves the best overall averages. It is worth noting that our method significantly improves LPIPS performance in all scenarios, indicating substantially improved perceptual quality and better preservation of fine textures and high-frequency appearance cues. These results reinforce a trend also observed in the prior experiments: methods that impose strong geometric regularization, such as 2DGS Huang et al. (2024) and GOF Yu et al. (2024b), typically lag behind vanilla 3DGS Kerbl et al. (2023) in rendering fidelity, suggesting a fundamental tension between geometry-focused constraints and appearance learning. In contrast, our dual-branch design avoids this trade-off and allows the appearance branch to retain the expressive rendering capacity of the standard Gaussians. Moreover, qualitative comparisons in Fig. 8 further support this advantage. Our method produces sharper edges, richer textures, and more coherent fine structures, particularly in challenging regions such as vegetation and cluttered backgrounds. Baseline approaches often exhibit noticeable blur or loss of detail in these areas,

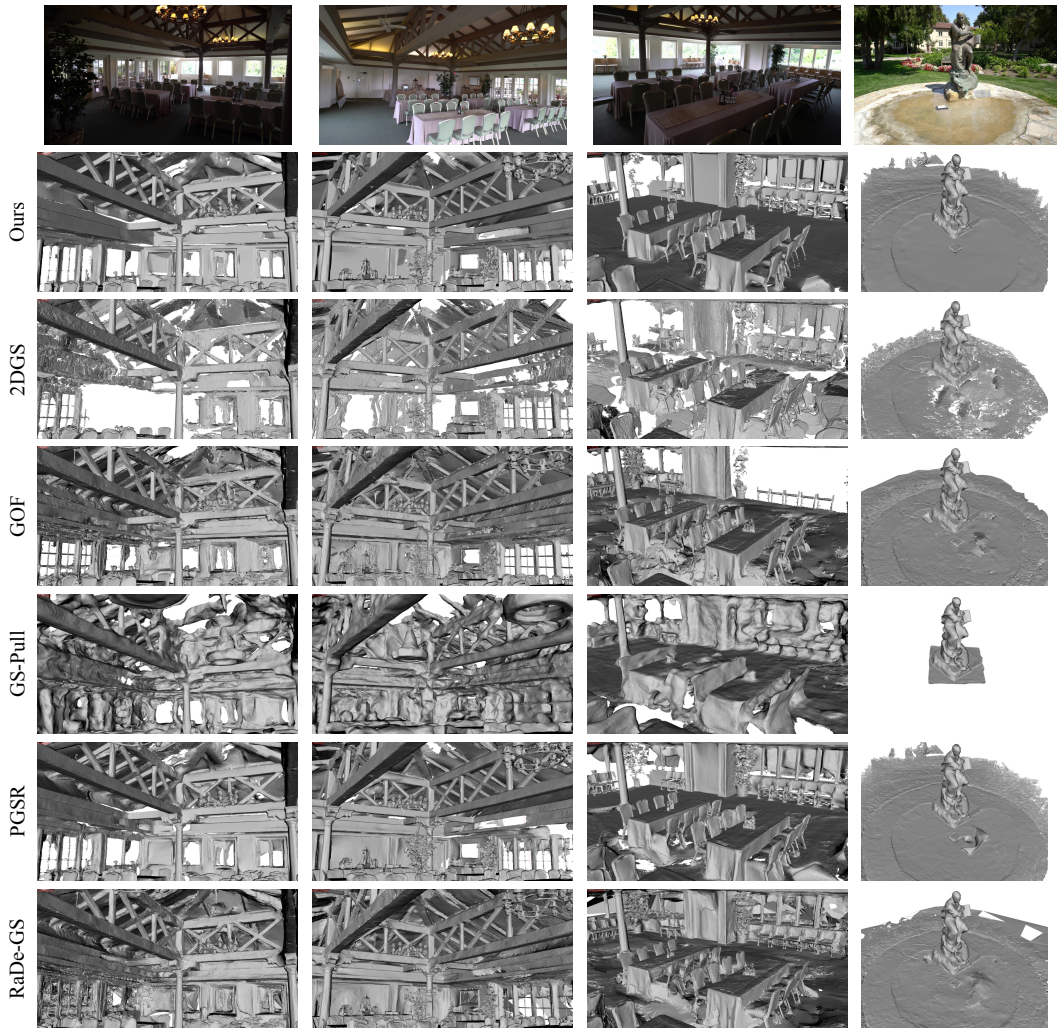

Figure 7: Qualitative comparison of surface reconstruction on the TNT dataset. Across large indoor and outdoor scenes, our method generates more continuous and topologically consistent meshes with sharper high-frequency structures, avoiding the breaks, smoothing artifacts, and non-manifold regions observed in other approaches.

whereas our results maintain crisp definition and visual realism even under complex scene content and lighting.

C.2.2 Results on DTU and TNT

We further evaluate our approach on the DTU Jensen et al. (2014) and TNT Knapitsch et al. (2017) datasets, which emphasize surface reconstruction while still providing a suitable benchmark for view synthesis. We first assess performance on the trained views. Following common practice, one out of every eight trained views is used for evaluation. Results are summarized in Tables 11 and 12. On the DTU dataset, our method does not achieve the highest PSNR and only attains the best average SSIM. This indicates that on seen viewpoints, methods such as 3DGS Kerbl et al. (2023) and RaDe-GS Zhang et al. (2024a) remain competitive. The relatively simple geometry of the DTU dataset allows PSNR to be improved mainly through color fitting, which limits the benefit of stronger geometric constraints. On the TNT dataset, our method achieves higher SSIM and LPIPS scores, demonstrating improved structural consistency and perceptual quality. While 3DGS Kerbl et al. (2023) still remains strong in PSNR metric, other methods such as RaDe-GS Zhang et al. (2024a)

