

---

# GenS: Generalizable Neural Surface Reconstruction from Multi-View Images

---

Rui Peng<sup>1,2</sup> Xiaodong Gu<sup>3</sup> Luyang Tang<sup>1</sup> Shihe Shen<sup>1</sup> Fanqi Yu<sup>1</sup> Ronggang Wang<sup>✉1,2</sup>

<sup>1</sup>School of Electronic and Computer Engineering, Peking University

<sup>2</sup>Peng Cheng Laboratory <sup>3</sup>Alibaba Group

ruipeng@stu.pku.edu.cn rgwang@pkusz.edu.cn

## Abstract

Combining the signed distance function (SDF) and differentiable volume rendering has emerged as a powerful paradigm for surface reconstruction from multi-view images without 3D supervision. However, current methods are impeded by requiring long-time per-scene optimizations and cannot generalize to new scenes. In this paper, we present GenS, an end-to-end generalizable neural surface reconstruction model. Unlike coordinate-based methods that train a separate network for each scene, we construct a generalized multi-scale volume to directly encode all scenes. Compared with existing solutions, our representation is more powerful, which can recover high-frequency details while maintaining global smoothness. Meanwhile, we introduce a multi-scale feature-metric consistency to impose the multi-view consistency in a more discriminative multi-scale feature space, which is robust to the failures of the photometric consistency. And the learnable feature can be self-enhanced to continuously improve the matching accuracy and mitigate aggregation ambiguity. Furthermore, we design a view contrast loss to force the model to be robust to those regions covered by few viewpoints through distilling the geometric prior from dense input to sparse input. Extensive experiments on popular benchmarks show that our model can generalize well to new scenes and outperform existing state-of-the-art methods even those employing ground-truth depth supervision. Code will be available at <https://github.com/prstrive/GenS>.

## 1 Introduction

Surface reconstruction from multi-view images is a cornerstone task in computer vision with many applications in virtual reality, autonomous driving, robotics, etc. Typical solutions [15, 16, 7, 43, 58, 56, 10, 36] in the past were mostly based on a multi-step pipeline, which includes depth estimation, depth fusion and meshing. Although they have demonstrated their excellent performance, the procedure is cumbersome and inevitably introduces cumulative errors. While several early works [31, 61] used differentiable surface rendering to directly reconstruct surfaces, recent works [34, 49, 60], inspired by the huge success of neural radiance field (NeRF) [27] in synthesizing novel views, follow the volume rendering [24] to represent the 3D geometry with an occupancy field [25] or signed distance function (SDF) [35] and can achieve more impressive results.

The key idea of these approaches is to train a compact multi-layer perceptrons (MLPs) to predict the implicit representation (e.g., SDF) of each sampled point on camera rays. The density of volume rendering is then regarded as a function of this implicit representation, and alpha-composition of samples is performed to produce the corresponding pixel color and geometry information. However, existing methods are hampered by requiring a lengthy per-scene optimization procedure and cannot generalize to new scenes, which makes them infeasible for many application scenarios. A recent method [23] attempts to address these issues through conditioning the SDF-induced model with

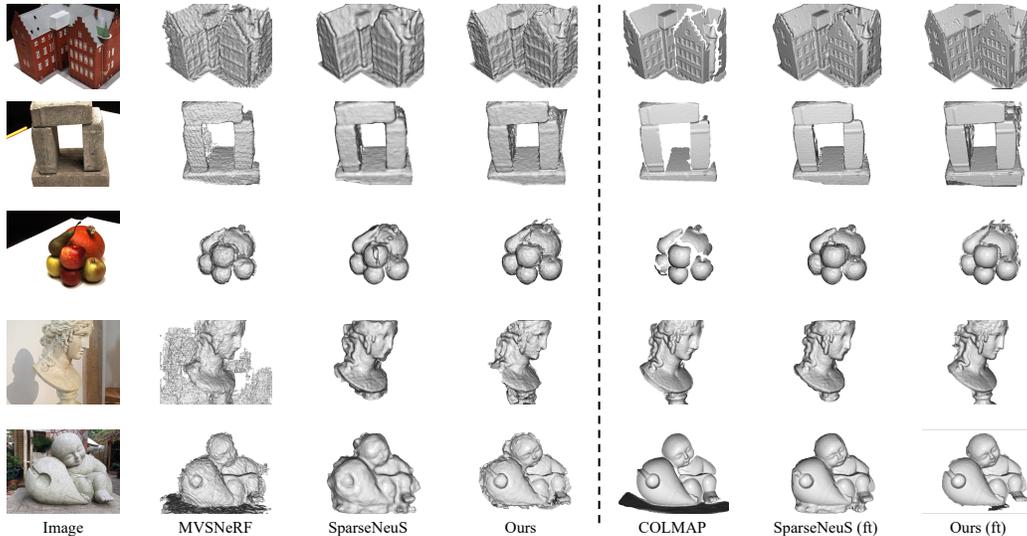


Figure 1: **Qualitative comparisons on DTU and BlendedMVS datasets with sparse inputs.**

features extracted from sparse nearby views. Nevertheless, its accuracy is limited due to the smooth reconstruction, and the multi-stage process it relies on is prone to introducing cumulative errors. In this paper, we seek to establish an end-to-end generalizable model which can efficiently infer finer 3D structure. Compared with existing methods [49, 60], this generalization system faces more challenging problems. First, it’s non-trivial to efficiently represent the scene. Previous methods [29, 23, 3, 50] either build a global volume or employ feature projections, but they have proven to be either lacking in detail or unsuitable for view independent surface reconstruction. Second, relying only on the rendering loss is difficult to reconstruct compact geometry, since the multi-view consistency is ignored. And we found that the ordinary photometric consistency also cannot effectively solve this problem for our generalizable model because of the existence of ambiguous areas such as low-texture and reflection. Last but not least, since generalizable models heavily rely on aggregation quality, how to infer smooth geometry when the input is sparse is a thorny issue.

