

NeUDF: Leaning Neural Unsigned Distance Fields with Volume Rendering

Yu-Tao Liu^{1,2} Li Wang^{1,2} Jie Yang¹ Weikai Chen³
Xiaoxu Meng³ Bo Yang³ Lin Gao^{1,2*}

¹Beijing Key Laboratory of Mobile Computing and Pervasive Device,
Institute of Computing Technology, Chinese Academy of Sciences

²University of Chinese Academy of Sciences

³Digital Content Technology Center, Tencent Games

liuyutao17@mails.ucas.ac.cn {wangli20s, yangjie01}@ict.ac.cn chenwk891@gmail.com
{xiaoxumeng, brandonyang}@global.tencent.com gaolin@ict.ac.cn

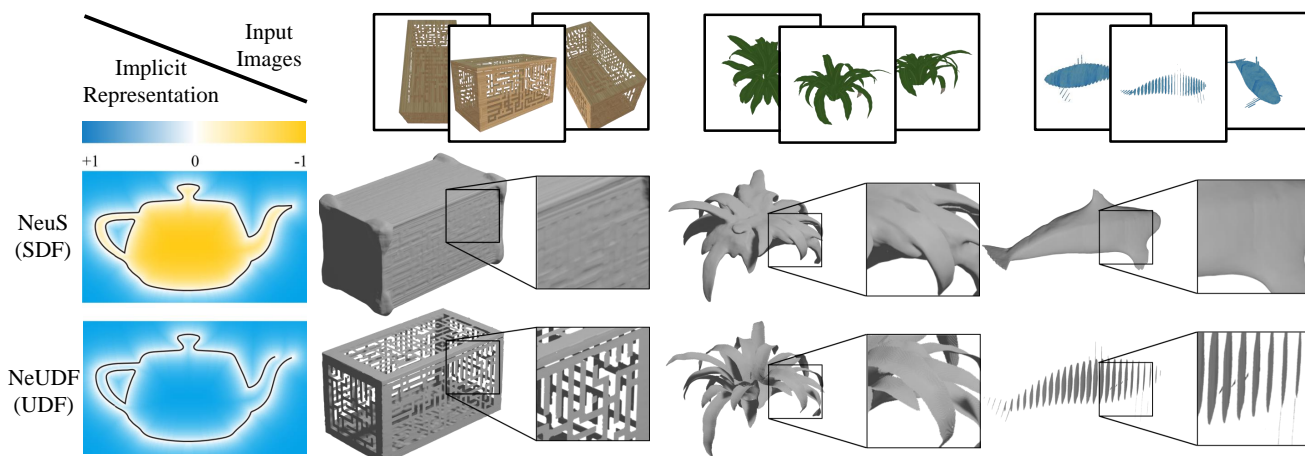


Figure 1. We show comparisons of the input multi-view images (top), watertight surfaces (middle) reconstructed with state-of-the-art SDF-based volume rendering method NeuS [53], and open surfaces (bottom) reconstructed with our method. Our method is capable of reconstructing high-fidelity shapes with both open and closed surfaces from multi-view images.

Abstract

Multi-view shape reconstruction has achieved impressive progresses thanks to the latest advances in neural implicit surface rendering. However, existing methods based on signed distance function (SDF) are limited to closed surfaces, failing to reconstruct a wide range of real-world objects that contain open-surface structures. In this work, we introduce a new neural rendering framework, coded NeUDF¹, that can reconstruct surfaces with arbitrary topologies solely from multi-view supervision. To gain the flexibility of representing arbitrary surfaces, NeUDF leverages the unsigned distance function (UDF) as surface representation. While a naive extension of an SDF-based neural renderer cannot scale to UDF, we propose two new formulations of weight function specially tailored for UDF-based volume rendering. Furthermore, to cope with open

surface rendering, where the in/out test is no longer valid, we present a dedicated normal regularization strategy to resolve the surface orientation ambiguity. We extensively evaluate our method over a number of challenging datasets, including DTU [21], MGN [5], and Deep Fashion 3D [61]. Experimental results demonstrate that NeUDF can significantly outperform the state-of-the-art method in the task of multi-view surface reconstruction, especially for complex shapes with open boundaries.

1. Introduction

Multi-view surface reconstruction is a long-standing and fundamental problem in computer vision and computer graphics. Conventional multi-view stereo based methods [43, 44] often underperform when the input images are sparse or appear textureless. Recent advances in neural implicit representation [9, 30, 32, 38] have brought impressive progress in achieving high-quality reconstruction

*Corresponding Author is Lin Gao (gaolin@ict.ac.cn).

¹Visit our project page at <http://geometrylearning.com/neudf/>

of intricate geometry even with sparse views. Specifically, they [13, 17, 28, 53, 54, 57, 62] leverage the volume rendering scheme to jointly learn the implicit geometry and color field by minimizing the discrepancy between the rendering results and the input images. However, since these methods represent surfaces using either signed distance function (SDF) [28, 53] or occupancy field [37], they can only reconstruct watertight shapes. This greatly limits their applications as shapes with open surfaces, such as garments, 3D-scanned scenes, *etc.*, are widely seen in the real world. Recent works, such as NDF [11], 3PSDF [8], and GIFS [59], have proposed new neural implicit functions to represent surfaces with arbitrary topologies. Nonetheless, none of these methods is compatible with existing neural rendering frameworks. Hence, how to leverage neural rendering to reconstruct non-watertight shapes, *e.g.* open surfaces, remains an open question.

We fill this gap by introducing NeUDF, a new volumetric rendering framework that can reconstruct shapes with arbitrary topologies only from multi-view image supervision. NeUDF is built upon the unsigned distance function (UDF), a straightforward implicit function that returns the absolute distance from a query point to the target surface. Despite its simplicity, we show that naively extending the SDF-based neural rendering mechanism to unsigned distance fields cannot ensure unbiased rendering of non-watertight surfaces. In particular, as shown in Figure 2, the SDF-based weighting function would generate spurious surfaces where the rendering weight triggers undesirable local maxima in the void region. To resolve this issue, we propose a new unbiased weighting paradigm specially tailored for UDF while being aware of surface occlusions. To accommodate the proposed weighting function, we further present a customized importance sampling strategy that ensure high-quality reconstruction of non-watertight surfaces. Furthermore, to tackle the inconsistent gradients of UDFs near the zero iso-surface, we introduce a normal regularization method to enhance the gradient consistency by leveraging normal information in the surface neighborhood.

To the best of our knowledge, NeUDF is the first attempt to reconstruct the surfaces with arbitrary topologies solely from 2D image supervision. Extensive experiments on the public datasets, *e.g.* MGN [5], Deep Fashion3D [61], and BMVS [56], demonstrate that NeUDF can significantly outperform the state-of-the-art methods in the task of open surface reconstruction while achieving comparable results in recovering watertight surfaces. We summarize our contributions as follows:

- The first UDF-based neural volume rendering framework, dubbed NeUDF, that can be used for multi-view reconstruction of shapes with arbitrary topologies, including complex shapes with open boundaries.

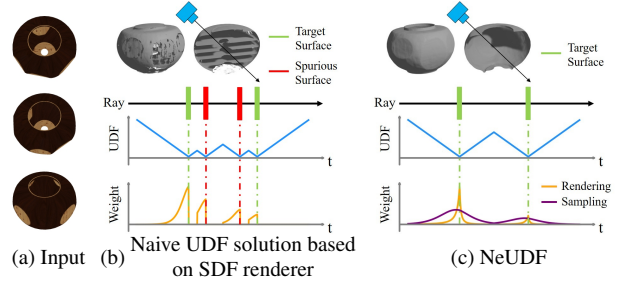


Figure 2. As shown in (b), the naive UDF solution based on the SDF renderer is biased, thus resulting in redundant surfaces in the reconstruction. NeUDF solves this problem by introducing a novel unbiased weighting mechanism as shown in (c).

- A novel unbiased weighting function and importance sampling strategy specially tailored for UDF rendering.
- The new state-of-the-art performance in the task of multi-view surface reconstruction over a number of challenging datasets with non-watertight 3D shapes.

2. Related Works

In this section, we first discuss classical implicit representations and neural rendering techniques. Next, we provide an overview of recent works on combining them to improve the performance of multi-view reconstruction tasks.

Neural Implicit Representation Recent developments in neural implicit representation [9, 31, 38, 41, 55] have surpassed the prior topological and resolution limit of explicit representations (*e.g.* point clouds, voxels, and meshes), setting a new state of the art for 3D modeling and reconstruction. Complex shapes can be implicitly represented by classifying the query points into inside or outside the shape (binary occupancy) [10, 12, 15, 18, 30, 39, 42] or predicting the signed distance (SDF) to the surface [9, 22, 31, 38, 41]. Due to the reliance on in/out the partition of 3D space, such methods can only model watertight objects. Methods based on unsigned distance function (UDF) [11, 50–52, 59] are proposed to overcome the limitation, enabling deep neural networks to properly represent and learn a much wider range of shapes with open surfaces. NDF [11] predicts an unsigned distance from an input query point and its position-aware shape feature which is encoded in a multi-scale manner. HSDF [52] simultaneously predicts a UDF field and a sign field to achieve better mesh fidelity. But they [11, 50–52, 59] require 3D supervision for mesh reconstruction.

Neural Rendering Besides the geometry information, appearance information is also needed to faithfully depict a scene, especially when the input observations take the

form of 2D pictures. Methods based on neural implicit surface rendering [24, 26, 27, 36, 47, 49, 58] find the intersection between a ray and the surface using differential sphere tracing [20] or its variants. They query the RGB color of the ray-surface intersection point using another network branch. Because the back-propagated gradients are influenced by the entire space, surface rendering methods like IDR [58] and DVR [36] struggle in reconstructing complex shapes without additional 2D mask supervision. In contrast, the methods based on neural volumetric rendering [29, 32, 34, 35, 40, 48, 60] imply that rather than a binary intersection case, rays can have a chance of interacting with the scene properties at every point in space. For machine learning pipelines that largely rely on the availability of well-behaved gradients for optimization, this continuous model performs well as a differentiable rendering framework.

Multi-view Reconstruction Multi-view stereo approaches [1, 6, 7, 14, 25, 44–46] before the advent of deep learning mainly rely on image feature matching [6, 44] across viewpoints or volumetric representation like voxel grids [1, 7, 14, 25, 46]. The former, like the widely used method COLMAP [44], highly relies on rich texture information and classic meshing techniques from point clouds because it computes multi-view depth maps from correspondence between images and fuses them into dense point clouds, while the latter is limited to low resolution due to the cubic memory growth of voxel representation.

Recent works [13, 17, 24, 27, 28, 36, 37, 53, 54, 57, 58] combining implicit representation and neural rendering outperform previous approaches in reconstructing watertight surfaces with high fidelity. Since these methods represent surfaces using either occupancy values [37] or signed distance function [28, 53] (SDF), their reconstruction results are limited to be watertight. Our NeUDF proposes a novel neural volume rendering algorithm for unsigned distance function (UDF) and thus can naturally extract the surface as the zero-level set of UDF, which is capable of representing complex shapes with open surfaces and thin structures.

3. Methodology

Given a set of calibrated images $\{\mathcal{I}_k | 1 \leq k \leq n\}$ of an object or scene, we aim to reconstruct arbitrary surfaces, including closed and open structures, only using 2D image supervision. In our paper, a surface is represented as a zero-level set of unsigned distance functions (UDFs). To learn the UDF representation of objects or scenes, we introduce a novel neural rendering architecture that incorporates unbiased formulation of weights for rendering. We first define our scene representation based on UDF (Sec. 3.1). Then we introduce NeUDF with two key formulations of weight function specially tailored for UDF-based volume rendering

(Sec. 3.2). Finally, we illustrate our normal regularization (Sec. 3.3) for alleviating the ambiguity from 2D images and our loss configuration (Sec. 3.4).

3.1. Scene Representation

Different from signed distance function (SDF), unsigned distance function (UDF) is sign-less and capable of representing open surfaces with arbitrary topologies, in addition to watertight surfaces. Given a 3D object $\mathcal{O} = \{V, F\}$, where V and F are the collections of vertices and faces, the UDF of an object \mathcal{O} can be formulated as a function $d = \Psi_{\mathcal{O}}(x) : \mathbb{R}^3 \mapsto \mathbb{R}^+$, which maps a point coordinate x to the Euclidean distance d to the surface. We define $\text{UDF}_{\mathcal{O}} = \{\Psi_{\mathcal{O}}(x) | d < \epsilon, d = \arg\min_{f \in F} (\|x - f\|_2)\}$, where ϵ is a small threshold, and the surface of the object can be modulated by the zero-level set of $\text{UDF}_{\mathcal{O}}$.

We introduce a differentiable volume rendering framework to predict UDF from input images. The framework is approximated by a neural network ψ , that predicts a UDF value d and the rendering color c according to a spatial location x along the sampling ray v :

$$(d, c) = \psi(v, x) : \mathbb{S}^2 \times \mathbb{R}^3 \mapsto (\mathbb{R}^+, [0, 1]^3) \quad (1)$$

With the help of volume rendering, the weights are optimized by minimizing the distance between the predicted images \mathcal{I}_k^l and ground-truths \mathcal{I}_k .

