GaussianUDF: Inferring Unsigned Distance Functions through 3D Gaussian Splatting

Anonymous CVPR submission

Paper ID 9787

Abstract

Reconstructing open surfaces from multi-view images is vital in digitalizing complex objects in daily life. A widely used strategy is to learn unsigned distance functions (UDFs) by checking if their appearance conforms to the image observations through neural rendering. However, it is still hard to learn the continuous and implicit UDF representations through 3D Gaussians splatting (3DGS) due to the discrete and explicit scene representations, i.e., 3D Gaussians. To resolve this issue, we propose a novel approach to bridge the gap between 3D Gaussians and UDFs. Our key idea is to overfit thin and flat 2D Gaussian planes on surfaces, and then, leverage the self-supervision and gradient-based inference to supervise unsigned distances in both near and far area to surfaces. To this end, we introduce novel constraints and strategies to constrain the learning of 2D Gaussians to pursue more stable optimization and more reliable self-supervision, addressing the challenges brought by complicated gradient field on or near the zero level set of UDFs. We report numerical and visual comparisons with the state-of-the-art on widely used benchmarks and real data to show our advantages in terms of accuracy, efficiency, completeness, and sharpness of reconstructed open surfaces with boundaries.

1. Introduction

It is vital but still challenging to reconstruct shapes with thin and open surfaces and sharp boundaries from multi-view images. A widely used strategy is to learn implicit representations, such as unsigned distance functions (UDFs), by minimizing rendering errors of UDFs with respect to multi-view observations, such as RGB images [9, 22, 23, 28, 49]. This strategy shows promising results because of the advantages of both implicit representations and the neural volume rendering, i.e., the ability of reconstructing arbitrary topology and the differentiability for back-propagating the gradients of rendering errors. Eventually, the open surfaces can be extracted from the zero level set of the learned UDF.

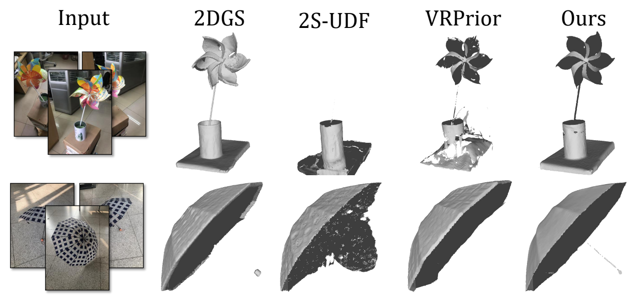

Figure 1. The comparisons with 2DGS [14], 2S-UDF [9], and VR-Prior [49]. Our method recovers the most accurate open surfaces without artifacts.

Recent methods [9, 22, 23, 28, 49] usually learn a UDF within a radiance field through the volume rendering introduced by NeRF [30]. They infer unsigned distances at sampled points along rays emitted from views through an additional transformation which bridges the gap between the UDF and the radiance field. However, NeRF-based rendering is not efficient due to the need of finding intersections with ray tracing. This makes 3D Gaussian Splatting (3DGS) a promising solution since rasterizing 3D Gaussians is not only differentiable but also faster than ray tracing in NeRF-based rendering. However, one obstacle coming from the discrete and explicit scene representations, i.e., 3D Gaussians, is that they are much different from the continuous and implicit radiance field. Therefore, how to overcome this obstacle is the most challenging problem to reveal complete, smooth, and continuous UDFs through 3DGS.

To resolve this problem, we introduce a novel approach to inferring UDFs from multi-view images through 3DGS, which can efficiently reconstruct high-quality surfaces with open structures as shown in Figure 1. Our key idea is to constrain 3D Gaussians to represent surfaces directly, based on which we estimate the unsigned distance field. Our novelty lies in two aspects: (1) the novel constraints that we imposed on the Gaussians, which overfits these Gaussians on the surfaces, (2) and the ways of inferring unsigned distances with self-supervision and gradient-based inference. To this end, we use 2D Gaussians in the 3D space which are thin enough to approximate the surface. We also align these 2D Gaussians on the surface using the gradient field of the implicit function, which involves a UDF in the differentiable rasterization procedure. Meanwhile, we introduce self-supervision along the normal of 2D Gaussians to infer unsigned distances near the 2D Gaussians, and infer unsigned distances far away from the surface with the gradient field of the UDF. Our evaluations show that our method successfully bridges the gap between discrete Gaussians and continuous UDFs in a fully differentiable manner, leading to more accurate, complete, and continuous open surface reconstructions than the state-of-the-art methods. Our contributions are summarized below.

- We present a novel approach to reconstruct thin and open surfaces from multi-view images with 3DGS, which bridges the gap between continuous UDFs and discrete 3D Gaussians in a differentiable manner.
- We introduce stable constraints to overfit 3D Gaussians on surfaces, and novel strategies to infer unsigned distances accurately in both near and far areas to the surface.
- Our method produces the state-of-the-art results in reconstructing shapes with open surfaces and sharp boundaries on the widely used benchmarks.

2. Related Work

2.1. Neural Implicit Representation

Neural implicit representations have shown great advantages in representing shapes using continuous functions due to their ability to represent surfaces with flexible topology in high resolutions. Typically, neural implicit functions map spatial query coordinates to occupancy probabilities [29] or signed/unsigned distances [6, 34]. Neural implicit functions can be learned from various 2D or 3D surface signals, such as RGB images [23, 36, 49], point clouds [3, 27, 32, 51], binary classification labels [29], and distance labels [6, 20, 34]. Among them, Neural-Pull [27] aims to pull the query points on the zero level set of the neural implicit function and achieves the learning of Signed Distance Functions (SDFs) from point clouds. The SDFs partition surfaces into exterior and interior regions, which limits such methods to modeling only watertight objects. To extend the capability of implicit functions to open surfaces, recent methods [6, 23, 41, 51] have been proposed to predict the unsigned distances from any query point to the surface to reconstruct high quality single layer surfaces. And some methods [5, 12, 47, 50] extend Marching Cubes [19, 24] or Dual Contouring [17] to efficiently and accurately extract meshes from unsigned distance field.

2.2. Novel View Synthesis

Neural Radiance Fields (NeRF) [30] have achieved promising results in novel view synthesis. The method adapts implicit field functions to encode view-dependent appearance. Specifically, NeRF maps the spatial points sampled on the ray to densities and colors with several Multi-Layer Perceptrons (MLPs), and then integrates the samples into pixel colors through volumetric rendering. Advancements [1, 2, 15, 31, 42, 43] following the development of NeRF have further extended its capabilities.

