3PSDF: Three-Pole Signed Distance Function for Learning Surfaces with Arbitrary Topologies

Weikai Chen   Cheng Lin   Weiyang Li   Bo Yang

Digital Content Technology Center, Tencent Games

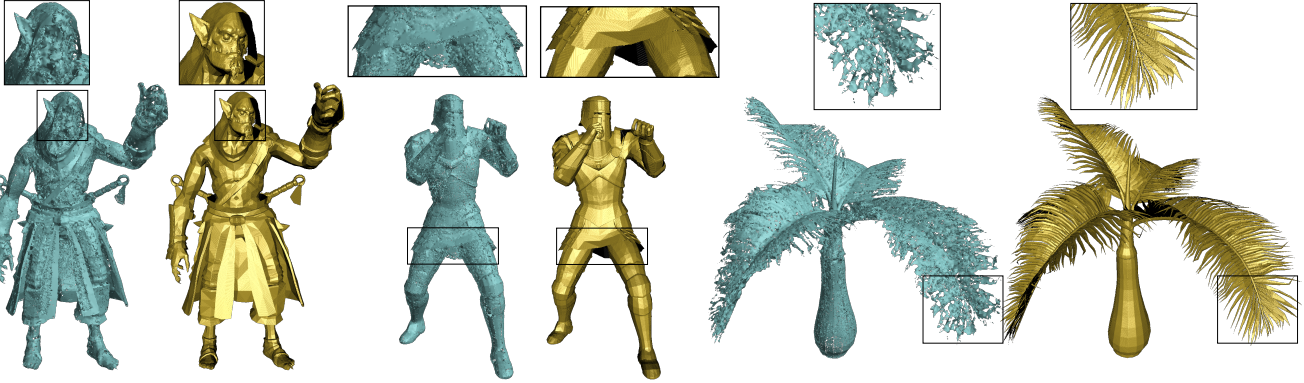{weikaichen,arnolin,kimonoli,brandonyang}@tencent.com

Figure 1. We show three groups of shape reconstruction results generated by NDF [10] (in cyan) and our proposed 3PSDF (in gold) respectively. Our method is able to faithfully reconstruct high-fidelity, intricate geometric details including both the closed and open surfaces, while NDF suffers from the meshing problems. Each NDF result is reconstructed from a dense point cloud containing 1 million points while ours are reconstructed using an equivalent resolution.

Abstract

Recent advances in learning 3D shapes using neural implicit functions have achieved impressive results by breaking the previous barrier of resolution and diversity for varying topologies. However, most of such approaches are limited to closed surfaces as they require the space to be divided into inside and outside. More recent works based on unsigned distance function have been proposed to handle complex geometry containing both the open and closed surfaces. Nonetheless, as their direct outputs are point clouds, robustly obtaining high-quality meshing results from discrete points remains an open question. We present a novel learnable implicit representation, called three-pole signed distance function (3PSDF), that can represent non-watertight 3D shapes with arbitrary topologies while supporting easy field-to-mesh conversion using the classic Marching Cubes algorithm. The key to our method is the introduction of a new sign, the NULL sign, in addition to the conventional in and out labels. The existence of the null sign could stop the formation of a closed isosurface derived from the bisector of the in/out regions. Further, we propose a dedicated learning framework to effectively learn 3PSDF without worrying about the vanishing gradient due to the null labels. Experimental results show that our approach outperforms the previous state-of-the-art methods in a wide range of benchmarks both quantitatively and qualitatively.

1. Introduction

The choice of representation for 3D shapes and surfaces has been a central topic for effective 3D learning. Various 3D representations, including mesh [18,41], voxels [36,42], and point cloud [31,32], have been extensively studied over the past years. Recently, the advent of neural implicit functions (NIF) [6,20,26,29] has brought impressive advances to the state-of-the-art of learning-based 3D reconstruction and modeling.

Classic NIF approaches are built upon the signed distance function (SDF); they train a deep neural network to classify continuous 3D locations as inside or outside the surface via occupancy prediction or regressing the SDF. However, they can only model *closed* surfaces that support the in/out test for level surface extraction. Recent advances that leverage unsigned distance function (UDF) [10,39,40] have made it possible to learn *open* surfaces from point clouds. But instantiating this field into an explicit mesh remains cumbersome and is prone to artifacts. It requires the gen-

eration of dense point cloud and leveraging UDF's gradient field to iteratively push the points onto the target surface. Such process is vulnerable to complex gradient landscape, e.g., parts with many details, and could easily get stuck at a local minima. In addition, reconstruction of mesh from UDF has to use the Ball Pivoting (BP) algorithm which has several drawbacks. 1) It is very sensitive to the input ball radius. A slightly larger or smaller radius would lead to an incomplete meshing result. 2) It is prone to generate self-intersections and disconnected face patches with inconsistent normals even with surfaces of moderate complexity (see the clothing result in Figure 3). 3) The BP algorithm is very time-consuming especially dealing with dense point clouds. Finally, learning UDF becomes a regression task instead of classification like for SDF, making the training more difficult. We show in the closeups of Figure 1 that NDF [8] cannot reconstruct the face details of the first character even with 1 million sampling points.