The learned surface $\mathcal{S}_{\mathcal{O}}$ can be represented by the zero-level set of the predicted UDF:

$$\mathcal{S}_{\mathcal{O}} = \{x \in \mathbb{R}^3 | d = 0, (d, c) = \psi(v, x)\} \quad (2)$$

3.2. NeUDF Rendering

Rendering procedure is the key to learning an accurate UDF as it connects the output color and the UDF value via integration along ray v :

$$C(o, v) = \int_0^{+\infty} w(t) c(p(t), v) dt, \quad (3)$$

where $C(o, v)$ is the output pixel color from the camera origin o along the view direction v , $w(t)$ is the weight function for the point $p(t)$, and $c(p(t), v)$ is the color at the point $p(t)$ along the view direction v .

To reconstruct UDFs via volume rendering, we first introduce a probability density function $\zeta_r'(\Psi(x))$, called *U-density*, where $\Psi(x)$ is the unsigned distance of x . The U-density function $\zeta_r'(\Psi(x))$ maps UDF field to a probability density distribution which assumes prominently high values near the surface for accurate reconstruction. Inspired by NeuS [53], we derive an unbiased and occlusion-aware weight function $w_r(t)$ and its opaque density $\tau_r(t)$ using

U-density function as:

$$w_r(t) = \tau_r(t) e^{-\int_0^t \tau_r(u) du} \quad (4)$$

$$\tau_r(t) = \left| \frac{\partial(\zeta_r \circ \Psi \circ p)(t)}{\partial t} \right| \quad (5)$$

where \circ is the function composition operator, and $\zeta_r(\cdot)$ must satisfy the following rules for valid UDF reconstruction:

$$\zeta_r(0) = 0, \lim_{d \rightarrow +\infty} \zeta_r(d) = 1 \quad (6)$$

$$\zeta'_r(d) > 0; \zeta''_r(d) < 0, \forall d > 0 \quad (7)$$

The $\zeta_r(d)$ can be any function shaped in the right figure. Since $\zeta_r(d)$ is the cumulative distribution function of U-density, $\zeta_r(0) = 0$ guarantees that there is no accumulated density from points with negative distances. Furthermore, $\zeta'_r(d) > 0$ and $\zeta''_r(d) < 0$ ensure U-density values are positive and prominently high for points near the surface. The parameter r in $\zeta_r(d)$ is learnable and controls the distribution of the density. This function structure addresses the volume-surface gap between the volume rendering and the surface reconstruction and guarantees global unbiased property. Please refer to our supplementary for detailed discussions.

We argue that a naive extension of SDF-based neural renderers would violate some of the above rules. For example, the cumulative distribution function of U-density in NeuS [53] is Φ_s (Sigmoid Function) and $\Phi_s(0) > 0$ violates Equ. 6. The violation would lead to bias in rendering weights and thus result in redundant floating faces and irregular noises shown in Fig. 2. Note that the local maximal constraint proposed in NeuS cannot address this rendering bias in UDF. Please check out the detailed discussion of the unbiased property and the global/local maximal constraint in our supplemental materials.

After extensive evaluations for different forms of $\zeta_r(d)$ in the ablation study (Sec. 4.3), we ultimately choose $\zeta_r(d) = \frac{rd}{1+rd}$ with r initialized to 0.05. Further, we adopt the α -compositing to discretize the weight function, which samples the points along the ray direction and accumulates the colors according to the weight integral. For the detailed discretization and proofs of the unbiased and the occlusion-aware properties of Eqn. 4 and Eqn. 5, please refer to our supplemental materials.

Importance points sampling. Points sampling that accommodates the rendering weight is an important step in

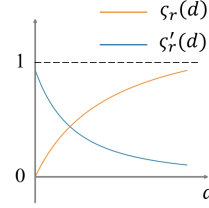


Figure 3. U-density function $\zeta'_r(d)$ and its cumulative distribution function $\zeta_r(d)$ satisfying the rules in Equ. 6 and Equ. 7.

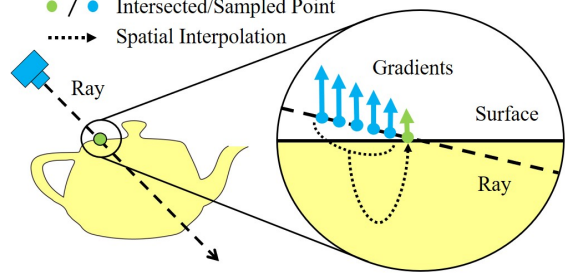


Figure 4. Normal Regularization Diagram. We use the gradients of points (in blue) with an offset from the surface to approximate the unstable surface normal (in green) of UDF representation.

volume rendering. Unlike SDF, to achieve unbiased rendering of UDF, the rendering function should distribute more weights before the intersection points (Fig. 2(c)). Hence, if both the rendering and sampling functions employ the same weights, the regularization (the Eikonal loss) on UDF gradients would lead to highly unbalanced gradient magnitudes on the two sides of the surface. This could significantly hamper the quality of the reconstructed UDF field. Therefore, we propose a specially-tailored sampling weight function (Fig. 2(c)) to achieve well-balanced regularization all over the space. The importance sampling $w_s(t)$ is formulated as follows:

$$w_s(t) = \tau_s(t) e^{-\int_0^t \tau_s(u) du}, \tau_s(t) = \zeta_s \circ \Psi \circ p(t), \quad (8)$$

where $\zeta_s(\cdot)$ satisfies the rules: $\zeta_s(d) > 0$ and $\zeta'_s(d) < 0, \forall d > 0$. Intuitively, $\zeta_s(\cdot)$ is a monotonically decreasing function in the first quadrant. In our paper, we use $\zeta_s(d) = \frac{se^{-sd}}{(1+e^{-sd})^2}$, where the parameter b in $\zeta_s(d)$ controls the intensity at $x = 0$. s starts from 0.05 and changes every sampling step z with the rate set to 2^{z-1} . Any sampling function that can achieve balanced regularization with the rendering function is compatible with our framework. For a detailed illustration of the above rules, please see our supplementary document. Further, we evaluate the necessity of the $\zeta_s(d)$ qualitatively and quantitatively in the ablation study (Sec. 4.3).

Overall, the weight functions are collaboratively used in rendering (Eqn. 4) and sampling (Eqn. 8) during volume rendering, which enables the high-fidelity open surface reconstruction with differentiable volume rendering.

3.3. Normal Regularization

Since points in UDF's zero-level set are cusps that are not first-order differentiable, the gradients of the sampled points in the vicinity of the learned surface are not numerically stable (jittered). As the rendering weight function takes as input the UDF gradient, unreliable gradients lead to inaccurate surface reconstruction. We introduce a nor-

mal regularization to perform spatial interpolation to alleviate this problem. The normal regularization replaces the naively sampled surface normal with an interpolated normal from its neighborhood. Figure 4 presents a detailed illustration. Since the unstable normal only exists near the surface, we use the point normal with an offset from the surface to approximate the unstable normal. We discretely formulate it at point $p(t_i)$ as follows:

$$\mathbf{n}(p(t_i)) = \frac{\sum_{k=1}^K w_{i-k} \Psi'(p(t_{i-k}))}{\sum_{k=1}^K w_{i-k}} \quad (9)$$

where $w_{i-k} = \|p(i) - p(i-k)\|_2^2$ is distance from $p(i)$ to $p(i-k)$. $\Psi'(\cdot)$ is the derivative of UDF $\Psi(\cdot)$, which returns the gradients of UDF. By leveraging normal regularization, our framework achieves smoother open surface reconstruction from the 2D images. We can adjust the normal regularization weight to obtain a more detailed geometry. Experiments show that normal regularization can prevent the highly bright and dark regions in 2D images from the high-quality reconstruction as shown in Fig. 10.

3.4. Training

To learn the high-fidelity open surface reconstruction, we optimize the network by minimizing the difference between the rendered images and groundtruth images with known camera poses, without any 3D supervision. Following NeuS [53], we also apply the three loss terms used in SDF volumetric rendering: Color loss \mathcal{L}_c , Eikonal loss [58] \mathcal{L}_e , and Mask loss \mathcal{L}_m . The color loss measures the difference between rendered image and input images under L1 loss. The Eikonal loss numerically regularizes the gradients of UDF on sampled points. If the masks are provided, the Mask loss also encourages the predicted mask to be close to the groundtruth mask under the BCE measurement. Overall, we use a loss that is composed of three parts:

$$\mathcal{L} = \mathcal{L}_c + \alpha \mathcal{L}_e + \beta \mathcal{L}_m \quad (10)$$

For detailed implementation and network architecture, please refer to our supplementary document.

4. Experiments & Evaluations

In this section, we validate NeUDF on multi-view reconstruction task qualitatively and quantitatively and further tested our method for real scenes. The experiments demonstrate that NeUDF outperforms the state-of-the-art techniques and can successfully reconstruct complex shapes with open boundaries. Lastly, we perform ablation studies and further discussions to demonstrate the importance of each key design.

4.1. Experimental Setup

Datasets. Since our method mainly focuses on open surface reconstruction under multi-view supervision, we

perform our experiments on three commonly used datasets, including Multi-Garment Net dataset (MGN) [4], Deep Fashion3D dataset (DF3D) [61], and DTU MVS dataset (DTU) [21]. For DTU MVS dataset, Each scene contains 49 or 64 images at 1600×1200 resolution and masks are from IDR [58]. And the DF3D and MGN contain some real-scanned garments with open boundaries, which are rendered as 200 colored images with 800×800 resolution for reconstruction. We respectively sampled 18 and 10 shapes from different categories for the two datasets. For the detailed camera poses, please refer to the supplementary document. Furthermore, we also collect some complex shapes with non-watertight structures² and rendered them to evaluate our framework. These shapes contain more intricate structures, which are composed of surfaces with open boundaries, *e.g.* plant leaves, and hollow structures (Fig. 1). Some datasets with diverse shapes (*e.g.* BMVS, Mixamo, and some real captured objects) are also tested.

Baselines. We compare NeUDF with several baselines for multi-view reconstruction task, including COLMAP [43, 44], IDR [58], NeuS [53], NeuralWarp [13], HF-NeuS [54]. COLMAP is a widely used MVS approach, where it reconstructs the point cloud from multi-vies images and extracts the explicit open surface by Ball-Pivoting Algorithm (BPA) [3]. IDR is the state-of-the-art surface rendering method, which can reconstruct high-quality meshes under mask supervision for training. NeuS is a pioneering work on surface reconstruction via SDF-based volume rendering, which achieves impressive results in surface reconstruction. The latest works, NeuralWarp, HF-NeuS achieve better performance for watertight shapes with improved high-frequency details or geometry consistency. However, they fail to model arbitrary surfaces with open boundaries. The straightforward solution mentioned in Sec. 1, which is a naive extension of NeuS renderer by adding an absolute operation on predicted SDF values and keeping all other configurations the same for UDF reconstruction, is also evaluated.

Metrics. To measure the accuracy of reconstructed shapes with regards to the ground truth, we adopt the commonly used metric – Chamfer Distance [2] (CD) for quantitative comparisons to state-of-the-art methods. We use a masked Poisson method for UDF mesh extraction, where we first sample one million points in the UDF and adopt SPSR [23] to extract a watertight mesh, and then mask out the spurious surfaces with non-zero UDF values. We scale all the meshes of different datasets into a unit sphere for a fair comparison. For the detailed calculation of the metrics, please refer to Fan *et al.* [16].

