Highlights

AND-GS: Adaptive Supervision of Normal and Depth in Gaussian Splatting for Accurate and Efficient Surface Reconstruction

Xiang Le,Qiang Zhao,Haofan Ren,Zhongtian Zheng,Tingyu Wang,Jiyong Zhang,Chenggang Yan

- We propose a new training strategy that adaptively adjusts the training stage according to the similarity of the low-frequency information of the rendered image, thereby using different levels of regularizers, which can greatly improve the training convergence speed and greatly shorten the training time.
- We fully incorporate geometric priors with the input training RGB images and exploit the inherent constraints between depth and normals through our proposed effective scene-level regularizers to achieve a more accurate Gaussian distribution and result in better reconstruction quality and rendering quality.
- Our proposed AND-GS has attained state-of-the-art results on multiple benchmark datasets, demonstrating superior performance in both geometric accuracy and rendering quality.

AND-GS: Adaptive Supervision of Normal and Depth in Gaussian Splatting for Accurate and Efficient Surface Reconstruction

Xiang Le[a], Qiang Zhao[b,*], Haofan Ren[a], Zhongtian Zheng[c], Tingyu Wang[b], Jiyong Zhang[a] and Chenggang Yan[a]

[a]The School of Automation, Hangzhou Dianzi University, Hangzhou, 310018, China
[b]The School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, 310018, China
[c]The School of Electrical Engineering and Computer Science, The University of Queensland, Brisbane QLD 4072, Australia

ARTICLE INFO

Keywords:
Surface Reconstruction
Multi-View-to-3D
Gaussian Splatting
Novel View Synthesis

ABSTRACT

Recently, 3D Gaussian Splatting (3DGS) has demonstrated stunning results in novel view synthesis, enabling real-time rendering of fine-grained images. Nevertheless, using 3D Gaussians for surface reconstruction presents prominent challenges. Previous methods either introduce the monocular predicted 3D scene cues, which usually lose the fine geometric details, or exploit the inherent constraints between depths and normals, which may fall into local minima and lead to low convergence speed. In this study, we present a new framework AND-GS that adaptively supervises the rendered depth and normal in the 3DGS optimization procedure for accurate and efficient surface reconstruction. Our method takes advantage of both the monocular predicted 3D scene cues and the inherent constraints between depths and normals, and proposes a new training strategy that adaptively switches between these two constraints according to the similarity between low-frequency information of the rendered images and the input image. This strategy can significantly improve the convergence speed and the final reconstruction quality. Our evaluations demonstrate that AND-GS achieves superior performance compared to existing 3DGS-based methods in both surface reconstruction and novel view synthesis. Furthermore, it delivers comparable or even better results than neural implicit methods, excelling in both quality and computational efficiency. We plan to release our code as open source upon acceptance of this paper.

1. Introduction

Accurate and efficient 3D geometry reconstruction from multiperspective images is a long-lasting but not fully resolved problem in computer vision. With the development of new 3D visualization technologies (such as Meta Quest, Apple Vision Pro, etc.), 3D reconstruction has become a more and more important task, where the reconstructed 3D objects can be integrated seamlessly into the real or virtual world through these head-mounted devices. Since Neural Radiance Field (NeRF) [38] has demonstrated excellent novel view synthesis (NVS) capabilities, it has also been extended to 3D geometry reconstruction through occupancy networks [40] and Signed Distance Functions (SDF) [44, 54]. However, 3D reconstruction based on neural implicit representation is time-consuming, as expensive random sampling is required during the optimization procedure. For example, 128 GPU hours are needed to represent a single scene by Neuralangelo [28].

Lately, 3D Gaussian Splatting (3DGS) [24] has gained prominence as an effective technique to represent complex scenes with 3D Gaussians, achieving remarkable results in novel view synthesis (NVS). It features real-time rendering capabilities and efficient training, making it a versatile approach that has been rapidly extended to surface reconstruction [10, 21, 18]. However, directly applying 3DGS to 3D reconstruction poses significant challenges. Optimizing solely with photometric losses often results in noisy and unreliable reconstructions, limiting the quality and precision of the reconstructed geometry. SuGar [18] further regularizes the Gaussians to fit the surface of the scene and employs Poisson surface reconstruction [23] to extract a mesh from the rendered depth maps. The absence of 3D cues and surface constraints during optimization leads to suboptimal reconstructions, characterized by artifacts and geometric inaccuracies. As a result, the extracted mesh of SuGaR is relatively coarse. To solve the problem that Gaussians do not

*Corresponding author
xiang-le@hdu.edu.cn (X. Le); qzhao@hdu.edu.cn (Q. Zhao); haofan@hdu.edu.cn (H. Ren); zt.zheng@student.uq.edu.au (Z. Zheng); tingyu.wang@hdu.edu.cn (T. Wang); jzhang@hdu.edu.cn (J. Zhang); cgyan@hdu.edu.cn (C. Yan)
ORCID(s):

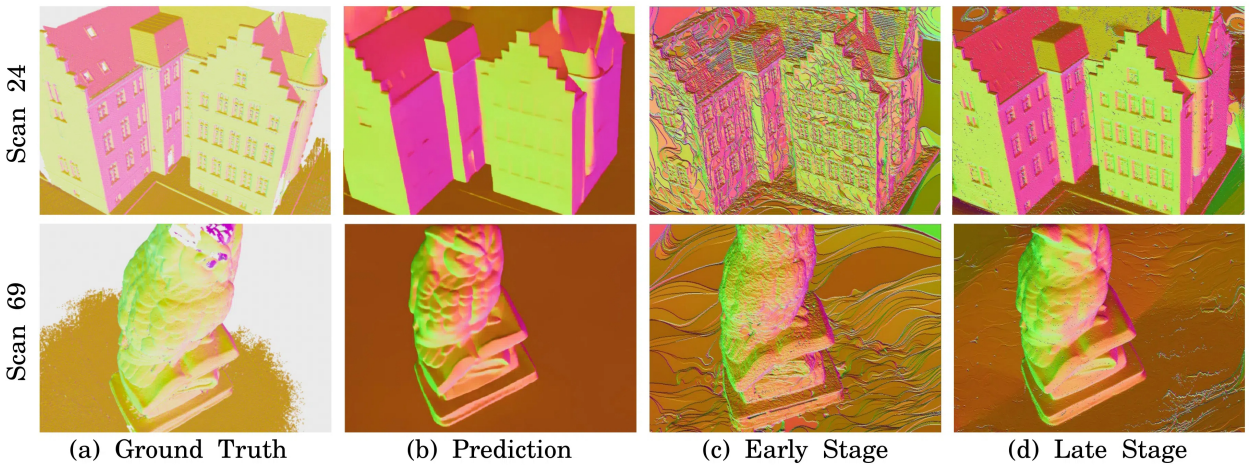


Figure 1: Normal map comparison. (a) The ground truth normal maps; (b) The normal maps predicted by the monocular estimation network have overall correct structure, but lose fine details compared with ground truth. (c) The normal maps rendered at the early stage of 3DGS optimization have many noises, while (d) those rendered at the late stage can preserve geometric details.

conform to the geometric surface, 2D Gaussian Splatting (2DGS) [21] uses 2D Gaussians instead of 3D Gaussians for better surface reconstruction and introduces depth distortion and normal consistency term to further boost the quality of the reconstructions. However, due to prior assumptions and overly strong constraints, the mesh is too smooth. GOF [59] derives Gaussian Opacity Fields directly from 3D Gaussians, which enables direct geometry extraction from 3D Gaussians by locating its levelset. The two regularization terms in 2DGS are also extended in GOF for better reconstruction quality.

Although the depth distortion and normal consistency terms in 2DGS and GOF provide different levels of improvement in reconstruction detail, they can be unreliable in the early stage of optimization. This is mainly due to the fact that these terms are defined with respect to the rendered depth maps and normal maps. Since the set of Gaussians is initialized from the sparse point cloud generated by SfM, the rendered depth map and the normal map contain many noises in the early stage, while preserving the geometric details in the late stage, as shown in Figure 1 (c) and (d). This may cause the reconstruction algorithms to fall into local minima in the beginning stage of the training and lead to poor reconstruction quality and low convergence speed.

To achieve high-fidelity 3D reconstruction, DN-Splatter [42] incorporates monocular depth and normal maps as direct supervision and aims to produce more physically accurate reconstructions. However, as most of the monocular estimation networks encourage smooth depth and normal maps, the predictions always lose fine geometric details compared with ground truth as shown in Figure 1 (a) and (b). Besides, DN-Splatter leverages depth and normal cues separately, and ignores the inherent relationship between depths and normals, which does not fully exploit the power of the supervision. VCR-GauS [11] also has the above problems, and it uses a single-source normal prediction pre-trained neural network, which makes the model performance more limited by the capabilities of the pre-trained network.

Motivated by these observations and the limitations of previous methods, we introduce AND-GS, an effective framework for accurate and efficient surface reconstruction. Our AND-GS fully takes advantage of both the predicted and the rendered depth and normal maps, and can adaptively switch between these two types of supervisions without manual intervention. Moreover, unlike existing works that obtain depth from the center position of 3D Gaussians, we compute the depth as the intersection of the ray. Subsequently, depth computation can be reduced to the intersection between a ray and a Gaussian. As a result, we can exploit the inherent constraints between depth and normals through our proposed effective scene-level regularizer to achieve a more accurate Gaussian distribution and result in better geometric accuracy and rendering quality.

We further integrate our adaptive training strategy to combine scale-invariant depth and normal supervision with rendered depth and normal supervision at different stages. Specifically, in the first stage, we use scale-invariant depth and normal maps as the scene-level geometric prior supervision, and regularize the Gaussians to be quickly distributed

to more accurate positions and to have the correct orientations. In the second stage, we exploit the fine geometric details of rendered depth and normal maps, and use depth distortion and normal consistency regularization terms for further supervision. These two stages are switched adaptively based on the similarity between the rendered and input images, which solves the shortcomings of previous 3DGS-based reconstruction methods. Remarkably, experiments reveal that our method outperforms Gaussian-based baselines in both reconstruction quality and rendering speed.