We overcome the above limitations by presenting a new learnable implicit representation, called *Three-Pole Signed Distance Function* (3PSDF), which is capable of representing highly intricate geometries containing both closed and open surfaces with high fidelity (see Figure 1). In addition, 3PSDF makes the learning an easy-to-train classification task, and is compatible with classic and efficient isosurface extraction techniques, e.g. the Marching Cubes algorithm. The key idea of our approach is the introduction of a direction-less sign, the *NULL* sign, into the conventional binary-sided signed distance function. Points with null sign will be assigned with nan value, preventing the decision boundary to be formed between them and their neighbors. Therefore, by properly distributing the null signs over the space, we are able to cast surfaces with arbitrary topologies (see Figure 2). Similar to previous works based on occupancy prediction [6, 26], we train a neural network to classify continuous points into 3 categories: *inside*, *outside*, and *null*. The resulting labels can be converted back to the 3PSDF using a simple mapping function to obtain meshing result.

We evaluate 3PSDF on three different tasks with gradually increased difficulty: shape reconstruction, point cloud completion and single-view reconstruction. 3PSDF can consistently outperform the state-of-the-art methods over a wide range of benchmarks, including ShapeNet [5], MGN [4], Maximo [1], and 3D-Front [16], both quantitatively and qualitatively. We also conduct comparisons of field-to-mesh conversion time with NDF and analyze the impact of different resolutions and sampling strategies on our approach. Our contributions can be summarized as:

- We present a new learnable 3D representation, 3PSDF, that can represent highly intricate shapes with both closed and open surfaces while being compatible with existing level surface extraction techniques.

- We propose a simple yet effective learning paradigm for 3PSDF that enables it to handle challenging task like single-view reconstruction.

- We obtain SOTA results on three applications across a wide range of benchmarks using 3PSDF.

2. Related Work

Learning with explicit representations. Explicit representations of 3D shapes are often well regularized and structured. Voxel based methods [11, 17, 19] are compatible with convolutional neural networks for learning; to reduce the high memory cost, octree-based partitions are adopted [23, 36, 42]. However, inner parts of objects usually occupy a large portion of the voxels, leading to compromised 3D accuracy due to memory limitation. Mesh-based methods mostly deform a pre-defined mesh to approximate a given 3D shape [14, 18, 28, 41]. One key limitation of such methods is the difficulty of changing mesh topologies, confining its 3D representation capability. Point clouds have achieved much attention recently [32, 33, 38, 44] due to its simplicity. Although such methods are convenient for shape analysis, generating 3D shapes with high precision remains difficult.

Implicit function learning. With the development of deep learning, implicit representation of 3D shapes has achieved great progress in recent years [6, 8, 15, 24, 27, 35]. A good example is signed distance field (SDF), which creates a continuous implicit field in 3D space [29, 30] where outside and inside points are denoted by positive and negative SDFs. Zero-isosurface, i.e., the object's surface, can be efficiently extracted by Marching Cubes [25]. Such representation supports infinite resolution and can simplify the SDF learning as a binary classification process [26]. However, SDF is only applicable to objects with closed surfaces.

To deal with open surfaces, unsigned distance field (UDF) [10] and deep unsigned distance embeddings [40] are proposed. These methods use absolute distance to describe point position, and the zero-isosurface is extracted by Ball-Pivoting algorithm [3]. However they have several major limitations: 1) learning UDF is a regression problem, harder than that in SDF; 2) ball pivoting [3] is more computational expensive and less stable than Marching Cubes [25]; 3) gradient vanishes on the surface, resulting in artifacts. Venkatesh et al [39] proposed Closest Surface-Point (CSP) representation to prevent gradient vanishing and improve the speed. Zhao et al. [45] proposed Anchor UDF to improve reconstruction accuracy. However, the first two limitations of UDF-type methods still remain.
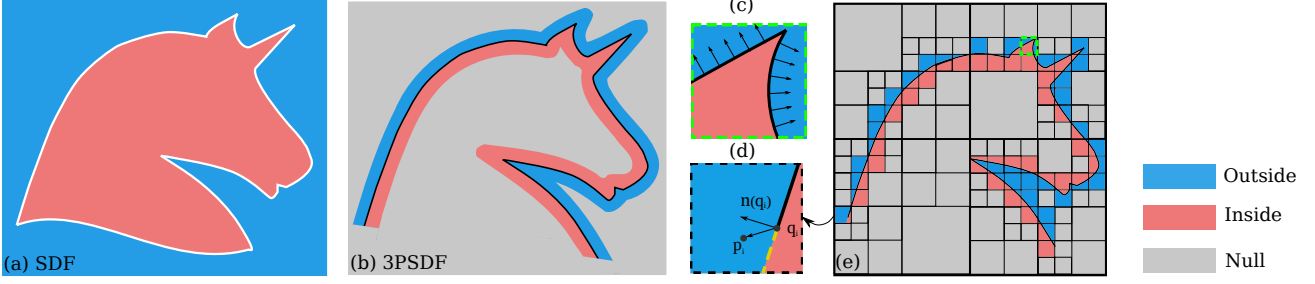
Figure 2. 2D illustration of 3PSDF. (a) Conventional signed distance function (SDF) can only represent closed surface. (b) By introducing the null sign into SDF, 3PSDF can disable specified decision boundaries to cast arbitrary topologies that contain open surfaces. We propose practical framework for computing 3PSDF based on local cells ((c) and (d)). While 3PSDF may introduce approximation error (the yellow dash line in (d)) for open surface enclosed within a cell, the approximation error can be significantly reduced with finer space decomposition. We propose octree-based subdivision approach (e) to improve approximation performance with high computation efficiency.