²<https://downloadfree3d.com/>, <https://archive3d.net/>

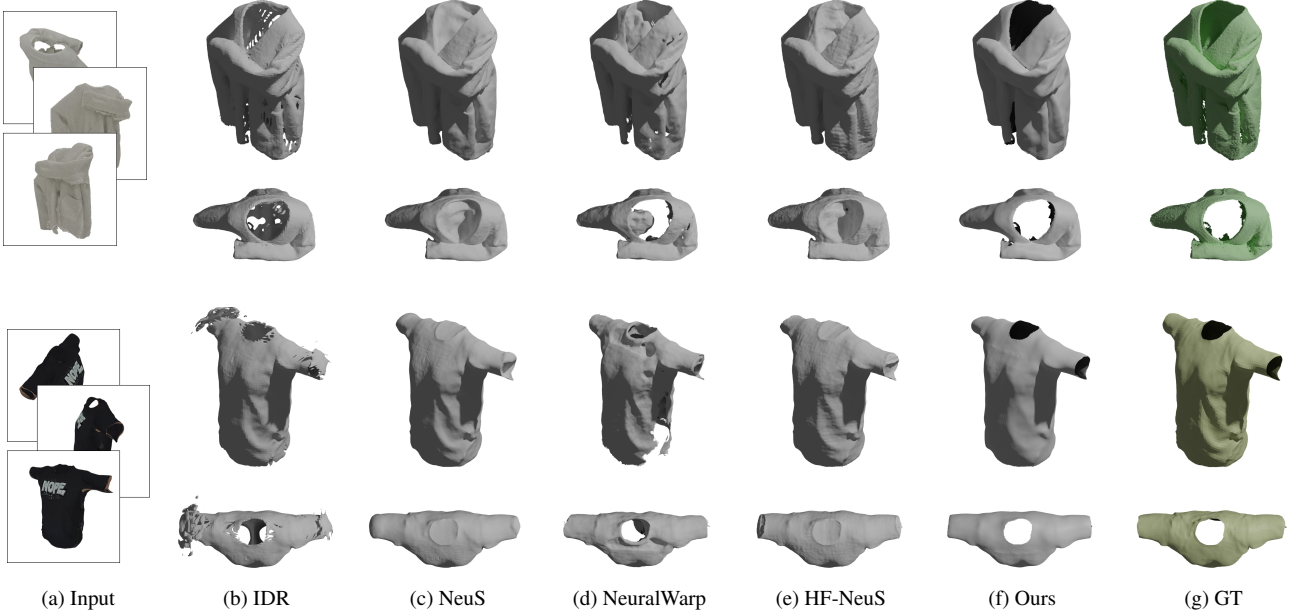


Figure 5. Qualitative comparison with IDR [58], NeuS [53], NeuralWarp [13] and HF-NeuS [54]. The GT of DF3D [61] data is point cloud (green) and the GT of MGN [5] data is open mesh (yellow). The back faces of the open surfaces are rendered in deep colors. The baselines are limited by the SDF representation and incorporate erroneous prior of closed surfaces. In contrast, our results can reconstruct complicated high-fidelity surfaces with open boundaries thanks to the UDF representation.

4.2. Comparisons on Multi-view Reconstruction

To demonstrate our reconstruction ability on diverse datasets (especially for the open surfaces), we perform quantitative and qualitative comparison to the SOTA methods on the above three datasets, including the open surface datasets varying in topology and geometry, as well as the watertight surface used in previous work. Note that IDR uses mask supervision for DTU [21], DF3D [61] and MGN [5] datasets, and ours uses mask supervision for DTU [21] dataset.

Quantitative Results. We report the average Chamfer Distance in Tab. 1. The results show that our method outperforms these baselines on the two open surface datasets (DF3D [61] and MGN [4]) by a large margin. Our method is the only one which is able to reconstruct high-fidelity open surfaces, while the baselines are subject to watertight shapes. For the watertight dataset (DTU [21]), our method is comparable with baselines. We also provide the evaluation of the naive extension of NeuS renderer. The naive extension results in a large Chamfer Distance on open surface samples (naive extension: 9.53 vs ours: **1.49**) due to the noisy surfaces and sometimes fails to converge (DTU_scan65).

Qualitative Results. The quantitative comparisons on the DF3D and MGN datasets are visualized in Fig. 5. As

Table 1. Quantitative comparison with the baselines on DF3D [61], MGN [4] and DTU [21] datasets. We split the open surface dataset (*i.e.* MGN, DF3D) into some sub-categories. For each dataset, we mark the evaluated number of scenes in the sub-categories. In this table, we report the average score of each category under the metric $\times 10^{-3}$. From the results, we can see that our method outperforms the IDR and NeuS on the two open surface datasets (MGN, DF3D) by a large margin.

DataSet	COLMAP	IDR	NeuS	NeuralWarp	HF-NeuS	Ours
MGN-upper (6)	12.32	19.68	11.65	15.40	9.16	6.78
MGN-pants (4)	30.62	23.70	17.95	22.26	24.02	16.43
DF3D-upper (6)	8.60	14.46	15.29	10.27	23.31	8.72
DF3D-pants (4)	25.91	16.91	16.00	7.99	12.29	5.77
DF3D-dress (8)	9.77	14.27	11.75	7.79	12.03	7.39
DTU (15)	3.75	4.92	4.46	3.78	5.60	4.98
Mean(open surface)	15.35	17.19	13.98	12.05	15.58	8.60
Mean(all)	11.31	12.91	10.80	9.16	12.10	7.34

shown in Fig. 5 (b) (c) (d) (e), the SDF-based methods (IDR [58], NeuS [53], NeuralWarp [13], HF-NeuS [54]) are subject to watertight shapes, and underperform with surfaces with open boundaries. In comparison, NeUDF can reconstruct high-fidelity meshes with open boundaries (such as the sleeves, collars and waists) without mask as shown in Fig. 5 (f).

We further conduct comparisons with NeuS [53] on the Mixamo [33] and BMVS [56] datasets. As shown in Figure 6 (*i.e.* mixamo-demon and bmvs-bear), our method is able to reconstruct geometries with open boundaries, such

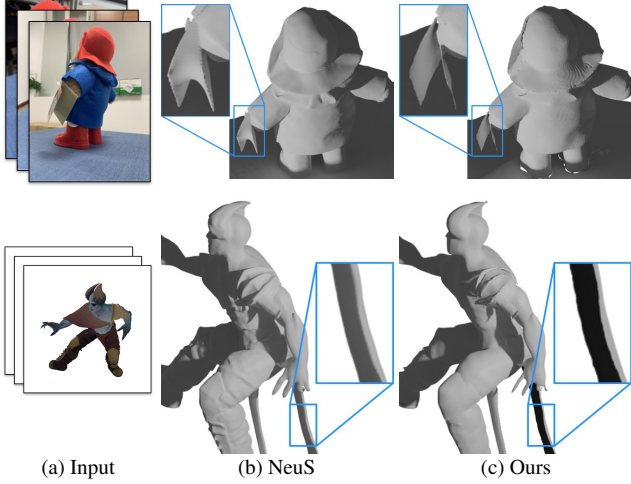


Figure 6. Comparisons with NeuS [53] on BMVS [56] dataset and Mixamo [33] dataset. Our NeUDF can reconstruct geometries with open boundaries (*e.g.* greeting card at the bear hand, clothes on the human character) while SDF-based method NeuS [53] is unable to properly represent. Further, our method also maintain comparable reconstruction quality for watertight parts such as bodies of the bear and the human character.

as the greeting card in the bear’s hand and the single-layer cloak. It is clear to observe that NeuS fails to synthesis the surface with open boundaries. In comparison, our reconstructed shape geometries are accurate, and the complex open structures are preserved. More qualitative results are presented in our supplementary document.

We additionally show some challenging cases with the complex structure in Fig. 1, such as the hollowed box, plant leaves, and patch-based fish. From the results on these objects with complex open boundaries, we see both detailed geometry and complex open-surface structures clearly, which validates that NeUDF learns a better UDF for multi-view reconstruction.

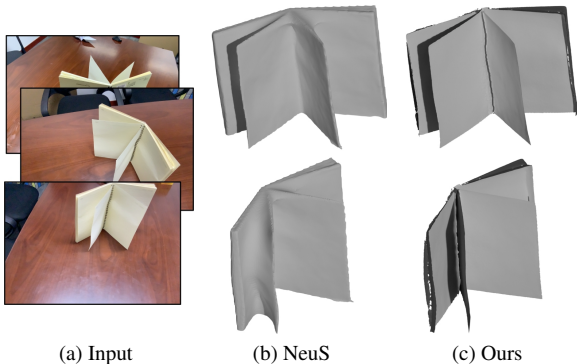


Figure 7. The evaluation on the captured real scenes. We conduct a comparison with NeuS on the real scenes. From the results, we can see that our method is capable of complex surface reconstruction with open boundaries, while NeuS encourages to merge the book pages together.

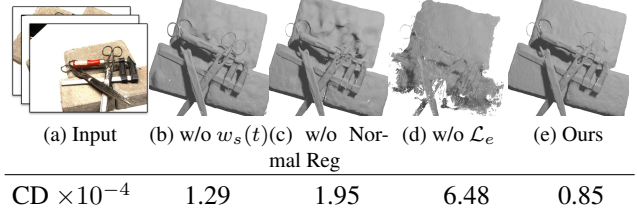


Figure 8. Ablation of different components in our NeUDF. (a) the input multi-view images; (b) w/o our importance points sampling, namely adopting color weight function for sampling instead; (c) w/o normal regularization; (d) w/o Eikonal loss; (e) Ours with full components. From all the results, it is clearly observed that all key designs are critical for our high-fidelity complex surface reconstruction.

Captured Real Scenes. We further evaluate our method on the captured data from real-world objects, including book pages, fan blades and plant leaves. For each scene, we use the mobile phone to capture a video surrounding the object and extract about 200 frames from the video. Then, we use COLMAP to estimate the camera poses and take the calibrated images as input to optimize the network parameters without mask supervision. Fig. 7 presents the reconstructed shape of book pages, and more captured real scenes are presented in our supplementary document. The results show that NeuS encourages merging the book pages together and causing unrealistic geometry, while ours achieves accurate surface reconstruction with open boundaries.

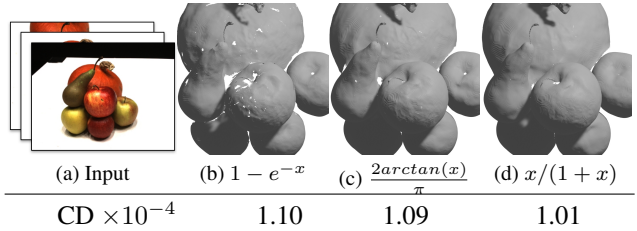


Figure 9. Ablation on different choice of ς_r in τ_r . According to the given rules (Eqn. 6, Eqn. 7), we have evaluated some similar function (*e.g.*, $1 - e^{-x}$, $2\arctan(x)/(\pi)$, $x/(1+x)$). From the results, we find the different ς_r only affects the speed of converge. The reconstructed shapes are similar until converge. In the figure, we present the visual results when ours converges. We can see that other ablated versions do not converge with several holes.

4.3. Further Discussions and Analysis

We conduct three ablation studies to validate our individual designs of our methods. First, evaluating different choices of ς_r in τ_r shows its effectiveness for UDF learning. Then, we also validate the necessity of our designed importance sampling and normal regularization for the accurate open surface reconstruction. All the ablation studies are conducted on multiple sample objects with diverse shapes.

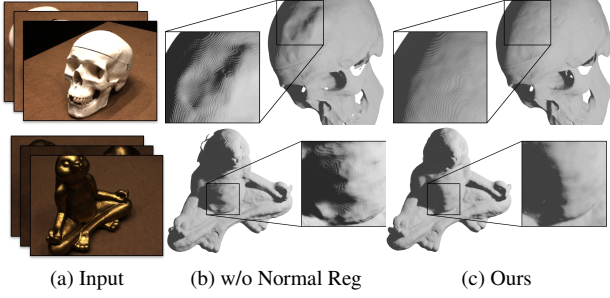


Figure 10. Evaluations on two extreme cases for effectiveness of normal regularization. (b): Without normal regularization, artifacts can be observed on the reconstructed skull with high lights and metal rabbit with dark shadow. (c): Our full method with normal regularization can produce more visually pleasing results.

The choice of ς_r in τ_r . Although we have given the rules (Eqn. 6, Eqn. 7) that ς_r should satisfy, there is a family of functions that satisfy the rules. All the functions in the family are suitable for UDF volume rendering, so we conduct validations on several different candidate functions to check the convergence aptitude of each function for network optimization, *i.e.*, with which function the network converges to the best results in a given training iterations. Fig 9 shows the visual results of three candidate functions ($1 - e^{-x}$, $\frac{2\arctan(x)}{\pi}$ and $\frac{x}{1+x}$) that follow the rules. After the given iterations (300k), the network using the function $\frac{x}{1+x}$ converges to the best result both qualitatively and quantitatively, while the other functions are not fully convergent and cause incomplete surfaces and slightly higher Chamfer-Distances. Evaluation on diverse shapes also demonstrates that all the functions work well and the chosen one ($\frac{x}{1+x}$) works the best (ours: **1.11** vs candidates: 1.13/1.18) in our setting.

Necessity of Importance Points Sampling $w_s(t)$ (Eqn. 8). To demonstrate the necessity of Eqn. 8, we design an ablated version that removes the importance of point sampling and uses the Eqn. 4 to sample the points for training. Fig. 8 (b) shows that the output surfaces using the same weight for both rendering and sampling are less smooth and with larger CD errors as expected. The errors come from the sampling points distribution is not balanced on both sides of the surface, and the network is not well regularized. Fig. 8 (e) shows that the network is well regularized with the importance sampling and produces better results.

