MeshSplat: Generalizable Sparse-View Surface Reconstruction via Gaussian Splatting

Hanzhi Chang[1*], **Ruijie Zhu**[1, 2*], **Wenjie Chang**[1], **Mulin Yu**[2],
Yanzhe Liang[1], **Jiahao Lu**[1], **Zhuoyuan Li**[1], **Tianzhu Zhang**[1, 3†]
[1]University of Science and Technology of China
[2]Shanghai AI Laboratory
[3]Hainan Aerospace Technology Innovation Center

Abstract

Surface reconstruction has been widely studied in computer vision and graphics. However, existing surface reconstruction works struggle to recover accurate scene geometry when the input views are extremely sparse. To address this issue, we propose MeshSplat, a generalizable sparse-view surface reconstruction framework via Gaussian Splatting. Our key idea is to leverage 2DGS as a bridge, which connects novel view synthesis to learned geometric priors and then transfers these priors to achieve surface reconstruction. Specifically, we incorporate a feed-forward network to predict per-view pixel-aligned 2DGS, which enables the network to synthesize novel view images and thus eliminates the need for direct 3D ground-truth supervision. To improve the accuracy of 2DGS position and orientation prediction, we propose a Weighted Chamfer Distance Loss to regularize the depth maps, especially in overlapping areas of input views, and also a normal prediction network to align the orientation of 2DGS with normal vectors predicted by a monocular normal estimator. Extensive experiments validate the effectiveness of our proposed improvement, demonstrating that our method achieves state-of-the-art performance in generalizable sparse-view mesh reconstruction tasks.

Project — https://hanzhichang.github.io/meshsplat_web

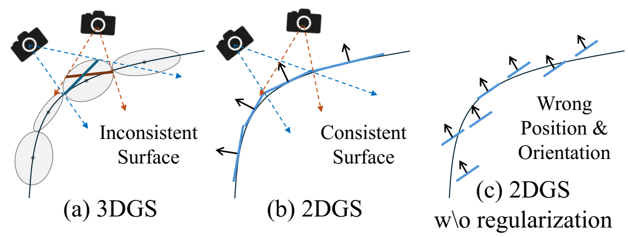


Figure 1: Motivation. (a) The ellipsoid shape of 3DGS leads to different intersection planes in different viewpoints, resulting in inconsistent surface. (b) 2DGS has consistent intersection planes in different viewpoints, which is more suitable for surface reconstruction. (c) When the positions and orientations of 2DGS are not regularized, there will be significant discrepancies between 2DGS and the contours of the surface, which hinders the reconstruction of scene surfaces.

1 Introduction

Surface reconstruction of 3D scenes is a fundamental task in 3D vision, with a wide range of applications in downstream tasks such as AR/VR and embodied AI. Recently, based on NeRF (Mildenhall et al. 2021) and 3DGS (Kerbl et al. 2023), numerous per-scene optimized surface reconstruction methods (Wang et al. 2021a; Huang et al. 2024a; Chen, Li, and Lee 2023; Yu et al. 2024; Guédon and Lepetit 2024) have been proposed. However, these methods struggle to robustly reconstruct scenes with only sparse views as input. This is because sparse views can only provide limited multi-view geometric constraints, which are insufficient to perform high-quality per-scene geometry optimization. Therefore, it is necessary to construct a feed-forward pipeline to learn geometric priors from diverse scenarios for generalizable sparse-view surface reconstruction.

To address this issue, several meaningful attempts have been made, which are mainly NeuS-based methods (Wang et al. 2021a). For instance, SparseNeuS (Long et al. 2022) constructs geometric volumes to estimate implicit SDF fields, which are then transformed to mesh. However, due to the inefficiency of implicit SDF representation and neural rendering, NeuS-based methods are limited to object-centric scenes and suffer from long rendering times. Compared with NeuS, Gaussian splatting can achieve not only faster rendering speeds but also better rendering quality. However, the vanilla Gaussian splatting framework falls short in representing surface geometry (Huang et al. 2024a). Therefore, existing Gaussian splatting-based methods (Charatan et al. 2024; Chen et al. 2024c; Wewer et al. 2024) mainly focus on generalizable novel view synthesis, while methods applying Gaussian splatting to generalizable sparse-view surface reconstruction remain unexplored.

To realize high-quality surface reconstruction, the network requires not only realistic appearance rendering but also precise geometry recovery. More importantly, the former task usually has more available training data, while the latter requires expensive geometric labeling. This raises a question: **is it possible to learn generalizable geometric**

[*]These authors contributed equally.
[†]Corresponding Author.