In conclusion, the main contributions of our work are outlined as follows:

- We propose a new training strategy that adaptively adjusts the training stage according to the similarity of the low-frequency information of the rendered image, thereby using different levels of regularizers, which can greatly improve the training convergence speed and greatly shorten the training time.
- We fully incorporate geometric priors with the input training RGB images and exploit the inherent constraints between depth and normals through our proposed effective scene-level regularizers to achieve a more accurate Gaussian distribution and result in better reconstruction quality and rendering quality.
- Our proposed AND-GS has attained state-of-the-art results on multiple benchmark datasets, demonstrating superior performance in both geometric accuracy and rendering quality.

2. Related Work

2.1. Novel View Synthesis

Neural Radiance Fields (NeRF) [38] have become a cornerstone in scene representation, utilizing a MLP to capture both geometry and view-dependent appearances [16, 17, 62, 12, 47]. The optimization of this MLP relies on photometric loss via volume rendering [13], achieving impressive results in novel view synthesis. However, this approach is computationally intensive, requiring extensive MLP calculations and training times exceeding 10 hours. To address these challenges, subsequent research has focused on enhancing NeRF's efficiency. Feature-grid representations have been introduced to optimize training [7, 15, 39], while methods such as baking have significantly accelerated rendering [19, 54, 46]. Beyond efficiency improvements, adaptations of NeRF have tackled issues like anti-aliasing [5] and unbounded scene modeling [4]. Despite these advancements, NeRF-based models remain constrained by their implicit representation. These limitations manifest as slow training speeds and restricted editability, presenting significant barriers to broader applications in areas requiring interactive or real-time performance.

In recent times, 3D Gaussian Splatting (3DGS) [24] has become a notable approach for modeling complex scenes using 3D Gaussians. It has demonstrated remarkable performance in novel view synthesis (NVS), empowering efficient optimization and on-the-fly rendering of high-resolution images. Despite its ability to produce high-quality reconstructions, there remains significant potential for improvement, particularly in surface reconstruction tasks. Follow-up research has sought to address these limitations by enhancing rendering quality through techniques such as anti-aliasing [56] and extending the method to dynamic scene modeling [63]. Other efforts have focused on improving computational efficiency and scalability [9, 37, 51, 50, 48, 20, 32, 43], broadening the applicability of 3DGS to diverse scenarios[29]. Recent Gaussian-based variants further broaden the design space of efficient reconstruction and prior-guided splatting. In particular, recent studies such as StegaNeRF [27], EndoGaussian [35], LGS [30], MonoSplat [34], and FlexGS explore [31], respectively, hidden-information embedding, deformable surgical reconstruction, lightweight dynamic Gaussian modeling, monocular-prior-driven generalizable splatting, and flexible many-in-one Gaussian parameterizations. Although these methods target different settings, they collectively indicate that the current research frontier is shaped by three closely related factors: how external priors are injected, how Gaussian parameters are regularized for geometry, and how efficiency is traded against fidelity. In this context, the main novelty of AND-GS lies in combining scene-level monocular priors with depth-normal coupling through an adaptive two-stage supervision schedule, rather than in introducing a new Gaussian primitive itself.

In this work, we build on these advancements by extending 3D Gaussian Splatting for high-quality surface reconstruction. Specifically, we leverage scale-invariant depth and normal priors to improve reconstruction accuracy and introduce an efficient adaptive training strategy that accelerates convergence while further enhancing rendering quality and reconstruction performance.

2.2. Multi-View Surface Reconstruction

Surface reconstruction from multi-perspective images is a fundamental challenge in computer graphics. Traditional multi-view stereo (MVS) methods [52, 57, 61, 64] typically rely on intricate multi-phase pipelines comprising feature

matching, depth estimation, point cloud fusion, and surface reconstruction from combined point clouds [23]. While effective, these approaches often suffer from high computational overhead and intricacies in pipeline integration. In contrast, neural implicit methods [40, 44] have introduced a more streamlined process by optimizing implicit surface representations via volume rendering. This enables seamless extraction of triangle meshes at arbitrary resolutions using Marching Cubes [36]. Recent advancements in this domain have focused on improving scene representation expressiveness [28], employing advanced training strategies [28], and incorporating monocular priors [58]. However, these methods are often constrained to foreground object reconstruction, are computationally expensive, and may produce excessively large meshes when high-resolution grids are used to capture fine details [28, 54].

To address these challenges, we propose an efficient approach using 3D Gaussians for surface reconstruction. This method enables mesh extraction and realistic novel view rendering while maintaining computational efficiency and effectively capturing fine details without generating unwieldy meshes.

2.3. Surface Reconstruction with Gaussians

Influenced by the outstanding novel view synthesis (NVS) capabilities of 3D Gaussian Splatting (3DGS) [24], researchers have explored the potential of 3D Gaussians for surface reconstruction. New techniques [58, 10] have merged 3D Gaussians with neural implicit surfaces by co-optimizing a Signed Distance Function (SDF) network and 3D Gaussians. Despite representing significant improvements, these methods also inherit the intrinsic limitations of implicit surfaces, including challenges in scalability and efficiency, as outlined earlier.

Further work has explored surface extraction from optimized Gaussian primitives through post-processing approaches [18, 21, 42]. For instance, Sugar [18] leverages Poisson surface reconstruction [23] to generate meshes from rendered depth maps, whereas 2DGS [21] utilizes TSDF fusion for surface extraction. Despite achieving enhanced reconstructions, these methods have difficulties in capturing detailed geometry and reconstructing background areas effectively. Meanwhile, GOF derives Gaussian Opacity Fields directly from 3D Gaussians, aligning with the volume rendering process to render RGB images and enabling surface extraction via level set identification. VCR-GauS [11] simplifies this process by using a single-source normal prediction network, but its reliance on a pre-trained normal prediction model limits reconstruction and rendering accuracy, often resulting in suboptimal results. PGSR [8] uses geometric constraints between multiple views to add a new loss function to optimize the scene.

Despite these advancements, a key trade-off persists: many methods for achieving accurate geometric reconstruction compromise rendering quality. To address this, our method focuses on better aligning Gaussians with the underlying scene geometry while introducing subsequent optimization processes to enhance both visual quality and reconstruction accuracy.

3. Preliminary

Inspired by the Gaussian Opacity Field [59], our work leverages the strengths of opacity fields while introducing a novel and effective strategy to integrate geometric priors into the training process. To ensure this paper is self-contained and accessible, this section offers a summary of the mathematical notation and rasterization algorithms used in our method.

3.1. 3D Gaussian Splatting

The standard 3D Gaussian Splatting [24] represents a scene as a collection of differentiable 3D Gaussians, enabling continuous and smooth modeling of scene geometry and appearance. Each 3D Gaussian is formulated as:

$$G(\mathbf{x}) = \alpha e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)} \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^3$ is the Gaussian center, $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ is the covariance matrix. It can be factorized into a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ and a scaling vector $\boldsymbol{S} \in \mathbb{R}^{3\times1}$ as $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^\top\boldsymbol{R}^\top$, opacity $\boldsymbol{\alpha} \in [0, 1]$ and color $\boldsymbol{c} \in \mathbb{R}^3$ encoded via spherical harmonics. To render a new view, the 3D Gaussians are projected into camera space as 2D Gaussians. These 2D Gaussians are then globally sorted by their z-depth values in a single sorting step. Finally, pixel colors $\hat{\mathbf{C}}$ are generated through alpha compositing, using the discrete volume rendering equation:

$$\hat{\mathbf{C}} = \sum_{i\in N} \mathbf{c}_i \alpha_i T_i, \text{ where } T_i = \prod_{j=1}^{i-1} \left(1 - \alpha_j\right) \tag{2}$$

where c_i is the view-dependent color, modeled with spherical harmonics, T_i is the total transmittance accumulated at pixel p. By parallelizing Gaussian projection and blending operations, real-time rendering performance is achieved.

3.2. Ray Gaussian Intersection

The intersection of the ray and Gaussian is defined as the location where the Gaussian reaches its maximum along the ray. Specifically, given a camera center at $\boldsymbol{o} \in \mathbb{R}^{3\times1}$ and a ray direction $\boldsymbol{r} \in \mathbb{R}^{3\times1}$, any point $\boldsymbol{x} \in \mathbb{R}^{3\times1}$ along the ray is defined as $x = o + tr$, where t is depth along the ray. The point x is first transformed into the local coordinate system of the 3D Gaussian and normalized by its scale:

$$\mathbf{o}_g = (\mathbf{R}(\mathbf{o} - \mu)) \odot \mathbf{S}^{-1},\ \mathbf{r}_g \quad = \mathbf{R}\mathbf{r} \odot \mathbf{S}^{-1},\ \mathbf{x}_g = \mathbf{o}_g + t\mathbf{r}_g. \tag{3}$$

indicates element-wise multiplication. Within this local coordinate system, the Gaussian value at any point on the ray becomes a 1D Gaussian, expressed as:

$$\mathbf{G}^{1D}(t) = \mathbf{G}\left(\mathbf{x}_g\right) = e^{-\frac{1}{2}\mathbf{x}_g^T\mathbf{x}_g} \tag{4}$$

The maximum of Equation (4) has a closed form solution at

$$t^* = -\frac{B}{A} \tag{5}$$

where $\mathbf{A} = \mathbf{r}_g^T\mathbf{r}_g$ and $\mathbf{B} = \mathbf{o}_g^T\mathbf{r}_g$.