Table 9: Quantitative comparison results of novel view synthesis for all scenes in the Mip-NeRF 360 dataset. The results are highlighted as 1st, 2nd and 3rd.

	bicycle	flowers	garden	stump	treehill	room	counter	kitchen	bonsai	**Mean**
PSNR ↑										
3DGS Kerbl et al. (2023)	25.17	21.45	27.18	26.56	22.30	31.34	28.89	30.71	31.98	27.29
2DGS Huang et al. (2024)	24.71	21.03	26.61	26.11	22.27	30.72	28.08	30.30	31.24	26.79
GS-Pull Zhang et al. (2024b)	24.19	20.56	26.08	25.24	22.60	30.84	26.41	26.06	29.10	25.68
RaDe-GS Zhang et al. (2024a)	25.56	21.67	27.39	27.10	22.34	30.84	28.73	31.26	31.73	27.40
GOF Yu et al. (2024b)	25.44	21.59	27.27	26.93	22.40	30.42	28.62	30.64	31.50	27.20
PGSR Chen et al. (2024a)	25.66	21.52	27.49	26.98	22.29	30.06	28.31	30.80	31.55	27.18
Ours	25.92	21.81	27.68	27.38	22.93	30.79	28.96	31.45	32.52	27.72
SSIM ↑										
3DGS Kerbl et al. (2023)	0.762	0.602	0.861	0.770	0.631	0.916	0.905	0.923	0.938	0.812
2DGS Huang et al. (2024)	0.729	0.569	0.838	0.753	0.614	0.906	0.892	0.915	0.929	0.794
GS-Pull Zhang et al. (2024b)	0.660	0.498	0.776	0.661	0.617	0.896	0.846	0.838	0.897	0.743
RaDe-GS Zhang et al. (2024a)	0.793	0.640	0.870	0.801	0.648	0.916	0.905	0.924	0.939	0.826
GOF Yu et al. (2024b)	0.786	0.634	0.864	0.790	0.641	0.911	0.900	0.915	0.935	0.820
PGSR Chen et al. (2024a)	0.793	0.635	0.873	0.798	0.660	0.927	0.914	0.932	0.945	0.831
Ours	0.804	0.650	0.879	0.808	0.674	0.931	0.921	0.935	0.950	0.839
LPIPS ↓										
3DGS Kerbl et al. (2023)	0.216	0.341	0.115	0.219	0.328	0.223	0.204	0.130	0.208	0.220
2DGS Huang et al. (2024)	0.276	0.380	0.149	0.263	0.381	0.243	0.230	0.146	0.228	0.255
GS-Pull Zhang et al. (2024b)	0.331	0.443	0.223	0.362	0.414	0.239	0.269	0.248	0.242	0.308
RaDe-GS Zhang et al. (2024a)	0.176	0.286	0.103	0.190	0.279	0.218	0.205	0.130	0.204	0.199
GOF Yu et al. (2024b)	0.182	0.282	0.109	0.197	0.282	0.221	0.205	0.137	0.200	0.202
PGSR Chen et al. (2024a)	0.186	0.264	0.103	0.192	0.273	0.180	0.172	0.113	0.169	0.184
Ours	0.164	0.254	0.093	0.174	0.242	0.172	0.157	0.107	0.154	0.169