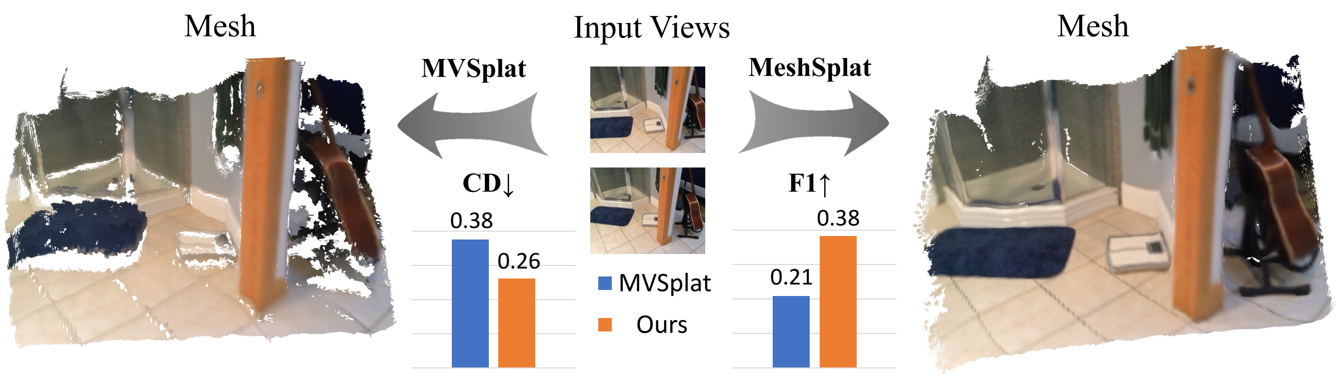


Figure 2: Given sparse-view images as input, MeshSplat can directly predict the scene geometry and efficiently extract the scene mesh. Compared to MVSplat (Chen et al. 2024c) and other state-of-the-art methods, Meshplat achieves more consistent and precise mesh extraction in generalizable sparse-view surface reconstruction.

priors from novel view synthesis task? Motivated by recent work 2DGS (Huang et al. 2024a), we find an effective way to address this issue: using 2DGS as a bridge between novel view synthesis (NVS) and surface reconstruction. Compared with 3DGS, 2DGS is naturally more suitable for representing thin surfaces and can be easily used to extract surface mesh beyond rendering novel views, as shown in Figure 1 (a) and (b). To this end, we decide to construct a feed-forward framework to predict pixel-aligned 2DGS for generalizable sparse-view surface reconstruction.

However, we also find that integrating 2DGS into generalizable sparse-view 3D scene reconstruction is not a trivial task. Compared to 3DGS, 2DGS is more sensitive to **position** and **orientation** estimation in mesh reconstruction, as illustrated in Figure 1 (c). For position, it can be iteratively optimized in a per-scene optimization setting. However, in an end-to-end framework, the Gaussian positions depend on the predicted depth maps. Due to the thin nature of 2DGS, mispredictions in the depth map directly lead to noticeable position shifts, unlike 3DGS. For orientation, unlike 3DGS whose orientations can be more flexible, the orientation of 2DGS directly determines the surface normals of the scene. Errors in predicting Gaussian orientation will ultimately result in distorted scene surfaces.

To address these challenges, we propose **MeshSplat**, a generalizable sparse-view surface reconstruction framework based on 2DGS, which consists of the following key designs: (1) Given sparse images as input, we build an **MVS-based feed-forward network to generate pixel-aligned 2DGS**, where the position and orientation of 2DGS are transformed according to the predicted depth and normal maps, and other attributes of 2DGS are decoded directly from a convolutional Gaussian head. With the predicted 2DGS, we can synthesize novel views for supervision and finally perform mesh extraction. (2) To improve the accuracy of 2DGS position prediction, we design a **Weighted Chamfer Distance Loss** to align point clouds generated from predicted depths across different views. Since adjacent views have varying overlaps, we introduce a confidence map to identify overlapping regions and weight the Chamfer Distance Loss accordingly, assigning higher weights to these overlapping areas. (3) To improve the accuracy of 2DGS orientation prediction, we design an **uncertainty-based normal prediction network** that predicts per-view normals and converts them to rotation quaternions for 2DGS. The per-view normals are supervised by an off-the-shelf monocular normal estimator, which incorporates monocular geometry priors to assist surface reconstruction.

Our main contributions can be summarized as follows:

- We propose MeshSplat, the first method, to the best of our knowledge, that focuses on generalizable sparse-view surface reconstruction via Gaussian splatting.
- By incorporating 2DGS instead of 3DGS as a bridge, MeshSplat can learn generalizable geometric priors from the novel view synthesis task in a self-supervised manner.
- To better integrate 2dgs into the feed-forward framework, we introduce a Weighted Chamfer Distance Loss to align Gaussian position across views and design an uncertainty-based normal prediction network that provides monocular supervision to Gaussian orientations.
- Extensive experiments show that our method achieves state-of-the-art performance in generalizable sparse-view surface reconstruction and cross-dataset generalization.

2 Related Work

2.1 3D Scene Representation

Early 3D scene methods (Jiang et al. 2020; Genova et al. 2020; Cheng et al. 2025) employ point clouds, signed distance functions or occupancy fields to represent 3D scenes. NeRF (Mildenhall et al. 2021) utilizes neural networks to construct the 3D radiance field for scene representation and renders RGB images by volume rendering, which inspires many subsequent works in 3D vision (Barron et al. 2021; Müller et al. 2022; Hong et al. 2023). However, NeRF-based methods suffer from long-time inefficient rendering. 3DGS (Kerbl et al. 2023) models 3D scenes by explicit ellipsoidal Gaussian primitives and introduces a differentiable rendering method based on alpha-blending. It has become the core 3D scene representation for many tasks, such as novel view synthesis (Ren et al. 2025; Lu et al. 2024), 3D assets generation (Tang et al. 2023) and dynamic scene reconstruction (Huang et al. 2024b; Zhu et al. 2024). Despite their success, original NeRF and 3DGS require per-scene optimization, lacking cross-scene generalization capabilities.