Necessity of Normal Regularization. We use the Normal Regularization (Sec 3.3) to address the unstable gradients at the zero level set of UDF, and we conduct a validation on the necessity of the Normal Regularization. As shown in Fig. 8(c), without the Normal Regularization the result suffers from rough surfaces and large Chamfer distance due to unstable gradient calculating. Further, the normal regularization

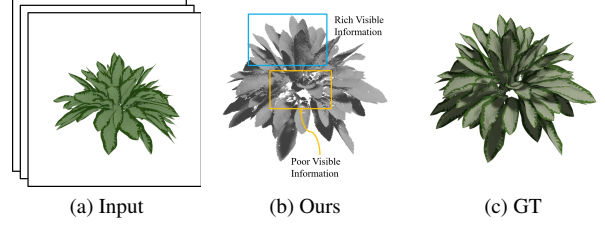


Figure 11. Failure Case. The severely occluded parts in this plant can't be well reconstructed.

benefits the surface reconstruction even with extreme cases, *e.g.* extremely bright or dark regions. Figure 10 shows two cases in extremely bright and dark light conditions. The visualization results indicate that normal regularization is critical to alleviating the geometric error induced by the ambiguity of light conditions (*e.g.* the artifacts on the DTU-skull and the DTU-metal-rabbit).

5. Discussions & Conclusions

Limitation. While our method can successfully reconstruct arbitrary surfaces with open boundaries, it still has several limitations. First, it is difficult to model transparent surfaces with our formulation. The reconstruction quality degrades when there are not enough visible information from input images (*e.g.* sparse in viewpoints or severely occluded) and an example of failure cases is given in Figure 11. There also exist trade-offs between the smoothness and high-frequency details due to the normal regularization, since it accumulates the vicinity information to alleviate the surface normal ambiguity. Further, since we introduce UDF for better representation ability, we need additional meshing tools like MeshUDF [19] or SPSR [23] which may introduce more reconstruction errors.

Conclusions. We propose NeUDF, a novel UDF-based volume rendering approach to achieve high-fidelity multi-view reconstruction for arbitrary shapes with both open and closed surfaces from 2D images with or without masks. NeUDF outperforms the state-of-the-art methods both qualitatively and quantitatively, especially on complex surfaces with open boundaries. Therefore, our NeUDF can play a crucial role in real-world 3D applications. In future work, we can extend our formulation for better reconstruction of transparent surfaces. Enhancing our NeUDF to support sparse input images is also an interesting future direction.

Acknowledgment

This work was supported by CCF-Tencent Open Fund, the Beijing Municipal Natural Science Foundation for Distinguished Young Scholars (No. JQ21013), the National Natural Science Foundation of China (No. 62061136007) and the Youth Innovation Promotion Association CAS.

References

- [1] Motilal Agrawal and Larry S. Davis. A probabilistic framework for surface reconstruction from multiple images. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 470–476. IEEE Computer Society, 2001.
- [2] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings: Image Understanding Workshop*, pages 21–27, 1977.
- [3] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.
- [4] Bharat Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5419–5429, 2019.
- [5] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 5419–5429. IEEE, 2019.
- [6] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In Jesse Hoey, Stephen J. McKenna, and Emanuele Trucco, editors, *British Machine Vision Conference, BMVC 2011, Dundee, UK, August 29 - September 2, 2011. Proceedings*, pages 1–11. BMVA Press, 2011.
- [7] Adrian Broadhurst, Tom Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), Vancouver, British Columbia, Canada, July 7-14, 2001 - Volume 1*, pages 388–393. IEEE Computer Society, 2001.
- [8] Weikai Chen, Cheng Lin, Weiyang Li, and Bo Yang. 3psdf: Three-pole signed distance function for learning surfaces with arbitrary topologies. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022.
- [9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.
- [10] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020.
- [11] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [12] Julian Chibane and Gerard Pons-Moll. Implicit feature networks for texture completion from partial 3d data. In *European Conference on Computer Vision*, pages 717–725. Springer, 2020.
- [13] 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.
- [14] Jeremy S De Bonet and Paul Viola. Voxels: Probabilistic voxelized volume reconstruction. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 2, 1999.
- [15] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 612–628. Springer, 2020.
- [16] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.
- [17] 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 (NeurIPS)*, 2022.
- [18] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas A Funkhouser. Deep structured implicit functions. 2019.
- [19] Benoît Guillard, Federico Stella, and Pascal Fua. Meshudf: Fast and differentiable meshing of unsigned distance field networks. *CoRR*, abs/2111.14549, 2021.
- [20] John C. Hart. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. *Vis. Comput.*, 12(10):527–545, 1996.
- [21] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.
- [22] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.
- [23] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.
- [24] Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4287–4297. Computer Vision Foundation / IEEE, 2021.

- [25] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. In *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*, pages 307–314. IEEE Computer Society, 1999.
- [26] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8293–8304, 2019.
- [27] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2016–2025. Computer Vision Foundation / IEEE, 2020.
- [28] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. *arXiv preprint arXiv:2206.05737*, 2022.
- [29] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 7210–7219. Computer Vision Foundation / IEEE, 2021.
- [30] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.
- [31] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv preprint arXiv:1901.06802*, 2019.
- [32] 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 Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2020.
- [33] Adobe’s Mixamo. Mixamo: Animated 3d characters. <https://www.mixamo.com>.
- [34] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. *Comput. Graph. Forum*, 40(4):45–59, 2021.
- [35] Michael Niemeyer and Andreas Geiger. GIRAFFE: representing scenes as compositional generative neural feature fields. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 11453–11464. Computer Vision Foundation / IEEE, 2021.
- [36] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3501–3512. Computer Vision Foundation / IEEE, 2020.
- [37] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5569–5579. IEEE, 2021.
- [38] 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 Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.
- [39] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European conference on computer vision (ECCV 2020)*, 2020.
- [40] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10318–10327. Computer Vision Foundation / IEEE, 2021.
- [41] Edoardo Remelli, Artem Lukoianov, Stephan R Richter, Benoît Guillard, Timur Bagautdinov, Pierre Baque, and Pascal Fua. Meshsdf: Differentiable iso-surface extraction. *arXiv preprint arXiv:2006.03997*, 2020.
- [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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019.
- [43] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4104–4113. IEEE Computer Society, 2016.
- [44] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, volume 9907 of *Lecture Notes in Computer Science*, pages 501–518. Springer, 2016.
- [45] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and*

- Pattern Recognition (CVPR 2006)*, 17-22 June 2006, New York, NY, USA, pages 519–528. IEEE Computer Society, 2006.
- [46] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, June 17-19, 1997, San Juan, Puerto Rico, pages 1067–1073. IEEE Computer Society, 1997.
 - [47] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1119–1130, 2019.
 - [48] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 7495–7504. Computer Vision Foundation / IEEE, 2021.
 - [49] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles T. Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 11358–11367. Computer Vision Foundation / IEEE, 2021.
 - [50] Rahul Venkatesh, Tejan Karmali, Sarthak Sharma, Aurobrata Ghosh, R. Venkatesh Babu, Laszlo A. Jeni, and Maneesh Singh. Deep implicit surface point prediction networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12653–12662, October 2021.
 - [51] Rahul Venkatesh, Sarthak Sharma, Aurobrata Ghosh, Laszlo Jeni, and Maneesh Singh. Dude: Deep unsigned distance embeddings for hi-fidelity representation of complex 3d surfaces. *arXiv preprint arXiv:2011.02570*, 2020.
 - [52] Li Wang, Jie Yang, Wei-Kai Chen, Xiao-Xu Meng, Bo Yang, Jin-Tao Li, and Lin Gao. Hsdf: Hybrid sign and distance field for modeling surfaces with arbitrary topologies. In *Neural Information Processing Systems (NeurIPS)*, 2022.
 - [53] 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 Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27171–27183, 2021.
 - [54] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. Hf-neus: Improved surface reconstruction using high-frequency details. *arXiv preprint arXiv:2206.07850*, 2022.
 - [55] Yujie Wang, Yixin Zhuang, Yunzhe Liu, and Baoquan Chen. Mdisn: Learning multiscale deformed implicit fields from single images. *Visual Informatics*, 6(2):41–49, 2022.
 - [56] 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. *Computer Vision and Pattern Recognition (CVPR)*, 2020.
 - [57] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
 - [58] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
 - [59] Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. GIFS: neural implicit function for general shape representation. *CoRR*, abs/2204.07126, 2022.
 - [60] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *CoRR*, abs/2010.07492, 2020.
 - [61] 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 *European Conference on Computer Vision*, pages 512–530. Springer, 2020.
 - [62] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022.

Appendices

A. Overview

In the main paper, we introduce a novel UDF-based volume rendering approach to achieve high-fidelity multi-view reconstruction for arbitrary shapes with both open and closed surfaces. This supplemental material consists of detailed proofs, implementation details and additional results of multi-view reconstruction. All the sections are organized as follows:

- Section B analyzes the inherent bias in color rendering of the naive UDF solution based on SDF renderer.
- Section C provides detailed proofs of the unbiased and occlusion-aware properties of our proposed NeUDF.
- Section D provides implementation details on network architecture (Section D.1), training details (Section D.2) and data preparation (Section D.3).
- Section E provides additional qualitative results of multi-view reconstruction.

B. Bias in Naive UDF solution based on SDF renderer

In this section we illustrate the bias of color rendering introduced by the naive UDF solution based on SDF renderer, which directly extends the weight of NeuS to UDF. The bias causes inherent geometric error like redundant surfaces and floating noises.

To apply the naive UDF solution based on the SDF renderer of NeuS, we denote the rendered color $C(o, v)$:

$$C(o, v) = \int_0^{+\infty} w_n(t) c(p(t), v) dt, \quad (11)$$

where (o, v) are the origin and view direction of the sample ray, $c(x, v)$ the color at position x along the view direction v , and $w_n(t)$ the rendering weight of NeuS:

$$w_n(t) = \rho_s(t) e^{-\int_0^t \rho_s(u) du} \quad (12)$$

$$\rho_s(t) = \max\left\{-\frac{\partial(\Phi_s \circ \Psi \circ p)}{\partial t}(t), 0\right\} \quad (13)$$

where $\rho_s(t)$ denotes the opaque density of NeuS, $\Phi_s(d)$ the Sigmoid function, and $\Psi(x)$ the UDF value at position x . The learnable parameter s controls the distribution of the Sigmoid function, which is expected to increase to infinity during training.

Assume that the ray linearly crosses the open surface in its local neighbor, *e.g.*, there exists an interval (t^l, t^r) , the intersection point $t^* \in (t^l, t^r)$, which satisfies:

$$\Psi \circ p(t) = |\cos \theta| \cdot |t - t^*|, \forall t \in (t^l, t^r), \quad (14)$$

where θ is the angle between the view direction and the surface normal.

In UDF, the color $C(o, v)$ rendered based on SDF renderer, Equ. 11, consists of inherent bias and inconsistency of the geometry. Denote the first intersection point t_0^* and its corresponding interval (t_0^l, t_0^r) , the bias can be formularized as below:

$$\lim_{s \rightarrow \infty} C(o, v) = 0.5c(p(t_0^*), v) + \frac{2^k - 1}{2^{k+1}} c_m + \frac{1}{2^{k+1}} c_n, \quad (15)$$

where k is the number of intersection points along the ray, c_m the undesired mixture of colors from invisible surfaces and c_n the colors from floating noise induced by the rendering bias. The parameter s decides the weight distribution of colors along the ray, and is supposed to increase towards infinity during training.

Note that the weight distribution corresponding to Equ. 15 satisfies the local maximal constraint discussed in NeuS, *i.e.* the weight attains local maxima at each intersection point (locally unbiased). But the local maximal constraint is not sufficient for an unbiased rendering for open surfaces due to the volume-surface representation discrepancy. The volume rendering relies on the volume-level color fusion for optimization, while the ground-truth color is exactly the surface color at the intersection point of the sample ray and the first intersected surface. A self-consistent rendering procedure should be able to address this volume-surface discrepancy, *i.e.* the color fusion range should be limited as close to the first intersection point as possible (globally unbiased). Otherwise the network is not able to converge to a surface representation through volume rendering. Note that the weight of NeuS is globally and locally unbiased for SDF, but not globally unbiased for UDF, and this difference comes from the difference of the value domains of SDF and UDF.

To illustrate the detailed causation of c_m and c_n , we first prove that:

$$\lim_{s \rightarrow \infty} \int_0^{t_0^l} w_n(t) dt = 0 \quad (16)$$

$$\lim_{s \rightarrow \infty} \int_{t_0^l}^{t_0^r} w_n(t) dt = 0.5, \quad (17)$$

which means that the output color consists of undesired bias whose weight sums to 0.5, and the bias cannot be corrected through training. Then we show the detailed distribution of the bias c_m and c_n for corroboration.