Table 10: Training stability and convergence on the DTU dataset.

	Iteration	5k	10k	15k	20k	25k	30k
CD	2DGS Huang et al. (2024)	1.46	0.889	0.787	0.774	0.764	0.761
	Ours	3.23	0.498	0.487	0.457	(0.458)	(0.456)
PSNR	2DGS Huang et al. (2024)	31.40	32.16	33.00	33.71	34.25	34.45
	Ours	24.60	26.45	27.41	36.22	(36.29)	(36.33)

and GOF Yu et al. (2024b) also show competitive behavior in some scenes but less stable overall performance.

To further test generalization, we construct a disjoint evaluation set on the DTU and TNT datasets by holding out one image out of every eight as unseen viewpoints before the training. Due to time constraints, we only compare against three strong baselines: 3DGS Kerbl et al. (2023), RaDe-GS Zhang et al. (2024a), and GOF Yu et al. (2024b). As shown in Tables 13 and 14, these baselines exhibit notable degradation when moving from trained views to unseen ones. In contrast, our method maintains more stable performance. On the DTU dataset, we obtain the highest PSNR and SSIM on unseen views. On the TNT dataset, our method achieves the best average SSIM and LPIPS, confirming stronger geometric consistency and detail preservation across views.

Overall, our approach performs competitively or better on trained views, and provides clearer improvements when evaluated on novel viewpoints. These results support the effectiveness of decoupling geometry and appearance reconstruction into separate branches and enforcing stronger geometric alignment, which leads to a more robust balance between reconstruction fidelity and view generalization.

C.3 Training stability and convergence

We analyze convergence on DTU and observe faster and stable optimization than prior geometry-aware Gaussian methods, as shown in Table 10. Our model converges within 20k iterations, com-

Table 11: Quantitative comparison results of view synthesis for trained views of the DTU dataset. The results are highlighted as 1st, 2nd and 3rd.