Recently, 3D Gaussian Splatting (3DGS) [18] has become an important breakthrough in the field. 3DGS represents the scene with 3D Gaussians including means, covariances, opacities and spherical harmonics parameters. The explicit representation avoids unnecessary computation cost in the empty space and achieves high quality and real-time novel view synthesis. Later, numerous methods [13, 25, 35, 38, 39, 45] extend this technique to a wide variety of fields.

2.3. Learning Neural SDFs with Multi-view Images

Combining implicit representations with neural rendering, NeRF-based methods [10, 33, 36, 37] can reconstruct watertight meshes well from multi-view images. These methods transform occupancy values [33] or signed distances [8, 10, 21, 36, 40] to density in volumetric rendering.

Recently, attempts [7, 11, 14, 46] have been made to reconstruct meshes from mutiple views with 3DGS. Several methods [7, 11, 14] have been developed to make 3D Gaussians approximate surfels and align with surfaces. And some methods [4, 26, 44, 48] optimize SDFs together with the 3D Gaussians. GOF [46] establishes a Gaussian opacity field from 3D Gaussians and extracts the surface from the levelset. However, these methods learn SDFs to model surfaces and are limited to reconstructing watertight meshes. In contrast, we aim to handle thin and open surfaces with 3D Gaussian Splatting, which can efficiently reconstruct non-watertight meshes. The recent method GSPull [48] also pulls the queries to the zero level set to learn SDF. However, the projection can not provide enough supervision to learn correct UDF due to the complexity of gradients on the isosurface. Therefore, we introduce self-supervision and other losses to overcome this challenge and reconstruct accurate and complete open surfaces.

2.4. Learning Neural UDFs with Volume Rendering

Unlike SDF modeling the surfaces as exterior and interior, UDF [6, 51] can handle arbitrary topologies. Recent methods [9, 22, 23, 28, 49] usually learn a UDF from multi-view images with volume rendering. NeuralUDF [23] flips the normal orientation behind the surface points. NeUDF [22] introduces a new probability density function. NeAT [28] learns additional validity to reconstruct open surfaces from

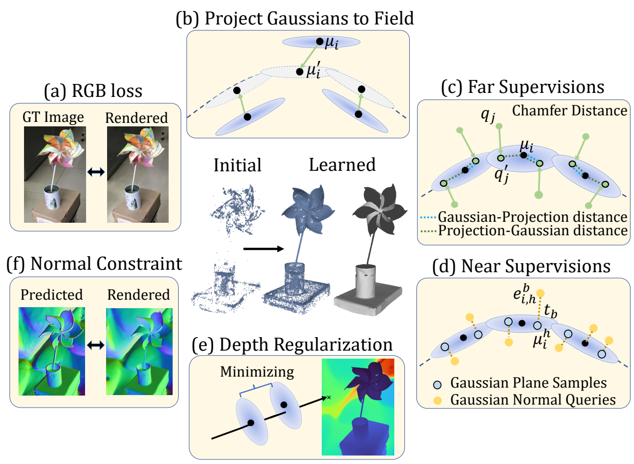

Figure 2. Overview of our method. (a) The UDF is optimized with the rendering process. To ensure Gaussians provide more accurate clues of the surfaces, (b) the Gaussians are projected to the zero level set of the UDF. (c) Projecting random queries to the Gaussian centers helps the UDF learn coarse shapes in far area. Moreover, (d) unsigned distances recovered near the Gaussian plane compensates for the sparsity of Gaussian centers. We adopt depth (e) and normal (f) regularization terms to make Gaussians align with surfaces well.

SDF. 2S-UDF [9] proposes a two-stage method to decouple density and weight. However, these methods need finding intersections and ray tracing in volume rendering, which leads to inefficiency. Our method is built on the point-based rendering of 3D Gaussian Splatting [18] without requiring any ray tracing process, resulting in improved efficiency.

3. Method

Overview. Figure 2 illustrates the framework of our approach. To overfit 3D Gaussians on surfaces, we follow 2DGS [14] to represent scenes using 2D Gaussians which are thin enough to represent open surfaces with sharp boundaries. We jointly infer a UDF f and learn 2D Gaussians $\{\boldsymbol{g}_i\}_{i=1}^{I}$ by minimizing rendering errors with respect to the observations through splatting a set of I 2D Gaussians. Besides the thin feature of 2D Gaussians, we also leverage the gradient field of the UDF to align 2D Gaussians to the zero level set of the UDF, which ensures these 2D Gaussians represent the surface faithfully. Based on this representation, we set up self-supervision along the normal of 2D Gaussians to supervise the learning of UDF around the surface, and also use the gradient field to infer unsigned distances, especially for the space far away from the surface. To this end, we also constrain the normal of 2D Gaussians and rendered depth images so that the 2D Gaussians can provide reliable self-supervisions and the gradient based inference for more accurate distance fields.

2D Gaussian Splatting. We leverage the differentiable splatting introduced by 2DGS [14] to render 2D Gaussians into images. Each 2D Gaussian $\boldsymbol{g}_i$ has several learnable parameters including the center $\boldsymbol{\mu}_i \in \mathbb{R}^{1\times3}$, the color $\boldsymbol{c}_i \in \mathbb{R}^{1\times3}$, the opacity α_i, the rotation matrix $\boldsymbol{r}_i \in \mathbb{R}^{3\times3}$, and scaling factors $\boldsymbol{s}_i \in \mathbb{R}^{1\times2}$, where $\boldsymbol{\mu}_i$ and $\boldsymbol{r}_i$ determine the location and pose of the Gaussian $\boldsymbol{g}_i$, $\boldsymbol{s}_i$ determines the variances along two axis of the Gaussian $\boldsymbol{g}_i$, the color $\boldsymbol{c}_i$ and the opacity α_i describe the appearance, and the last column of $\boldsymbol{r}_i$ represents the normal $\boldsymbol{n}_i$ of the flat $\boldsymbol{g}_i$.