2.2 Generalizable 3D Reconstruction

Extensive works have proposed generalizable 3D reconstruction frameworks by training neural networks on large-scale datasets to learn cross-scene priors. Early approaches (Chen et al. 2021; Wang et al. 2021b; Xu et al. 2024) primarily combined feed-forward neural networks with Neural Radiance Fields (NeRF). However, these methods often suffer from the inefficiencies inherent in NeRF training and rendering, which limit their performance. More recently, methods such as DUSt3R and its subsequent extensions (Wang et al. 2024a; Duisterhof et al. 2025; Wang et al. 2025a) leverage Transformer-based architectures trained on large-scale 3D-annotated datasets to predict 3D point maps from input 2D images. While effective in generating point-based 3D representations, these approaches cannot generalize to novel views and are not suitable for surface reconstruction tasks. Methods like LVSM (Jin et al. 2024) support novel view synthesis (NVS) tasks, but they are limited to generating RGB images and thus cannot be directly applied to surface reconstruction. With the introduction of 3D Gaussian Splatting (3DGS), a new class of methods (Charatan et al. 2024; Chen et al. 2024c; Wang et al. 2024b, 2025b), where MVSplat (Chen et al. 2024c) represents a prominent approach, employ 3DGS as the 3D scene representation for generalizable 3D reconstruction. They usually first predict depth maps from input views, and then use back-projected 3D points as centers to construct pixel-aligned 3D Gaussian representations. PM loss (Shi et al. 2025) further regularizes the predicted 3DGS positions by introducing VGGT priors. However, despite their promising performance in NVS tasks, these generalizable frameworks remain inadequate for mesh extraction and surface reconstruction tasks.

2.3 Surface Reconstruction

Traditional surface reconstruction methods (Kazhdan and Hoppe 2013; Furukawa and Ponce 2009) primarily model 3D scene geometry through explicit feature matching. With the rise of neural networks, many works (Wang et al. 2021a; Reiser et al. 2024; Li et al. 2023; Yariv et al. 2021) reconstruct scene surface by estimating neural occupancy fields or signed distance functions (SDF) by neural networks, and render novel views through surface rendering or volume rendering techniques. Recently, many works have extended Gaussian Splatting to mesh extraction. These methods can be categorized into two types: methods that directly use ellipsoidal 3DGS to extract meshes (Guédon and Lepetit 2024; Yu et al. 2024; Yu, Sattler, and Geiger 2024; Zhu et al. 2025), and methods that use flattened 3DGS or 2DGS (Huang et al. 2024a; Chen, Li, and Lee 2023; Chen et al. 2024a). However, these methods fail to generate high-quality meshes under sparse viewpoint inputs and lack cross-scene generalization. SparseNeuS (Long et al. 2022) and subsequent works (Peng et al. 2023; Xu et al. 2023; Chen et al. 2024b), inspired by generalizable novel view synthesis approaches, achieve generalizable surface reconstruction based on NeuS (Wang et al. 2021a). These methods extract feature maps from input images to construct a 3D geometry volume, which is used to obtain neural SDF fields and output the mesh of the scene. However, these methods exhibit poor efficiency in mesh extraction and are limited to object-centric scenes. 2DGS (Huang et al. 2024a) outperforms NeuS-based methods in mesh extraction tasks, but how to generalize 2DGS to different scenes with sparse inputs still remains unexplored. Therefore, we propose MeshSplat, an end-to-end generalizable sparse-view surface reconstruction framework based on 2DGS.

3 Methods

3.1 Overall Framework

The framework of MeshSplat is illustrated in Figure 3. Given two images $\{I_i\}_{i=1}^{2}$ and corresponding projection matrices$\{\Pi_i\}_{i=1}^{2}$, $\Pi_i = K_i[R_i|t_i]$, where K_i denotes for camera intrinsics, R_i denotes for camera rotation matrices and t_i denotes for translation matrices, MeshSplat begins with an encoder composed of a CNN and Multi-View Transformer to extract feature maps. From these multi-view feature maps, per-view cost volumes are constructed via plane sweeping. To enhance the multi-view consistency, we propose a Weighted Chamfer Distance Loss to constrain the cost volumes. Subsequently, we use a Gaussian Prediction Network, which includes a normal prediction network to predict Gaussian orientations, and a depth refinement network to predict Gaussian positions. In this way, we can obtain the pixel-aligned 2DGS to represent the scene. This process can be formulated as:

$$\{I_i, \Pi_i\}_{i=1}^{2} \rightarrow \{\mu_j, s_j, r_j, \alpha_j, c_j\}_{j=1}^{2\times H\times W} \tag{1}$$

Finally, these predicted 2DGS can render novel views and reconstruct the scene mesh.

3.2 Cost Volume Construction

Following MVSplat (Chen et al. 2024c), we first use a CNN to extract a feature map from each input view. Then, we employ a Transformer that incorporates multi-view cross-attention to enable information exchanges across different views. As a result, we obtain the feature map $\{F_i\}_{i=1}^{2}$ for each view with a size of $\frac{H}{4} \times \frac{W}{4} \times C$.

To intuitively obtain the matching correspondences between the two input images, we construct per-view cost volumes via plane sweeping. Specifically, for input view i, we first divide the depth ranges into several depth bins $\{d_k\}_{k=1}^{D}$. The feature map F_j from the other view j is then warped to view i using the projection matrices P_i and P_j along with the current depth candidate d_k, denoted as $F_{j\rightarrow i}^{d_k}$. Next, we compute the dot product between the two feature maps and concatenate the results from all depth candidates. After two convolutional layers ϕ_{CV}, we finally obtain the cost volume V_i with a size of $\frac{H}{4} \times \frac{W}{4} \times D$:

$$V_i = \phi_{\mathrm{CV}}(\{V_i^{d_k}\}_{k=1}^{D}), \quad V_i^{d_k} = \frac{F_i \cdot F_{j\rightarrow i}^{d_k}}{\sqrt{C}} \tag{2}$$