To this end, we introduce GenS to tackle these challenges. The main ideas behind are as follows: 1) We first construct a generalized multi-scale volume to represent the scene, which preserves global smoothness through the low-resolution volumes and recovers geometric details from high-resolution volumes. Meanwhile, low-dimensional features make our model more lightweight compared to a single large-width volume. 2) We introduce the multi-scale feature-metric consistency, which enforces multi-view consistency in the multi-scale feature space, to replace the common photometric consistency. Compared with the original image space, learnable multi-scale features can provide more discriminative representation, and the feature space can be self-enhanced during the generalization training process to continuously improve the matching accuracy. 3) Inspired by the fact that the reconstruction with dense inputs is more accurate, we propose a view contrast loss to force the model to better perceive the geometry of regions visible by few viewpoints through teaching the reconstruction from sparse inputs with dense inputs.

To demonstrate the quantitative and qualitative effectiveness of GenS, we conduct extensive experiments on DTU [12] and BlendedMVS [59] datasets. Results show that our model can outperform existing state-of-the-art generalizable method [23], and even recent method [40] which adopts the ground-truth depth for supervision. Compared with the per-scene overfitting methods [49, 60, 61, 34, 23], we can also achieve comparable or superior results with dense inputs. Some comparisons are shown in Fig. 1. In summary, our main contributions are highlighted below: **a)** We present a powerful representation based on our generalized multi-scale volume, which can efficiently reconstruct smooth and detail surfaces from multi-view images. **b)** We introduce a more discriminative multi-scale feature-metric consistency to successfully constrain the geometry, which helps the generalization model converge to the optimum. **c)** We propose a view contrast loss to improve the geometric smoothness and accuracy when the visible viewpoint is limited. **d)** Our model can be trained end-to-end and achieve state-of-the-art reconstructions in both generic setting and per-scene optimization setting.

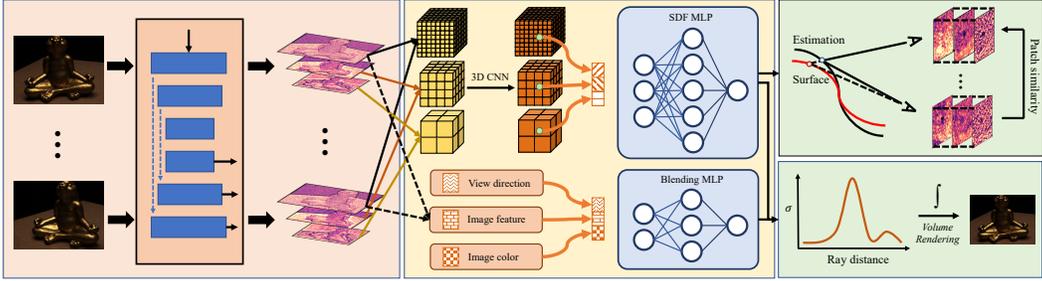


Figure 2: **Illustration of GenS.** We first extract multi-scale features through a FPN network. The generalized multi-scale volume is then reconstructed with the corresponding scale feature. We employ the same blending strategy as [50] to estimate the appearance of each point on a ray, and adopt the volume rendering to recover the color of a pixel. We design the multi-scale feature-metric consistency to constrain the geometry as shown in the top right. For convenience, we omit some losses that will be detailed later.

## 2 Related work

**Classical multi-view reconstruction.** Reconstructing 3D geometry from multi-view images is a longstanding problem in the field of 3D vision. Classical algorithms mainly adopt depth-based or voxel-based methodology to solve this problem. Multi-view stereo (MVS) is a typical class of depth-based methods, which takes stereo correspondence from multiple images as the main cue to reconstruct depth maps. While previous traditional MVS methods [1, 43, 7, 6, 56, 44] relied on the hand-crafted similarity metrics, many recent learning-based methods [58, 10, 48, 36] apply deep learning to achieve more discriminative matching. These methods go through complicated procedures to retrieve surface, including depth filtering, fusion and meshing [15, 2], and are prone to cumulative errors. On the other hand, voxel-based methods [45, 18, 11, 32] directly model objects in a volume, but they are restricted to memory, which is the common drawback of the volumetric representation, and cannot achieve high accuracy.

**Neural surface.** Due to the notable advantages of being able to achieve high spatial resolution, neural implicit functions have recently gained a lot of attention and have emerged as an effective representation of 3D geometry [47, 25, 35, 8, 26, 30, 37, 42] and appearance [20, 27, 21, 33, 46, 62, 38, 28]. Furthermore, some works [27, 22, 31] have proposed to train models without 3D supervision via differentiable rendering, e.g., surface rendering and volume rendering. Methods adopt surface rendering [31, 61, 65] only consider a single surface intersection point for each ray and fail to reconstruct complex objects, and they are restricted by the need of accurate object masks and careful weight initialization. On the contrary, recent methods use volume rendering [34, 49, 60, 64] to take multiple points along the ray into consideration and achieve more impressive results. However, either type of method requires an expensive per-scene optimization and cannot generalize to new scenes.

**Generalizable neural surface.** In the field of novel view synthesis, some methods [3, 50, 63, 13] have successfully introduced the generalization into rendering methods. These methods suffer from the same problem as NeRF: the geometry is ambiguous. Few works have focused on the generalization of neural surface reconstruction. A recent study, SparseNeuS [23], is the first attempt to achieve this by reconstructing the surface from nearby viewpoints in a multi-stage manner. Nevertheless, its reconstruction lacks details, and same to the classical 3D reconstruction, the multi-stage pipeline may accumulate errors at each stage. On the contrary, our designed model can be trained end-to-end and reconstruct smoother and more refined geometries.

## 3 Method

Given  $N$  posed images of an object taken from different viewpoints, our goal is to reconstruct the surface as an implicit function without expensive per-scene optimization or only by fast fine-tuning. Our overall framework is depicted in Fig 2. We first introduce how to infer the geometry and appearance from the generalized multi-scale volume in Sec. 3.1, then elaborate on the necessity and implementation of the multi-scale feature-metric consistency in Sec. 3.2, and finally detail the realization of view contrast loss in Sec. 3.3 and the overall pipeline in Sec. 3.4.

### 3.1 Geometry and color reasoning from the generalized multi-scale volume

Compared with existing solution [23], which relies on a single volume and multi-stage strategy, we have three main advantages. First of all, our generalized multi-scale volume is a more powerful representation, which implicitly decouples geometry into base structures in low-resolution volumes and high-frequency details in high-resolution volumes. Second, with the low-dimensional features, we can construct multi-scale volumes with higher resolution and less memory consumption than a single large-width volume. Besides, our model can be trained end-to-end, avoiding cumulative errors.