	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
PSNR ↑																
3DGS Kerbl et al. (2023)	38.49	35.19	39.12	37.04	40.65	37.11	35.04	36.06	34.33	37.33	38.50	36.15	37.03	39.23	38.42	37.31
2DGS Huang et al. (2024)	33.95	31.16	33.62	33.89	37.88	34.49	32.46	34.16	31.69	34.75	36.49	34.20	33.24	37.80	37.07	34.45
GSurfels Dai et al. (2024)	26.98	22.62	22.30	27.34	22.51	29.03	28.27	21.55	24.26	22.56	33.06	29.55	30.36	33.08	32.16	27.04
GS-Pull Zhang et al. (2024b)	27.57	25.82	26.39	29.98	30.91	30.64	28.41	31.36	27.42	29.67	34.26	30.58	30.40	31.32	35.39	30.01
RaDe-GS Zhang et al. (2024a)	36.22	33.00	35.75	35.77	38.91	35.85	33.30	31.21	31.44	32.89	38.50	35.27	34.72	39.45	38.21	35.37
GOF Yu et al. (2024b)	35.18	32.74	34.89	34.92	38.37	36.28	33.82	34.70	30.50	35.50	37.75	34.42	33.81	38.76	38.49	35.34
PGSR Chen et al. (2024a)	32.20	27.80	30.57	34.44	36.49	34.68	32.21	32.85	31.51	34.85	36.53	34.65	33.20	38.09	37.19	33.82
Ours	34.80	31.09	33.45	34.88	36.07	37.40	39.52	34.68	32.34	35.36	37.30	40.61	36.43	39.94	39.38	36.22
SSIM ↑																
3DGS Kerbl et al. (2023)	0.977	0.977	0.977	0.945	0.981	0.946	0.947	0.930	0.939	0.943	0.950	0.940	0.949	0.944	0.941	0.952
2DGS Huang et al. (2024)	0.961	0.959	0.957	0.930	0.973	0.932	0.930	0.916	0.921	0.925	0.939	0.924	0.931	0.934	0.931	0.938
GSurfels Dai et al. (2024)	0.918	0.895	0.896	0.833	0.928	0.837	0.855	0.820	0.855	0.826	0.899	0.869	0.880	0.892	0.889	0.873
GS-Pull Zhang et al. (2024b)	0.916	0.906	0.892	0.898	0.943	0.866	0.856	0.901	0.869	0.872	0.931	0.887	0.885	0.925	0.932	0.899
RaDe-GS Zhang et al. (2024a)	0.972	0.969	0.969	0.957	0.977	0.951	0.949	0.931	0.936	0.938	0.957	0.947	0.949	0.955	0.953	0.954
GOF Yu et al. (2024b)	0.967	0.963	0.961	0.950	0.973	0.944	0.943	0.925	0.929	0.932	0.953	0.943	0.942	0.950	0.949	0.948
PGSR Chen et al. (2024a)	0.963	0.955	0.956	0.943	0.975	0.940	0.936	0.920	0.926	0.933	0.948	0.942	0.942	0.947	0.943	0.945
Ours	0.965	0.958	0.959	0.941	0.971	0.968	0.967	0.924	0.926	0.933	0.951	0.974	0.973	0.969	0.968	0.956
LPIPS ↓																
3DGS Kerbl et al. (2023)	0.040	0.075	0.066	0.131	0.063	0.132	0.116	0.201	0.155	0.153	0.202	0.219	0.176	0.222	0.236	0.146
2DGS Huang et al. (2024)	0.065	0.096	0.101	0.158	0.078	0.165	0.146	0.237	0.190	0.192	0.208	0.237	0.196	0.227	0.243	0.169
GSurfels Dai et al. (2024)	0.158	0.214	0.214	0.334	0.196	0.373	0.348	0.435	0.348	0.390	0.339	0.391	0.328	0.353	0.356	0.319
GS-Pull Zhang et al. (2024b)	0.121	0.156	0.182	0.198	0.117	0.307	0.286	0.270	0.274	0.299	0.190	0.322	0.276	0.199	0.192	0.226
RaDe-GS Zhang et al. (2024a)	0.044	0.087	0.074	0.091	0.069	0.110	0.106	0.187	0.147	0.156	0.145	0.172	0.143	0.149	0.158	0.123
GOF Yu et al. (2024b)	0.049	0.087	0.079	0.104	0.068	0.124	0.117	0.197	0.148	0.157	0.153	0.184	0.158	0.165	0.177	0.131
PGSR Chen et al. (2024a)	0.058	0.098	0.099	0.121	0.075	0.136	0.129	0.233	0.182	0.183	0.175	0.189	0.161	0.184	0.203	0.148
Ours	0.057	0.098	0.095	0.127	0.084	0.145	0.110	0.219	0.186	0.173	0.162	0.152	0.120	0.169	0.173	0.138

pared to 30k for classic 2DGS. Geometry-aware constraints of 2D branch introduced at 7k iterations significantly improve geometry (CD), while activating the 3D branch at 15k iterations yields clear rendering gains (PSNR) without degrading geometry. After 20k iterations, further training yields only marginal improvements in rendering quality while geometric metrics saturate. We therefore stop training at this point to balance efficiency and performance. Overall, these results demonstrate that our dual-branch optimization is stable, well-behaved, and more efficient than comparable baselines.