We render $\{\boldsymbol{g}_i\}$ into a RGB color at each pixel (u, v) using α blending through a differentiable splatting procedure,

$$\boldsymbol{C}'(u,v) = \sum_{i=1}^{I} \boldsymbol{c}_i \alpha_i p_i(u,v) \prod_{k=1}^{i-1} (1 - \alpha_k p_k(u,v)), \quad (1)$$

where $\boldsymbol{C}'(u,v)$ is the color at the pixel location (u,v) on the rendered image $\boldsymbol{C}'$, and $p_i(u,v)$ is the probability of contributing to pixel (u,v) from the projection of $\boldsymbol{g}_i$. Similarly, we can also render depth or normal maps by replacing the color with projection distances or the normal of 2D Gaussians in the above equation. We learn the Gaussians $\{\boldsymbol{g}_i\}$ by minimizing rendering errors with respect to the observations $\boldsymbol{C}$,

$$L_{rgb} = ||\boldsymbol{C}'(u,v) - \boldsymbol{C}(u,v)||_1. \quad (2)$$

Unsigned Distance Functions. An unsigned distance function f describes a distance field, indicating the distance d to the nearest surface in a scene at an arbitrary location $\boldsymbol{q} = (x, y, z)$, i.e., $d = f(\boldsymbol{q})$. A gradient field can be derived from f, where the gradient $\nabla f(\boldsymbol{q})$ at each query $\boldsymbol{q}$ points to a direction that is far away from the nearest surface.

The gradient field of f provides good clues to reveal surfaces which are indicated by the zero level set of f. Neural-Pull [27] has shown that one can infer signed distances by pulling randomly sampled points against the direction of gradient to the surface. However, UDF has pretty complex gradient field near both sides of the surface, due to the absence of gradient on the surface. This fact becomes a serious problem in learning UDF from multi-view images.

To resolve this issue, we employ two kinds of supervisions to infer unsigned distances with 2D Gaussians. One is to use the gradient field to pull queries onto the zero level set of f, which pays more attention to the space far away from surfaces. The other is to leverage the normal of the Gaussians to produce self-supervision covering the whole flat plane, which focuses on the area closed to surfaces.

Self-supervision and Inference. For the first supervision, we randomly sample J queries $\{\boldsymbol{q}_j\}_{j=1}^{J}$ around the centers $\boldsymbol{\mu}_i$ of Gaussians $\{\boldsymbol{g}_i\}$ using the sampling strategy introduced in Neural-Pull [27]. We project $\{\boldsymbol{q}_j\}$ onto the zero level set of f below,

$$\boldsymbol{q}'_j = \boldsymbol{q}_j - d_j \cdot \frac{\nabla f(\boldsymbol{q}_j)}{|\nabla f(\boldsymbol{q}_j)|}, \quad (3)$$

CVPR #9787

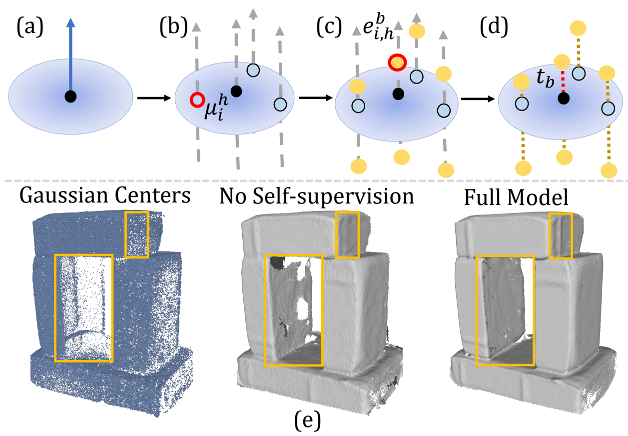

Figure 3. Self-supervision loss. For a Gaussian in (a), (b) we first sample root point μ_i^h on the plane. (c) Then we randomly move the root point to position $e_{i,h}^b$ along the positive direction or negative direction of the normal with offset t_b. (d) We use $\{e_{i,h}^b, t_b\}$ as a training sample pair to train the UDF network. (e) The below reconstructed meshes show that the 2D Gaussian planes provide more surface information for the UDF, which helps to fill the holes and capture more details.

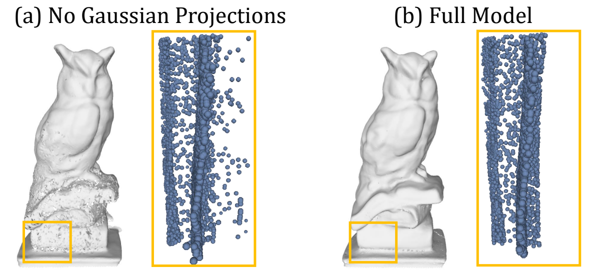

Figure 4. We project the Gaussian centers to the zero level set with a constraint, which makes the point cloud have less noises and the UDF have more accurate surface.

where $\boldsymbol{q}_j'$ is the projection of $\boldsymbol{q}_j$ and $d_j = f(\boldsymbol{q}_j)$ is the unsigned distance. We leverage the centers of Gaussians to supervise the projections,

$$\begin{aligned} L_{far} =& \frac{1}{J} \sum_{\boldsymbol{q}' \in \{\boldsymbol{q}_j'\}} \min_{\boldsymbol{\mu} \in \{\boldsymbol{\mu}_i\}} ||\boldsymbol{q}' - \boldsymbol{\mu}||_2^2 \\ &+ \frac{1}{I} \sum_{\boldsymbol{\mu} \in \{\boldsymbol{\mu}_i\}} \min_{\boldsymbol{q}' \in \{\boldsymbol{q}_j'\}} ||\boldsymbol{\mu} - \boldsymbol{q}'||_2^2, \end{aligned} \tag{4}$$

where L_{far} evaluates the Chamfer distance between the set of projections $\{\boldsymbol{q}_j'\}$ and $\{\boldsymbol{\mu}_i\}$, encouraging the UDF f to conform to the surface represented by the Gaussian centers. To relief the computational burden during optimization, we only use a batch of $\boldsymbol{g}_i$ and query points sampled around them to evaluate this loss in each iteration.

Gaussians are sparse in some regions, which limits their ability to represent surfaces, so relying solely on their centers with L_{far} is inadequate. Hence, the first supervision merely provide a coarse supervision which is helpful for inferring unsigned distances in areas far away from the surface. As a complement, our self-supervision will provide the second kind of supervision over the whole Gaussian plane near the surface.