**Generalized multi-scale volume construction.** Suppose there are  $N$  images  $\{I_i\}_{i=0}^{N-1}$  of an object, we first apply the FPN network [19] to extract multi-scale feature maps  $\{F_i^j\}_{i,j=0,0}^{N-1,L-1}$  for all images with shared weights, and different volumes are then constructed from features at corresponding scales. In this paper, we define a bounding box of interest in the reference frustum like [23] and in the world coordinate system like [49, 66] when dense inputs are available. We adopt a combination of  $L$  volumes  $\{V_j\}_{j=0}^{L-1}$ , which cover the same region but with different resolutions  $Ch \times \frac{D}{2^j} \times \frac{H}{2^j} \times \frac{W}{2^j}$ .

Here, we discuss at the first scale and omit the scale subscript  $j$  for convenience. Given camera intrinsics  $\{K_i\}_{i=0}^{N-1}$  and extrinsics  $\{[R, t]_i\}_{i=0}^{N-1}$ , we first project the voxel  $v = (x, y, z)$  onto viewpoint  $i$ 's pixel position:

$$q_i(v) = \pi(K_i R_i^T (v - t_i)), \quad (1)$$

where  $\pi((x, y, z)^T) = (\frac{x}{z}, \frac{y}{z})^T$  is an operator to convert homogeneous coordinates to cartesian coordinates. Then we can get the corresponding feature of each voxel on  $i_{th}$  viewpoint through bilinear interpolation  $F_i(v) = F_i < q_i(v) >$ . To fuse features from all viewpoints  $\{F_i(v)\}_{i=0}^{N-1}$ , we adopt the same aggregation strategies to generate cost volume as in [50] that concatenates mean and variance to simultaneously capture statistical and semantic information:  $B(v) = [Mean(v), Var(v)]$ .

Simply repeating the above process on features and volumes of all  $L$  scales, we can get the multi-scale cost volumes  $\{B_j\}_{j=0}^{L-1}$ . Next, we further design an efficient multi-scale 3D network  $\psi$  to refine these cost volumes in one forward, starting from the finest volume and injecting the others into different stages of the model to save memory. The output of the 3D network  $\{V_j\}_{j=0}^{L-1} = \psi(\{B_j\}_{j=0}^{L-1})$  is the multi-scale volume that we need to infer the geometry.

**Geometry reasoning.** For an arbitrary 3D point  $p = (x, y, z)$ , we first get the interpolation of volumes at all scales  $\{V_j(p)\}_{j=0}^{L-1}$  through trilinear sampling, and then concatenate them as the final feature  $\mathcal{F}(p) \in \mathbb{R}^{Ch_1}$ , where  $Ch_1 = L \times Ch$ . Combining the feature and the point position, an MLP network is applied to predict the corresponding SDF value:  $sdf_\theta : \mathbb{R}^3 \times \mathbb{R}^{Ch_1} \rightarrow \mathbb{R}$ . And the surface is represented by the zero-level set of the SDF value:

$$S = \{p \in \mathbb{R}^3 | sdf_\theta(p, \mathcal{F}(p)) = 0\}. \quad (2)$$

**Color prediction.** We refer to the first viewpoint  $I_0$  as the reference image. To predict the color of each point on a ray, we employ the blending strategy similar to [50]. We first project the 3D point  $p$  to source views' pixel position according to Eq. 1, and interpolate the corresponding colors  $\{I_i(p)\}_{i=1}^{N-1}$  and features  $\{F_i(p)\}_{i=1}^{N-1}$ . Here, we only use the highest resolution features to predict the color. Next, an MLP network take the concatenation of features and viewing direction differences  $\Delta d = d - d_i$  as input, to predict the softmax-activated blending weights  $\{w_i(p)\}_{i=1}^{N-1}$  of each source view, and the final color is blended as the weighted sum of source colors:

$$c(p) = \sum_{i=1}^{N-1} I_i(p) w_i(p). \quad (3)$$

**SDF-based volume rendering.** Given the density  $\{\sigma_i\}_{i=1}^M$  and color  $\{c_i\}_{i=1}^M$  of  $M$  samples along the ray  $p(t) = o + td$  emitting from camera center  $o$  to pixel  $q$  in view direction  $d$ , NeRF [27] approximates the color using numerical quadrature:

$$\hat{C} = \sum_{i=1}^M T_i \alpha_i c_i, \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (4)$$

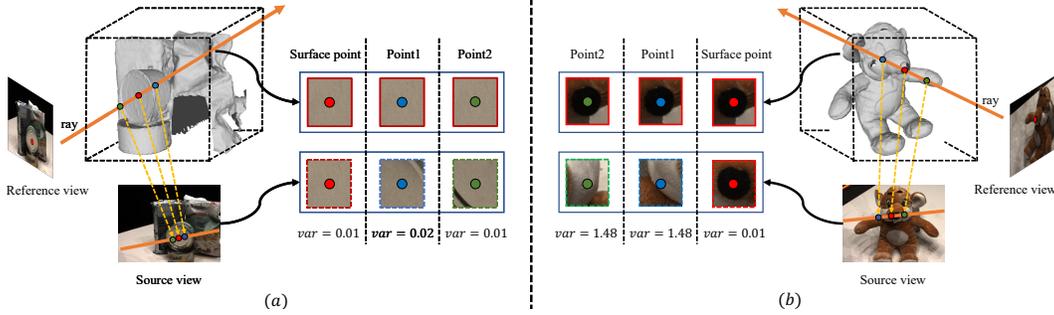


Figure 3: **Multi-view aggregation ambiguity.** Here, we take two viewpoints as an example. (a) For those low-texture regions, sampling points near the surface may get the same aggregation and lack discriminability. (b) The aggregation of points away from the surface are random and hard to infer the accurate geometry, e.g., two sampling points may get the same aggregation even with different SDF value.

where  $T_i$  is the accumulated transmittance, and  $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$  in original volume rendering. To better approximation the geometry of the scene, NeuS [49] proposed an unbiased and occlusion-aware weighting method to incorporate signed distance, and the  $\alpha_i$  is formulated as:

$$\alpha_i = \max\left(\frac{\Phi_s(\text{sdf}(p(t_i))) - \Phi_s(\text{sdf}(p(t_{i+1})))}{\Phi_s(\text{sdf}(p(t_i)))}, 0\right). \quad (5)$$

Here,  $\Phi_s(x) = (1 + e^{-sx})^{-1}$  is the sigmoid function and  $s$  is an anneal factor. Readers can refer to [49] for more details.