3.3. Gaussian Opacity Fields

One key benefit of using explicit Ray-Gaussian intersection rather than projection is that it enables the calculation of opacity or transmittance at any point along the ray. Let's first analyze the case where only a single Gaussian G exists along the ray. Here, the opacity of any point along the ray is:

$$\mathbf{O}(\mathbf{G}, \mathbf{o}, \mathbf{r}, t) = \begin{cases} \mathbf{G}^{1D}(t) & \text{if } t \le t^* \\ \mathbf{G}^{1D}(t^*) & \text{if } t > t^* \end{cases} \tag{6}$$

The opacity intuitively increases until it reaches its maximum value, after which it remains unchanged as the transmittance, e.g., $1 - \mathrm{O}$, decreases monotonically along the view ray. Hence, the opacity at any point along a ray, given a set of Gaussians, can be defined similarly to the volume rendering process in Equation (2) as:

$$\mathbf{O}_k = \mathbf{O}_k\left(\mathbf{G}_k, \mathbf{o}, \mathbf{r}, t\right) \tag{7}$$

$$\mathbf{O}(\mathbf{o}, \mathbf{r}, t) = \sum_{k=1}^{K} \alpha_k \mathbf{O}_k \prod_{j=1}^{k-1}\left(1 - \alpha_j \mathbf{O}_k\right) \tag{8}$$

The opacity of a 3D point x can now be defined as the smallest opacity value among all training views or viewing directions:

$$\mathbf{O}(\mathbf{x}) = \min_{(\mathbf{r},t)} \mathrm{O}(\mathbf{o}, \mathbf{r}, t) \tag{9}$$

Finally, the scene initialization begins with Gaussian means and colors derived from Structure-from-Motion (SfM). The initial covariance matrices are configured so that their axes align with the average distance from each Gaussian to its three nearest neighbors. Throughout optimization, the parameters of each Gaussian—including means, colors, and covariances—are optimized via gradient descent across multiple rendering iterations to best match the training dataset. To maintain efficiency and adaptiveness, the algorithm performs scene updates at fixed intervals. This dynamic adjustment of Gaussians ensures accurate representation of scene geometry and appearance while maintaining computational efficiency.

4. Our Method

With multiple posed and calibrated images provided, our goal is to efficiently reconstruct the 3D scene. This encompasses detailed and compact surface extraction along with photorealistic novel view synthesis. For this purpose, as showed in Figure 2, we firstly construct flatten Gaussians from SfM. In the meantime, inspired by space carving [26], we use depth and normal priors obtained from two different pre-trained network to carve the Gaussian opacity field, which effectively solves the problem that the previous work DN-splatter cannot effectively back-propagate the loss function to optimize the Gaussian properties. Moreover, our method also solves the problem that the performance of the VCR-GauS model is limited by a single pre-trained neural network, and simultaneously utilizes the intrinsic connection between depth and normal to achieve the best effect by optimizing different properties of Gaussian separately. Then we use the scene-level regularization term we proposed to optimize the Gaussian. Those scene-level supervision enable Gaussians distribution more precise, get better rendering quality and geometric accuracy as shown in Figure 3 and Figure 4. Due to the length limitation of the article, the visual comparison results only show the comparison with the SOTA method, which also reflects the effect and performance of our method. Next, we propose a new training strategy that adaptively adjusts different training stages according to the similarity of low-frequency information, which improves the training convergence speed, as shown in Figure 5.

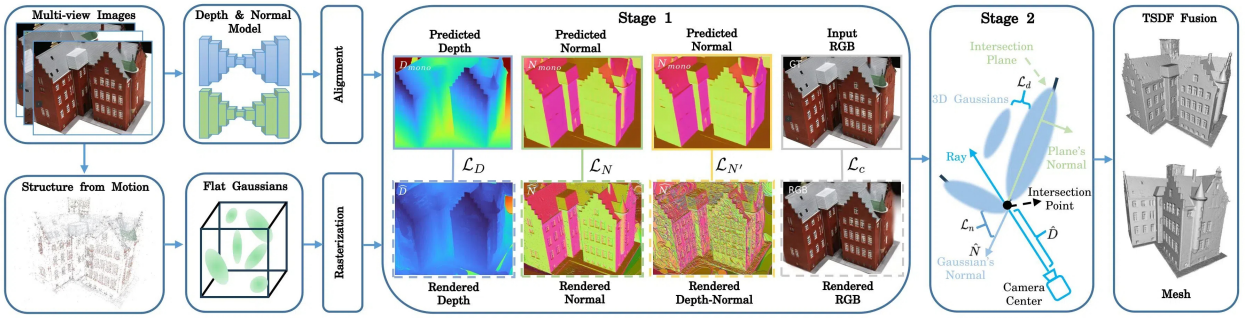


Figure 2: The overview of AND-GS: We use scale-invariant depth and normal map as geometry priors at stage 1, which provide overall correct scene structure and leads to fast convergence. At stage 2, we take the advantage of inherent contrains between rendered depth and normal map, which can recover the fine details of the surface.

4.1. Scene-level Geometric Supervision

From our previous tests and experiments, we found that Depth Distortion Loss and Normal Consistency Loss [59] are more effective in supervising high-frequency details, but have little effect on low-frequency contours in the beginning stage. Therefore, we use scene-level low-frequency depth and normal supervision to replace the original loss function to effectively reconstruct low-frequency contours.

4.1.1. Scale Invariant Depth Regularization

Through experiments, it is found that the least squares method used by DN-Splatter to adjust the scale and offset of the predicted depth map has problems and cannot effectively align the predicted depth map of the real scene, especially the outdoor open scene. Traditional depth regularization methods often assume that the predicted depth maps are directly aligned with the underlying 3D geometry. However, in practice, depth maps generated from multi-view images or different modalities can exhibit local scale variations, slight geometric distortions, or non-rigid misalignments caused by noise, camera calibration inaccuracies, or modality-specific characteristics. These issues may lead to suboptimal regularization and prevent the network from learning a consistent depth representation across views. Therefore, we use a learnable affine transformation between 3D depth points to solve this problem and propose a new depth regularization. The specific effect is demonstrated in the ablation experiment in Table 4.

We employ DepthAnything [49] to obtain dense per-pixel depth priors and resolve scale ambiguity between estimated depths and the scene by comparing with sparse SfM points. Besides, we propose $\hat{D}$ as per-pixel depth map which is the depth of the intersection point in Eq.7. Specifically, for each misaligned monocular depth estimate D_{MONO}, we adjust its scale to align with the sparse depth map D_{sparse} obtained by projecting SfM points into the

camera view and solve for the linear part of affine transformation $A \in \mathbb{R}^{3\times3}$ and translation vector $b \in \mathbb{R}^3$:

$$\begin{aligned}\mathcal{L}_{point} &= \frac{1}{N}\sum_{i=1}^{N}\sum \min ||A_i \cdot D_{MONO} + b_i - D_{sparse}||^2 \\ \mathcal{L}_{reg} &= \sum_{i=1}^{N} ||A_i - I||^2 + ||b_i||^2 \\ \mathcal{L}_{align} &= \alpha \cdot \mathcal{L}_{point} + \beta \cdot \mathcal{L}_{reg}\end{aligned} \tag{10}$$

To minimize $\mathcal{L}_{align}$ loss, we use stochastic gradient descent optimization. We denote $\mathcal{L}_{point}$ for the point cloud projection error and $\mathcal{L}_{reg}$ for affine parameter regularization. N is the number of depth maps, I is the identity matrix, D_{sparse} and D_{MONO} are the initial SfM depth maps and misaligned monocular depth maps, $\alpha = 10$ and $\beta = 0.5$. In practice, the affine parameters are estimated independently for each training image rather than globally for a whole scene or locally for image patches. We initialize every A_i as the identity matrix and every b_i as a zero vector, and optimize them only from the sparse SfM correspondences that are visible in the current image. The regularizer $\mathcal{L}_{reg}$ keeps the solution close to a scale-and-shift mapping and penalizes physically implausible shearing or excessive translation. After convergence of this pre-alignment step, the affine parameters are frozen and the aligned depth priors are passed to Gaussian optimization; therefore, implausible deformations cannot continue to grow during the later reconstruction stage. The number of valid SfM correspondences is scene-dependent and follows the visibility pattern of COLMAP points. In our experiments, the typical number of visible correspondences per image is approximately 180–260 on DTU, 120–210 on Tanks & Temples, and 90–170 on Mip-NeRF360. When the correspondences are sparse or unevenly distributed, the identity regularizer dominates and the alignment gracefully falls back to a near-rigid correction instead of overfitting a poorly constrained affine warp.

After learnable affine transformation alignment, we obtain more consistent predicted depth maps from multiple views as scene-level depth supervision. In order to minimize the gradient error caused by the depth scale, we use a scale-invariant depth loss to make the Gaussian distribution fit the actual surface. The scale-invariant depth loss takes into account the scale of both monocular and render depth, which smooths the abnormal depth value and encourages Gaussian close to the actual scene distribution and effectively improves rendering quality.

$$\mathcal{L}_D = \frac{1}{2|\hat{D}|}\sum(log\hat{D} - log D_{mono} + \mathcal{F}(\hat{D}, D_{mono}))^2 \tag{11}$$

Where $\mathcal{F}(\hat{D}, D_{mono}) = \frac{1}{|\hat{D}|}\sum(log D_{mono} - log\hat{D})$, $\hat{D}$ is the render depth map and D_{mono} is scale aligned predicted depth map, $|\hat{D}|$ indicates the total number of pixels in $\hat{D}$.

4.1.2. Normal Regularization

As optimization progresses, Gaussians tend to become flat and disc-like, with one scaling axis much smaller than the other two. This scaling axis acts as an approximation of a normal direction. It is worth noting that the assumption in Equation (14) that the normal is the axis of minimum scaling does not strictly apply to a completely isotropic Gaussian function (at initialization), since in this case there is no single dominant axis. However, in practice, an isotropic Gaussian function may not exist after just a few rounds of optimization, as the anisotropic scaling introduced during training allows the Gaussian function to adapt to the surface geometry. Furthermore, since our method does not compute normals at initialization, this does not affect the final performance. In particular, we define a geometric normal for a Gaussian based on its rotation matrix $R \in SO(3)$, obtained from its quaternion q, and scaling coefficients $s \in \mathbf{R}^3$:

$$\hat{n}_i = R \cdot \text{OneHot}(\arg\min(s_1, s_2, s_3)) \tag{12}$$

Normals are mapped into camera space using the current camera transform and alpha-composited according to the rendering equation to obtain a single per-pixel normal estimate.

$$\hat{N} = \sum_{i\in N} \hat{n}_i \alpha_i T_i \tag{13}$$

Unlike previous research, which adds an extra learnable parameter per Gaussian for normal prediction, our approach calculates normals directly based on the geometry. This facilitates the back-propagation process to optimize the Gaussian scale and rotation parameters, represented by the covariance matrices, and refine the normal estimates.