Our self-supervision is illustrated in Figure 3. We set up the self-supervision using the normal $\boldsymbol{n}_i$ of each Gaussian $\boldsymbol{g}_i$ and the samples on its flat plane, which makes sure the Gaussian plane can cover enough space to overfit surfaces regardless of the sparsity of Gaussian centers. As shown in Figure 3 (b), we sample root points $\{\boldsymbol{\mu}_i^h\}_{h=1}^H$ on the flat plane, and randomly sample samples $\{\boldsymbol{e}_{i,h}^b, t_b\}_{b=1}^B$ along the direction of normal $\boldsymbol{n}_i$ by $\boldsymbol{e}_{i,h}^b = \boldsymbol{\mu}_i^h + t_b \cdot \boldsymbol{n}_i / ||\boldsymbol{n}_i||_2$, as shown in Fig. 3 (c), where t_b is randomly sampled from $[-T, T]$, which makes sure we have training samples on both sides of the Gaussian. We record $\boldsymbol{e}_{i,h}^b$ and t_b as a training sample $\{\boldsymbol{e}_{i,h}^b, t_b\}$ in Fig. 3 (d), where t_b is regarded as the ground truth unsigned distances at $\boldsymbol{e}_{i,h}^b$. We will introduce another constraint L_{norm} to keep the normal of Gaussians orthogonal to surfaces, which also makes the self-supervision more reliable to use.

Eventually, we use $\{\boldsymbol{e}_i^b, t_b\}$ as self-supervision to train the UDF f through a L1 loss,

$$L_{near} = ||f(\boldsymbol{e}_i^b) - t_b||_1, \tag{5}$$

Overfitting Gaussians to Surfaces. Besides the thin feature of 2D Gaussian, we also move 2D Gaussians to the zero level set of f, which ensures to overfit 2D Gaussians to surfaces. Since the gradient field nearby the zero level set of UDFs is very complicated, we do not directly pull the center $\boldsymbol{\mu}_i$ of 2D Gaussians $\boldsymbol{g}_i$ using Eq. (3), which avoids the incorrect gradients that destablizes the optimization when most of 2D Gaussians are near the surface, as shown in Figure 4 (a). We notice concurrent work [48] that also involves gradients of SDF to constrain locations of Gaussians. But gradients of SDF near the zero level set is much more stable than UDF. Therefore, we propose to use an explicit constraint to project Gaussians on the zero level set of f. We run Eq. (3) and stop back-propagating the gradient through f, obtaining the projection of Gaussian $\boldsymbol{\mu}_i'$. Then, we regard $\boldsymbol{\mu}_i'$ as target and minimize the distance to directly update the location $\boldsymbol{\mu}_i$ of Gaussians below, which stabilizes the optimization near the zero level set, as shown in Figure 4 (b),

$$L_{proj} = ||\boldsymbol{\mu}_i' - \boldsymbol{\mu}_i||_2. \tag{6}$$

Constraints on Depth and Normals. To make all 2D Gaussians get closer to the surface, we adopt a depth distortion loss [14] to constrain Gaussian positions. Along each ray, we monitor the depth of intersections to Gaussians, and constraints their interval between two intersections,

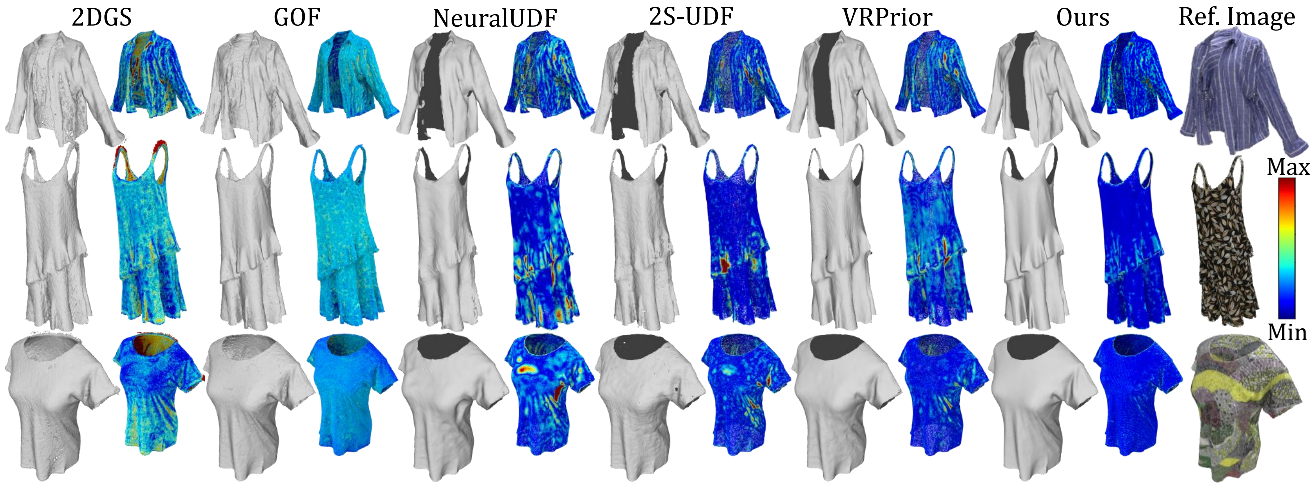

Figure 5. Qualitative comparison with 2DGS [14], GOF [46], NeuralUDF [23], 2S-UDF [9], and VRPrior [49] in DF3D [52] dataset. Note that VRPrior needs additional depth images to learn priors. The dark color on meshes represents the back faces of open surfaces, and the error map is shown next to the mesh. Our method obtains more accurate surfaces and captures more details such as the folds in the clothing.