### 3.2 Multi-scale feature-metric consistency

Rendering loss tends to trap the model into sub-optimization since it only considers a single point and ignores the consistency among multiple viewpoints. To mitigate this problem, a straightforward practice is to project the image patches of multiple views to the estimated surface location based on the local planar assumption and rely on the photometric consistency to enforce the multi-view consistency. However, we found this solution works well for per-scene overfitting training [4, 5] but brings limited benefits to generalization training.

We analyze that the main reason may be the failure of photometric consistency, which becomes more challenging for generalization training. As proven in recent self-supervised multi-view stereo methods [55, 54, 57, 39], the assumption of photometric consistency isn't always effective, and the predicted geometry still has significant holes even in combination with the robust patch similarity like SSIM [52]. As the coordinate-based methods train models separately for each scene to directly overfit the scene, they have greater potential to converge to the optimum. However, our generalization model encodes all scenes with one model, and it requires image features to infer geometry, which makes the model rely heavily on the discriminability of features, e.g., regions like low-texture and reflection become more critical for degrading results. As shown in Fig 3 (a), those regions violating photometric consistency not only reduce the accuracy of multi-view matching, but also decrease the discriminability of generalization model's input (we call this aggregation ambiguity), while the input of overfitting methods are distinct (3D coordinate).

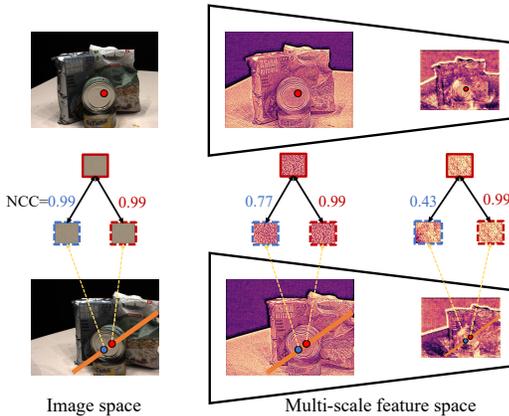


Figure 4: **Multi-scale feature space.** The feature space is more discriminative than ordinary image space, and is more potential to find the corresponding point during matching.

To overcome these challenges, we propose the multi-scale feature-metric consistency to measure the consistency between views in a multi-scale feature space, as shown in Fig. 4. There are three main advantages of doing this way. First of all, the learnable feature is proven to be more discriminative than the original image [14], especially on those ambiguous regions like low-texture and reflection. Second, due to the larger receptive field, multi-scale information is conducive to improving the matching accuracy, and allows the model to be assisted by global information while recovering details. More importantly, the feature discriminability can be continuously self-enhanced in the process of generalization training. The multi-scale feature space can train a powerful model through more accurate matching, and the more powerful model can in turn lead to a more discriminative feature space. And the enhanced feature can further mitigate the aforementioned aggregation ambiguity. These advantages have been proven in Tab. 3.

To generate the geometry, we adopt the same approximate method as [5] to directly locate the zero-level set of the SDF. As shown in Fig. 5, We first find the interval where a ray intersects the surface by checking whether the signs of the SDFs of two adjacent sampling points are different. To handle occlusion, we only extract the surface within the first interval. Suppose the two samples of the interval are  $p_1$  and  $p_2$ , and their distances to the camera center are  $t_1$  and  $t_2$  respectively, our goal is to compute the position of  $p_s$ . Here, we rely on an assumption that two adjacent samples are close enough that the near surface can be regarded as a local plane. In this way, we can get two similar triangles:

$$\triangle p_2 p_s s_2 \sim \triangle p_1 p_s s_1, \quad (6)$$

Therefore, we can approximate the distance from the surface to the camera center  $t_s$  as:

$$\frac{-sdf(p_2)}{sdf(p_1)} = \frac{t_2 - t_s}{t_s - t_1} \Rightarrow t_s = \frac{sdf(p_1)t_2 - sdf(p_2)t_1}{t_1 - t_2}. \quad (7)$$

We thus can get the coordinate of the surface point  $p_s = o + t_s d$ .

Through the automatic differentiation of the SDF network at  $p_s$ , we can get the corresponding normal  $n_s$ . Based on the assumption that the local surface centered at  $p_s$  is a plane of normal  $n_s$ , we can find the corresponding pixel position  $q_i$  in  $i$ th source view that correspond to the pixel  $q_0$  in reference view:

$$q_i = H_i q_0, \quad H_i = K_i (R_i R_0^T + \frac{R_i (R_i^T t_i - R_0^T t_0) n_s^T}{n_s^T p_s}) K_0^{-1}. \quad (8)$$

For a pixel patch  $\mathbf{q}_0$  in the reference view, we can find the corresponding source patch through passing all pixels to Eq. 8 like  $\mathbf{q}_i = H_i \mathbf{q}_0$ . Regardless of occlusion, if the estimated surface  $p_s$  is accurate, then these corresponding patches should also be consistent. In this paper, we measure patch consistency in a multi-scale feature space. We only apply features at the top 3 scales, since features at lower scales lose a lot of structural information. Therefore, for a pixel patch at a certain view, we can get the multi-scale patches  $\{F_j < \mathbf{q} >\}_{j=0}^2$  through bilinear interpolation, and we upsample and concatenate them together as input  $F'$ , whose channel is  $Ch_2 = 3 \times Ch$ , for patch similarity measure. Here, we employ the normalization cross correlation (NCC) to compute the feature-space consistency:

$$NCC_i = \frac{1}{Ch_2} \sum_{l=0}^{Ch_2-1} \frac{Cov(F'_{0l}, F'_{il})}{\sqrt{Var(F'_{0l})Var(F'_{il})}}, \quad (9)$$

where  $Cov$  denotes covariance and  $Var$  refers to variance. Following the common solution in multi-view stereo field [7], we compute the final multi-scale feature-space consistency loss as the average of the best  $K$  NCCs:

$$L_{mfc} = \frac{1}{K} \sum_{k=0}^{K-1} (1 - NCC_k). \quad (10)$$

### 3.3 View contrast loss

For a 3D structure captured by multiple viewpoints, there is a fact that some regions are covered by enough viewpoints, while some regions are only visible to a few viewpoints. Compared with the

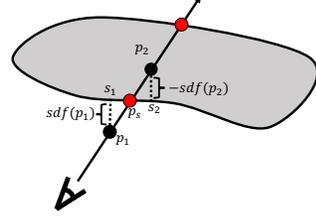


Figure 5: Locating the surface of a ray.

former, the aggregated features of the latter are more likely to be polluted by irrelevant rays, making them less predictable. To solve this problem, we design a view contrast loss to improve the accuracy of the reconstruction when visible views are limited, which enforces the geometric estimation to be the same under different inputs of the same scene.