Figure 3: Some novel view synthesis rendering visualization comparison on Mip-NeRF360 and Tanks & Temples datasets. Although our work focuses on surface reconstruction, there is still improvement for NVS. We use NVIDIA's FLIP tool to highlight visual differences, where brighter parts represent significant differences, and darker parts represent the same as GT. Our method is more robust in both indoor and outdoor scenes, with better PSNR, and more dark parts indicating the same as the GT.

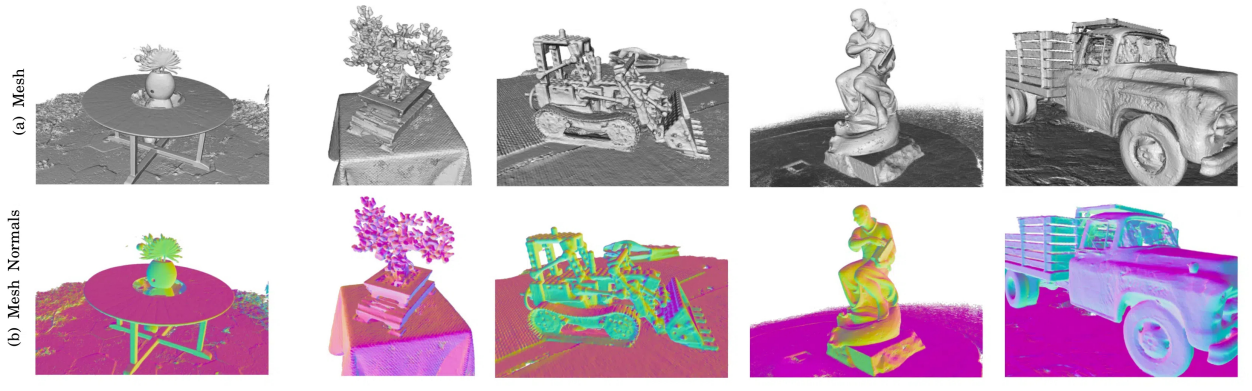


Figure 4: The mesh reconstruction outcomes and normals obtained across diverse indoor and outdoor scenes demonstrate the effectiveness of our method. AND-GS delivers accurate geometric reconstructions from a sequence of RGB images.

Hence, no additional learnable parameters are required. This method intuitively promotes Gaussians to better fit the scene's geometry, as their orientations and scales are naturally aligned with the surface normals during the optimization process.

Noise in depth maps, especially in complex scenes, causes artifacts when pseudo-ground truth normal maps estimated from the gradient of rendered depths are used for supervision. Rather, we supervise the normals using monocular cues provided by DSINE [3], which provide much smoother estimates for normals and reduce artifacts of normal caused by incorrect depth values. The normal loss smooths normal map and encourages Gaussian close to

the actual scene distribution in scene-level rapidly, also more effective for reconstruction.

$$\mathcal{L}_N = \frac{1}{|\hat{N}|} \sum log(1 + \|\hat{N} - N_{mono}\|_1) \tag{14}$$

Where $\hat{N}$ is the render normal map and N_{mono} is the predicted normal map, $|\hat{N}|$ indicates the total number of pixels in $\hat{N}$. This scene-level normal supervision is intentionally used only in the early stage, when monocular priors are more reliable than the noisy rendered normals. As a consequence, the supervision is not circular: the optimization first anchors the Gaussian distribution with external depth and normal cues, and only after this coarse geometry becomes stable do we activate the rendered depth-normal coupling terms.

4.1.3. Depth-Normal Consistency Regularization

The depth-normal consistency loss smooth the difference of normal map calculated from depth map and predicted normal map.This loss function optimizes the depth while optimizing the rendered normal map, making the Gaussian pose closer to the real surface and ultimately reconstructing fine geometry.

$$\mathcal{L}_{N'} = \sum_i \omega_i (1 - N_{mono} \cdot N) \tag{15}$$

where $\omega_i = \alpha_i \prod_{j=1}^{i-1} \left(1 - \alpha_j\right)$ is the blending weight of $i-$th Gaussian, N_{mono} is the predicted normal map and N is the normal estimated from the gradient of the depth map.

Figure 5: Our adaptive training strategy convergence speed compare with GOF on the Tanks & Temples dataset. It can be clearly seen from the figure that our method has a PSNR that is more than 1dB higher than GOF, a SSIM that is 0.025 higher, and a LPIPS that is 0.02 lower at 6000 iterations.

4.2. Adaptive Optimization Strategy

To accelerate training and tackle the problem of Gaussians not always corresponding to real geometric structures, we propose an adaptive geometry-guided optimization strategy. One of our key starting points is that we want Gaussian to learn the low-frequency information of the scene (contours, scene structure) first during training, so we use a low-pass filter to check the similarity of the low-frequency information of the rendered image.

To assess the similarity of low-frequency content between the rendered image I_r and the reference image I_{ref}, we apply an adaptive low-pass filter whose radius r_{LF} is determined from the spectral energy distribution. Let $F_r(f)$ and $F_{\text{ref}}(f)$ denote the Fourier transforms of I_r and I_{ref}, respectively. Define the cumulative normalized spectral energy as:

$$E(r) = \frac{\sum_{\|f\| \le r} |F(f)|^2}{\sum_{\forall f} |F(f)|^2}, \tag{16}$$

where f is the frequency coordinate, and the summations are taken over all frequencies within radius r. The adaptive radius r_{LF} is chosen as the minimal radius satisfying:

$$E(r_{\text{LF}}) \ge 0.9. \tag{17}$$

Using this radius, a low-pass filter $\mathcal{LPF}_{r_{\mathrm{LF}}}(\cdot)$ is applied to both images to obtain their low-frequency components:

$$\tilde{I}_r = \mathcal{LPF}_{r_{\mathrm{LF}}}(I_r), \quad \tilde{I}_{\mathrm{ref}} = \mathcal{LPF}_{r_{\mathrm{LF}}}(I_{\mathrm{ref}}). \tag{18}$$

We then compute the Structural Similarity Index Measure (SSIM) between $\tilde{I}_r$ and $\tilde{I}_{\mathrm{ref}}$. Empirically, if SSIM($\tilde{I}_r$, $\tilde{I}_{\mathrm{ref}}$)>0.85, we consider the Gaussian primitives to have adequately captured the low-frequency structure of the scene. This condition is used as the threshold in Equation (21) to switch the training phase from low-frequency learning to high-frequency detail refinement. The switching decision is made at the scene level during training, based on the low-frequency similarity accumulated from rendered training views, rather than independently for each image. In our implementation, the policy is one-way: once the criterion is satisfied, the optimization permanently enters the second stage and does not oscillate back to the first stage. Table 5 partially serves as a sensitivity study for this mechanism, because fixed switching points at 0k, 10k, 15k, and 20k all underperform the adaptive policy in geometric accuracy, indicating that the gain is not tied to a single manually tuned switching iteration. Similarly, the strong degradation of the w/o Strategy setting in Chamfer Distance shows that the adaptive policy mainly protects geometry, even when pure photometric optimization can occasionally yield a slightly higher PSNR.

In general, our new adaptive training strategy optimizes different attributes of Gaussian with effective loss functions at different stages, which comprises smoothly scale invariant depth loss and normal loss and depth-normal consistency loss, depth distortion loss and normal consistency loss proposed by 2D Gaussian Splatting. We also introduce depth distortion loss and normal consistency loss to make this paper self-contained.

4.2.1. Depth Distortion Loss

With inspiration of Mip-NeRF 360 [4] and 2DGS [21], the depth distortion loss promotes the proximity of different Gaussian splats along a ray by reducing the disparity in their depths, as expressed by the following equation:

$$\mathcal{L}_d = \sum_{i,j} \omega_i \omega_j (d_i - d_j)^2 \tag{19}$$

where $\omega_i = \alpha_i \prod_{j=1}^{i-1} \left(1 - \alpha_j\right)$ is the blending weight of $i-$th Gaussian and d_i is its depth.

4.2.2. Normal Consistency Loss

The normal consistency loss ensures that the Gaussian splats align with the surface by evaluating the agreement between the normals derived from the Gaussian and those from the depth-normal map, respectively,

$$\mathcal{L}_n = \sum_i \omega_i (1 - \hat{n}_i^{\top} N) \tag{20}$$

where i indexes over intersected Gaussians along the ray, ω denotes the blending weight, and N is the normal estimated by the gradient of the depth map. The minimum-axis normal proxy is most reliable once Gaussians become anisotropic, and is naturally less certain in low-opacity regions, thin structures, or heavily overlapping splats. For this reason, we do not use this proxy as a dominant signal at initialization; instead, it acts as a refinement term after the scene-level priors have already stabilized the coarse geometry. This design reduces the risk of circular supervision from noisy depth gradients and explains why the depth-normal consistency term improves fine details in the late stage but is harmful when applied from iteration zero.