	Method	30	92	117	133	164	204	300	320	448	522	591	598	Mean	Time
SDF	NeuS[36]	3.18	4.82	4.78	4.99	3.73	5.71	5.89	2.21	5.89	3.60	2.44	5.13	4.36	5.7h
	2DGS[14]	3.79	3.66	4.24	3.75	3.91	4.01	4.02	3.74	3.51	3.89	3.21	4.01	3.81	6min
	GOF[46]	3.15	2.47	2.49	2.23	2.38	2.65	2.40	2.41	2.14	3.00	2.18	2.37	2.49	47min
UDF	NeralUDF[23]	1.92	2.05	2.36	1.58	1.33	4.11	2.47	1.50	1.63	2.47	2.16	2.15	2.15	8.6h
	2S-UDF [9]	1.92	1.97	1.77	1.58	**1.32**	2.46	3.43	1.47	2.00	2.14	1.84	1.91	1.98	7.8h
	VRPrior[49]	**1.59**	1.73	2.06	1.63	1.44	2.07	1.66	1.60	**1.39**	2.14	**1.50**	1.67	1.71	9.2h
	Ours	1.85	**1.69**	**1.18**	**1.32**	1.59	**1.59**	**1.51**	**1.27**	2.62	**1.65**	1.74	**1.22**	**1.60**	1.6h

Table 1. Quantitative results of Chamfer Distance ($\times 10^{-3}$) of each object in DF3D [52] dataset.

$$L_{depth} = \sum_{k1,k2} g_{k1} g_{k2} |z_{k1} - z_{k2}|, \tag{7}$$

where $g_{k1} = \alpha_{k1} p_{k1}(u,v) \prod_{k=1}^{k1-1}(1 - \alpha_k p_k(u,v))$.

Furthermore, to make the self-supervision more reliable, we add supervision on the normals of Gaussians $\boldsymbol{n}_i$. We estimate normal maps from the depth gradients on the rendered depth images. Along each ray, we align the normal $\boldsymbol{n}_i$ of Gaussians hit by the ray with the estimated normal $\boldsymbol{N}_i$ on the rendered depth maps,

$$L_{norm} = \sum_k g_k (1 - \boldsymbol{n}_k^T \boldsymbol{N}_k). \tag{8}$$

Loss Function. We optimize 2D Gaussians in a scene by minimizing the following loss function,

$$\begin{aligned} L =& (1-\lambda_1) L_{rgb} + \lambda_1 L_{ssim} + \lambda_2 L_{far} + \lambda_3 L_{near} \\ &+ \lambda_4 L_{proj} + \lambda_5 L_{depth} + \lambda_6 L_{norm}, \end{aligned} \tag{9}$$

where L_{ssim} is a rendering quality loss inherited from 3DGS [18], and all these loss terms are balanced by weights λ_{1-6}.

4. Experiments

4.1. Experiment Settings

Details. The weights are set as $\lambda_1 = 0.2, \lambda_2 = 1.0, \lambda_3 = 1.0, \lambda_4 = 0.15$ on DTU [16] and DF3D [52], $\lambda_4 = 0.0001$ on real scans, $\lambda_5 = 1000$ on DTU, $\lambda_5 = 0$ on other scenes, and $\lambda_6 = 0.05$. We optimize the model for 30k iterations for all datasets. For the self-supervision, we sample 500 Gaussian planes per batch and sample 10 root points per plane. The offset t_b is sampled from a uniform distribution that is bounded by zero and T, and we set $T = 0.01$ in DF3D dataset and $T = 0.02$ in DTU dataset. Similar to NeuralUDF [23] and VRPrior [49], we tune the reconstruction using an additional warp loss [8, 10] on DTU dataset. The UDF f is parameterized by a 8-layer MLP with 256 hidden units and ReLU activation functions, and the activation of the last layer is an absolute value function. We apply positional encoding [30] to the input query point coordinates. We use an initial learning 1×10^{-3} with cosine learning rate decay strategy for training the UDF network. We conduct all experiments on a single NVIDIA 3090 GPU.

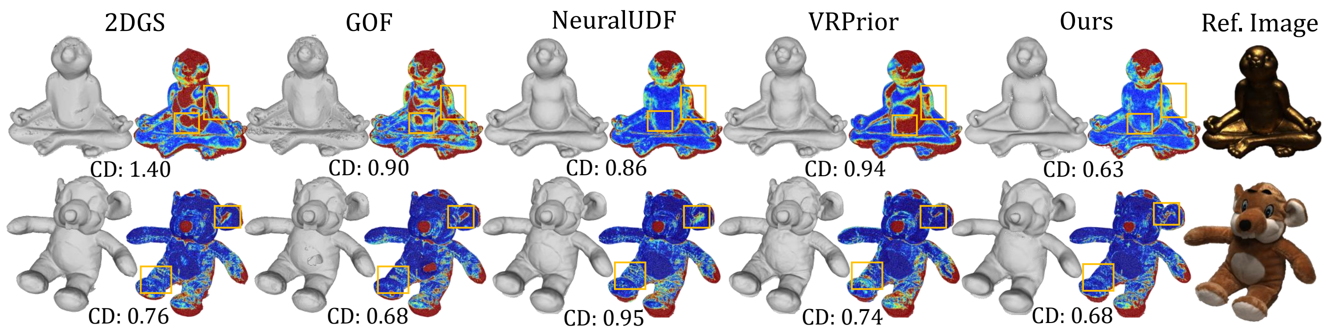

Figure 6. Visual comparisons of reconstruction and error maps on DTU [16] dataset. Larger errors are shown in warmer colors. Our method obtains visual-appealing results with small errors.

Method	2DGS	GOF	NeuralUDF	VRPrior	Ours
Average	0.80	0.74	0.75	0.71	**0.68**

Table 2. Numerical comparisons with 2DGS [14], GOF [46], NeuralUDF [23] and VRPrior [49] in terms of CD on DTU [16] dataset. Detailed comparisons can be found in the appendix.

Datasets and Evaluation Metrics. We evaluate the proposed method on DeepFasion3D (DF3D) [52] dataset, DTU [16] dataset, NeUDF [22] dataset, and our real-captured dataset. For DF3D dataset, we use the same 12 garments as previous methods [23, 49], each garment is scanned with 72 images in a resolution of 1024×1024 and is provided with the ground truth point cloud for evaluation. For DTU [16] dataset, we use the widely used 15 scenes that are all watertight and each scene contains 49 or 64 images in a resolution of 1600×1200. We use two real scans in NeUDF [22] dataset, and captures four real scenes. In our experiments, we train our models without mask supervision in all datasets. For a fair comparison, we use the MeshUDF [12] algorithm to extract open surfaces from unsigned distance fields like previous methods [9, 22, 23, 49], and use the Chamfer Distance (CD) as the metric for DF3D dataset and DTU dataset that provide ground truth.