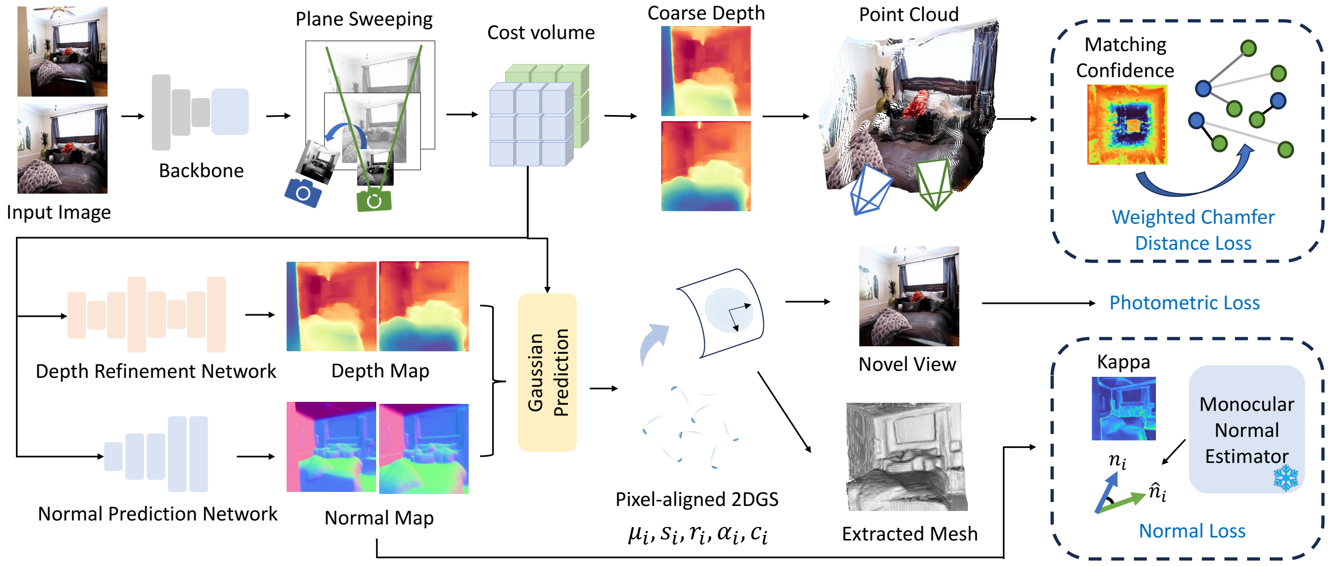


Figure 3: Overall Architecture. Taken a pair of images as input, MeshSplat begins with a multi-view backbone to extract per-view feature maps. Then we construct per-view cost volumes via the plane-sweeping to generate coarse depth maps, which can be projected to 3D point clouds and be constrained by our proposed Weighted Chamfer Distance Loss. We further feed cost volumes into our gaussian prediction network, together with a depth refinement network and a normal prediction network, to obtain pixel-aligned 2DGS. Finally, we use these 2DGS to render novel view for supervision and reconstruct the scene mesh.

Since the correlation calculation directly reflects the matching confidence between the input images, we can directly apply Softmax operation to compute the probabilities of the depth values to be each depth candidates. The coarse depth maps can be obtained by weighted summation:

$$W_i = \mathrm{Softmax}_D(V_i), \quad D_i^{\mathrm{coarse}} = \sum_k W_i^k d_k \tag{3}$$

3.3 Gaussian Prediction Network

Next, we discuss how to derive the attributes of pixel-aligned 2DGS from cost volumes. Following MVSplat (Chen et al. 2024c), we first use U-Net with attention layers to refine the cost volume, taking feature maps as additional inputs. Using the refined cost volume, we can compute the final depth maps using equation (3), and then perform back-projection to obtain the 3D positions of the pixel-aligned 2DGS.

In addition, 2DGS requires more precise orientation prediction than the ellipsoid-like 3DGS. Based on this, we introduce an additional normal prediction network to predict the normal maps of the input images, which are further converted to the 2DGS rotation quaternions. We use a lightweight CNN ϕ_{rot} to predict the orientations of the pixel-aligned 2DGS. The network takes the refined cost volumes, feature maps, and original RGB images at three different scales as input, and processes them through convolutional layers with residual connections. Finally the rotation quaternions q for all three scales are obtained, which are further converted into normal vectors n for each scale, along with kappa κ which reflect the uncertainty of predicted normal vectors following (Bae, Budvytis, and Cipolla 2021):

$$\{q, \kappa\} = \phi_{\mathrm{rot}}(V_i || F_i || I_i), \quad n = R(q) \cdot [0, 0, 1]^T \tag{4}$$

where $R(q)$ represents the rotation matrix computed from the rotation quaternion q. The resulting normal vectors n and kappa κ will be used in subsequent loss calculations.

For the remaining attributes of 2DGS, following (Chen et al. 2024c), we predict them through a two-layer convolutional gaussian head, taking the refined cost volumes, feature maps, and original RGB images as input.

3.4 Weighted Chamfer Distance Loss

In the ideal case, the Gaussian positions predicted from adjacent views should have significant overlaps. Considering this property, one approach is to use the Chamfer Distance (CD) Loss as regularization on these 3D points. Chamfer Distance measures the overlaps between two point clouds, which can be expressed as:

$$\mathcal{L}_{\mathrm{CD}} = \frac{1}{2}\left(\frac{1}{N_1}\sum_{i=1}^{N_1}\min_j ||p_1^i - p_2^j|| + \frac{1}{N_2}\sum_{i=1}^{N_2}\min_j ||p_2^i - p_1^j||\right) \tag{5}$$

where $\{p_k^i\}_{i=1}^{N_k}$ are the predicted point clouds calculated by the projection matrices Π_k and coarse depth maps D_k^{coarse} of the k-th view. The original CD Loss assigns equal weights to all points. However, due to occlusion and view differences, those pixels which do not have their corresponding pixels will have far chamfer distances. Applying CD loss uniformly to all points may result in incorrect and unreasonable constraints.