D Limitations and Future Works

While our dual-branch Gaussian framework achieves significant improvements in both surface accuracy and rendering fidelity, several limitations remain. First, the current training pipeline involves alternating optimization between the appearance and surface branches, which increases computational cost and memory usage compared with single-branch approaches. More efficient joint training or parameter sharing strategies could further enhance scalability. Second, our method assumes moderate viewpoint overlap and photometric consistency across views. Under extreme lighting changes or sparse-view conditions, the supervision of multi-view geometry may become less reliable, leading to degraded reconstruction in occluded or low-texture regions. Third, the geometry refinement currently relies on view-dependent cues rather than explicit semantic priors. Incorporating high-level geometric or material priors from large-scale foundation models could improve robustness and generalization to complex real-world scenes. In future work, we plan to explore unified optimization schemes for faster convergence, integrate semantic and physical priors to enhance generality, and extend the dual-branch formulation toward dynamic or large-scale outdoor environments.

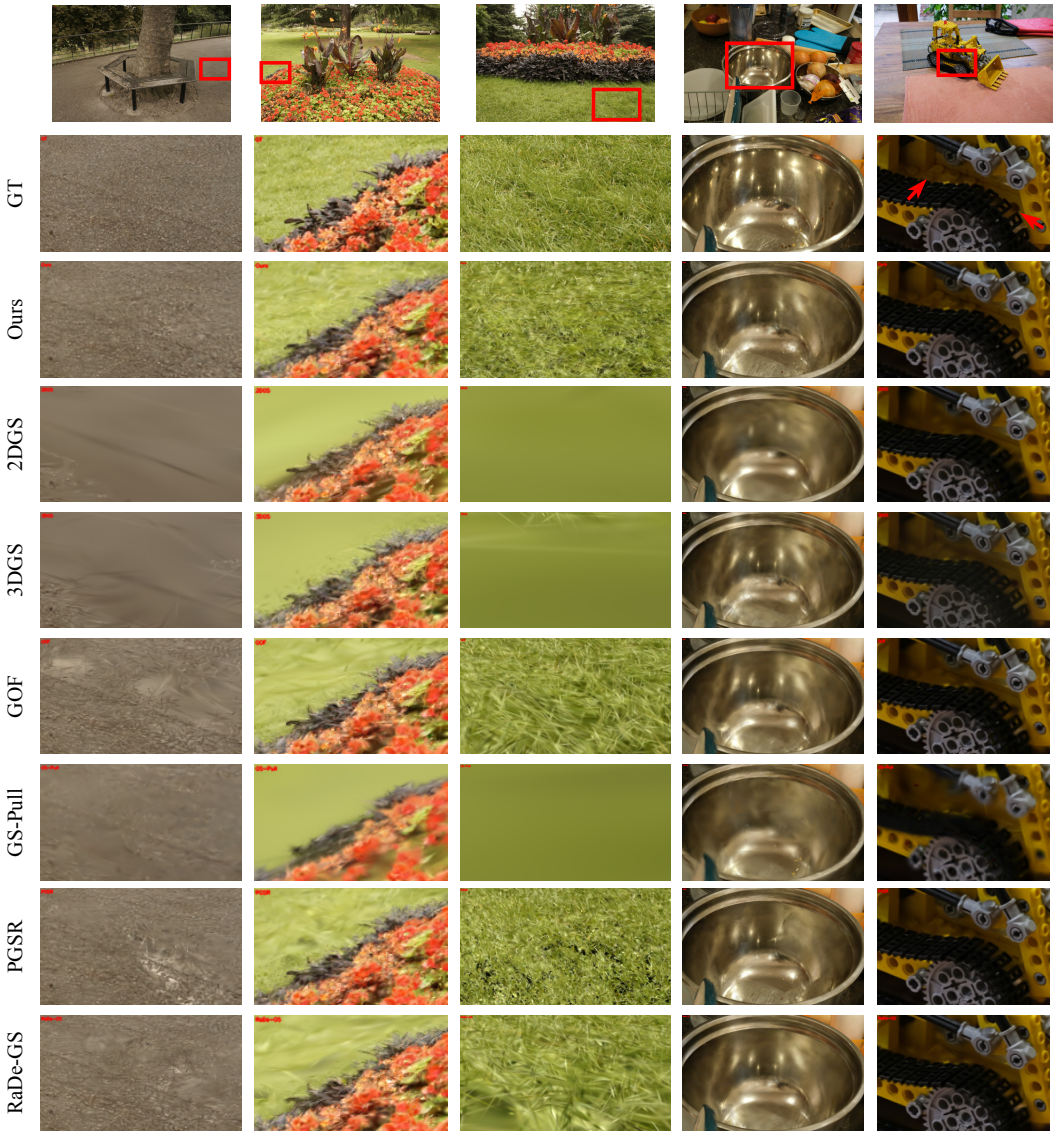

Figure 8: Comparison of novel view synthesis results on the Mip-NeRF 360 dataset. Our method renders sharper and more detailed images, preserving complex structures such as vegetation, while baselines often produce noticeably blurrier outputs in these challenging regions.

Table 12: Quantitative comparison results of view synthesis for trained views of the TNT dataset. The results are highlighted as 1st, 2nd and 3rd.

	Barn	Caterpillar	Courthouse	Ignatius	Meetingroom	Truck	**Mean**