3. Three-Pole Signed Distance Function

3.1. Definition

A watertight 3D shape can be implicitly represented by a signed distance function. Given a 3D query point $\mathbf{p} \in \mathbb{R}^3$, previous works apply deep neural networks to either predict the occupancy of $\mathbf{p}$ as $f(\mathbf{p}) : \mathbb{R}^3 \mapsto [0, 1]$ [20] or directly regress SDF as $f(\mathbf{p}) : \mathbb{R}^3 \mapsto \mathbb{R}$ [29, 43]. Our key observation is that the formation of closed surface is inevitable as long as both the positive and negative signs exist in the space (note that we do not consider space clipping where SDF is only computed in a limited bounding area). To resolve this issue, we introduce the third direction-less pole – the NULL sign into the field such that the "curse" of closeness can be lifted: no iso-surfaces can be formed at the bisector of either positive/null or negative/null pairs. Therefore, the null sign acts as a surface eliminator that cancels out unwanted surfaces and thus can flexibly cast arbitrary topologies including those with open surfaces.

Formally, for a 3D point $\mathbf{p} \in \mathbb{R}^3$, we propose that in addition to a continuous signed distance, it can be also be mapped to null value: $\Psi(\mathbf{p}) : \mathbb{R}^3 \mapsto \{\mathbb{R}, nan\}$. Hence, given an input surface $\mathcal{S}$, we aim to learn such a mapping function Ψ so that

$$\underset{\Psi}{\text{argmin}} \, ||\mathcal{S} - \mathcal{M}(\Psi(\mathbf{p}))||, \qquad (1)$$

where $\mathcal{M}$ is the meshing operator that converts the resulting field into an explicit surface and $|| \cdot ||$ returns the surface to surface distance. Next, we will introduce how to compute the corresponding 3PSDF for a given shape.

3.2. Field Computation

For non-watertight surface without closed boundaries, it is not possible to perform in/out test for a query point. Hence, we leverage the surface normal to determine the sign of the distance. In particular, we decompose the 3D space

into grid of local cells. As shown in Figure 2, for each cell $\mathcal{C}_i$, if it does not contain any surface of interest, we set its enclosed space as null region and any sample point $\mathbf{p}_i$ that lies inside $\mathcal{C}_i$ has nan distance to the target surface $\mathcal{S}$:

$$\Psi(\mathbf{p}_i, \mathcal{S}) = nan, \text{ if } \mathbf{p}_i \in \mathcal{C}_i \text{ and } \mathcal{C}_i \cap \mathcal{S} = \emptyset \qquad (2)$$

For a local cell $\mathcal{C}_i$ that encloses a surface patch $\mathcal{S}_i$, given a query point $\mathbf{p}_i \in \mathcal{C}_i$, we find $\mathbf{p}_i$'s closest point $\mathbf{q}_i$ on $\mathcal{S}_i$. We set the surface normal at $\mathbf{q}_i$ as $\mathbf{n}(\mathbf{q}_i)$. If vector $\overrightarrow{\mathbf{q}_i\mathbf{p}_i}$ aligns with $\mathbf{n}(\mathbf{q}_i)$, i.e. $\mathbf{n}(\mathbf{q}_i) \cdot \overrightarrow{\mathbf{q}_i\mathbf{p}_i} \geq 0$, we set $\mathbf{p}_i$'s distance to the input surface $\mathcal{S}$ as positive; otherwise, it is negative. The computation can be summarized as:

$$\Psi(\mathbf{p}_i, \mathcal{S}_i) = \begin{cases} \mathbf{d}(\mathbf{p}_i, \mathcal{S}_i) & \text{if } \mathbf{n}(\mathbf{q}_i) \cdot \overrightarrow{\mathbf{q}_i\mathbf{p}_i} \geq 0, \\ -\mathbf{d}(\mathbf{p}_i, \mathcal{S}_i) & \text{otherwise,} \end{cases} \qquad (3)$$

where $\mathbf{d}(\mathbf{p}, \mathcal{S}_i)$ returns the absolute distance between $\mathbf{p}_i$ and $\mathcal{S}_i$. With finer decomposition of 3D space, cells containing geometry would only distribute around the surface of interest while the null cells would occupy the majority of the space. This differs a lot from the conventional signed distance field, where the entirety of the space is filled with distances of either positive or negative sign. Our proposed 3PSDF better reflects the nature of 3D surface of any topology – the high sparsity of surface occupancy.

Surface approximation ability. If an enclosed surface subdivides its hosting cell into several closed sub-regions, our implicit representation can faithfully approximate the original shape without loss of accuracy (Figure 2(c)). If a local cell contains protruding open surface(s), our approach is prone to generate elongated surface patch (Figure 2(d)). However, such approximation error only happen locally and is limited to the size of the local cell. Hence, with a denser 3D decomposition, we can significantly reduce the approximation error. We provide additional experiments in Sec-

3

tion 4.5 showing different reconstruction performance with respect to varying sampling resolutions.

3.3. Learning Framework

Though the introduction of the null sign provides the flexibility of eliminating unwanted surface, the nan value prohibits computing meaningful gradient required for updating a deep neural network. To resolve this issue, a straightforward way is to combine binary classification (nan v.s. non-nan) and regression, where the former generates a mask of the valid narrow band around the surface and the later regresses the surface within this narrow band. While we experimentally validate that it is possible to learn 3PSDF via this approach, additional challenges would arise in aligning the narrow-band mask from binary classification and the regressed decision boundary from the regression branch. A misalignment of the two branches' results would lead to discontinuity in the final reconstruction. Hence, we propose an alternative learning framework that formulates the learning of 3PSDF as a 3-way classification problem as elaborated below. While we provide the method and results of the 3-way classification framework in the main paper, we provide detailed comparisons between the two learning methods in the supplemental materials.

Similar to the previous works on occupancy prediction [6, 26], the 3-way classification method proposes to approximate the target function (Equation 2 and 3) with a neural network that infers per-point label: $\{in, out, null\}$. We represent the label semantics using discrete numbers without loss of generality. Formally, we aim to learn a mapping function $o : \mathbb{R}^3 \mapsto \{0, 1, 2\}$, where the labels $\{0, 1, 2\}$ represent inside, outside, and null respectively.

When applying such a network for downstream tasks (e.g. 3D reconstruction) based on observation of the object (e.g. point cloud, image, etc.), the network must be conditioned on the input. Therefore, in addition to the coordinate of points $\mathbf{p} \in \mathbb{R}^3$, the network also consumes the observation of object $\mathbf{x} \in \mathcal{X}$ as input. Specifically, such a mapping function can be parameterized by a neural network Φ_θ that takes a pair $(\mathbf{p}, \mathbf{x})$ as input and outputs its 3-class label:

$$\Phi_\theta(\mathbf{p}, \mathbf{x}) : \mathbb{R}^3 \times \mathcal{X} \mapsto \{0, 1, 2\}. \quad (4)$$

Training. To learn the parameters θ of the neural network $\Phi_\theta(\mathbf{p}, \mathbf{x})$, we train the network using batches of point samples. For the i-th sample in a training batch, we sample N points $p_{ij} \in \mathbb{R}^3, j = 1, \ldots, N$. The mini-batch loss $\mathcal{L}_\mathcal{B}$ is:

$$\mathcal{L}_\mathcal{B} = \frac{1}{|\mathcal{B}|N} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{N} \mathcal{L}(\Phi_\theta(p_{ij}, x_i), y_{ij}), \quad (5)$$

where $\mathcal{L}(\cdot, \cdot)$ computes the cross-entropy loss, x_i is the i-th observation of batch $\mathcal{B}$, y_{ij} denotes the ground-truth label for point p_{ij}.

Octree-based subdivision. Since the computation of 3PSDF is done locally, to ensure a high reconstruction accuracy, it would be preferable not to include too many intricate geometric details and open surfaces in one cell. We propose an octree-based subdivision [37, 42] method as shown in Figure 2(e). We only subdivide a local cell if it intersects with the input shape. As the subdivision depth increases, the complexity of surface patch contained by each local cell decreases, leading to better approximation accuracy. In addition, since regions containing no shapes will not be further divided, we are able to accomplish a balanced trade-off between the computational complexity and reconstruction accuracy. In all of our experiments, we use the octree-based subdivision for ground truth computation unless otherwise stated. Our experiments in Section 4.5 validate the benefits of performance from octree-based sampling.