To address this issue, we design confidence maps to measure the matching confidence of each pixel and use them to weight CD loss, resulting in our proposed Weighted Chamfer Distance Loss (WCD Loss). We argue that confidence maps can be derived from the cost volumes, since they have already captured feature similarities between pixels. The confidence map M_i for view i is defined by taking the maximum value along depth dimension in the cost volume:

$$M_i = \max_{d_k} \mathrm{Softmax}_D(V_i), \quad i = 1, 2 \tag{6}$$

With the confidence maps, we define the WCD Loss as:

$$\begin{aligned}\mathcal{L}_{\mathrm{WCD}} = \frac{1}{2}(\frac{1}{N_1}\sum_{i=1}^{N_1} M_1(i) \min_j ||p_1^i - p_2^j|| \\ + \frac{1}{N_2}\sum_{i=1}^{N_2} M_2(i) \min_j ||p_2^i - p_1^j||)\end{aligned} \tag{7}$$

where $M_{1/2}(i)$ denotes the confidence map value for pixel i in the first or second view.

Our proposed WCD loss ensures that the regions with higher matching confidence overlap as much as possible. This regularizes the generation of cost volumes, and leads to a more accurate prediction of 2DGS positions.

3.5 Uncertainty-Guided NLL Normal Loss

Due to the lack of regularization for normals, the predicted orientations of 2DGS still suffer from degradation, preventing the 2DGS from aligning precisely with the actual surfaces. To address this issue, we use the uncertainty-guided normal negative log-likelihood (NLL) loss proposed by (Bae, Budvytis, and Cipolla 2021) to supervise predicted normal maps at each scale, which is defined as:

$$\begin{aligned}\mathcal{L}_{\mathrm{AngMF}}(n_i, \hat{n}_i, \kappa_i) = -\log(\kappa_i^2 + 1) + \log(1 + \\ \exp(-\kappa_i \pi)) + \kappa_i \cos^{-1} n_i^T \hat{n}_i\end{aligned} \tag{8}$$

where n_i, κ_i, $\hat{n}_i$ stands for the predicted normal vector, kappa and pseudo ground-truth normal vector of pixel i. Following (Bae, Budvytis, and Cipolla 2021), we sample pixels based on their κ_i, forming the sampled sets P_{sample}, and only apply NLL loss on P_{sample}. We use an off-the-shelf monocular normal estimation model Omnidata (Eftekhar et al. 2021) to provide pseudo ground-truth normal maps as supervision. Finally, the normal loss $\mathcal{L}_{\mathrm{normal}}$ is computed as the average of NLL losses in sampled points. Details can be found in supplementary materials.

3.6 Training and Inference

Finally, our overall training loss is defined as:

$$\mathcal{L} = w_1 \mathcal{L}_{\mathrm{pho}} + w_2 \mathcal{L}_{\mathrm{WCD}} + w_3 \mathcal{L}_{\mathrm{normal}} \tag{9}$$

$\mathcal{L}_{\mathrm{pho}}$ consists of MSE loss and LPIPS loss calculated from ground truth RGB image I and rendered image $\hat{I}$:

$$\mathcal{L}_{\mathrm{pho}} = w_{11}\,\mathrm{MSE}(I, \hat{I}) + w_{12}\,\mathrm{LPIPS}(I, \hat{I}) \tag{10}$$

During inference, 2DGS is first reconstructed through our network. Then we follow 2DGS original paper (Huang et al. 2024a) to extract the scene mesh.

4 Experiments

4.1 Settings

Datasets. We train and evaluate our model on Re10K (Zhou et al. 2018) and Scannet (Dai et al. 2017) datasets. Re10K is a large-scale home walkthrough multi-view dataset obtained from YouTube, containing 67,477 training scenes and 7,289 testing scenes. Scannet is a multi-view real-world indoor dataset, and following (Zhang et al. 2022), we use 100 scenes for training and 8 scenes for testing. To evaluate the model generalizability, we also perform cross-dataset evaluations from Re10K to Scannet and Replica (Straub et al. 2019), a multi-view synthetic indoor dataset. Following (Zhi et al. 2021), we use 8 scenes for testing in Replica dataset.

Implementation Details. We employ different training strategies for Re10K and Scannet. For experiments on Re10K, we crop the input images to 256×256 and train for 200,000 steps with a batch size of 12. For experiments on Scannet, we crop the input images to 512×384 and train for 75,000 steps with a batch size of 4.

4.2 Main Results

Mesh Reconstruction. To enable comparisons with the state-of-the-arts, we transfer four methods from generalizable novel view synthesis to mesh reconstruction as our comparison. For fair comparison, we also apply TSDF fusion for the compared methods. Table 1 presents the experimental results on the two datasets. The experiments demonstrate that our method significantly outperforms these methods in all metrics, indicating that our approach has strong capabilities in surface reconstruction with sparse-view input.

Figures 4 and 5 show the visualizations on Re10K and Scannet. For the compared methods, their extracted meshes suffer from holes and uneven surfaces. In contrast, MeshSplat can accurately reconstruct the meshes of complex scenes with smoother surfaces, demonstrating that our model can capture scene structure well even with sparse-view inputs. More comparisons can be found in Appendix.

Cross-Dataset Generalization. To validate the generalization ability, We also conduct zero-shot transfer experiments on the Scannet and Replica datasets respectively, using the model trained on the Re10K dataset. As shown in Table 2, our method still provides significant improvements compared to the baselines, demonstrating the satisfactory generalizability of MeshSplat across diverse scenarios.