4.2.3. Final Loss

Finally, we optimize our model from an initial sparse point cloud using multiple posed images by minimizing the following loss:

$$\mathcal{L} = \begin{cases} \mathcal{L}_c + \gamma \mathcal{L}_D + \delta \mathcal{L}_N + \beta \mathcal{L}_{N'} & \text{if } T \leq Threshold \\ \mathcal{L}_c + \alpha \mathcal{L}_d + \beta \mathcal{L}_n & \text{if } T > Threshold \end{cases} \tag{21}$$

where $\gamma = 0.1$ and $\delta = 0.01$, $\alpha = 100$ and $\beta = 0.05$, T is training iteration, $Threshold$ is adaptively determined by our proposed optimization strategy according to different scenes, $\mathcal{L}_c$ is an RGB reconstruction loss combining $\mathcal{L}_1$ with the D-SSIM term from [24], while $\mathcal{L}_D$, $\mathcal{L}_N$, $\mathcal{L}_{N'}$ and $\mathcal{L}_d$, $\mathcal{L}_n$ are regularization terms. Our final adaptive geometry-guided optimization strategy aims to enable the model to quickly learn an accurate Gaussian distribution. Through prior

Table 1
Quantitative evaluation on the DTU Dataset. We evaluate the Chamfer Distance of mesh. Our method achieves the best reconstruction accuracy among other explicit methods. † Indicates results directly reported from the original paper.

		24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean↓	Time↓
implicit	NeRF	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	>4h
	VolSDF	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86	>6h
	NeuS	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	>6h
	MonoSDF	0.66	0.88	0.43	0.40	0.87	0.78	0.81	1.23	1.18	0.66	0.66	0.96	0.41	0.57	0.51	0.73	>12h
	Neuralangelo	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	>128h
explicit	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	1.97	~0.2h
	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	1.32	~1.3h
	2DGS	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	~0.3h
	GOF	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	~ 2h
	AtomGS	0.51	0.77	0.53	0.40	1.07	0.81	0.87	1.21	1.14	0.47	0.70	1.36	0.36	0.58	0.43	0.75	~0.2h
	VCR-GauS†	0.55	0.91	0.40	0.43	0.97	0.95	0.84	1.39	1.30	0.90	0.76	0.92	0.44	0.75	0.54	0.80	~ 1h
	Ours	0.45	0.68	0.31	0.35	0.91	0.71	0.70	1.13	1.27	0.66	0.63	0.85	0.39	0.65	0.49	0.68	~ 0.5h

knowledge of geometry to guide Gaussian's densification at the beginning stage, then reconstruct the high-frequency details of the scene by adding a few Gaussians through the gradient of the input RGB images. This adaptive training strategy ensures that enhancements focus on maintaining geometric accuracy without affecting the RGB field fidelity as shown in Figure 3. Meantime, it accelerates the convergence speed as shown in Figure 5, reconstructing low-frequency contours and high-frequency texture details simultaneously.

5. Experiments

We perform a comprehensive evaluation of our method, benchmarking its surface reconstruction and novel view synthesis against state-of-the-art approaches. Additionally, we validate the effectiveness of its core components and adaptive training strategy through ablation studies.

5.1. Implementation Details

Our method is built upon the publicly available codebases of 3DGS [24] and GOF [59]. We adopt the default parameters from GOF and integrate the 3D filtering technique proposed in Mip-Splatting [56]. Following a strategy similar to 3DGS, we terminate densification at 15K iterations and optimize all models for a total of 30K iterations. We set $\gamma = 0.1$ and $\delta = 0.01$, $Threshold$ is adaptively determined by our proposed optimization strategy according to different scenes. Following GOF, in all our experiments, we set $\alpha = 100$ and $\beta = 0.05$, and we detach the gradient propagation of the blending weight ω when computing the depth distortion loss. Depth maps are rendered for all training views, followed by mesh extraction using TSDF. All experiments are performed on a single NVIDIA GeForce RTX 4090 GPU. The time consumption of our method includes all data pre-processing and post-processing. Unless otherwise noted, reproduced Gaussian-based baselines are evaluated with the same mesh extraction and metric scripts used for our method, while results that could not be reproduced reliably are explicitly quoted from the original papers and discussed as such in the text. The reported runtime of AND-GS includes monocular prior generation, Gaussian optimization, TSDF fusion, and marching cubes. In our pipeline, Gaussian optimization remains the dominant cost, whereas TSDF fusion and marching cubes add a comparatively short post-processing stage (about 1–3 minutes for large scenes, as discussed in Section 6). This breakdown is important because the efficiency gain over neural implicit methods mainly comes from the optimization stage, not from omitting downstream mesh extraction.

Baselines: We compare our method with state-of-the-art Gaussian Splatting approaches for surface reconstruction, including 3DGS [24], SuGaR [18], 2DGS [21], GOF [59], AtomGS [33], and VCR-GauS [11]. Additionally, we benchmark against NeRF-based implicit methods such as NeRF [38], VolSDF [53], NeuS [44], and Neuralangelo [28]. These implicit methods use a Signed Distance Function (SDF) to represent the scene and convert the SDF into opacity for ray tracing-based volume rendering. Unless otherwise stated, all reported results are reproduced in our environment under the same hardware setting and with matched evaluation parameters; only the methods explicitly marked as unreproducible are reported using the numbers quoted from the original papers.

Datasets: We assess the performance of our method on multiple datasets. Surface reconstruction experiments are conducted on the DTU [1] and Tanks & Temples [25] datasets. Following previous works, we select 15 scenes from DTU and 6 scenes from Tanks & Temples for evaluation. Using the camera poses provided by the datasets, we employ COLMAP [41] to generate sparse point clouds for initialization. For novel view synthesis, we experiment on the Mip-NeRF360 [4] and Tanks & Temples [25] datasets. Mip-NeRF360 includes large indoor and outdoor scenes, while Tanks & Temples features large outdoor scenes with complex lighting, and DTU contains object-level scenes with challenging reflections and detailed shapes.

Metrics: To facilitate comparisons with existing methods, we provide Chamfer Distance (**CD**) results on the DTU dataset and **F-1 Score** results on the Tanks & Temples dataset. The rendering quality is assessed using three standard metrics: Peak Signal-to-Noise Ratio (**PSNR**), Structural Similarity Index Measure (**SSIM**) [45], and Learned Perceptual Image Patch Similarity (**LPIPS**) [60], evaluated on the Mip-NeRF360 and Tanks & Temples datasets.

Data Preprocess: Using the camera poses provided by the datasets, we apply COLMAP to generate a sparse point cloud for each scene as initialization. For datasets lacking depth and normal data, we use scale-aligned monocular estimation networks for regularization. We download the pre-trained weights and obtain the depth and normal through monocular depth networks DepthAnything [49] for dense per-pixel depth priors and monocular normal networks DSINE [3] for precise and smooth normal priors. We resolve the scale ambiguity between the estimated depth and the scene by comparing the monocular predictions with sparse SfM points obtained from COLMAP. By using these two different prediction networks simultaneously, we can better exploit the intrinsic relationship between depth and normal by optimizing different Gaussian properties separately. This reduces normal artifacts caused by incorrect depth values, such as in the contemporary work VCR-GauS, which only uses a single prediction neural network without considering intrinsic connections.

5.2. Geometry Evaluation

We begin by comparing our method with existing approaches on the DTU dataset. As shown in Table 1, our method outperforms all Gaussian splatting-based methods and achieves competitive results with Neuralangelo in terms of Chamfer Distance error. However, Neuralangelo requires over 128 GPU hours to train a single scene. In contrast, our method is compact and over **100 times** faster. In addition, compared with MonoSDF[58], which also uses monocular prior information, the reconstruction accuracy is higher and the training time is greatly shortened by more than 10 times.

At the same time, our method solves the problem that DN-Splatter and the concurrent work VCR-GauS use depth and normal supervision respectively. Since the reconstruction results of DN-Splatter in the DTU dataset are very rough and the authors claim that they mainly focus on capturing larger exterior-facing indoor environments [1]. In addition, due to bugs in the VCR-GauS code, we are unable to reproduce the author's rendering and reconstruction results [2]. The numerical results are excerpted from the original paper. We did not present the visualization results of the above methods in the visual comparison of Figure 6. To make the comparison protocol explicit, we distinguish between reproduced and quoted baselines throughout our evaluation. Methods with stable public implementations and compatible extraction pipelines are re-run in our environment, while methods whose code is unavailable, unstable, or designed for substantially different settings are reported using the original numbers and discussed qualitatively with this caveat. While not all methods can be evaluated under a fully uniform setup, this protocol follows common practice and enables fair comparison among methods with compatible assumptions. Importantly, our main conclusion does not rely on quoted results: AND-GS consistently improves over reproduced Gaussian-based baselines under controlled settings, while remaining competitive with the best quoted implicit reconstruction methods.

It is worth noting that we consider the intrinsic correlation between depth and normal, jointly optimize different attributes of Gaussian, and comprehensively improve reconstruction accuracy and rendering quality. Figure 6 visualizes some results generated by different Gaussian splitting based methods. Our method produces smooth and precise shapes. In contrast, 2DGS generates noisy meshes due to the lack of geometric prior guidance for early optimization. At the same time, GOF tends to be unstable in specular and highlight regions, resulting in inaccurate surface predictions, whereas our method is more robust to these issues.

We additionally compare our method against state-of-the-art surface reconstruction techniques on the Tanks & Temples dataset. Evaluations are limited to foreground objects, as the ground truth point clouds exclude background regions. As shown in Table 2, our method outperforms all Gaussian-based methods and much more efficient than

[1] https://github.com/maturk/dn-splatter/issues/37
[2] https://github.com/HLinChen/VCR-GauS/issues/3

Table 2
Quantitative results on the Tanks & Temples dataset show that reconstructions are evaluated using the official evaluation scripts, with reported F1-score and average optimization time. Our method outperforms all 3DGS-based surface reconstruction methods and surpasses neural implicit methods by a large margin, while optimizing significantly faster. † Indicates results directly reported from the original paper.