Proof of Equ. 16. Specifically, to prove Equ. 16, we have:

$$\begin{aligned}
& \int_0^{t_0^l} w_n(t) dt \\
&= \int_0^{t_0^l} \rho_s(t) e^{-\int_0^t \rho_s(u) du} dt \\
&= \int_0^{t_0^l} -\frac{\partial}{\partial t} e^{-\int_0^t \rho_s(u) du} dt \\
&= -e^{-\int_0^{t_0^l} \rho_s(u) du} \Big|_0^{t_0^l} \\
&= -e^{-\int_0^{t_0^l} \rho_s(u) du} + 1 \\
&= -e^{-\int_0^{t_0^l} \max\{\frac{\partial(\Phi_s \circ \Psi \circ p)}{\partial u}(u), 0\} du} + 1
\end{aligned} \tag{18}$$

It follows that:

$$\begin{aligned}
& \int_0^{t_0^l} w_n(t) dt \\
&\leq -e^{-\int_0^{t_0^l} \left| \frac{\partial(\Phi_s \circ \Psi \circ p)}{\partial u}(u) \right| du} + 1 \\
&= -e^{-\int_0^{t_0^l} \left| \frac{\partial \Phi_s \circ \Psi \circ p(u)}{\partial \Psi \circ p(u)} \cdot \frac{\partial \Psi \circ p(u)}{\partial u} \right| du} + 1 \\
&= -e^{-\int_0^{t_0^l} \left| \frac{\Phi'_s \circ \Psi \circ p(u)}{\Phi_s \circ \Psi \circ p(u)} \cdot \frac{\partial \Psi \circ p(u)}{\partial u} \right| du} + 1 \\
&= -e^{-\int_0^{t_0^l} \frac{|\Phi'_s \circ \Psi \circ p(u)| \cdot \left| \frac{\partial \Psi \circ p(u)}{\partial u} \right|}{|\Phi_s \circ \Psi \circ p(u)|} du} + 1
\end{aligned} \tag{19}$$

Denote that:

$$A = |\Phi'_s \circ \Psi \circ p(u)| \tag{20}$$

$$B = \left| \frac{\partial \Psi \circ p(u)}{\partial u} \right| \tag{21}$$

$$C = |\Phi_s \circ \Psi \circ p(u)| \tag{22}$$

We have:

$$\int_0^{t_0^l} w_n(t) dt = -e^{-\int_0^{t_0^l} \frac{A \cdot B}{C} du} + 1 \tag{23}$$

Because t_0^* is the first zero point of $\Psi \circ p(t)$ and $\Psi(x)$ is a continuous function, there is:

$$\exists \Psi_{min} > 0, s.t., \Psi \circ p(t) > \Psi_{min}, \forall t \in (0, t_0^l).$$

Note that $\Phi_s(x)$ is the Sigmoid function $\Phi_s(x) = (1 + e^{-s \cdot x})^{-1}$, and $\frac{\partial \Psi \circ p(u)}{\partial u}$ is the gradient of the UDF along the

ray. We have:

$$C = |\Phi_s \circ \Psi \circ p(u)| \tag{24}$$

$$= (1 + e^{-s \cdot \Psi \circ p(u)})^{-1} \tag{25}$$

$$> (1 + e^{-s \cdot \Psi_{min}})^{-1} \tag{26}$$

$$> 0.5 \tag{27}$$

$$B = \left| \frac{\partial \Psi \circ p(u)}{\partial u} \right| < 1 \tag{28}$$

and $\forall \epsilon > 0, \exists S = \max\{1, \frac{-4t_0^l}{\ln(1-\epsilon) \cdot \Psi_{min}^2}\}, s.t., \forall s > S$, there is:

$$\begin{aligned}
A &= |\Phi'_s \circ \Psi \circ p(u)| \\
&= \frac{s \cdot e^{-s \cdot \Psi \circ p(t)}}{(1 + s \cdot e^{-s \cdot \Psi \circ p(t)})^2} \\
&\leq \frac{2}{\Psi^2 \circ p(t) \cdot s} \\
&\leq \frac{2}{\Psi_{min}^2 \cdot s} \\
&\leq \frac{2}{\Psi_{min}^2 \cdot \frac{-4t_0^l}{\ln(1-\epsilon) \cdot \Psi_{min}^2}} \\
&= \frac{-0.5 \ln(1-\epsilon)}{t_0^l}
\end{aligned} \tag{29}$$

It follows that:

$$\begin{aligned}
\int_0^{t_0^l} w_n(t) dt &= -e^{-\int_0^{t_0^l} \frac{A \cdot B}{C} du} + 1 \\
&< -e^{-\int_0^{t_0^l} \frac{0.5 \ln(1-\epsilon) \cdot 1}{t_0^l \cdot 0.5} du} + 1 \\
&= -e^{-\int_0^{t_0^l} \frac{\ln(1-\epsilon)}{t_0^l} du} + 1 \\
&= -e^{-t_0^l \cdot \frac{\ln(1-\epsilon)}{t_0^l}} + 1 \\
&= -e^{\ln(1-\epsilon)} + 1 \\
&= -(1-\epsilon) + 1 \\
&= \epsilon
\end{aligned} \tag{30}$$

This leads to:

$$\begin{aligned}
& \lim_{s \rightarrow \infty} \int_0^{t_0^l} w_n(t) dt \\
&= \lim_{s \rightarrow \infty} (-e^{-\int_0^{t_0^l} \frac{A \cdot B}{C} du} + 1) \\
&= 0
\end{aligned} \tag{31}$$

The Equ. 31 means that the weight before the first intersection of the ray converges against zero during training, so the output color composites no color before the first intersected surface. This completes the proof of Equ. 16.

Proof of Equ. 17. Then we give the proof of Equ. 17. Same as the derivation of Equ. 18, we have:

$$\begin{aligned}
& \int_{t_0^l}^{t_0^*} w_n(t) dt \\
&= \int_{t_0^*}^{t_0^*} \rho_s(t) e^{-\int_0^t \rho_s(u) du} dt \\
&= \int_{t_0^l}^{t_0^*} -\frac{\partial}{\partial t} e^{-\int_0^t \rho_s(u) du} dt \\
&= -e^{-\int_0^{t_0^*} \rho_s(u) du} \Big|_{t_0^l}^{t_0^*} \\
&= -e^{-\int_0^{t_0^*} \rho_s(u) du} + e^{-\int_0^{t_0^l} \rho_s(u) du} \\
&= -e^{-\int_0^{t_0^*} \rho_s(u) du - \int_{t_0^l}^{t_0^*} \rho_s(u) du} + e^{-\int_0^{t_0^l} \rho_s(u) du} \\
&= e^{-\int_0^{t_0^l} \rho_s(u) du} (-e^{-\int_{t_0^l}^{t_0^*} \rho_s(u) du} + 1)
\end{aligned} \tag{32}$$

Note that when $t \in (t_0^l, t_0^*)$, we have:

$$\frac{\partial \Psi \circ p(t)}{\partial t} = -|\cos \theta|. \tag{33}$$

It follows that:

$$\begin{aligned}
& -e^{-\int_{t_0^l}^{t_0^*} \rho_s(u) du} + 1 \\
&= -e^{-\int_{t_0^l}^{t_0^*} \max\left\{\frac{\partial(\Phi_s \circ \Psi \circ p)(u)}{\partial u}, 0\right\} du} + 1 \\
&= -e^{-\int_{t_0^l}^{t_0^*} \left|\frac{\partial(\Phi_s \circ \Psi \circ p)(u)}{\partial u}\right| du} + 1 \\
&= -e^{-\int_{t_0^l}^{t_0^*} \left|\frac{\partial}{\partial u} \ln \Phi_s \circ \Psi \circ p(u)\right| du} + 1 \\
&= -e^{-\int_{t_0^l}^{t_0^*} -\frac{\partial}{\partial u} \ln \Phi_s \circ \Psi \circ p(u) du} + 1 \\
&= -e^{\ln \Phi_s \circ \Psi \circ p(t_0^*) - \ln \Phi_s \circ \Psi \circ p(t_0^l)} + 1 \\
&= -\frac{e^{\ln \Phi_s \circ \Psi \circ p(t_0^*)}}{e^{\ln \Phi_s \circ \Psi \circ p(t_0^l)}} + 1 \\
&= -\frac{\Phi_s \circ \Psi \circ p(t_0^*)}{\Phi_s \circ \Psi \circ p(t_0^l)} + 1
\end{aligned} \tag{34}$$

Since t_0^* is the intersection point, we have $\Psi \circ p(t_0^*) = 0$ and $\Phi_s \circ \Psi \circ p(t_0^*) = 0.5$. It follows that:

$$\begin{aligned}
& -e^{-\int_{t_0^l}^{t_0^*} \rho_s(u) du} + 1 \\
&= -\frac{\Phi_s \circ \Psi \circ p(t_0^*)}{\Phi_s \circ \Psi \circ p(t_0^l)} + 1 \\
&= -\frac{0.5}{\Phi_s \circ \Psi \circ p(t_0^l)} + 1 \\
&\leq -\frac{0.5}{1} + 1 = 0.5
\end{aligned} \tag{35}$$

$$\forall \epsilon > 0, \exists S = \frac{-\ln 2\epsilon}{\Psi \circ p(t_0^l)}, s.t., \forall s > S,$$

$$\begin{aligned}
& -e^{-\int_{t_0^l}^{t_0^*} \rho_s(u) du} + 1 \\
&= -\frac{0.5}{\Phi_s \circ \Psi \circ p(t_0^l)} + 1 \\
&= -\frac{0.5}{(1 + e^{-s \cdot \Psi \circ p(t_0^l)})^{-1}} + 1 \\
&\geq -\frac{0.5}{(1 + e^{-\frac{-\ln 2\epsilon}{\Psi \circ p(t_0^l)} \Psi \circ p(t_0^l)})^{-1}} + 1 \\
&= -\frac{0.5}{(1 + e^{\ln 2\epsilon})^{-1}} + 1 \\
&= -\frac{0.5}{(1 + 2\epsilon)^{-1}} + 1 \\
&= -\epsilon + 0.5
\end{aligned} \tag{36}$$

The Equ. 35 and 36 derive that:

$$\lim_{s \rightarrow \infty} (-e^{-\int_{t_0^l}^{t_0^*} \rho_s(u) du} + 1) = 0.5 \tag{37}$$

It has been proved in Equ. 30 that:

$$\lim_{s \rightarrow \infty} (-e^{-\int_0^{t_0^l} \rho(u) du} + 1) = 0, i.e., \tag{38}$$

$$\lim_{s \rightarrow \infty} (e^{-\int_0^{t_0^l} \rho(u) du} - 1) = 1 \tag{39}$$

The equations 32, 37 and 39 together derive that:

$$\begin{aligned}
& \lim_{s \rightarrow \infty} \int_{t_0^l}^{t_0^*} w_n(t) dt \\
&= \lim_{s \rightarrow \infty} (e^{-\int_0^{t_0^l} \rho_s(u) du} (-e^{-\int_{t_0^l}^{t_0^*} \rho_s(u) du} + 1)) \\
&= 0.5
\end{aligned} \tag{40}$$

The Equ. 40 determines that the rendered color $C(o, v)$ of NeuS in UDF cannot converge to the ground-truth color $c(p(t_0^*), v)$ as up to half of the weight is not constrained, which causes the mixed rendering color with undesired bias and inherent geometric error. This completes the proof of Equ. 17.

Distribution of Bias. Further, we illustrate the components of the bias, e.g., c_m and c_n , and show the corresponding distribution.

For $t \in (t_0^*, t_1^*)$, where t_0^* and t_1^* denotes the first and second intersection points along the ray $p(t)$. Consider that:

$$\begin{aligned}
w_n(t) &= \rho_s(t) e^{-\int_0^t \rho_s(u) du} \\
&= \rho_s(t) e^{-\int_{t_0^*}^t \rho_s(u) du} \cdot e^{-\int_0^{t_0^*} \rho_s(u) du}
\end{aligned} \tag{41}$$

As is proved, $\lim_{s \rightarrow \infty} e^{-\int_0^{t_0^*} \rho_s(u) du} = 0.5$, there is:

$$w_n(t_1^*) = 0.5 \rho_s(t) e^{-\int_{t_0^*}^{t_1^*} \rho_s(u) du} \quad (42)$$

According to the assumption that $\exists(t_1^l, t_1^r) \ni t_1^*$, the UDF value $\Psi(t)$ along the ray is linear for $t \in (t_1^l, t_1^r)$. So similarly we can prove that:

$$\begin{aligned} & \lim_{s \rightarrow \infty} \int_0^{t_1^*} w_n(t) dt \\ &= 0.5 \lim_{s \rightarrow \infty} \int_0^{t_1^*} \rho_s(t) e^{-\int_{t_0^*}^t \rho_s(u) du} dt \\ &= 0.25 \end{aligned} \quad (43)$$

Consequently, for any given $k > 0$, we have:

$$\lim_{s \rightarrow \infty} \int_0^{t_k^*} w_n(t) dt = \frac{1}{2^{k+1}} \quad (44)$$

The colors of the k invisible surfaces are mixed to the output color $C(o, v)$, whose weight sums to $\frac{2^k - 1}{2^{k+1}}$. The mixed colors integral c_m leads to the undesired bias $\frac{2^k - 1}{2^{k+1}} c_m$, which cannot be corrected during training. The last weight $1 - 0.5 - \frac{2^k - 1}{2^{k+1}} = \frac{1}{2^{k+1}}$ comes from the disturbance besides the neighborhood of surfaces, and leads to new redundant surfaces during training. The bias c_m and c_n case inherent geometric error like redundant surfaces and floating noises in invisible space.