We empirically let results from dense inputs to supervise results of sparse inputs. Specially, taking a set of multi-view images as input, we first reconstruct a multi-scale volume as a teacher, which is used to infer the finer SDF value  $s$  for a set of 3D points  $P$ . Then we build a student multi-scale volume from sparse input views and estimate the corresponding SDF value  $s'$ . Meanwhile, as shown in Fig. 3 (b), we found that only the sampling points falling on the surface have positive epipolar correspondences, and their aggregated features are more meaningful, while other samples are more random, and may obtain the same aggregation even if their SDF values are different. As shown in Fig. 6, we thus only calculate the consistency loss for near-surface points, whose finer SDF values are more accurate:

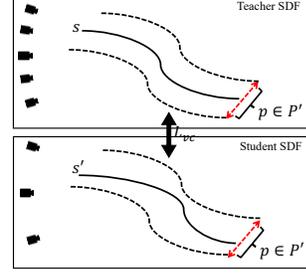


Figure 6: **Visualization of view contrast loss.**

$$L_{vc} = \frac{1}{|P'|} \sum_{p \in P'} |s(p) - s'(p)|, \quad (11)$$

where  $P'$  is a set of points close to the surface inferred from the fine SDF according to Eq. 7.

### 3.4 Overall pipeline

This section will introduce some implementation details and crucial components of our model including generalization training and fine-tuning.

**Loss function.** The overall loss function is defined as:

$$L = L_{color} + \alpha L_{mfc} + \beta L_{vc} + L_{reg}. \quad (12)$$

For a batch of sampled pixel set  $Q$ , the color loss is computed as the L1 distance between the rendered color and the ground-truth:

$$L_{color} = \frac{1}{|Q|} \sum_{q \in Q} |C(q) - \hat{C}(q)|. \quad (13)$$

To make the geometry more compact and accurate, we apply the regularization loss which is composed of four terms:

$$L_{reg} = \gamma L_{ek} + \eta L_{smooth} + \lambda L_{tv} + \delta L_{sparse}. \quad (14)$$

Eikonal loss [9] is employed to regularize SDF values of all sampled points  $P$ :

$$L_{ek} = \frac{1}{|P|} \sum_{p \in P} (\|\nabla sdf(p)\|_2 - 1)^2. \quad (15)$$

To maintain the smooth of the surface, we introduce a regularization to the gradient of the normal:

$$L_{smooth} = \frac{1}{|Q|} \sum_{q \in Q} \|n_{grad}(q)\|_2, \quad (16)$$

where  $n_{grad}(q)$  is the alpha composition of normal gradient  $\nabla^2 sdf(q)$  in a ray through pixel  $q$ . Besides, we also adopt the total variation (TV) regularization [41] for our multi-scale volumes:

$$L_{tv} = \sum_{j=0}^{L-1} \sqrt{\Delta_x^2(V_j) + \Delta_y^2(V_j) + \Delta_z^2(V_j)}. \quad (17)$$

To clean the geometric estimation, we introduce a sparsity prior:

$$L_{sparse} = \frac{1}{|P|} \sum_{p \in P} \exp(-\tau |sdf(p)|). \quad (18)$$

Method	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
VolRecon* [40]	1.20	2.59	1.56	1.08	1.43	1.92	1.11	1.48	1.42	1.05	1.19	1.38	0.74	1.23	1.27	1.38
PixelNerf [63]	5.13	8.07	5.85	4.40	7.11	4.64	5.68	6.76	9.05	6.11	3.95	5.92	6.26	6.89	6.93	6.28
IBRNet [50]	2.29	3.70	2.66	1.83	3.02	2.83	1.77	2.28	2.73	1.96	1.87	2.13	1.58	2.05	2.09	2.32
MVSNerf [3]	1.96	3.27	2.54	1.93	2.57	2.71	1.82	1.72	2.29	1.75	1.72	1.47	1.29	2.09	2.26	2.09
SparseNeuS [23]	1.68	3.06	2.25	1.10	2.37	2.18	1.28	<b>1.47</b>	1.80	1.23	1.19	1.17	0.75	1.56	1.55	1.64
<b>GenS</b>	<b>1.45</b>	<b>2.77</b>	<b>1.69</b>	<b>0.97</b>	<b>1.54</b>	<b>1.90</b>	<b>1.03</b>	1.49	<b>1.36</b>	<b>0.97</b>	<b>1.07</b>	<b>0.97</b>	<b>0.62</b>	<b>1.14</b>	<b>1.16</b>	<b>1.34</b>
NeuS [49]	4.57	4.49	3.97	4.32	4.63	1.95	4.68	3.83	4.15	2.50	1.52	6.47	1.26	5.57	6.11	4.00
VolSDF [60]	4.03	4.21	6.12	0.91	8.24	1.73	2.74	1.82	5.14	3.09	2.08	4.81	0.60	3.51	2.18	3.41
IBRNet (ft)	1.67	2.97	2.26	1.56	2.52	2.30	1.50	2.05	2.02	1.73	1.66	1.63	1.17	1.84	1.61	1.90
COLMAP [44]	<b>0.90</b>	2.89	1.63	1.08	2.18	1.94	1.61	1.30	2.34	1.28	1.10	1.42	0.76	1.17	1.14	1.52
SparseNeuS (ft)	1.29	<b>2.27</b>	1.57	0.88	1.61	1.86	1.06	1.27	1.42	1.07	0.99	0.87	0.54	1.15	1.18	1.27
<b>GenS (ft)</b>	0.91	2.33	<b>1.46</b>	<b>0.75</b>	<b>1.02</b>	<b>1.58</b>	<b>0.74</b>	<b>1.16</b>	<b>1.05</b>	<b>0.77</b>	<b>0.88</b>	<b>0.56</b>	<b>0.49</b>	<b>0.78</b>	<b>0.93</b>	<b>1.03</b>

Table 1: **Quantitative results of Chamfer Distance on DTU dataset with sparse inputs.** “\*” denotes that the method needs the ground-truth depth for supervision.