	NeuS-based			Gaussian-based					
	NeuS	MonoSDF	Geo-Neus	SuGaR	2DGS	3DGS	VCR-GauS†	GOF	Ours
Barn	0.29	0.49	0.33	0.14	0.36	0.13	0.62	0.51	0.55
Caterpillar	0.29	0.31	0.26	0.16	0.23	0.08	0.26	0.41	0.42
Courthouse	0.17	0.12	0.12	0.08	0.13	0.09	0.19	0.28	0.26
Ignatius	0.83	0.78	0.72	0.33	0.44	0.04	0.61	0.68	0.76
Meetingroom	0.24	0.23	0.20	0.15	0.16	0.01	0.19	0.28	0.26
Truck	0.45	0.42	0.45	0.26	0.26	0.19	0.52	0.59	0.58
Mean↑	0.38	0.39	0.35	0.19	0.30	0.09	0.40	0.46	0.47
Time↓	>24h	>24h	>24h	>2h	~0.6h	~0.3h	~1h	~2h	~ 0.6h

Table 3
Quantitative evaluation of 2D rendering results on the Mip-NeRF360 and Tanks & Temples datasets shows that our method achieves state-of-the-art NVS results, particularly on the Mip-NeRF360 dataset in terms of LPIPS. † Indicates results directly reported from the original paper.

Methods	Mip-NeRF360			Tanks&Temples		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Plenoxels	23.08	0.626	0.463	21.08	0.719	0.379
Instant-NGP	25.59	0.699	0.311	21.92	0.745	0.305
MipNeRF360	27.69	0.792	0.237	22.22	0.759	0.257
TRIPS	25.94	0.772	0.233	24.64	0.808	0.213
3DGS	27.21	0.815	0.214	23.14	0.841	0.183
SuGaR	25.51	0.756	0.268	22.68	0.794	0.217
2DGS	27.02	0.804	0.238	-	-	-
GOF	27.80	0.837	0.193	23.58	0.850	0.169
AtomGS	27.38	0.816	0.211	23.70	0.849	0.166
VCR-GauS†	27.42	0.814	0.232	-	-	-
Ours	27.84	0.843	0.189	23.72	0.858	0.167

NeuS-based methods. Notably, our model demonstrates exceptional efficiency, offering a reconstruction speed that is approximately 15 times faster compared to NeuS-based reconstruction methods. Compared to the contemporary work VCR-GauS, our method perform much better on F1-score (0.40 vs. 0.47). Moreover, our method improves both efficiency and reconstruction accuracy compared to the previous SOTA work GOF. The visualization results of Mesh and Mesh Normal for some indoor and outdoor scenes are shown in Figure 4.

5.3. Novel View Synthesis

We also evaluate our method against existing techniques on the Mip-NeRF360 datasets to measure their novel view synthesis capabilities. Table 3 presents quantitative results. As we can see from these tables, our method achieves the best average SSIM on the Tanks & Temples dataset and gets the best results on the Mip-NeRF360 dataset. In comparison, the SuGaR, 2DGS and VCR-GauS generate poorer novel view rendering than the standard 3DGS as shown in Table 3. This reveals that these methods have a negative impact on rendering accuracy while achieving improvement in geometric reconstruction, and cannot achieve excellent geometric reconstruction accuracy and rendering quality at the same time. Meanwhile, this comparison demonstrates that our method not only provides superior surface reconstruction quality but also achieves excellent novel view synthesis results compared to current state-of-the-art methods. Some visual comparison results are shown in Figure 3.

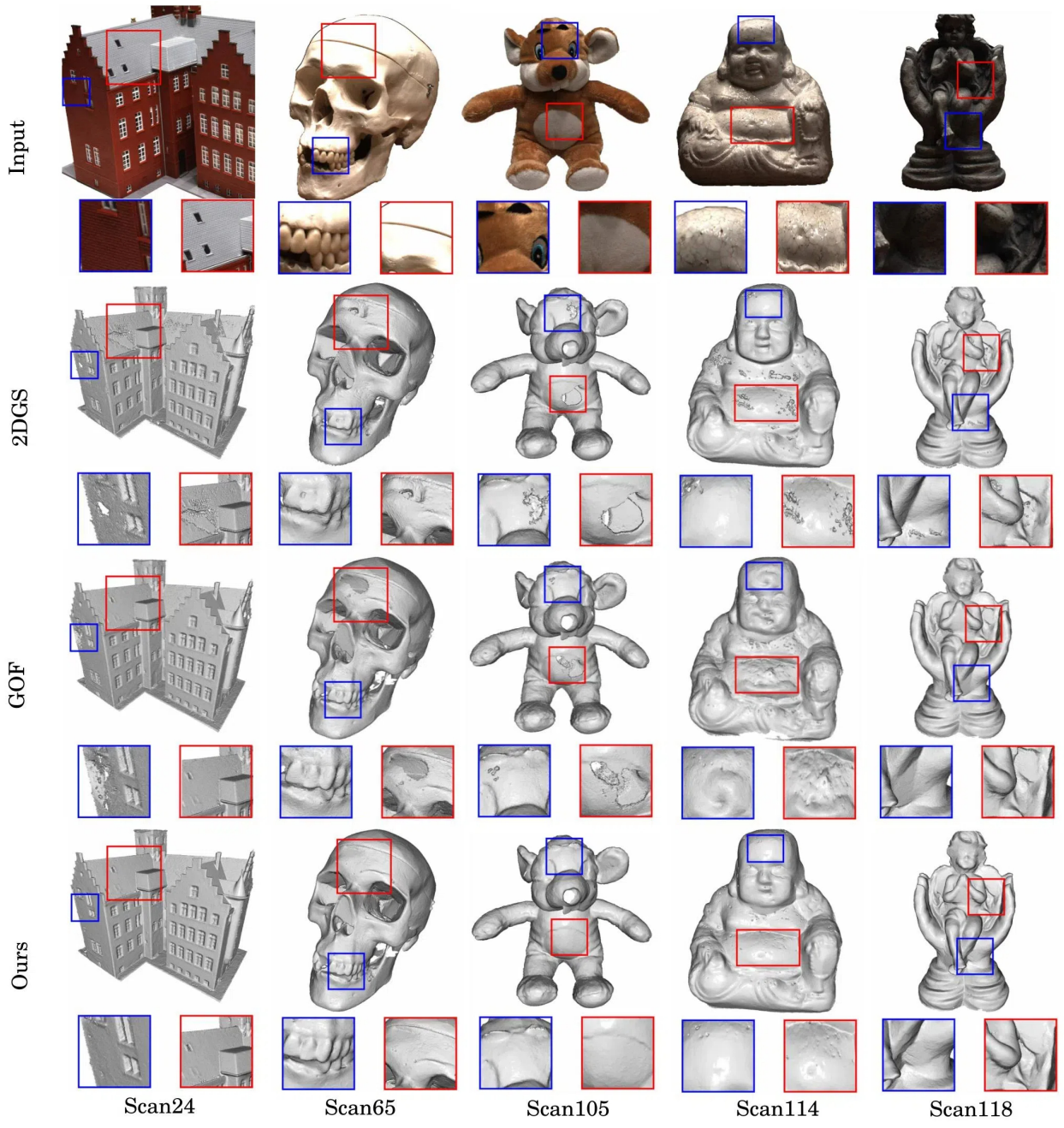


Figure 6: Visual comparison of our method with previous Gaussian-based approaches on the DTU dataset reveals that AND-GS outperforms other baseline methods in capturing scene details, while the baseline methods show missing or noisy surfaces.

5.4. Ablation Study

We conduct a series of ablation studies on Tanks & Temples and DTU to isolate the contribution of each component in AND-GS. Table 4 evaluates the effect of removing the learnable affine alignment and the geometry-aware regularization terms while keeping the remaining optimization pipeline unchanged. The results show that all components contribute positively to the final performance. In particular, removing the affine alignment leads to the largest degradation, which confirms that reliable depth-prior alignment is essential for stabilizing the subsequent

Table 4
Regularization Term Ablation Study on the Tanks & Temples and DTU Datasets (10 Times Average)

	Tanks & Temples			DTU
	PSNR↑	SSIM↑	LPIPS↓	CD↓
w/o learnable affine	23.43±0.03	0.839±0.004	0.181±0.002	0.82±0.02
w/o $\mathcal{L}_D$+$\mathcal{L}_N$+$\mathcal{L}_{N'}$	23.58±0.02	0.850±0.003	0.177±0.003	0.74±0.02
w/o $\mathcal{L}_D$	23.65±0.01	0.855±0.001	0.169±0.001	0.73±0.01
w/o $\mathcal{L}_N$	23.62±0.01	0.853±0.001	0.169±0.001	0.71±0.01
w/o $\mathcal{L}_{N'}$	23.60±0.02	0.851±0.002	0.170±0.001	0.70±0.01
Full Model	**23.72±0.01**	**0.858±0.001**	**0.167±0.001**	**0.68±0.01**

Table 5
Training Strategy Ablation Study on the Tanks & Temples and DTU Datasets (10 Times Average)

	Tanks & Temples			DTU
	PSNR↑	SSIM↑	LPIPS↓	CD↓
w/o Strategy(30k)	**23.76±0.03**	0.852±0.003	0.168±0.002	2.09±0.02
Fixed Strategy(20k)	23.64±0.01	0.853±0.001	0.169±0.001	0.79±0.02
Fixed Strategy(15k)	23.60±0.02	0.851±0.001	0.169±0.001	0.74±0.01
Fixed Strategy(10k)	23.59±0.02	0.851±0.002	0.169±0.001	0.73±0.01
Fixed Strategy(0k)	23.58±0.04	0.850±0.003	0.169±0.001	0.71±0.01
Adaptive Strategy(Ours)	23.72±0.01	**0.858±0.001**	**0.167±0.001**	**0.68±0.01**

optimization. Similarly, removing any of the depth- or normal-related geometric terms leads to consistent drops in both rendering quality and reconstruction accuracy, indicating that these terms play complementary roles rather than acting as redundant regularizers.

Table 5 analyzes the effect of different training schedules and directly validates the proposed adaptive geometry-guided optimization strategy. The non-staged setting applies only scene-level supervision throughout the full 30k iterations, whereas the fixed-strategy variants switch to the second-stage losses at predetermined iterations. In contrast, our adaptive strategy determines the switching point according to the actual state of scene convergence. The results show that all fixed schedules underperform the adaptive policy, which suggests that different scenes require different transition times rather than a single manually selected switching iteration. This observation is consistent with the fact that rendered depth and normal maps remain noisy in the early stage, as illustrated in Figure 1(c), and therefore can easily drive the optimization toward poor local minima if activated too early.