Depth and Normal Maps Prediction. To further show that MeshSplat can capture scene geometry accurately, we show the experiments on depth and normal maps prediction in Table 3. MeshSplat surpasses MVSplat by a large margin in all metrics, demonstrating that MeshSplat can better recover the scene geometry. We also show the visualizations of depth and normal maps in Figure 6, along with matching confidence maps in the WCD loss and kappa maps used in the normal loss. MeshSplat can accurately estimate depth and normal maps of the input views. The confidence maps clearly indicate regions of low confidence, such as textureless areas and non-overlapped regions. For kappa maps, the higher uncertainty areas are typically object borders.

	Re10K*				Scannet			
	CD↓	Precision↑	Recall↑	F1↑	CD↓	Precision↑	Recall↑	F1↑
SparseNeuS (Long et al. 2022)	6.0473	0.0012	0.0097	0.0020	7.1860	0.0056	0.1964	0.0107
MVSNeRF (Chen et al. 2021)	0.6139	0.1390	0.1548	0.1407	0.5761	0.1320	0.2053	0.1514
pixelSplat (Charatan et al. 2024)	1.4423	0.1067	0.0903	0.0944	0.3285	0.2599	0.3597	0.2948
MVSplat** (Chen et al. 2024c)	0.4038	0.3986	0.2633	0.3139	-	-	-	-
MVSplat (Chen et al. 2024c)	0.4015	0.3949	0.2607	0.3100	0.3748	0.1992	0.2282	0.2095
MeshSplat (Ours)	**0.3566**	**0.5289**	**0.2953**	**0.3758**	**0.2606**	**0.3901**	**0.3849**	**0.3824**

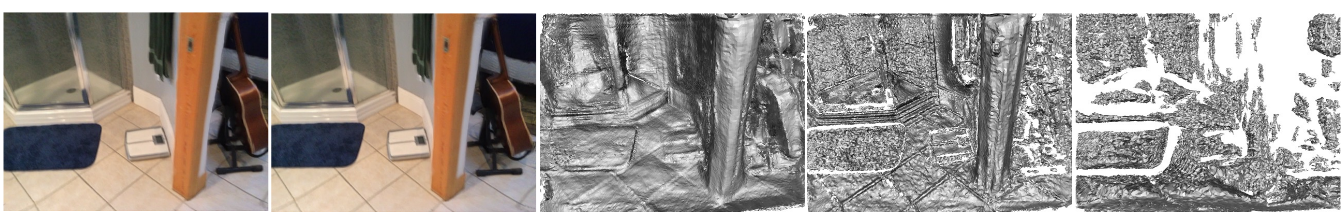
Table 1: Quantitative Comparisons. * denotes that we use the dense reconstruction results of COLMAP with dense view inputs as ground-truth point clouds of Re10K. ** denotes for the 300k-training-step version according to its original setting.

	Re10K → Scannet				Re10K → Replica			
	CD↓	Precision↑	Recall↑	F1↑	CD↓	Precision↑	Recall↑	F1↑
SparseNeuS (Long et al. 2022)	8.3055	0.0003	0.0166	0.0006	12.72	0.0002	0.0099	0.0003
MVSplat (Chen et al. 2024c)	0.4506	0.1333	0.1591	0.1418	1.089	0.0528	0.0695	0.0564
MeshSplat (Ours)	**0.3148**	**0.2868**	**0.3176**	**0.2956**	**0.921**	**0.0749**	**0.0984**	**0.0809**

Table 2: Quantitative Comparisons in Zero-Shot Transfer Experiments on Scannet and Replica Datasets. We use the models trained on Re10K only to perform cross-dataset generalization experiments. MeshSplat still shows promising results.

Ground Truth Ours (RGB) Ours MVSplat pixelSplat MVSNeRF

Figure 4: Quanlitative Comparisons on Re10K Dataset. While the baseline methods provide meshes with holes and uneven surfaces, MeshSplat successfully reconstruct the scene with smoother and more complete meshes.



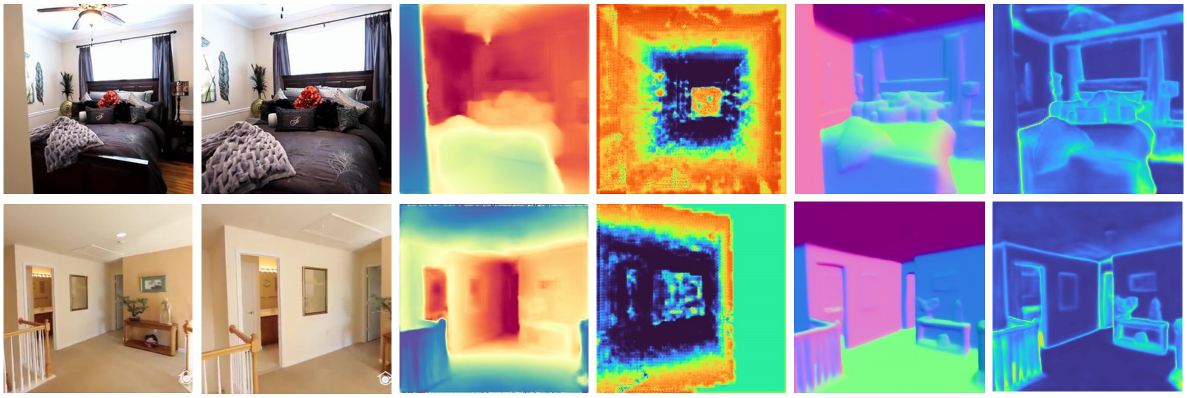
Ground Truth Ours (RGB) Ours MVSplat MVSNeRF

Figure 5: Quanlitative Comparisons on Scannet Dataset. While the baseline methods provide meshes with holes and uneven surfaces, MeshSplat successfully reconstruct the scene with smoother and more complete meshes.