PSNR ↑							
3DGS Kerbl et al. (2023)	30.42	26.65	24.49	26.15	29.12	27.59	27.40
2DGS Huang et al. (2024)	27.14	23.57	20.08	23.77	26.70	26.56	24.64
GSurfels Dai et al. (2024)	22.73	20.75	16.26	18.89	20.79	21.17	20.10
GS-Pull Zhang et al. (2024b)	18.66	20.41	19.15	23.51	21.39	20.23	20.56
RaDe-GS Zhang et al. (2024a)	26.73	23.43	21.86	22.71	27.02	24.24	24.33
GOF Yu et al. (2024b)	26.95	22.46	21.42	22.30	25.91	24.14	23.86
PGSR Chen et al. (2024a)	29.17	23.22	11.22	22.44	24.92	26.59	22.93
Ours	29.69	27.32	24.21	26.67	27.85	26.98	27.12
SSIM ↑							
3DGS Kerbl et al. (2023)	0.895	0.862	0.826	0.873	0.921	0.915	0.882
2DGS Huang et al. (2024)	0.860	0.809	0.719	0.848	0.900	0.902	0.840
GSurfels Dai et al. (2024)	0.794	0.696	0.645	0.675	0.757	0.766	0.722
GS-Pull Zhang et al. (2024b)	0.801	0.889	0.663	0.923	0.727	0.665	0.778
RaDe-GS Zhang et al. (2024a)	0.905	0.849	0.768	0.858	0.912	0.920	0.869
GOF Yu et al. (2024b)	0.905	0.849	0.772	0.860	0.902	0.919	0.868
PGSR Chen et al. (2024a)	0.917	0.848	0.405	0.853	0.898	0.916	0.806
Ours	0.923	0.878	0.826	0.880	0.912	0.919	0.889
LPIPS ↓							
3DGS Kerbl et al. (2023)	0.161	0.179	0.216	0.141	0.142	0.125	0.161
2DGS Huang et al. (2024)	0.208	0.237	0.354	0.167	0.172	0.147	0.214
GSurfels Dai et al. (2024)	0.285	0.354	0.440	0.342	0.386	0.307	0.352
GS-Pull Zhang et al. (2024b)	0.278	0.158	0.407	0.108	0.471	0.447	0.311
RaDe-GS Zhang et al. (2024a)	0.147	0.183	0.290	0.144	0.147	0.108	0.170
GOF Yu et al. (2024b)	0.145	0.170	0.279	0.139	0.153	0.107	0.166
PGSR Chen et al. (2024a)	0.119	0.171	0.561	0.137	0.142	0.096	0.204
Ours	0.107	0.147	0.210	0.114	0.130	0.089	0.133

Table 13: Quantitative comparison results of novel view synthesis for test views of the DTU dataset. We only compare the methods that perform better in Table 11. The results are highlighted as 1st and 2nd.