3.4. Surface Extraction

Once the network is learned, we are able to label each query point with our predictions. To extract the iso-surface, we first convert the inferred discrete labels back to the original 3PSDF representation. Points with labels 0, 1, and 2 are assigned with sdf values as -1, 1, and nan, respectively. The reconstructed surface can then be extracted as zero-level surface. Note that the iso-surface represented by 3PSDF can be directly extracted using the classic Marching Cubes (MC) algorithm. The existence of null value would naturally prevent MC from extracting valid iso-surfaces at locations that contain no shapes. In the meantime, in the vicinity of target surface, the iso-surface extraction can be performed normally just as the conventional signed distance field. After MC computation, we only need to remove all the nan vertices and faces generated by the null cubes. The remaining vertices and faces serve as the meshing result.

4. Experiments

4.1. Experimental Setup

Tasks and datasets. We validate the proposed 3PSDF using three types of experiments. First, we analyze the representation power of 3PSDF by examining how the 3PSDF can reconstruct complex 3D shapes from a learned latent embedding. This gives us an upper bound on the results we can achieve when conditioned on other inputs. Second, we condition the learning of 3PSDF on sparse point cloud and test its performance by feeding 3D features. Finally, we use image features as input and provide validation on the challenging task of singe-view reconstruction. All experiments are compared with the SOTA methods for better verification. The experiments are conducted on a wide range of 3D datasets including ShapeNet [5], MGN [4], 3D-Front [16], and Maximo [1]. The specific settings are detailed in the following experiments.
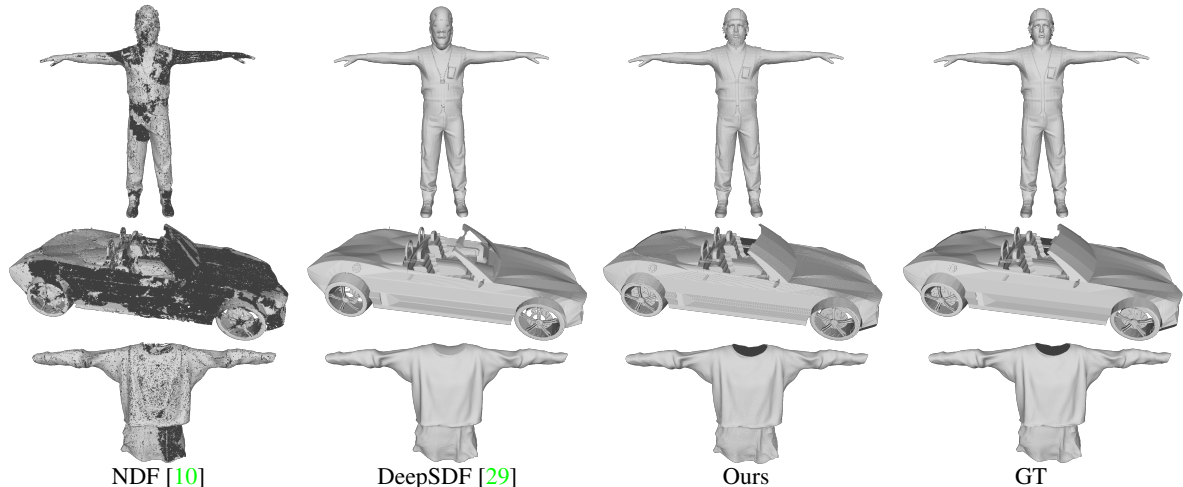
4

Figure 3. Visual comparisons of shape reconstruction result using different neural implicit representations.

Metric	Method	ShapeNet					MGN	Mixamo
		car	plane	boat	lamp	chair		
CD ($\times 10^{-5}$)↓	NDF	0.63	0.25	0.33	0.34	0.45	0.08	0.52
	DeepSDF	2.71	0.58	0.61	1.99	0.91	0.09	1.82
	Ours	**0.44**	**0.21**	**0.24**	**0.30**	**0.35**	**0.07**	**0.32**
EMD ($\times 10^2$)↓	NDF	2.39	2.46	2.11	2.05	1.47	0.33	2.81
	DeepSDF	4.23	2.56	2.29	2.78	1.66	0.62	4.56
	Ours	**2.10**	**2.23**	**2.04**	**1.92**	**1.38**	**0.21**	**2.55**

Table 1. Quantitative comparisons of shape reconstruction using different neural implicit representations.