An additional observation from Table 5 is that the non-staged variant attains a slightly higher PSNR but collapses badly in Chamfer Distance. This indicates that our adaptive schedule is not designed to maximize photometric fidelity at all costs; instead, it explicitly prioritizes geometric correctness once the two objectives diverge. We consider this trade-off desirable for surface reconstruction, because a small photometric gain is not meaningful if it is accompanied by severe geometric distortion.

We further examine the influence of hyperparameter choices and pre-trained priors. Table 7 reports the ablation over the loss weights and shows that the setting $(\alpha, \beta) = (10, 0.5)$ achieves the best overall balance between rendering and reconstruction quality, which is therefore adopted in the final model. Table 6 compares different pre-trained depth models and shows that the performance differences are relatively small, indicating that the improvement of AND-GS does not depend on a single specific prior model. Taken together, these results support the overall design of a two-stage optimization scheme in which scene-level geometric supervision first establishes reliable low-frequency structure and the later stage progressively refines high-frequency geometric details. Table 4 also clarifies that the benefit does not come solely from strong monocular priors. Removing the learnable affine alignment or any of the three geometry terms degrades both rendering and reconstruction, which suggests that the final gain comes from how the priors are reconciled with the Gaussian optimization rather than from simply injecting an external depth predictor.

We further analyze the behavior of the adaptive switching mechanism in Table 8 and Table 9. As shown in Table 8, the observed switching iterations vary across Mip-NeRF360, DTU, and Tanks & Temples, indicating that scenes with different geometric complexity and appearance characteristics require different transition times from coarse geometric

Table 6
Pretrain Model Ablation Study on the Tanks & Temples and DTU Datasets

	Tanks & Temples			DTU
	PSNR↑	SSIM↑	LPIPS↓	CD↓
Omnidata[14]	23.66	0.854	0.169	0.71
Metric3D[55]	23.69	0.857	0.169	0.69
ZoeDepth[6]	23.71	0.858	0.168	0.68
Ours	23.72	0.858	0.167	0.68

Table 7
Loss Function Weight Ablation Study on the Tanks & Temples and DTU Datasets

α	β	Tanks & Temples			DTU
		PSNR↑	SSIM↑	LPIPS↓	CD↓
1	0.5	23.54	0.849	0.171	0.76
10	0.5	**23.72**	**0.858**	**0.167**	**0.68**
50	0.5	23.45	0.841	0.179	0.80
1	1	23.52	0.844	0.176	0.74
10	1	23.57	0.853	0.170	0.72
50	1	23.55	0.851	0.172	0.75

Table 8
Observed switching iterations of the adaptive strategy across datasets.

Dataset	Mean switch iter.	Std.	Range
Mip-NeRF360	10092	723	9033-11005
DTU	7024	351	6505-7521
Tanks & Temples	9324	462	8521-10052

Table 9
Sensitivity analysis for the adaptive switching threshold and spectral-energy cutoff on Tanks & Temples.

SSIM Thr.	Energy cutoff	PSNR↑	SSIM↑	LPIPS↓	CD↓
0.80	90%	23.43	0.849	0.169	0.76
0.85	95%	23.72	0.858	0.167	0.68
0.90	98%	23.51	0.852	0.169	0.72

supervision to detail-oriented refinement. This observation explains why a single fixed switching point is suboptimal across datasets. Table 9 further evaluates the sensitivity of the method to the SSIM threshold and the spectral-energy cutoff. The best performance is achieved with the default setting of SSIM threshold 0.85 and energy cutoff 95%, while neighboring configurations lead only to limited performance variation. This result suggests that the proposed switching criterion is not only effective but also reasonably robust to moderate hyperparameter changes.

We additionally provide two complementary analyses to better characterize the robustness and efficiency of the proposed framework. Table 10 compares different monocular normal priors and shows that DSINE yields the strongest overall performance. Nevertheless, the performance gap between different priors remains moderate, which indicates that the advantage of AND-GS does not rely exclusively on a single normal predictor but instead stems from the overall optimization framework. Table 11 reports the end-to-end runtime and peak memory usage of each stage. The results show that Gaussian optimization accounts for most of the total runtime, whereas TSDF fusion determines the peak memory consumption. This breakdown helps clarify that the efficiency advantage of AND-GS mainly originates from the optimization stage rather than from excluding post-processing.

Table 10
Ablation over normal priors on Tanks & Temples.

Normal prior	PSNR↑	SSIM↑	LPIPS↓	CD↓
DSINE [3]	23.72	0.858	0.167	0.68
Omnidata [22]	23.61	0.851	0.170	0.72
EESNU [2]	23.58	0.848	0.171	0.76

Table 11
End-to-end runtime and memory breakdown.

Stage	Runtime (min)	Peak memory (GB)
Monocular depth prior generation	1	7
Monocular normal prior generation	1	6
Gaussian optimization	25	15
TSDF fusion	3	21
Marching cubes	1	8
Total	31	21

6. Limitation

Although our AND-GS can efficiently and faithfully reconstruct geometry, it also faces some challenges. First, most neural networks used to predict depth and normals are trained based on low resolution (about 512x512). Therefore, when inputting ultra-high resolution images (such as 4096x4096), the prediction network cannot output high-definition and detailed results, which may cause the model to be limited by the performance of the prediction network in the initial stage. However, this affects the first stage more directly than the second stage, which further supports the effectiveness of our adaptive staged optimization strategy. Secondly, due to GPU memory limitations, our current TSDF fusion is restricted to low-resolution voxel grids for large-scale scenes, thus hindering accurate surface extraction from Gaussians. For large-scale scenes, our method typically requires 16-24 GB of GPU memory for TSDF fusion and mesh extraction at a voxel size of 2-4 mm. The mesh extraction process, implemented with parallelized marching cubes, takes approximately 1-3 minutes per scene depending on the scene size and voxel resolution. However, we will address this limitation in the future by integrating an adaptive TSDF fusion technique, which can be used to extract meshes in a coarse-to-fine manner, thus reducing memory consumption but achieving better geometric quality. We further observe failure cases in thin structures, reflective regions, textureless surfaces, and views where the monocular depth prior is severely biased. In these scenarios, the first stage may inherit over-smoothed normals or erroneous depth ordering, while the final mesh can still be limited by the resolution of TSDF fusion. Therefore, the final error can arise from two different sources: inaccurate monocular priors and the downstream mesh extraction bottleneck. This failure decomposition is important because it shows that AND-GS improves robustness but does not eliminate the dependency on prior quality and extraction resolution.

7. Conclusion

In this paper, we introduce AND-GS, a novel approach that adaptively supervises the rendered depth and normal in 3DGS optimization procedure for accurate and efficient surface reconstruction. Our method takes advantage of both the monocular predicted 3D scene cues and the inherent constraints between depths and normals, and proposes a new training strategy that adaptively switches between these two constraints according to the similarity between low-frequency information of the rendered images and the input image. This strategy can significantly improve the convergence speed and the final reconstruction quality. Our method, based on general Gaussian primitives, can be seamlessly integrated into any Gaussian-Splatting-based approach. We validate the rendering and reconstruction quality on the MipNeRF360, DTU, and Tanks & Temples datasets. Experiments across multiple public datasets show that our AND-GS outperforms previous implicit and explicit methods. We position the method primarily as a geometry-oriented Gaussian reconstruction framework whose adaptive supervision schedule preserves competitive novel view synthesis instead of optimizing photometric quality in isolation.

References

[1] Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B., 2016. Large-scale data for multiple-view stereopsis. International Journal of Computer Vision 120, 153–168.

[2] Bae, G., Budvytis, I., Cipolla, R., 2021. Estimating and exploiting the aleatoric uncertainty in surface normal estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13137–13146.

[3] Bae, G., Davison, A.J., 2024. Rethinking inductive biases for surface normal estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9535–9545.

[4] Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, Peter, 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5470–5479.

[5] Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P., 2023. Zip-nerf: Anti-aliased grid-based neural radiance fields, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 19697–19705.

[6] Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M., 2023. Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 .

[7] Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H., 2022. Tensorf: Tensorial radiance fields, in: European conference on computer vision, Springer. pp. 333–350.

[8] Chen, D., Li, H., Ye, W., Wang, Y., Xie, W., Zhai, S., Wang, N., Liu, H., Bao, H., Zhang, G., 2024a. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. IEEE Transactions on Visualization and Computer Graphics .

[9] Chen, G., Wang, W., 2024. A survey on 3d gaussian splatting. arXiv preprint arXiv:2401.03890 .

[10] Chen, H., Li, C., Lee, G.H., 2023. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. arXiv preprint arXiv:2312.00846 .

[11] Chen, H., Wei, F., Li, C., Huang, T., Wang, Y., Lee, G.H., 2024b. Vcr-gaus: View consistent depth-normal regularizer for gaussian surface reconstruction. arXiv preprint arXiv:2406.05774 .

[12] Dai, F., Zhu, C., Ma, Y., Cao, J., Zhao, Q., Zhang, Y., 2019. Freely explore the scene with 360 field of view, in: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), IEEE. pp. 888–889.

[13] Drebin, R.A., Carpenter, L., Hanrahan, P., 1988. Volume rendering. ACM Siggraph Computer Graphics 22, 65–74.

[14] Eftekhar, A., Sax, A., Malik, J., Zamir, A., 2021. 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.

[15] Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A., 2022. Plenoxels: Radiance fields without neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5501–5510.

[16] Fu, Y., Hong, Y., Zou, Y., Liu, Q., Zhang, Y., Liu, N., Yan, C., 2023. Raw image based over-exposure correction using channel-guidance strategy. IEEE Transactions on Circuits and Systems for Video Technology .