4.3 Ablations

Main Components. Table 4 and Figure 7 (a) present the results of our ablation studies conducted on the Scannet dataset. The first two lines of the table indicate that 2DGS is more suitable for mesh reconstruction tasks compared to 3DGS. Moreover, by incorporating our proposed WCD loss and normal prediction network, 2DGS can align better with the actual surfaces. These enhancements significantly improve surface fidelity and reduce geometric artifacts in the final output. Overall, these results demonstrate that each proposed component plays a complementary role in boosting the reconstruction quality.

Gaussian normals. To further demonstrate the necessities of the normal prediction network, we present comparisons on normal maps rendered by Gaussian Splatting in Figure 7 (b). Following (Chen, Li, and Lee 2023), we define the normal vector of 3DGS as the direction of the axis which has the minimum value of scales. 2DGS suffer from more sig-



View View II Depth Confidence Normal Kappa

Figure 6: Visualizations of Output Depth and Normal Maps, Confidence Maps Used in WCD Loss and Kappa Maps Used in Normal Loss. The confidence maps reflect the unconfident matching areas like texture-less areas and non-overlapped areas between the two views. For kappa maps, areas with higher uncertainty typically correspond to object boundaries.

	Depth		Normal	
	AbsRel↓	AbsDiff↓	Mean↓	<30°↑
MVSplat	0.1692	0.3197	57.16	0.1357
MeshSplat	**0.0910**	**0.1680**	**33.84**	**0.6026**

Table 3: Results on Depth and Normal Prediction. MeshSplat can predict depth and normal maps correctly.

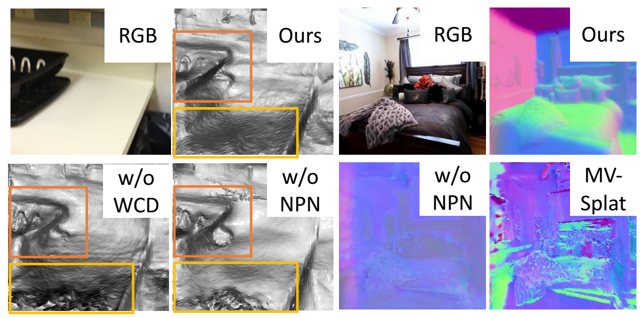
#	2DGS	WCD Loss	NPN	CD↓
1				0.3748
2	✓			0.2948
3	✓	✓		0.2769
4	✓		✓	0.2642
5	✓	✓	✓	**0.2606**

Table 4: Ablation Studies on Scannet Dataset. NPN stands for the normal prediction network and NLL loss.

nificant degradation in normals compared to 3DGS when no regularization is applied. After incorporating the normal prediction network and its corresponding regularization loss, the predicted orientations of 2DGS become more accurate.

5 Limitations

Although our model achieves promising results in generalizable sparse-view surface reconstruction, it still has some limitations. MeshSplat sometimes may predict discontinuous depth maps in weakly textured areas, as shown in the Appendix, although the rendered RGB image is reliable. This is probably due to the ambiguity of feature matching in these areas. Additionally, MeshSplat cannot reconstruct surfaces in regions not visible from the input views. Incorporating generative approaches might be a good choice.



(a) (b)

Figure 7: Qualitative Ablation Studies. (a) Ablation studies on proposed modules. (b) Comparisons of rendered normal maps by Gaussian Splatting.

6 Conclusion

We present MeshSplat, a generalizable surface reconstruction framework designed for sparse-view inputs. We use 2DGS as scene representation and construct an end-to-end surface reconstruction network. To better meet the needs for more accurate prediction of 2DGS, we propose the WCD loss to regularize coarse point maps, and also use a normal prediction network to predict the orientations of 2DGS. Experiments demonstrate that MeshSplat outperforms existing methods in generalizable sparse-view mesh extraction.

Acknowledgements

This work was supported by Hainan Province Science and Technology Special Fund (Grant No. ATIC-2023010004) Youth Innovation Promotion Association of CAS.

References

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

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5855–5864.

Charatan, D.; Li, S. L.; Tagliasacchi, A.; and Sitzmann, V. 2024. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19457–19467.

Chen, A.; Xu, Z.; Zhao, F.; Zhang, X.; Xiang, F.; Yu, J.; and Su, H. 2021. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, 14124–14133.

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

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

Chen, X.; Sun, J.; Xie, Y.; Bao, H.; and Zhou, X. 2024b. Neuralrecon: Real-time coherent 3d scene reconstruction from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 7542–7555.

Chen, Y.; Xu, H.; Zheng, C.; Zhuang, B.; Pollefeys, M.; Geiger, A.; Cham, T.-J.; and Cai, J. 2024c. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, 370–386. Springer.

Cheng, Z.; Deng, J.; Li, X.; Yin, B.; and Zhang, T. 2025. Bridge 2D-3D: Uncertainty-aware Hierarchical Registration Network with Domain Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 2491–2499.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

Duisterhof, B. P.; Zust, L.; Weinzaepfel, P.; Leroy, V.; Cabon, Y.; and Revaud, J. 2025. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *2025 International Conference on 3D Vision (3DV)*, 1–10. IEEE.

Eftekhar, A.; Sax, A.; Malik, J.; and 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*, 10786–10796.

Furukawa, Y.; and Ponce, J. 2009. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8): 1362–1376.

Genova, K.; Cole, F.; Sud, A.; Sarna, A.; and Funkhouser, T. 2020. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4857–4866.

Guédon, A.; and 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*, 5354–5363.