**Generalization training.** We select  $N = 4$  for sparse setting and  $N = 19$  for dense setting. We use Adam optimizer [17] with the base learning rate of  $1e-3$  for feature network and  $5e-4$  for other MLPs. We train the joint loss for 16 epochs on two A100 GPUs. We increase the value of  $\alpha$  from 0 to 1 and in the first 2 epochs. In our implementation, we generate the surface points  $P'$  of each image of the model trained with dense input first, and then distill the model with sparse input, with  $\beta$  set to 1. We build the generalized multi-scale volume with 5 scales, whose resolution increase from  $2^4$  to  $2^8$ . Each volume is equipped with thin features with only 4 feature channels, which allows us to save memory compared to general single volume methods.

**Fine-tuning.** After generalization training, we first reconstruct the generalized multi-scale volume, which has encoded the geometry information. Then we sparse the multi-scale volume by pruning voxels far from the surface. During fine-tuning, we abandon the feature network, and directly optimize the multi-scale volume and MLPs. With the generalization prior, we can achieve state-of-the-art performance in only about 20 minutes of fine-tuning.

## 4 Experiments

We demonstrate the state-of-the-art performance of GenS with comprehensive experiments and verify the effectiveness of each module through ablation studies. We first introduce the datasets and then analyze our results.

**Datasets.** We conduct experiments on both DTU [12] and BlendedMVS [59] datasets as previous methods [49, 60, 23]. Our generalization model is trained on DTU dataset, which is an indoor MVS dataset with 124 different scenes scanned from 49 or 64 views with fixed camera trajectories. Following [61, 49, 23], we take the same 15 scenes for testing. The training set is defined as in [58, 36], and the test scenes contained therein are removed. We also evaluate our model on BlendedMVS, which is a large-scale synthetic dataset. Each scene is scanned from different number of views, and all images has a resolution of  $768 \times 576$ . We report the Chamfer Distance for DTU, and show some visual effects for BlendedMVS.

### 4.1 Comparisons

**Results on DTU.** We first adopt the same testing split and configuration as [23] to compare with existing generalizable methods [3, 50, 23, 63]. The results shown in Tab. 1 indicate that our model outperforms existing methods by a significant margin, and this advantage can be amplified after rapid fine-tuning (about 20 mins). Even compared with recent method [40], which adopts the ground-truth depth for supervision, our model can achieve superior results. The qualitative results in Fig. 1 show that our reconstruction exhibits finer details. We further conducted more experiments on DTU to compare with per-scene overfitting methods with more input views. The quantitative comparisons in Tab. 2 show that our model can surpass some methods [61, 34, 49, 60] just through a very fast network inference, i.e., we can achieve more than 34% improvement on scene 24 compared with [49]. After a fast fine-tuning, the performance can be significantly improved, and even surpassing recent SOTA works [4, 5, 51]. Some visualization results in Fig. 7 depict that our model trained on large amounts of data is more robust to ambiguous regions.

Method	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
IDR [61]	1.63	1.87	0.63	0.48	1.04	0.79	0.77	1.33	1.16	0.76	0.67	0.90	0.42	0.51	0.53	0.90
MVSDF [65]	0.83	1.76	0.88	0.44	1.11	0.90	0.75	1.26	1.02	1.35	0.87	0.84	0.34	0.47	0.46	0.88
COLMAP [44]	<b>0.45</b>	0.91	0.37	<b>0.37</b>	0.90	1.00	0.54	1.22	1.08	<b>0.64</b>	<b>0.48</b>	<b>0.59</b>	0.32	0.45	0.43	0.65
NeRF [27]	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
UNISURF [34]	1.32	1.36	1.72	0.44	1.35	0.79	0.80	1.49	1.37	0.89	0.59	1.47	0.46	0.59	0.62	1.02
VolSDF [60]	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
NeuS [49]	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
HF-NeuS [51]	0.76	1.32	0.70	0.39	1.06	0.63	0.63	<b>1.15</b>	1.12	0.80	0.52	1.22	0.33	0.49	0.50	0.77
Voxurf [53]	0.65	0.74	0.39	<b>0.35</b>	0.96	0.64	0.85	1.58	1.01	0.68	0.60	1.11	0.37	0.45	0.47	0.72
NeuralWarp [4]	0.49	<b>0.71</b>	0.38	0.38	<b>0.79</b>	0.81	0.82	1.20	1.06	0.68	0.66	0.74	0.41	0.63	0.51	0.68
Geo-NeuS* [5]	0.46	0.83	0.38	0.39	0.88	<b>0.61</b>	<b>0.51</b>	1.26	<b>0.92</b>	0.68	0.57	<b>0.82</b>	<b>0.30</b>	<b>0.41</b>	<b>0.42</b>	0.63
<b>GenS</b>	0.66	1.01	0.71	0.43	1.06	0.99	0.73	1.43	1.18	0.78	0.64	0.93	0.38	0.54	0.54	0.80
<b>GenS (ft)</b>	0.55	<b>0.71</b>	0.39	0.38	<b>0.79</b>	0.65	0.57	1.29	0.96	<b>0.64</b>	0.49	<b>0.59</b>	0.33	0.44	0.45	<b>0.62</b>

Table 2: **Quantitative results of Chamfer Distance on DTU dataset with dense inputs.** ‘\*’ denotes that we retrain the Geo-NeuS without sparse geometric supervision.