Baselines. We compare the proposed method with the following state-of-the-art methods: 1) SDF-based surface reconstruction methods including NeuS [36], 2DGS [14], and GOF [46], and 2)UDF-based surface reconstruction methods for open surfaces including NeuralUDF [23], 2S-UDF [9], and VRPrior [49]. For the open surface dataset DF3D [52], we trained GOF [46] and 2S-UDF [9] with the default parameters. Since we share Gaussian optimization parameters with 2DGS [14], we keep these parameters the same. The other quantitative metrics are borrowed from the original papers.

4.2. Evaluations

Comparisons in Reconstructing Open Surfaces. We evaluate our method on the DF3D[52] dataset which includes shapes with open surfaces. The CD $(\times 10^{-3})$ in Table 1 indicates that we achieve the best performance compared to baseline methods. The reconstruction errors with SDF-based baselines including NeuS [36], 2DGS [14], and GOF [46] are large because they try to either wrap the surface with closed mesh or excessively smooth out the details on the clothing. The visual comparisons in Figure 5 show that our method can reconstruct open surfaces with more details. The methods 2DGS and GOF inherit the shortcoming of SDF-based methods which learn to reconstruct closed surfaces. This results the double-layered faces and increases the reconstruction errors. The UDF-based baselines reconstruct the open surface correctly, but they fail to capture details, resulting in over-smoothed results. Thanks to the quick convergence of 3D Gaussian splatting, the speed of training our method can be much faster than the NeRF-based methods for open surface reconstruction.

Comparisons in Reconstructing Closed Surfaces. We further conduct evaluations on DTU dataset, and report the quantitative and visual comparisons in Table 2 and Figure 6, respectively. Our method achieves the best performance in terms of average CD compared among baseline methods, demonstrating its overall robustness. The complex gradients near the surface make the learning of UDF more challenging than SDF. Without assuming closed surfaces, our method still achieves comparable results or even better results in some scenes to SDF-based methods that are specifically designed for closed surfaces. Moreover, our approach achieves better quantification on some relatively complex shapes than baseline methods. As shown in the error map in Figure 6, our method accurately reconstruct surface even with complex light conditions. The underlying reason is that the geometric information of the UDF is derived from the positions of Gaussians, making it less sensitive to appearance attributes like opacity.

Results on Real Scans. We first conduct evaluation on the public real-captured NeUDF [22] dataset. As shown in Figure 7, our method can reconstruct extremely flat and thin surfaces. Due to the detail-capturing capability of Gaussian

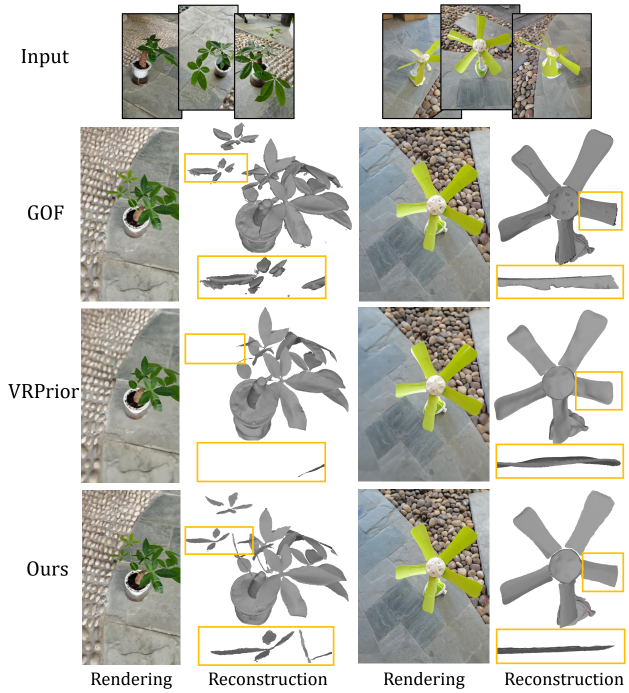

Figure 7. The reconstruction results on NeUDF [22] dataset. Our method accurately reconstructs the open surfaces in real scans.

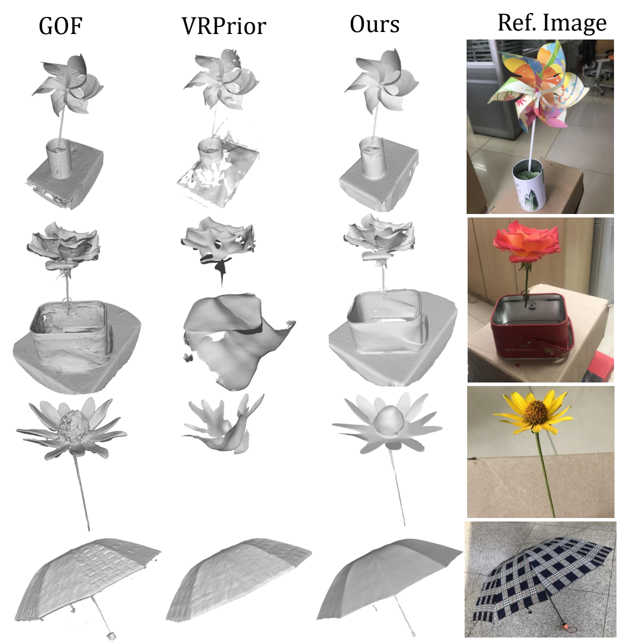

Figure 8. The reconstruction results on real scans. Our method reconstructs accurate and complete surfaces.

Splatting, our method achieves more complete geometry reconstruction compared to the NeRF-based state-of-the-art VRPrior [49], such as the plant leaves, even if it uses additional data-driven learned priors. We further report our results on our self-captured four scenes with thin and open surfaces. As shown in Figure 8, VRPrior [49] struggles to reconstruct correct structures for objects with relatively simple textures, and GOF [46] reconstructs double-layer surfaces without smoothness. Instead, our method can reconstruct more complete, accurate, and smoother meshes.

Settings	Far	Near	Proj	Warp	CD ↓
Only Far	✓				0.99
Far & Near	✓	✓			0.78
Far & Proj	✓		✓		0.88
w/o Warp	✓	✓	✓		0.74
w/o Near	✓		✓	✓	0.77
w/o Proj	✓	✓		✓	0.76
Full Model	✓	✓	✓	✓	0.68

Table 3. Ablation studies on DTU dataset. The results show that all designs in our method are effective.