[17] Fu, Z., Mao, Z., Yan, C., Liu, A.A., Xie, H., Zhang, Y., 2021. Self-supervised synthesis ranking for deep metric learning. IEEE Transactions on Circuits and Systems for Video Technology 32, 4736–4750.

[18] Guédon, A., Lepetit, V., 2024. 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.

[19] Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P., 2021. Baking neural radiance fields for real-time view synthesis, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 5875–5884.

[20] Hu, R., Wang, X., Zhao, C., 2025. Identity aware 3d face reconstruction from in-the-wild images. Neurocomputing , 130299.

[21] Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S., 2024. 2d gaussian splatting for geometrically accurate radiance fields. arXiv preprint arXiv:2403.17888 .

[22] Kar, O.F., Yeo, T., Atanov, A., Zamir, A., 2022. 3d common corruptions and data augmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18963–18974.

[23] Kazhdan, M., Hoppe, H., 2013. Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG) 32, 1–13.

[24] Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., 2023. 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. 42, 139–1.

[25] Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V., 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) 36, 1–13.

[26] Kutulakos, K.N., Seitz, S.M., 2000. A theory of shape by space carving. International journal of computer vision 38, 199–218.

[27] Li, C., Feng, B.Y., Fan, Z., Pan, P., Wang, Z., 2023a. Steganerf: Embedding invisible information within neural radiance fields, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 441–453.

[28] Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H., 2023b. Neuralangelo: High-fidelity neural surface reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8456–8465.

[29] Li, Z., Yao, S., Chu, Y., Garcia-Fernandez, A.F., Yue, Y., Ding, W., Zhu, X., 2025. Mvg-splatting: Multi-view guided gaussian splatting with adaptive quantile-based geometric consistency densification. Information Fusion , 103540.

[30] Liu, H., Liu, Y., Li, C., Li, W., Yuan, Y., 2024a. Lgs: A light-weight 4d gaussian splatting for efficient surgical scene reconstruction, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 660–670.

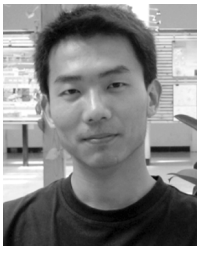
[31] Liu, H., Wang, Y., Li, C., Cai, R., Wang, K., Li, W., Molchanov, P., Wang, P., Wang, Z., 2025a. Flexgs: Train once, deploy everywhere with many-in-one flexible 3d gaussian splatting, in: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 16336–16345.

[32] Liu, J., Cheng, H., Wang, S., Zhao, F., Li, M., 2025b. Nerf dynamic scene reconstruction based on motion, semantic information and inpainting. Neurocomputing 630, 129653.

[33] Liu, R., Xu, R., Hu, Y., Chen, M., Feng, A., 2024b. Atomgs: Atomizing gaussian splatting for high-fidelity radiance field. arXiv preprint arXiv:2405.12369 .

[34] Liu, Y., Fan, K., Yu, W., Li, C., Lu, H., Yuan, Y., 2025c. Monosplat: Generalizable 3d gaussian splatting from monocular depth foundation models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21570–21579.
[35] Liu, Y., Li, C., Yang, C., Yuan, Y., 2024c. Endogaussian: Gaussian splatting for deformable surgical scene reconstruction. arXiv preprint arXiv:2401.12561 3, 8.
[36] Lorensen, W.E., Cline, H.E., 1998. Marching cubes: A high resolution 3d surface construction algorithm, in: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353.
[37] Lu, R., Chen, H., Zhu, Z., Qin, Y., Lu, M., Zhang, L., Yan, C., Xue, A., 2024. Thermalgaussian: Thermal 3d gaussian splatting. arXiv preprint arXiv:2409.07200 .
[38] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R., 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM 65, 99–106.
[39] Müller, T., Evans, A., Schied, C., Keller, A., 2022. Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) 41, 1–15.
[40] Oechsle, M., Peng, S., Geiger, A., 2021. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5589–5599.
[41] Schonberger, J.L., Frahm, J.M., 2016. Structure-from-motion revisited, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[42] Turkulainen, M., Ren, X., Melekhov, I., Seiskari, O., Rahtu, E., Kannala, J., 2024. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. arXiv preprint arXiv:2403.17822 .
[43] Wang, C., Reza, M.A., Vats, V., Ju, Y., Thakurdesai, N., Wang, Y., Crandall, D.J., Jung, S.h., Seo, J., 2024. Deep learning-based 3d reconstruction from multiple images: A survey. Neurocomputing 597, 128018.
[44] Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W., 2021. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 .
[45] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13, 600–612.
[46] Wu, J., Wyman, O., Tang, Y., Pasini, D., Wang, W., 2024. Multi-view 3d reconstruction based on deep learning: A survey and comparison of methods. Neurocomputing 582, 127553.
[47] Yan, C., Li, L., Zhang, C., Liu, B., Zhang, Y., Dai, Q., 2019. Cross-modality bridging and knowledge transferring for image understanding. IEEE Transactions on Multimedia 21, 2675–2685.
[48] Yan, C., Shao, B., Zhao, H., Ning, R., Zhang, Y., Xu, F., 2020. 3d room layout estimation from a single rgb image. IEEE Transactions on Multimedia 22, 3014–3024.
[49] Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H., 2024. Depth anything: Unleashing the power of large-scale unlabeled data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10371–10381.
[50] Yang, X., Mohamed, A.S.A., 2025. Gaussian-based r-cnn with large selective kernel for rotated object detection in remote sensing images. Neurocomputing 620, 129248.
[51] Yang, X., Xie, W., Peng, S., Fu, Y., Fan, W., Yang, B., Dong, X., 2025. 4d gaussian splatting for high-fidelity dynamic reconstruction of single-view scenes. Neurocomputing , 130262.
[52] Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L., 2018. Mvsnet: Depth inference for unstructured multi-view stereo, in: Proceedings of the European conference on computer vision (ECCV), pp. 767–783.
[53] Yariv, L., Gu, J., Kasten, Y., Lipman, Y., 2021. Volume rendering of neural implicit surfaces, in: Thirty-Fifth Conference on Neural Information Processing Systems.
[54] Yariv, L., Hedman, P., Reiser, C., Verbin, D., Srinivasan, P.P., Szeliski, R., Barron, J.T., Mildenhall, B., 2023. Bakedsdf: Meshing neural sdfs for real-time view synthesis, in: ACM SIGGRAPH 2023 Conference Proceedings, pp. 1–9.
[55] Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., Shen, C., 2023. Metric3d: Towards zero-shot metric 3d prediction from a single image, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9043–9053.
[56] Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A., 2024a. Mip-splatting: Alias-free 3d gaussian splatting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19447–19456.
[57] Yu, Z., Gao, S., 2020. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1949–1958.
[58] Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A., 2022. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. Advances in neural information processing systems 35, 25018–25032.
[59] Yu, Z., Sattler, T., Geiger, A., 2024b. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. arXiv preprint arXiv:2404.10772 .
[60] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 586–595.
[61] Zhao, Q., Dai, F., Lv, J., Ma, Y., Zhang, Y., 2019. Panoramic light field from hand-held video and its sampling for real-time rendering. IEEE Transactions on Circuits and Systems for Video Technology 30, 1011–1021.
[62] Zhao, Q., Wan, L., Feng, W., Zhang, J., Wong, T.T., 2013. Cube2video: Navigate between cubic panoramas in real-time. IEEE Transactions on Multimedia 15, 1745–1754.
[63] Zhou, H., Shao, J., Xu, L., Bai, D., Qiu, W., Liu, B., Wang, Y., Geiger, A., Liao, Y., 2024. Hugs: Holistic urban 3d scene understanding via gaussian splatting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21336–21345.
[64] Zhu, Z., Wei, Y., Lu, R., Xu, C., Le, X., Zheng, B., Yan, C., Xu, F., 2024. Indoor scene reconstruction using a rotating device and multiple rgb-d cameras. IEEE Transactions on Instrumentation and Measurement .

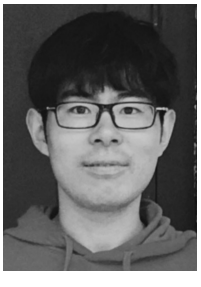
Xiang Le received the B.S. degree in automation from Hangzhou Dianzi University, Hangzhou, China, in 2023, where he is currently pursuing the Ph.D. degree with the Department of Automation. His research interests include surface reconstruction, novel view synthesis, gaussian splatting and SLAM.



Qiang Zhao received the B.Eng. degree in software engineering and the Ph.D. degree in computer science and technology from Tianjin University, Tianjin, China, in 2009 and 2016, respectively. He is currently a Professor with the school of communication engineering, Hangzhou Dianzi University, Hangzhou, china. Before that, he was an Assistant Professor and Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His main research interests include image based rendering, feature extraction, and panoramic image processing.

Haofan Ren received the B.S degree in automation from Hangzhou Dianzi University, Hangzhou, China, in 2020. He is currently working toward the master's degree with the Department of Automation, Hangzhou Dianzi University. His research insterests include neural point field, 3D reconstruction, simultaneous location, and mapping.

Zhongtian Zheng received the B.S. degree from the University of Manchester in 2023. He is currently pursuing the master's degree with the University of Queensland. His main research interests include novel view synthesis and radiance fields.



Tingyu Wang received the Ph.D. degree from the Lab of Intelligent Information Processing, Hangzhou Dianzi University, in 2023. He is a research assistant at School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China. His research interests include deep learning, image retrieval and remote sensing.

Jiyong Zhang received the B.S. degree and M.S. degree in computer science from Tsinghua University in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the Swiss Federal Institute of Technology at Lausanne (EPFL) in 2008. He is currently a Distinguished Professor at Hangzhou Dianzi University. His research interests include artificial intelligence, machine learning, data mining and image processing.

Chenggang Yan received the B.S. degree in computer science from Shandong University in 2008 and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2013. He is currently a Professor with Hangzhou Dianzi University. His research interests include intelligent information processing, machine learning, image processing, computational biology, and computational photography.