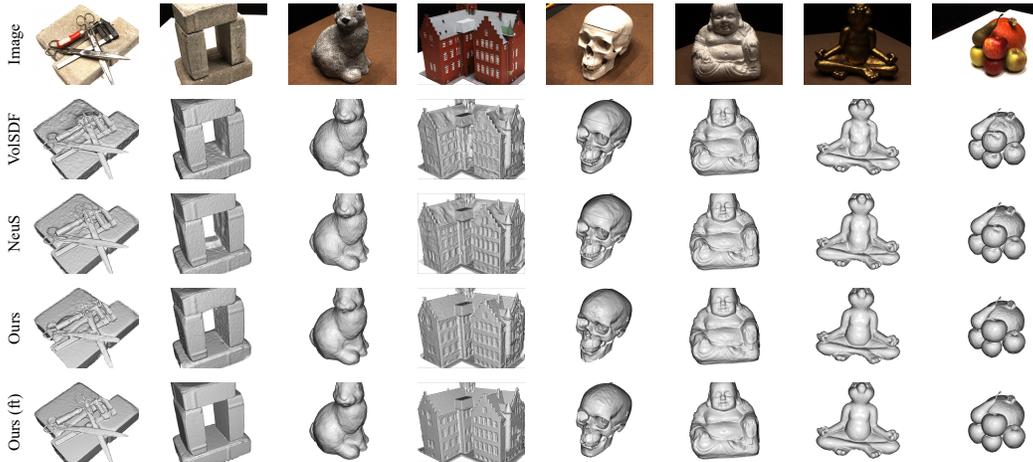


Figure 7: **Qualitative comparisons with per-scene overfitting methods on DTU dataset with dense inputs.**

**Generalizing to BlendedMVS.** Like [23], we conduct additional verification on BlendedMVS to showcase the generalization ability of our model. We also employ the same evaluation strategy as [23] for a fair comparison. As shown in Fig. 1, just through a fast network inference, our model can recover more geometric details than SparseNeuS [23]. Through the fine-tuning of 5k iterations, the effect will be significantly improved and better than SparseNeuS with 10k iterations.

## 4.2 Analysis

**Ablation studies.** We conduct ablation studies on DTU dataset to understand how the components of our model contribute to the overall performance. We start with our baseline model SparseNeuS [23], and gradually insert our contributions. The results in Tab. 4 show that our full model combining all components has the best mean score, and the baseline model, without any of our contributions, performs the worst. **Multi-scale Feature-metric Consistency (MFC):** Our self-enhanced MFC can continuously improve the multi-view consistency of the model, and we also elaborated the ablation results of its three main characteristics in Tab. 3. And the baseline is based on the pixel-wise feature consistency proposed in [65]. It can be seen that our strategy is more robust and efficiency. **Generalized Multi-scale Volume (GMV):**

Patch-similarity	Multi-scale	Self-enhanced	Mean
✗	✗	✗	1.86
✓	✗	✗	1.76
✓	✓	✗	1.73
✓	✓	✓	<b>1.62</b>

Table 3: **Some ablation studies on MFC.**



Figure 8: **Reconstruction from volumes with different resolution.**

MFC	GMV	VCL	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
✗	✗	✗	2.26	3.39	2.04	1.27	2.47	2.65	1.62	1.84	1.61	1.32	1.82	1.94	0.91	1.78	1.62	1.90
✓	✗	✗	1.61	3.12	1.99	1.16	2.00	2.21	1.30	1.58	1.45	1.18	1.48	1.53	0.80	1.54	1.43	1.62
✓	✓	✗	1.51	3.07	1.88	0.97	1.56	2.11	1.12	<b>1.45</b>	<b>1.31</b>	<b>0.95</b>	1.20	1.02	0.64	1.32	1.24	1.42
✓	✓	✓	<b>1.45</b>	<b>2.77</b>	<b>1.69</b>	<b>0.97</b>	<b>1.54</b>	<b>1.90</b>	<b>1.03</b>	1.49	1.36	0.97	<b>1.07</b>	<b>0.97</b>	<b>0.62</b>	<b>1.14</b>	<b>1.16</b>	<b>1.34</b>

Table 4: Ablation results on DTU.

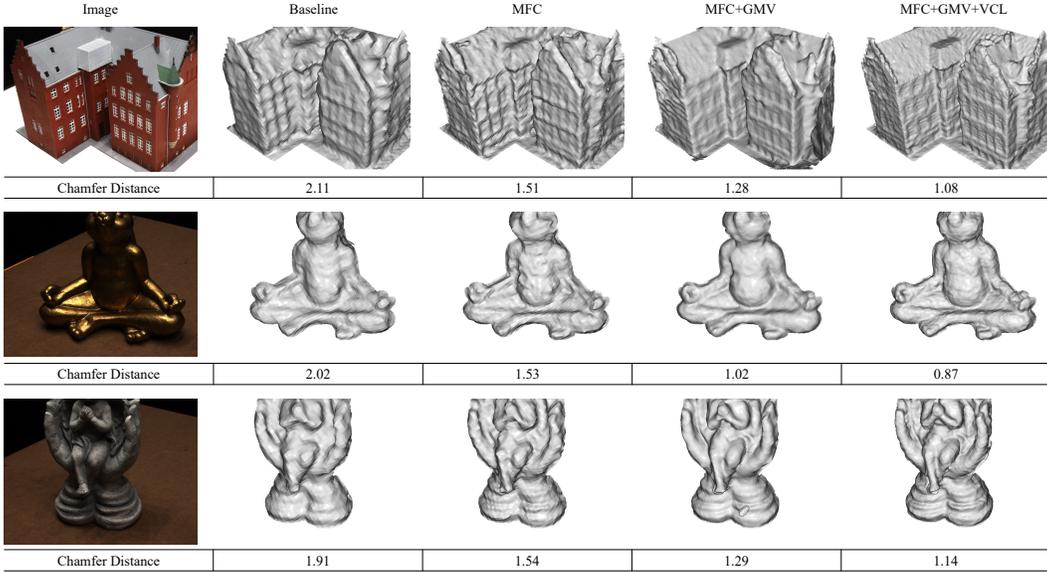


Figure 9: Visualization of some ablation results on DTU.

We show some results of the models have different resolutions in Fig. 8. We can see that the reconstruction of a single high-resolution volume is unbearably noisy (higher-resolution volume will lead to more empty voxels, which is more tricky for generalizable models.) and overly smooth at low resolution, whereas our GMV reconstructs clean and detailed geometry. And our representation is lighter than a single volume with a resolution of  $32 \times 192^3$  due to our thin feature. **View Contrast Loss (VCL):** As the results shown in Tab. 1 and Tab. 2, the reconstruction with dense inputs is more accurate than the sparse reconstruction, we therefore treat the former as a teacher to teach the latter. The results shown in Tab. 4 validate that this strategy can indeed improve the reconstruction quality of the model. The visualization of the ablation results are shown in Fig. 9, which further depicts that every contributions we propose can continuously improve performance.