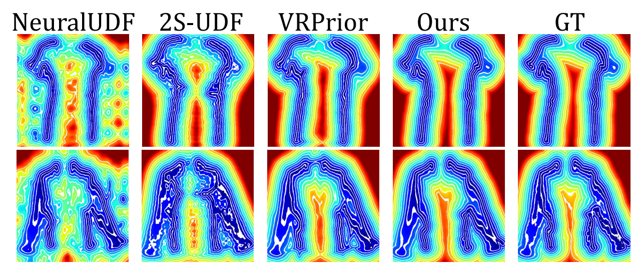

Figure 9. The learned UDFs for different methods. Our method learned more complete and smoother level sets in the field.

4.3. Visual Analysis in Unsigned Distance Fields

Visualization of Unsigned Distance Fields. We visualize the learned unsigned distance fields in Figure 9. We use the unsigned distances from UDFs learned by different methods and map these distances in colors. Points near the surface are close to blue, while points far from the surface are close to red. NeuralUDF [23] learns zero UDF values far from the surface, which increases the difficulty of convergence. 2S-UDF [9] learns a complex function close to the surface due to overfitting on textures. With the help of depth prior, VRPrior [49] learns better fields. However, it fails to capture the correct boundaries and almost closes the adjacent open surfaces. Our method learns the most accurate implicit functions without any extra prior.

Point Cloud Deformation. With the learned unsigned distance function, we can obtain the distance and the direction pointing to the surface for any point. Therefore, the UDF can deform source point clouds into the shape represented by the UDF. As shown in Figure 10, we gradually pull the input point clouds into the garments with Eq. (3), which validates that the implicit function has learned correct surface

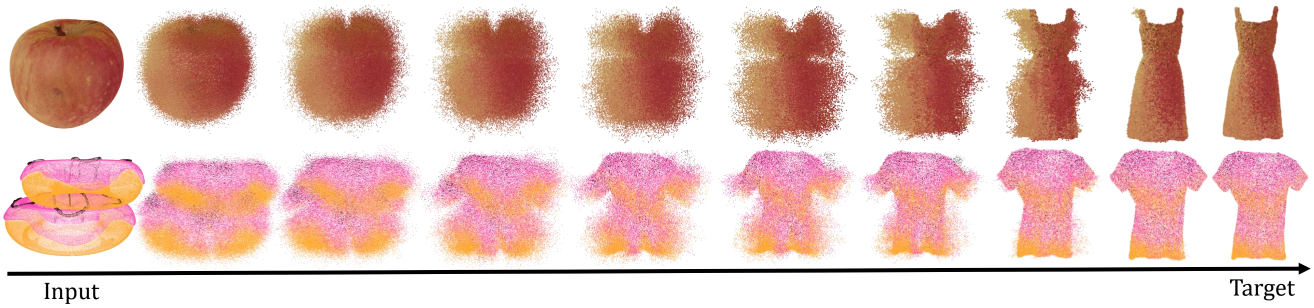

Figure 10. Point cloud deformation in the learned unsigned distance field. Accurate field can deform point clouds with any shapes (such as apple and donut) into the target shapes represented by the UDFs.

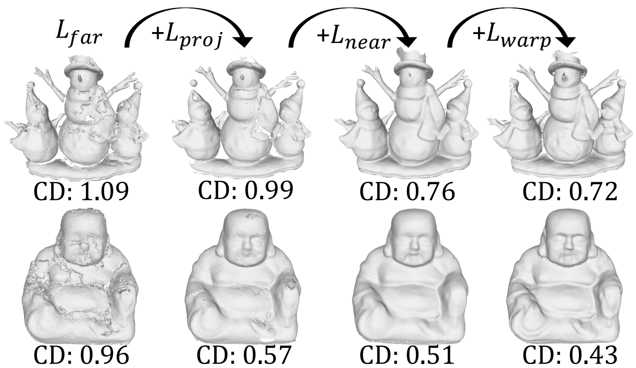

Figure 11. Visual changes for adding different constraints. The results show all components in our method are critical for our accurate surface reconstruction.

information at any point in space.

4.4. Ablation Studies

We conduct ablation studies on the DTU dataset [16] to show the impact of each module on the performance, and the full quantitative results are reported in Table 3.

Firstly, We try to learn the unsigned distance fields directly from the Gaussian point clouds, which is similar to the target of point cloud reconstruction [27, 51]. As shown by the row "Only Far" in Table 3, the performances drop significantly and the reason is that the point clouds of Gaussians are noisy, sparse and uneven, which cannot provide accurate geometry information. Overfitting a low-quality point cloud results in a poor surface, as shown in the first picture in Figure 11. We also combine the L_{far} with L_{proj} and L_{near} respectively. The results in "Far & Near" and "Far & Proj" show that both losses are critical for the accurate reconstruction and L_{near} plays a more important role.

To show how each loss affects our method, we remove the terms one by one and report the metrics as "w/o Near", "w/o Proj" and "w/o Warp". The results show that each loss plays a positive role in the final result, verifying the effectiveness of different parts of our method. Besides, removing the L_{near} loss leads to the largest drop in average metrics, which also proves that the self-supervision loss provides the most important information for learning UDF.

We gradually add different losses in the order of L_{far}, L_{proj}, L_{near}, and L_{warp}, and show the changes in results in Figure 11. Projecting Gaussians to the surface helps to learn a smooth surface, and self-supervision can fill the holes in the meshes. The warp loss captures more details. All loss terms contribute to more accurate surface reconstruction.

Limitations. Compared to SDF-based reconstruction methods, our approach demonstrates reduced performance in reconstructing textureless structures. This limitation arises from the high flexibility of UDF, which introduces complexities into the optimization process. Moreover, extracting surfaces from UDF fields is still an ongoing challenge [47, 51], which constrains the quality of the reconstructed open mesh. These factors result in a lack of detail in the surfaces reconstructed by our method, particularly for complex structures. In future work, incorporating additional priors, such as normals, masks, and depth, could help capture higher-frequency signals. Furthermore, integrating our approach with the latest UDF extraction methods [5, 47] may also enhance the quality of the reconstructed mesh.