C. Proofs of Unbiased and Occlusion-aware properties of NeUDF

In this subsection we illustrate the capability of NeUDF for UDF learning from three aspects. First we show that different from NeuS, NeUDF avoids the c_m and c_n which cause the biased rendering color and inherent geometric error in UDF. Then we give the proofs of the unbiased and occlusion-aware properties of NeUDF respectively.

C.1. Avoidance of c_m and c_n in NeUDF.

Before providing the detailed proofs of the unbiased and occlusion-aware properties of NeUDF, we briefly show that NeUDF is free from the undesired colors c_m and c_n by introducing the new rendering weight function:

$$w_r(t) = \tau_r(t) e^{-\int_0^t \tau_r(u) du}, \quad (45)$$

$$\tau_r(t) = \left| \frac{\frac{\partial \varsigma_r \circ \Psi \circ p}{\partial t}(t)}{\varsigma_r \circ \Psi \circ p(t)} \right|, \quad (46)$$

where $\varsigma_r(d)$ satisfies that:

$$\varsigma_r(0) = 0, \lim_{d \rightarrow \infty} \varsigma_r(d) = 1, \quad (47)$$

$$\forall d > 0, \varsigma_r'(d) > 0, \varsigma_r''(d) < 0. \quad (48)$$

Similar to the derivation in B, there is:

$$\lim_{r \rightarrow \infty} \int_0^{t_0^l} w_r(t) dt = 0 \quad (49)$$

and

$$\begin{aligned} & \lim_{r \rightarrow \infty} \int_{t_0^l}^{t_0^*} w_n(t) dt \\ &= \lim_{r \rightarrow \infty} e^{-\int_0^{t_0^l} \tau_r(u) du} (-e^{-\int_{t_0^l}^{t_0^*} \tau_r(u) du} + 1) \\ &= \lim_{r \rightarrow \infty} -e^{-\int_{t_0^l}^{t_0^*} \tau_r(u) du} + 1 \end{aligned} \quad (50)$$

When $t \in (t_0^l, t_0^r)$, there is:

$$\frac{\partial \Psi \circ p(t)}{\partial t} = -|\cos \theta| < 0 \quad (51)$$

We have:

$$\begin{aligned} & -e^{-\int_{t_0^l}^{t_0^*} \tau_r(u) du} + 1 \\ &= -e^{-\int_{t_0^l}^{t_0^*} \left| \frac{\frac{\partial \varsigma_r \circ \Psi \circ p}{\partial u}(u)}{\varsigma_r \circ \Psi \circ p(u)} \right| du} + 1 \\ &= -e^{-\int_{t_0^l}^{t_0^*} \left| \frac{\partial}{\partial u} \ln \varsigma_r \circ \Psi \circ p(u) \right| du} + 1 \\ &= -e^{\int_{t_0^l}^{t_0^*} \frac{\partial}{\partial u} \ln \varsigma_r \circ \Psi \circ p(u) du} + 1 \\ &= -e^{\ln \varsigma_r \circ \Psi \circ p(t_0^*) - \ln \varsigma_r \circ \Psi \circ p(t_0^l)} + 1 \\ &= -\frac{\varsigma_r \circ \Psi \circ p(t_0^*)}{\varsigma_r \circ \Psi \circ p(t_0^l)} + 1 \\ &= -0 + 1 \\ &= 1 \end{aligned} \quad (52)$$

So we have:

$$\lim_{r \rightarrow \infty} \int_{t_0^l}^{t_0^*} w_n(t) dt = 1 \quad (53)$$

It follows that:

$$\begin{aligned} \lim_{r \rightarrow \infty} C(o, v) &= \lim_{r \rightarrow \infty} \int_{t_0^l}^{t_0^*} w_n(t) dt \cdot c(p(t_0^*), v) \\ &\quad + (1 - \lim_{r \rightarrow \infty} \int_{t_0^l}^{t_0^*} w_n(t) dt) \cdot c_m \\ &= c(p(t_0^*), v) \end{aligned} \quad (54)$$

It indicates that NeUDF avoids the limitation introduced by the undesired mixture c_m (and c_n). The detailed proof of unbiased property of NeUDF is provided in the next section.

C.2. Proof of Unbiased Property in NeUDF.

Intuitively, the rendering weight function should be unbiased, *i.e.*, more contribution should come from the intersection point than its neighbor. In this subsection we prove that NeUDF is unbiased:

- Given the ray $p(t)$ and the UDF $\Psi(x)$, the weight of rendering $w_r(t)$ in NeUDF attains a locally maximum value at a intersection point t^* .

Assume that the weight $w_r(t)$ is a linear function within the local neighborhood (t^l, t^r) of the zero point $t^* \in (t^l, t^r)$. We consider the intervals (t^l, t^*) and (t^*, t^r) respectively. For $t \in (t^l, t^*)$, we have:

$$\begin{aligned}
w_r(t) &= \tau_r(t) e^{-\int_0^t \tau(u) du} \\
&= \tau_r(t) e^{-\int_0^{t^l} \tau(u) du} e^{-\int_{t^l}^t \tau(u) du} \\
&= \tau_r(t) e^{-\int_0^{t^l} \tau(u) du} e^{-\int_{t^l}^t \left| \frac{\partial \varsigma_r \circ \Psi \circ p(u)}{\partial u} \right| du} \\
&= \tau_r(t) e^{-\int_0^{t^l} \tau(u) du} e^{-\int_{t^l}^t \left| \frac{\partial}{\partial u} \ln \varsigma_r \circ \Psi \circ p(u) \right| du} \\
&= \tau_r(t) e^{-\int_0^{t^l} \tau(u) du} e^{-\int_{t^l}^t \frac{\partial}{\partial u} \ln \varsigma_r \circ \Psi \circ p(u) du} \\
&= \tau_r(t) e^{-\int_0^{t^l} \tau(u) du} e^{\ln \varsigma_r \circ \Psi \circ p(t) - \ln \varsigma_r \circ \Psi \circ p(t^l)} \\
&= \tau_r(t) e^{-\int_0^{t^l} \tau(u) du} \frac{e^{\ln \varsigma_r \circ \Psi \circ p(t)}}{e^{\ln \varsigma_r \circ \Psi \circ p(t^l)}} \\
&= \tau_r(t) e^{-\int_0^{t^l} \tau(u) du} \frac{\varsigma_r \circ \Psi \circ p(t)}{\varsigma_r \circ \Psi \circ p(t^l)} \\
&= \left| \frac{\partial \varsigma_r \circ \Psi \circ p(u)}{\partial u} \right| \frac{e^{-\int_0^{t^l} \tau(u) du} \varsigma_r \circ \Psi \circ p(t)}{\varsigma_r \circ \Psi \circ p(t^l)} \\
&= \frac{\left| \frac{\partial \varsigma_r \circ \Psi \circ p(t)}{\partial u} \right| \cdot \left| \frac{\partial \Psi \circ p(t)}{\partial t} \right|}{|\varsigma_r \circ \Psi \circ p(t)|} e^{-\int_0^{t^l} \tau(u) du} \frac{\varsigma_r \circ \Psi \circ p(t)}{\varsigma_r \circ \Psi \circ p(t^l)} \\
&= \frac{|\varsigma'_r \circ \Psi \circ p(t)| \cdot |\cos \theta|}{|\varsigma_r \circ \Psi \circ p(t)|} e^{-\int_0^{t^l} \tau(u) du} \frac{\varsigma_r \circ \Psi \circ p(t)}{\varsigma_r \circ \Psi \circ p(t^l)} \\
&= \frac{\varsigma'_r \circ \Psi \circ p(t) \cdot |\cos \theta|}{\varsigma_r \circ \Psi \circ p(t)} e^{-\int_0^{t^l} \tau(u) du} \frac{\varsigma_r \circ \Psi \circ p(t)}{\varsigma_r \circ \Psi \circ p(t^l)} \\
&= \frac{\varsigma'_r \circ \Psi \circ p(t) \cdot |\cos \theta| \cdot e^{-\int_0^{t^l} \tau(u) du}}{\varsigma_r \circ \Psi \circ p(t^l)}
\end{aligned} \tag{55}$$

For a given parameter r , $\varsigma_r \circ \Psi \circ p(t^l)$, $e^{-\int_0^{t^l} \tau_r(u) du}$ and $|\cos \theta|$ are all constant. So we have:

$$w_r(t) = A \cdot \varsigma'_r \circ \Psi \circ p(t), A = \frac{|\cos \theta| \cdot e^{-\int_0^{t^l} \tau_r(u) du}}{\varsigma_r \circ \Psi \circ p(t^l)}, \tag{56}$$

where A is a fixed positive number for any given r .

Note that $\varsigma'_r(d) > 0$, $\varsigma''_r(d) < 0$, it follows that:

$$w_r(t_1) > w_r(t_2), \forall t_1 > t_2, t_1, t_2 \in (t^l, t^*). \tag{57}$$

For $t \in (t^*, t^r)$, we have:

$$\tau_r(t) = \left| \frac{\partial \varsigma_r \circ \Psi \circ p(t)}{\partial t} \right| = \frac{\varsigma'_r \circ \Psi \circ p(t) \cdot |\cos \theta|}{\varsigma_r \circ \Psi \circ p(t)} \tag{58}$$

$\forall t_1 > t_2, t_1, t_2 \in (t^*, t^r)$, there is:

$$\tau_r(t_1) < \tau_r(t_2) \tag{59}$$

$$e^{-\int_0^{t_1} \tau_r(u) du} < e^{-\int_0^{t_2} \tau_r(u) du} \tag{60}$$

It follows that:

$$w_r(t_1) < w_r(t_2), \forall t_1 > t_2, t_1, t_2 \in (t^l, t^*). \tag{61}$$

The Equ. 57 and 61 indicates that the point closer to the zero point is with higher weight value. Note that the proof does not require a strict zero point t^* , *i.e.*, the property holds true when there is a small perturbation Δ to the zero point t^* : $\Psi \circ p(t^*) = \Delta > 0$.

Empirically, the zero point of the UDF is encoded as a small positive number, so the weight function $w_r(t)$ is continuous along the ray. Therefore we have:

$$w_r(t^*) > w_r(t), \forall t \in (t^l, t^r), t \neq t^* \tag{62}$$

This completes the proof.

C.3. Proof of Occlusion-aware Property in NeUDF.

In this subsection we prove that NeUDF is occlusion-aware. Intuitively, for two parts of the sample ray with the same UDF value, we hope that more contribution of the output colors is from the part closer to the camera. That is, the closer surfaces are more likely to have higher weight.

Specifically, given two surfaces S_1 and S_2 such that S_1 is closer to the camera, for two corresponding points $p(t_1)$ and $p(t_2)$ with the same UDF value, we have:

$$\int_{t_1}^{t_1+\delta} w_r(t) dd_1(t) > \int_{t_2}^{t_2+\delta} w_r(t) dd_2(t), \tag{63}$$

where $d_i(t)$ denotes the distance between the location $p(t)$ and the surface S_i , and δ denotes the small step length.

$$\tau_r(t) = \left| \frac{\partial \varsigma_r \circ \Psi \circ p(t)}{\partial t} \right| = \frac{|\varsigma'_r \circ \Psi \circ p(t)| \cdot |\cos \theta|}{\varsigma_r \circ \Psi \circ p(t)} \tag{64}$$

For $t_1 < t_2$, $\Psi(t_1) = \Psi(t_2)$, $w_r(t_1), w_r(t_2) > 0$, we have:

$$\frac{\tau_r(t_1)}{|\cos \theta_1|} = \frac{|\varsigma'_r \circ \Psi \circ p(t_1)|}{\varsigma_r \circ \Psi \circ p(t_1)} = \frac{|\varsigma'_r \circ \Psi \circ p(t_2)|}{\varsigma_r \circ \Psi \circ p(t_2)} = \frac{\tau_r(t_2)}{|\cos \theta_2|} \tag{65}$$

$$e^{-\int_0^{t_1} \tau_r(u) du} > e^{-\int_0^{t_2} \tau_r(u) du} \quad (66)$$

There is:

$$\begin{aligned} \frac{w_r(t_1)}{|\cos \theta|} &= \frac{\tau_r(t_1) e^{-\int_0^{t_1} \tau_r(u) du}}{|\cos \theta|} \\ &> \frac{\tau_r(t_2) e^{-\int_0^{t_2} \tau_r(u) du}}{|\cos \theta|} = \frac{w_r(t_2)}{|\cos \theta|} \end{aligned} \quad (67)$$

It follows that:

$$\int_{t_1}^{t_1+\delta} w_r(t) dd_1(t) = \int_{t_1}^{t_1+\delta} \frac{w_r(t)}{|\cos \theta|} dt \quad (68)$$

$$\int_{t_2}^{t_2+\delta} w_r(t) dd_2(t) = \int_{t_2}^{t_2+\delta} \frac{w_r(t)}{|\cos \theta|} dt \quad (69)$$

$$\int_{t_1}^{t_1+\delta} w_r(t) dd_1(t) > \int_{t_2}^{t_2+\delta} w_r(t) dd_2(t), \quad (70)$$

where $d_i(t)$ denotes the distance between the location $p(t_i)$ and the surface S_i .