Implementation details. For the task of reconstruction from point cloud, we use the same point encoder (IF-Net) and hyper-parameters with NDF [10]. For single-view reconstruction, we use VGG16 [34] with batch normalization as the image encoder. Similar to DISN [43], we use both multi-scale local and global features to predict the 3PSDF value. We re-orient the normals of the ground-truth surfaces based on visibility [13] to make them consistent. We refine the results by filling small holes and smoothing the surfaces. The ground-truth 3PSDF values are generated with resolution 128^3 and the results are evaluated using resolution 256^3. We use octree-based importance sampling for all experiments. To ease the learning of 3PSDF, we ensure the size of the minimum leaf octree cell to be consistent across different objects by using a unified bounding box for all samples.

4.2. Shape Reconstruction

To evaluate the capability of 3PSDF of modeling complex geometry, we perform the shape reconstruction experiment comparing with other SOTA neural implicit representations: DeepSDF [29] and NDF [10]. Similar to the auto-encoding method in [29], we embed each training sample with a 512 dimensional latent code and train neural networks to reconstruct the 3D shape from the embedding. We perform evaluations on five representative categories of

ShapeNet that contain the most intricate geometry, and two datasets with open surfaces: MGN [4] and Mixamo [1].

Since we are only interested in reconstructing the training data, we do not use validation and test set for this experiment. As DeepSDF cannot handle open surfaces, we generate its ground-truth SDF value using [21] which converts complex open surfaces into closed ones using winding number. For training and evaluation, we use 10 as the depth for octree-based sampling for our method and the equivalent resolution of 1024 for DeepSDF. To ensure similar density of sampling, we generate 1 million surface points for the NDF. All the NDF results (including the following experiments) are generated using the post-processing scripts released by the authors to ensure fair comparison. We show the visual comparisons in Figure 1 and 3 and the quantitative comparisons in Table 1. While DeepSDF is able to reconstruct fine details, it cannot handle open surfaces like hair, clothing, and the windshield. NDF can deal with all topologies, but suffers from meshing problems – lots of self-intersections and flipped faces are introduced. Our method can faithfully reconstruct all the intricate geometries while achieving the best performance in quantitative comparisons.

4.3. Reconstruction from Point Cloud

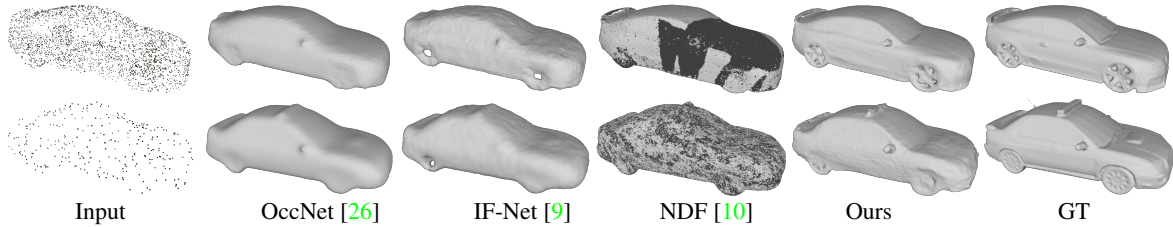We further validate 3PSDF on the task of shape reconstruction from sparse point clouds. Following NDF [10],

| Input | OccNet [26] | IF-Net [9] | NDF [10] | Ours | GT |

Figure 4. Comparisons of point cloud completion trained on watertight shapes (with inner structure removed).
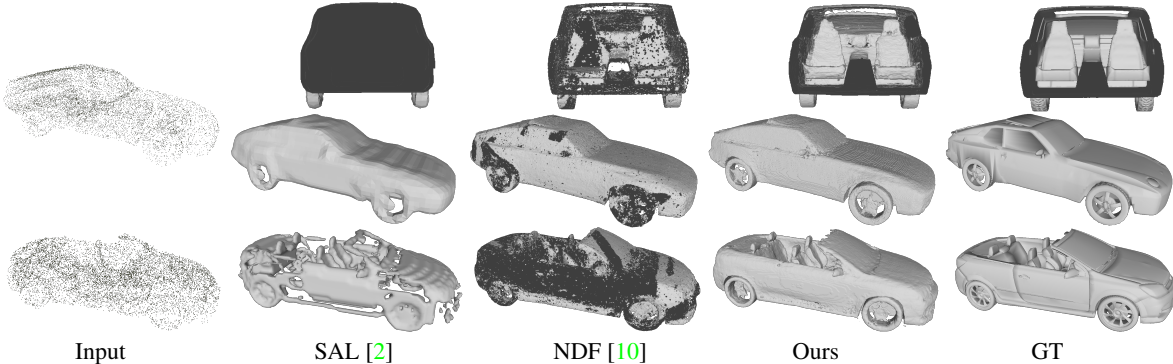

| Input | SAL [2] | NDF [10] | Ours | GT |

Figure 5. Comparisons of point cloud completion trained on non-watertight shapes (with inner structure and open surface). The first row shows the inner structure of the reconstructed results in the second row.