Hong, Y.; Zhang, K.; Gu, J.; Bi, S.; Zhou, Y.; Liu, D.; Liu, F.; Sunkavalli, K.; Bui, T.; and Tan, H. 2023. LRM: Large Reconstruction Model for Single Image to 3D. In *The Twelfth International Conference on Learning Representations*.

Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; and Gao, S. 2024a. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, 1–11.

Huang, Y.-H.; Sun, Y.-T.; Yang, Z.; Lyu, X.; Cao, Y.-P.; and Qi, X. 2024b. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4220–4230.

Jiang, C.; Sud, A.; Makadia, A.; Huang, J.; Nießner, M.; Funkhouser, T.; et al. 2020. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6001–6010.

Jin, H.; Jiang, H.; Tan, H.; Zhang, K.; Bi, S.; Zhang, T.; Luan, F.; Snavely, N.; and Xu, Z. 2024. LVSM: A Large View Synthesis Model with Minimal 3D Inductive Bias. In *The Thirteenth International Conference on Learning Representations*.

Kazhdan, M.; and Hoppe, H. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3): 1–13.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4): 1–14.

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

Long, X.; Lin, C.; Wang, P.; Komura, T.; and Wang, W. 2022. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*, 210–227. Springer.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20654–20664.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15.

Peng, R.; Gu, X.; Tang, L.; Shen, S.; Yu, F.; and Wang, R. 2023. Gens: Generalizable neural surface reconstruction from multi-view images. *Advances in Neural Information Processing Systems*, 36: 56932–56945.

Reiser, C.; Garbin, S.; Srinivasan, P.; Verbin, D.; Szeliski, R.; Mildenhall, B.; Barron, J.; Hedman, P.; and Geiger, A. 2024. Binary opacity grids: Capturing fine geometric detail for mesh-based view synthesis. *ACM Transactions on Graphics (TOG)*, 43(4): 1–14.

Ren, K.; Jiang, L.; Lu, T.; Yu, M.; Xu, L.; Ni, Z.; and Dai, B. 2025. Octree-GS: Towards Consistent Real-time Rendering with LOD-Structured 3D Gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shi, D.; Wang, W.; Chen, D. Y.; Zhang, Z.; Bian, J.-W.; Zhuang, B.; and Shen, C. 2025. Revisiting Depth Representations for Feed-Forward 3D Gaussian Splatting. *arXiv preprint arXiv:2506.05327*.

Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. 2019. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*.

Tang, J.; Ren, J.; Zhou, H.; Liu, Z.; and Zeng, G. 2023. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. In *The Twelfth International Conference on Learning Representations*.

Wang, J.; Chen, M.; Karaev, N.; Vedaldi, A.; Rupprecht, C.; and Novotny, D. 2025a. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 5294–5306.

Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; and Wang, W. 2021a. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *Advances in Neural Information Processing Systems: 35th conference on neural information processing systems (Neurips 2021)*. Curran Associates.

Wang, Q.; Wang, Z.; Genova, K.; Srinivasan, P. P.; Zhou, H.; Barron, J. T.; Martin-Brualla, R.; Snavely, N.; and Funkhouser, T. 2021b. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4690–4699.

Wang, S.; Leroy, V.; Cabon, Y.; Chidlovskii, B.; and Revaud, J. 2024a. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20697–20709.

Wang, W.; Chen, D. Y.; Zhang, Z.; Shi, D.; Liu, A.; and Zhuang, B. 2025b. ZPressor: Bottleneck-Aware Compression for Scalable Feed-Forward 3DGS. *arXiv preprint arXiv:2505.23734*.

Wang, Y.; Huang, T.; Chen, H.; and Lee, G. H. 2024b. Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. *Advances in Neural Information Processing Systems*, 37: 107326–107349.

Wewer, C.; Raj, K.; Ilg, E.; Schiele, B.; and Lenssen, J. E. 2024. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *European conference on computer vision*, 456–473. Springer.

Xu, H.; Chen, A.; Chen, Y.; Sakaridis, C.; Zhang, Y.; Pollefeys, M.; Geiger, A.; and Yu, F. 2024. MuRF: multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20041–20050.

Xu, L.; Guan, T.; Wang, Y.; Liu, W.; Zeng, Z.; Wang, J.; and Yang, W. 2023. C2f2neus: Cascade cost frustum fusion for high fidelity and generalizable neural surface reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18291–18301.

Yariv, L.; Gu, J.; Kasten, Y.; and Lipman, Y. 2021. Volume rendering of neural implicit surfaces. *Advances in neural information processing systems*, 34: 4805–4815.

Yu, M.; Lu, T.; Xu, L.; Jiang, L.; Xiangli, Y.; and Dai, B. 2024. Gsdf: 3dgs meets sdf for improved neural rendering and reconstruction. *Advances in Neural Information Processing Systems*, 37: 129507–129530.

Yu, Z.; Sattler, T.; and Geiger, A. 2024. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics (ToG)*, 43(6): 1–13.

Zhang, X.; Bi, S.; Sunkavalli, K.; Su, H.; and Xu, Z. 2022. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5449–5458.

Zhi, S.; Laidlow, T.; Leutenegger, S.; and Davison, A. J. 2021. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15838–15847.

Zhou, T.; Tucker, R.; Flynn, J.; Fyffe, G.; and Snavely, N. 2018. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4): 1–12.

Zhu, R.; Liang, Y.; Chang, H.; Deng, J.; Lu, J.; Yang, W.; Zhang, T.; and Zhang, Y. 2024. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37: 101790–101817.

Zhu, R.; Yu, M.; Xu, L.; Jiang, L.; Li, Y.; Zhang, T.; Pang, J.; and Dai, B. 2025. Objectgs: Object-aware scene reconstruction and scene understanding via gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8350–8360.