The Equ. 70 indicates that the cumulative weight near the first intersected surface are higher than the second one. This means that more concentration are on the former surface. Note that no prior assumption of the existence of other intersected surfaces is required, *i.e.*, the property of occlusion-aware holds true for more than two surface intersections along the ray. This completes the proof of the occlusion-aware property.

D. Implementation Details

D.1. Network Architecture

Similar to IDR [53] and NeuS [53], we use two MLP networks to respectively encode the UDF and the color. The input of the UDF network is the spatial location $p(t)$ and the output is the corresponding UDF value along with a 256-dimensional feature vector. The UDF network $\Psi(x)$ consists of 8 hidden layers with hidden size of 256, and the activation function is chosen as the Softplus with $\beta = 100$ for all hidden layers and the output layer. A skip connection is also used to connect the input with the output of the fourth layer. The inputs of the color network are the spatial location $p(t)$, the view direction v , the gradient n of the UDF network at the spatial location $p(t)$ and the corresponding feature vector derived by the UDF network. The color network $c(x, v)$ consists of 4 hidden layers with hidden size of 256. Normal regularization is applied before the gradient n of the UDF network is used as the input of the color network. Same positional encoding and weight normalization are adopted as in NeuS.

D.2. Training Details

Discretization. We adopt the α -compositing to discretize the weight function, which divides the sample ray into bins by sampling n points $p(t_i) = o + t_i |i = 1, \dots, n, t_i < t_{i+1}$ and accumulate colors within each bin according to the weight integral:

$$\begin{aligned} \alpha_i &= 1 - e^{-\int_{t_i}^{t_{i+1}} \tau_r(t) dt} \\ &= \frac{|\zeta_r \circ \Psi \circ p(t_i) - \zeta_r \circ \Psi \circ p(t_{i+1})|}{\zeta_r \circ \Psi \circ p(t_i)}. \end{aligned} \quad (71)$$

We slightly modify Equ. 71 by:

$$\alpha_i = \frac{\zeta_i^{max} - \zeta_i^{min}}{\zeta_i^{max}}, \quad (72)$$

where ζ_i^{max} and ζ_i^{min} is the maximum and minimum of the set $\{\zeta_r \circ \Psi \circ p(t_i), \zeta_r \circ \Psi \circ p(t_{i+1})\}$.

Up Sampling. We first formally sample 64 points per ray, and then hierarchically conduct importance sampling on top of the sampling weight $w_s(t)$ for another 64 points:

$$w_s(t) = \tau_s(t) e^{-\int_0^t \tau_s(u) du}, \tau_s(t) = \zeta_s \circ \Psi \circ p(t) \quad (73)$$

And $\zeta_s(\cdot)$ satisfies the rules: $\zeta_s(d) > 0$ and $\zeta'_s(d) < 0, \forall d > 0$. Intuitively, the $\tau_s(t)$ derived by the monotonically decreasing function is a view-invariant sampling density, and the density has positive correlation with the UDF value. To derive the sampling weight $w_s(t)$, the classical volume rendering scheme is applied.

The weight of the i^{th} sample point $w_s(t_i)$ is slightly modified by:

$$w'_s(t_i) = \max\{w_s(t_{i+k}), k = -1, 0, 1\} \quad (74)$$

And then the weight $w'_s(t)$ is normalized so that the integral equals to one:

$$w''_s(t) = \frac{w'_s(t)}{\sum_{i=0}^{n-1} w'_s(t_i)} \quad (75)$$

For each iteration we hierarchically conduct the importance sampling for two times, and each time 32 points are sampled. The total number of sampling points are 128. If no masks are provided, 32 points are randomly sampled in addition outside the unit sphere per ray to represent the outside scene. The outside scene is represented with NeRF++ [60], as used in NeuS [53].

Platform. The network is trained with ADAM optimizer, and the learning rate warms up to 2×10^{-4} in the first 5k iterations, and decreases to 1×10^{-5} by the end of training. For each iteration, 512 random rays are sampled from 8 input camera poses randomly selected. We train each model for 400k iterations in total for 9 hours for the setting of with mask, and 11 hours for the setting of without mask on a single Nvidia 3090 GPU.

D.3. Data Preparation

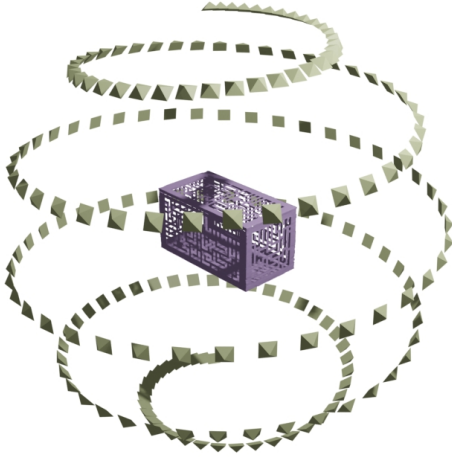


Figure 12. Poses of the camera. The camera poses are represented as the yellow pyramid, and the object to reconstruct is represented in purple.

Rendered Data. To generate the customized data, we use the pyrender package to render images from the ground-truth objects. We rendered 200 views at 800×800 pixels for each textured mesh or colored point cloud. Fig 12 visualizes the camera poses. Corresponding masks with black background are provided optionally. Only the rendered images and the masks are used as inputs of the network.

Captured Data. We additionally captured several real-world objects using the mobile phone. The captured images are extracted from the captured videos around the object. For the book object we captured 200 images at the resolution of 1920×1440 . For the fan object we captured 59 images at the resolution of 3456×4608 . For the plant object we captured 200 images at the resolution of 720×1280 . All the camera poses are estimated by COLMAP [43,44] and no masks are provided.

E. Additional Results

We visualize more reconstruction results of NeUDF on DF3D [61], MGN [4], DTU [21], BMVS [56] datasets and real-captured data. Fig. 13 shows the comparison with NeuS on the DF3D dataset without mask supervision. Fig. 14 shows the comparison with NeuS on the DF3D dataset with mask supervision. Fig. 15 shows the comparison with NeuS on the MGN dataset without mask supervision. Fig. 16 shows the comparison with NeuS on the MGN dataset with mask supervision. Fig. 17 shows the comparison with NeuS on the DTU and BMVS datasets with mask supervision. Fig. 18 shows the additional results of the real-captured scenes with open surfaces.

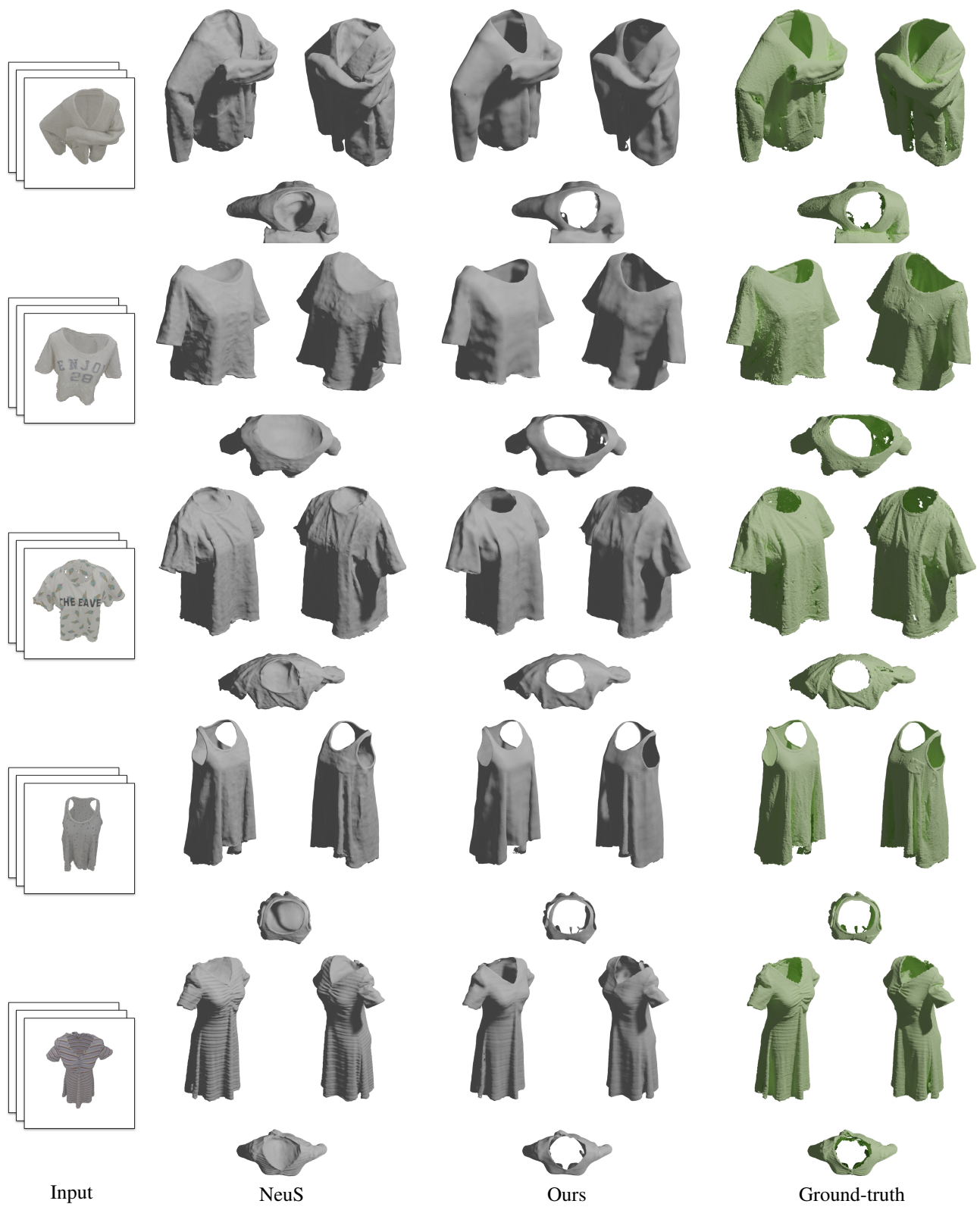


Figure 13. Additional results on the DF3D [61] dataset without mask supervision.



Figure 14. Additional results on the DF3D [61] dataset with mask supervision.

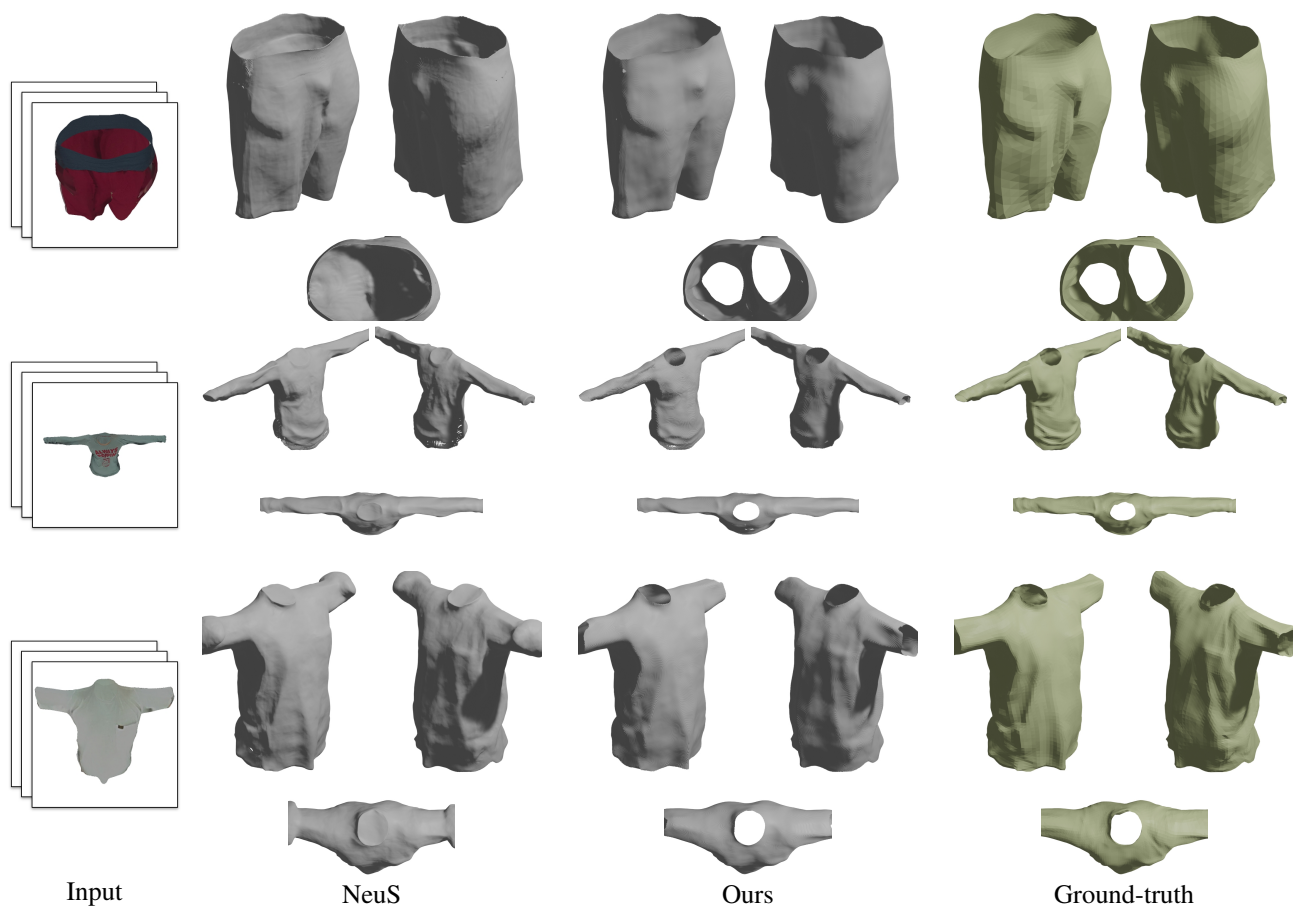


Figure 15. Additional results on the MGN [4] dataset without mask supervision.