	Chamfer-L_2			Chamfer-L_2	
	3K	300		10K	3K
DMC	1.255	2.417	SAL	6.39	7.39
OccNet	0.938	1.009	NDF	0.074	0.275
IF-Net	0.326	1.147	Ours	**0.071**	**0.258**
NDF	0.127	0.626			
Ours	**0.112**	**0.595**			

Table 2. Left: results of point cloud completion for closed watertight cars from 3000 and 300 points. Right: results of point cloud completion for unprocessed cars from 10000 and 3000 points. Chamfer distance is reported in $\times 10^{-4}$.

we first evaluate 3PSDF on reconstructing closed surfaces, and then demonstrate that 3PSDF can represent complex surfaces with inner structures and open surfaces.

Reconstruction of closed shapes. To compare with the SOTA methods: OccNet [26], IF-Net [9], and DMC [22], we train on the ShapeNet car category pre-processed by [43] with all open surfaces closed and inner structures removed. We show the reconstruction results using 300 and 3000 points as input both qualitatively and quantitatively in Figure 4 and Table 2 respectively. Compared to the other methods, our approach can better reconstruct the sharp geometry details while outperforming all baselines in quantitative measurement.

Reconstruction of complex surfaces. To validate the ability of 3PSDF of handling raw, unprocessed data, we train 3PSDF to reconstruct complex shapes from sparse point clouds on three datasets: unprocessed cars from ShapeNet [5], garments with open surfaces from MGN [4], and the living room scenes from 3D-Front [16]. We use NDF [10] and SAL [2] as the baselines for reconstructing unprocessed cars. Since SAL is built upon traditional SDF, we use the closed shapes as ground truth. We provide visual comparisons of the reconstructed results in Figure 5 and 6. SAL struggles to model the open surfaces, e.g. the windshield and the thin outer structure of car. NDF can generate dense point clouds close to the target surface. However, the output points are prone to be clustered (as shown in the closeups of Figure 6) which prevent the BP algorithm from generating high-quality meshing results. In contrast, 3PSDF is able to faithfully reconstruct the interior structures as well as the open surfaces. The quantitative comparisons in Table 2 and Figure 6 also validates our advantage over the baselines.

4.4. Single-view 3D Reconstruction

In this experiment, we apply 3PSDF to single-view 3D reconstruction (SVR) tasks to further demonstrate its representational ability. We evaluate on the MGN dataset [4] and ShapeNet [5]. We use Chamfer-L_2 distance and F-score ($\tau = 1\%$ volume diagonal length) as the evaluation metrics.

We compare against the representative SVR methods using implicit fields, including IMNet [7], OccNet [26] and
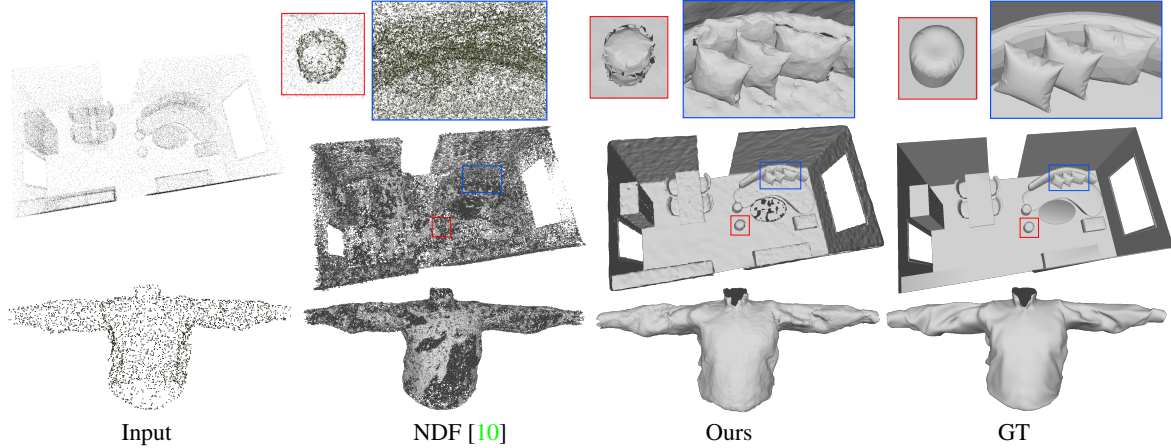
6

Figure 6. Comparisons of point cloud completion performance on MGN and 3D-Front. Chamfer-L_2 ($\times 10^{-4}$) comparison: MGN: NDF - 0.035; ours - **0.033**; 3D-Front: NDF - 1.452; ours - **1.378**.
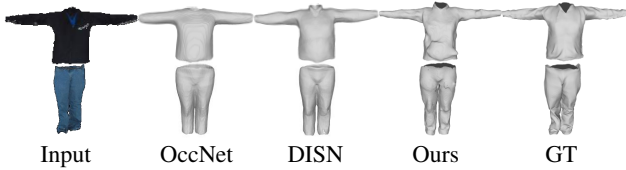

Figure 7. Qualitative comparison on MGN dataset with state-of-the-art single-view reconstruction methods based on implicit functions. The quantitative evaluation results in terms of CD ($\times 10^{-3}$) and F-score ($\times 10^{-2}$) metrics on the testing set of MGN are: 1.03 and 69.8 (DISN); 1.01 and 71.0 (OccNet); **0.98** and **71.2** (Ours).