5. Conclusion

We introduce an approach to reconstruct shapes with open surfaces and sharp boundaries from multi-view images with 3DGS. Our method can not only benefit from the high training efficiency of 3DGS, but also recover more accurate, complete, and continuous UDFs from discrete 3D Gaussians. The proposed constraints effectively overfit 3D Gaussians on surfaces, based on which our strategies for unsigned distance inference can recover more accurate unsigned distances. Our evaluations justify the effectiveness of each module, and show advantages over the latest methods in terms of accuracy, completeness, and sharpness on reconstructed open surfaces.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 2

[3] Chao Chen, Yu-Shen Liu, and Zhizhong Han. Inferring neural signed distance functions by overfitting on single noisy point clouds through finetuning data-driven based priors. In *Advances in Neural Information Processing Systems*, 2024. 2

[4] Hanlin Chen, Chen Li, and Gim Hee Lee. NeuSG: Neural implicit surface reconstruction with 3D Gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023. 2

[5] Zhiqin Chen, Andrea Tagliasacchi, Thomas Funkhouser, and Hao Zhang. Neural dual contouring. *ACM Trans. Graph.*, 41 (4):1–13, 2022. 2, 8

[6] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020. 2

[7] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using Gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[8] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6260–6269, 2022. 2, 5

[9] Junkai Deng, Fei Hou, Xuhui Chen, Wencheng Wang, and Ying He. 2S-UDF: A novel two-stage UDF learning method for robust non-watertight model reconstruction from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5084–5093, 2024. 1, 2, 3, 5, 6, 7

[10] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-NeuS: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 35:3403–3416, 2022. 2, 5

[11] Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned Gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024. 2

[12] Benoit Guillard, Federico Stella, and Pascal Fua. MeshUDF: Fast and differentiable meshing of unsigned distance field networks. In *Proceedings of the European Conference on Computer Vision*, pages 576–592. Springer, 2022. 2, 6

[13] Liang Han, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Binocular-guided 3D Gaussian splatting with view consistency for sparse view synthesis. In *Advances in Neural Information Processing Systems*, 2024. 2

[14] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 1, 2, 3, 4, 5, 6

[15] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a Diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 2

[16] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2014. 5, 6, 8

[17] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 339–346, 2002. 2

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 5

[19] Leif P Kobbelt, Mario Botsch, Ulrich Schwanecke, and Hans-Peter Seidel. Feature sensitive surface extraction from volume data. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 57–66, 2001. 2

[20] Tianyang Li, Xin Wen, Yu-Shen Liu, Hua Su, and Zhizhong Han. Learning deep implicit functions for 3D shapes with dynamic code clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12840–12850, 2022. 2

[21] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 2

[22] Yu-Tao Liu, Li Wang, Jie Yang, Weikai Chen, Xiaoxu Meng, Bo Yang, and Lin Gao. NeUDF: Leaning neural unsigned distance fields with volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 237–247, 2023. 1, 2, 6, 7

[23] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Yuan Liu, Peng Wang, Christian Theobalt, Taku Komura, and Wenping Wang. NeuralUDF: Learning unsigned distance fields for multi-view reconstruction of surfaces with arbitrary topologies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20834–20843, 2023. 1, 2, 5, 6, 7

[24] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 2

[25] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D

Gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2

[26] Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 3DGSR: Implicit surface reconstruction with 3D Gaussian splatting. *arXiv preprint arXiv:2404.00409*, 2024. 2

[27] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-Pull: Learning signed distance function from point clouds by learning to pull space onto surface. In *International Conference on Machine Learning*, pages 7246–7257. PMLR, 2021. 2, 3, 8

[28] Xiaoxu Meng, Weikai Chen, and Bo Yang. NeAT: Learning neural implicit surfaces with arbitrary topologies from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–258, 2023. 1, 2

[29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[30] 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 *Proceedings of the European Conference on Computer Vision*, pages 405–421. Springer, 2020. 1, 2, 5

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):1–15, 2022. 2

[32] Takeshi Noda, Chao Chen, Xinhai Liu Weiqi Zhang and, Yu-Shen Liu, and Zhizhong Han. MultiPull: Detailing signed distance functions by pulling multi-level queries at multi-step. In *Advances in Neural Information Processing Systems*, 2024. 2

[33] Michael Oechsle, Songyou Peng, and Andreas Geiger. UniSurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2

[34] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2

[35] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative Gaussian splatting for efficient 3D content creation. In *The Twelfth International Conference on Learning Representations*. 2

[36] 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. In *Advances in Neural Information Processing Systems*, pages 27171–27183, 2021. 2, 5, 6

[37] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. HF-NeuS: Improved surface reconstruction using high-frequency details. *Advances in Neural Information Processing Systems*, 35:1966–1978, 2022. 2

[38] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4D Gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 2

[39] Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. Recent advances in 3D Gaussian splatting. *Computational Visual Media*, 10(4): 613–642, 2024. 2

[40] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2

[41] Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. GIFS: Neural implicit function for general shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12829–12839, 2022. 2

[42] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2

[43] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[44] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. GSDF: 3DGS meets sdf for improved rendering and reconstruction. In *Advances in Neural Information Processing Systems*, 2024. 2

[45] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024. 2

[46] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Trans. Graph.*, 2024. 2, 5, 6, 7

[47] Congyi Zhang, Guying Lin, Lei Yang, Xin Li, Taku Komura, Scott Schaefer, John Keyser, and Wenping Wang. Surface extraction from neural unsigned distance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22531–22540, 2023. 2, 8

[48] Wenyuan Zhang, Yu-Shen Liu, and Zhizhong Han. Neural signed distance function inference through splatting 3D Gaussians pulled on zero-level set. In *Advances in Neural Information Processing Systems*, 2024. 2, 4

[49] Wenyuan Zhang, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Learning unsigned distance functions from multi-view images with volume rendering priors. In *Proceedings of the European Conference on Computer Vision*, 2024. 1, 2, 5, 6, 7

[50] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-aware unsigned dis-

tance functions progressively from raw point clouds. *Advances in Neural Information Processing Systems*, 35: 16481–16494, 2022. 2

[51] Junsheng Zhou, Baorui Ma, Shujuan Li, Yu-Shen Liu, Yi Fang, and Zhizhong Han. CAP-UDF: Learning unsigned distance functions progressively from raw point clouds with consistency-aware field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (01):1–18, 2024. 2, 8

[52] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep Fashion3D: A dataset and benchmark for 3D garment reconstruction from single images. In *Proceedings of the European Conference on Computer Vision*, pages 512–530. Springer, 2020. 5, 6