**Limitation.** Although our model exhibits excellent generalization performance in multi-view reconstruction, we found that it cannot satisfactorily handle scenes with large camera motion, such as surrounding cases. Because in these scenarios, the aggregation features will be polluted by the ray features shooting from behind. Our current solution is to first predict the local structure covered by some adjacent viewpoints like [23, 58, 36], and finally fuse them together.

## 5 Conclusion

In this paper, we introduced GenS, an end-to-end generalizable neural surface reconstruction model. We first encode all scenes into our generalized multi-scale volume, a more powerful representation that can reconstruct clean and detailed 3D structures. Then we introduce the multi-scale feature-metric consistency to combat the challenge of the photometric consistency failure. The learnable multi-scale feature can provide more discriminative representation and can be self-enhanced during the generalization training. And we finally designed a view contrast loss to improve the accuracy of the reconstruction through distilling the finer reconstruction from dense inputs to the reconstruction from sparse inputs. Experimental results on both DTU and BlendedMVS datasets show that our model possess stronger generalization ability and can achieve start-of-the-art reconstruction through fast network inference or efficient fine-tuning. In the future, we will focus on improving the performance of the model in difficult scenarios.

## Acknowledgments and Disclosure of Funding

This work is financially supported by National Natural Science Foundation of China U21B2012 and 62072013, Shenzhen Science and Technology Program-Shenzhen Cultivation of Excellent Scientific and Technological Innovation Talents project(Grant No. RCJC20200714114435057) , Shenzhen Science and Technology Program-Shenzhen Hong Kong joint funding project (Grant No. SGDX20211123144400001), this work is also financially supported for Outstanding Talents Training Fund in Shenzhen. In addition, we thank our collaborators in Alibaba Group and the anonymous reviewers for their valuable comments.

## References

- [1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 28(3):24, 2009.
- [2] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE TVCG*, 5(4):349–359, 1999.
- [3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, pages 14124–14133, 2021.
- [4] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *CVPR*, pages 6260–6269, 2022.
- [5] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-neus: geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *NeurIPS*, 2022.
- [6] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE TPAMI*, 32(8):1362–1376, 2009.
- [7] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, pages 873–881, 2015.
- [8] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *ICCV*, pages 7154–7164, 2019.
- [9] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.
- [10] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, pages 2495–2504, 2020.
- [11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *UIST*, pages 559–568, 2011.
- [12] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, pages 406–413, 2014.
- [13] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *CVPR*, pages 18365–18375, 2022.
- [14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016.
- [15] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *SGP*, volume 7, page 0, 2006.

- [16] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM ToG*, 32(3):1–13, 2013.
- [17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [18] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *IJCV*, 38:199–218, 2000.
- [19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.
- [20] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 33:15651–15663, 2020.
- [21] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *CVPR*, pages 2019–2028, 2020.
- [22] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. *NeurIPS*, 32, 2019.
- [23] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *ECCV*, pages 210–227. Springer, 2022.
- [24] Nelson Max. Optical models for direct volume rendering. *IEEE TVCG*, 1(2):99–108, 1995.
- [25] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, pages 4460–4470, 2019.
- [26] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *ICCV*, pages 4743–4752, 2019.
- [27] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421, 2020.
- [28] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM ToG*, 41(4):1–15, 2022.
- [29] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *ECCV*, pages 414–431. Springer, 2020.
- [30] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *ICCV*, pages 5379–5389, 2019.
- [31] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, pages 3504–3515, 2020.
- [32] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM ToG*, 32(6):1–11, 2013.
- [33] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *ICCV*, pages 4531–4540, 2019.
- [34] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, pages 5589–5599, 2021.

- [35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019.
- [36] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *CVPR*, pages 8645–8654, 2022.
- [37] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, pages 523–540. Springer, 2020.
- [38] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, pages 10318–10327, 2021.
- [39] Ke Qiu, Yawen Lai, Shiyi Liu, and Ronggang Wang. Self-supervised multi-view stereo via inter and intra network pseudo depth. In *ACMMM*, pages 2305–2313, 2022.
- [40] Yufan Ren, Fangjinhua Wang, Tong Zhang, Marc Pollefeys, and Sabine Süsstrunk. Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In *CVPR*, 2023.
- [41] Leonid I Rudin and Stanley Osher. Total variation based image restoration with free local constraints. In *ICIP*, volume 1, pages 31–35. IEEE, 1994.
- [42] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, pages 2304–2314, 2019.
- [43] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016.
- [44] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, pages 501–518. Springer, 2016.
- [45] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 35:151–173, 1999.
- [46] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, pages 5459–5469, 2022.
- [47] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *CVPR*, pages 11358–11367, 2021.
- [48] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *CVPR*, pages 14194–14203, 2021.
- [49] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021.
- [50] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, pages 4690–4699, 2021.
- [51] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. Hf-neus: Improved surface reconstruction using high-frequency details. *NeurIPS*, 35:1966–1978, 2022.
- [52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.
- [53] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. In *arXiv preprint arXiv:2208.12697*, 2022.

- [54] Hongbin Xu, Zhipeng Zhou, Yu Qiao, Wenxiong Kang, and Qiuxia Wu. Self-supervised multi-view stereo via effective co-segmentation and data-augmentation.
- [55] Hongbin Xu, Zhipeng Zhou, Yali Wang, Wenxiong Kang, Baigui Sun, Hao Li, and Yu Qiao. Digging into uncertainty in self-supervised multi-view stereo. In *ICCV*, pages 6078–6087, 2021.
- [56] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *CVPR*, pages 5483–5492, 2019.
- [57] Jiayu Yang, Jose M Alvarez, and Miaomiao Liu. Self-supervised learning of depth inference for multi-view stereo. In *CVPR*, pages 7526–7534, 2021.
- [58] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, pages 767–783, 2018.
- [59] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, pages 1790–1799, 2020.
- [60] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 34:4805–4815, 2021.
- [61] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 33:2492–2502, 2020.
- [62] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *ICCV*, pages 5752–5761, 2021.
- [63] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021.
- [64] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, volume 35, pages 25018–25032, 2022.
- [65] Jingyang Zhang, Yao Yao, and Long Quan. Learning signed distance field for multi-view surface reconstruction. In *ICCV*, pages 6525–6534, 2021.
- [66] Kai Zhang, Gernot Riegler, Noah Snaveley, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.