	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
PSNR ↑																
3DGS Kerbl et al. (2023)	25.16	24.75	23.03	28.00	29.09	28.80	26.66	29.03	26.96	29.27	33.44	32.16	30.01	34.47	33.78	28.97
RaDe-GS Zhang et al. (2024a)	25.19	25.04	23.86	28.50	28.26	29.69	26.29	27.32	26.44	28.56	33.68	31.97	29.61	33.82	33.87	28.81
GOF Yu et al. (2024b)	24.99	25.01	24.03	28.33	28.98	28.75	26.14	27.24	26.17	28.05	33.54	31.89	29.40	33.67	34.19	28.69
Ours	25.57	25.17	25.37	29.15	30.20	31.52	32.92	28.84	27.16	29.85	33.43	36.51	31.89	34.77	34.81	30.48
SSIM ↑																
3DGS Kerbl et al. (2023)	0.926	0.903	0.894	0.885	0.953	0.892	0.886	0.899	0.886	0.904	0.922	0.919	0.910	0.925	0.919	0.908
RaDe-GS Zhang et al. (2024a)	0.932	0.908	0.912	0.897	0.942	0.905	0.882	0.901	0.888	0.911	0.937	0.930	0.924	0.935	0.934	0.916
GOF Yu et al. (2024b)	0.927	0.904	0.912	0.888	0.953	0.893	0.878	0.897	0.885	0.907	0.932	0.926	0.916	0.927	0.928	0.912
Ours	0.932	0.913	0.916	0.902	0.958	0.948	0.938	0.903	0.893	0.914	0.933	0.956	0.952	0.951	0.950	0.931
LPIPS ↓																
3DGS Kerbl et al. (2023)	0.081	0.131	0.148	0.192	0.094	0.196	0.174	0.237	0.206	0.191	0.224	0.236	0.209	0.241	0.254	0.188
RaDe-GS Zhang et al. (2024a)	0.076	0.127	0.129	0.155	0.094	0.169	0.169	0.223	0.195	0.192	0.163	0.196	0.166	0.174	0.184	0.161
GOF Yu et al. (2024b)	0.082	0.127	0.129	0.166	0.092	0.182	0.178	0.224	0.194	0.192	0.174	0.201	0.172	0.200	0.196	0.167
Ours	0.080	0.126	0.132	0.167	0.092	0.175	0.142	0.242	0.214	0.194	0.180	0.169	0.138	0.188	0.188	0.162

Table 14: Quantitative comparison results of novel view synthesis for test views of the TNT dataset. We only compare the methods that perform better in Table 12. The results are highlighted as 1st and 2nd.

	Barn	Caterpillar	Courthouse	Ignatius	Meetingroom	Truck	**Mean**
PSNR ↑							
3DGS Kerbl et al. (2023)	28.91	24.08	23.32	22.82	25.92	25.55	25.10
RaDe-GS Zhang et al. (2024a)	15.40	20.66	19.74	13.75	11.49	10.10	15.19
GOF Yu et al. (2024b)	26.42	21.91	20.06	20.94	24.97	23.20	22.91
Ours	28.79	23.68	21.54	22.68	25.47	25.32	24.58
SSIM ↑							
3DGS Kerbl et al. (2023)	0.879	0.822	0.812	0.821	0.886	0.884	0.851
RaDe-GS Zhang et al. (2024a)	0.466	0.701	0.740	0.319	0.407	0.316	0.492
GOF Yu et al. (2024b)	0.894	0.819	0.754	0.816	0.885	0.894	0.844
Ours	0.901	0.831	0.785	0.824	0.878	0.886	0.851
LPIPS ↓							
3DGS Kerbl et al. (2023)	0.168	0.196	0.223	0.164	0.160	0.139	0.175
RaDe-GS Zhang et al. (2024a)	0.379	0.215	0.312	0.413	0.572	0.608	0.417
GOF Yu et al. (2024b)	0.149	0.190	0.289	0.157	0.158	0.117	0.177
Ours	0.119	0.180	0.233	0.146	0.163	0.107	0.158