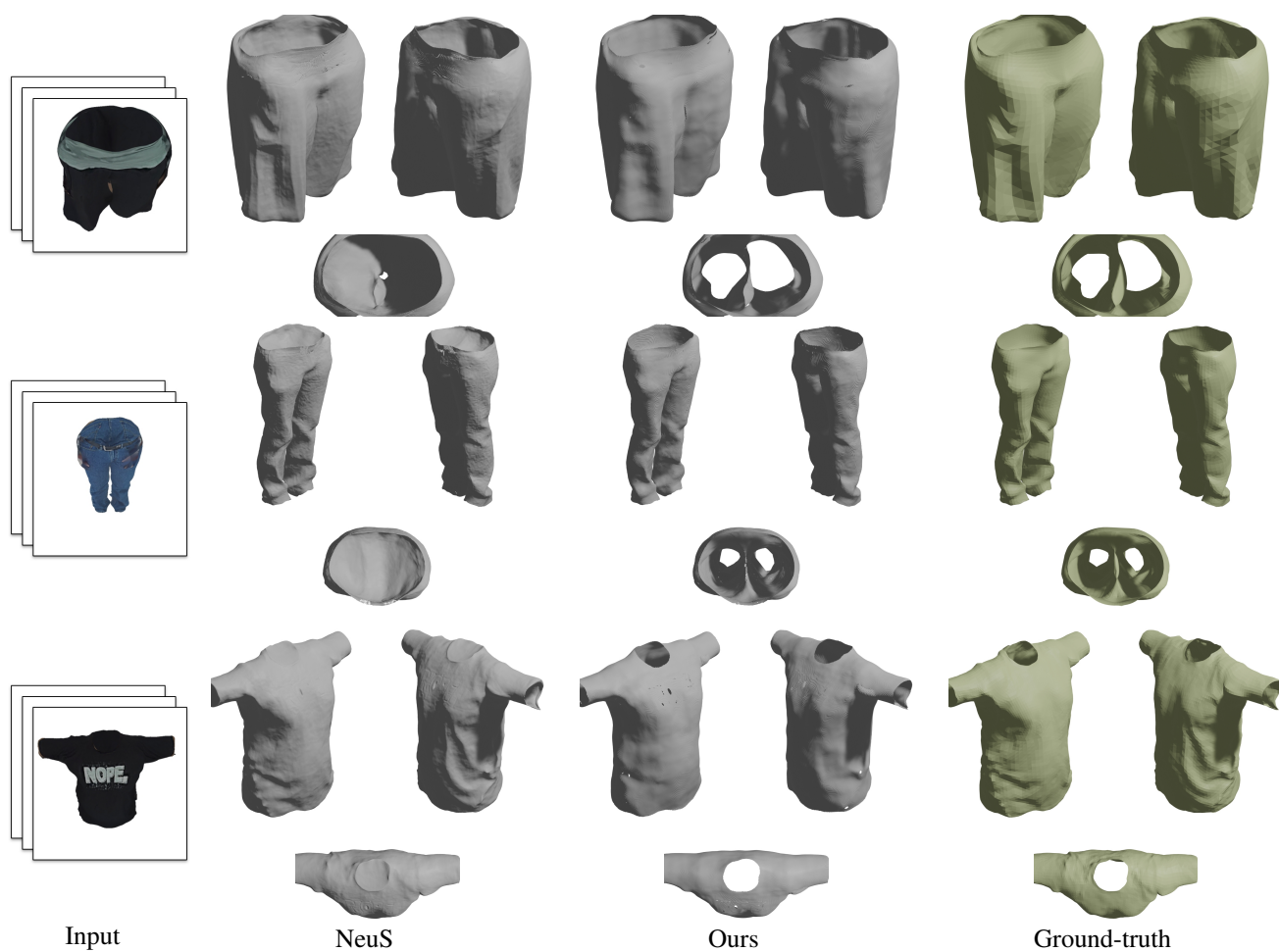


Figure 16. Additional results on the MGN [4] dataset with mask supervision.

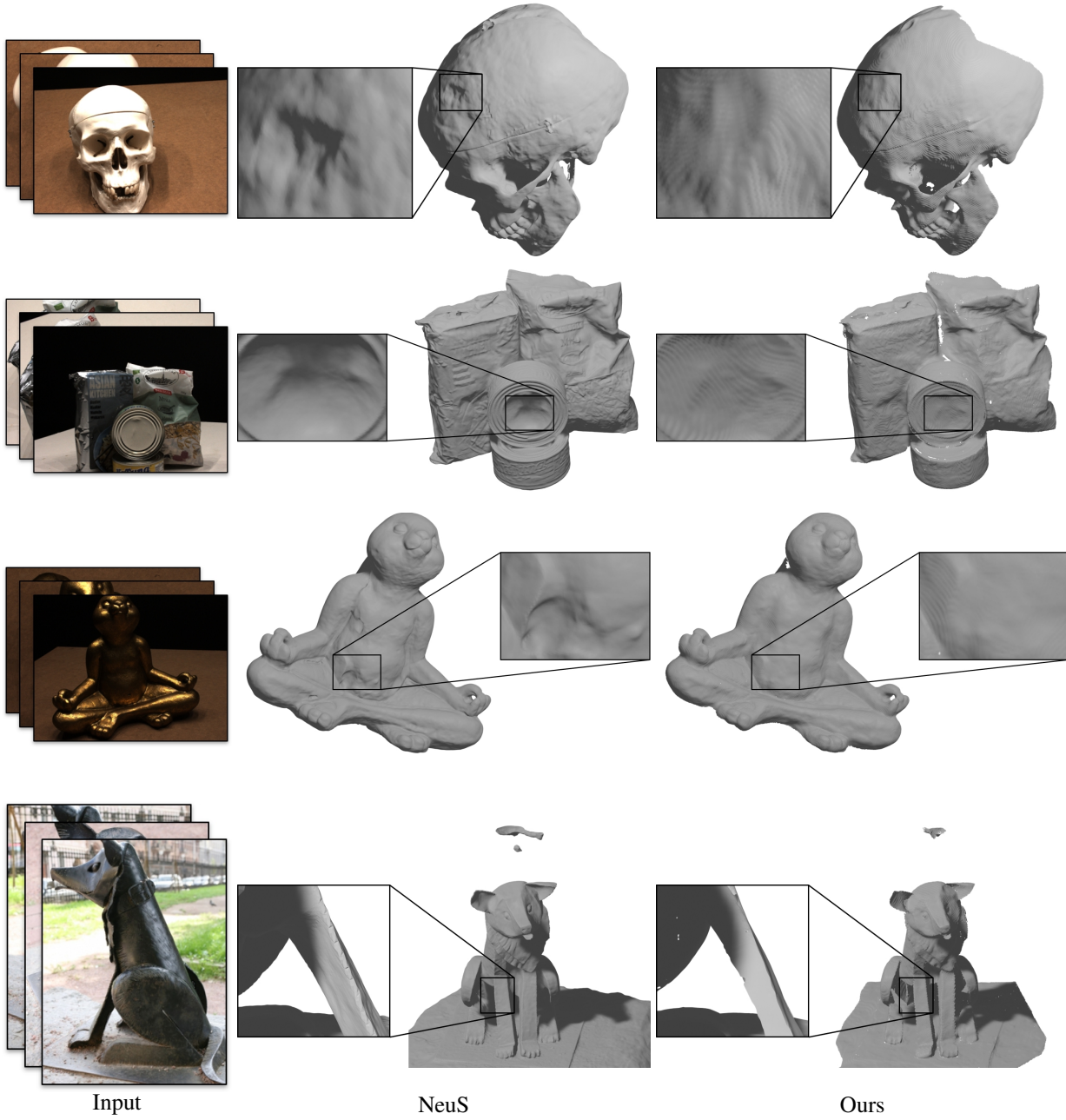


Figure 17. Additional results on the DTU [21] dataset (the first three scenes) and BMVS [56] dataset (the last one scene) with mask supervision.

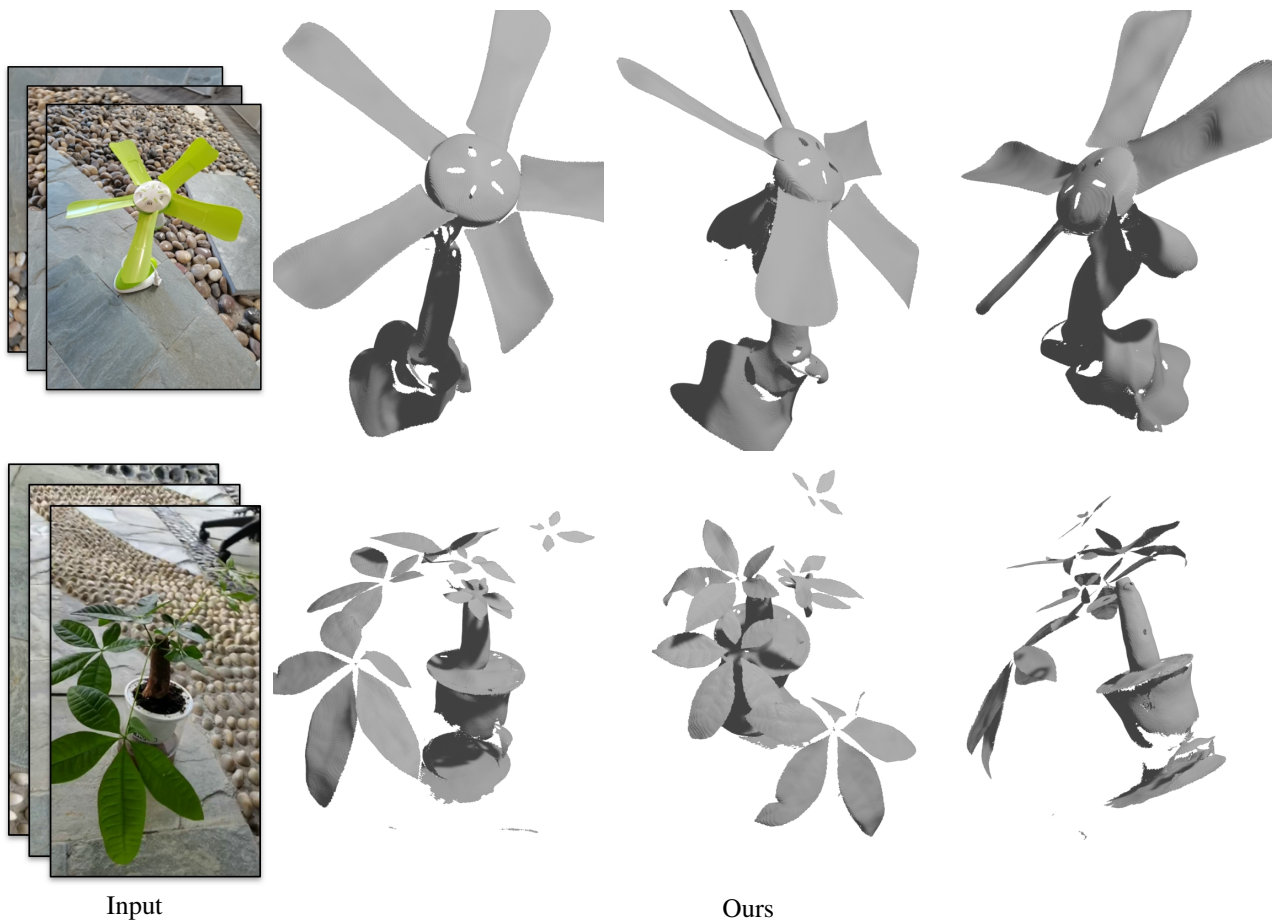


Figure 18. Additional results of the real-captured data without mask supervision.