Figure 8. Qualitative comparison results with SOTA single-view reconstruction methods based on implicit functions.

DISN [43]. We further implement an image-based NDF [10] estimator but find reasonable results cannot be generated by solely using image features. Since the models in these two datasets usually contain non-watertight surfaces which cannot be directly handled by the baseline methods, we first convert these models to watertight ones. Note that our representation is directly trained on the original shapes without this extremely time-consuming process.

Single-view reconstruction on MGN. The models in the MGN dataset [4] are represented as open freeform surfaces with single sheets, which is challenging to the existing single-view reconstruction methods with implicit functions. We render an RGB image using the textured mesh for each garment model, and train a network conditioned on images to predict the shape representations. As shown in Figure 7, our results capture the original open-surface structure as well as more high frequency geometric features such as the wrinkles. The 3PSDF representation also achieves the best quantitative results on the testing set.

Single-view reconstruction on ShapeNet. We use a subset of ShapeNet [5] for evaluation, from which we choose 5 categories (ai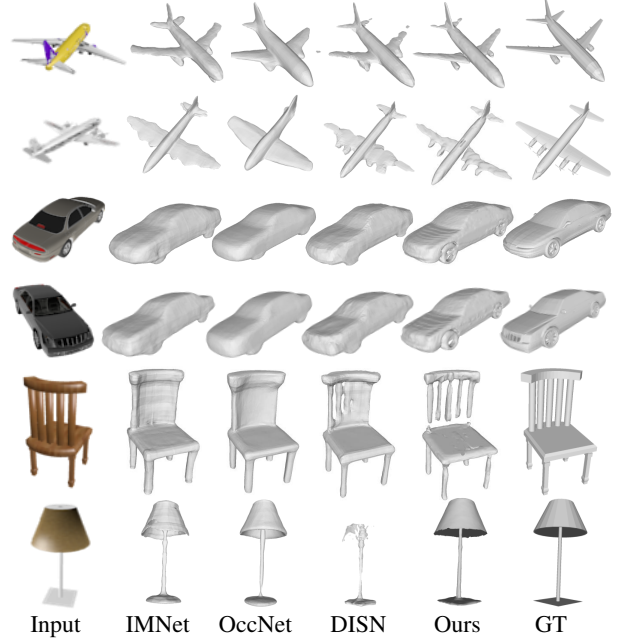rplane, car, lamp, chair, boat) resulting in 17803 shapes. We use the same image renderings (24 views per shape) and train/test split as Choy et al. [12]. Figure 8 shows a set of qualitative comparisons. Despite being designed for handling open surfaces, 3PSDF is still a versatile representation for reconstructing various 3D shapes in the ShapeNet with either closed or open surfaces. We not only faithfully preserve the original structure of the target shape, but also captures more detailed geometries. Instead, the existing implicit functions always rely on watertight shapes, which substantially limits their representational ability and usually leads to over-smoothed geometries, lack of details, as well as inconsistent typologies. As shown in Table 3, 3PSDF achieves state-of-the-art performance compared to

	Method	ShapeNet				
		car	plane	boat	lamp	chair
CD ↓	IMNet	3.48	5.07	4.17	9.51	1.81
	OccNet	1.74	1.74	3.48	14.55	2.22
	DISN	1.23	1.71	4.84	**6.11**	**1.54**
	Ours	**0.76**	**1.66**	**3.27**	7.67	3.29
FS ↑	IMNet	31.8	33.7	39.8	34.3	61.1
	OccNet	54.4	59.7	44.9	**50.6**	59.6
	DISN	65.8	**77.2**	57.8	50.4	**63.8**
	Ours	**77.0**	72.8	**66.6**	49.3	58.5

Table 3. Quantitative comparisons of single-view reconstruction. Chamfer-L_2 and F-score are reported in $\times 10^{-3}$ and $\times 10^{-2}$ respectively.

the existing methods, where it has 5 metrics ranking the first and comparable results for the remaining metrics.

4.5. Further Discussions

Reconstruction accuracy/appearance with different resolutions. Since 3PSDF is continuously defined in the 3D space, it can represent a shape using arbitrary resolution. Figure 9 gives a coarse-to-fine shape approximation results, where we discretize the volumetric space and use different grid resolutions to represent a 3D shape. The experimental results show that the approximation quality of 3PSDF increases as the resolution grows, leading to smoother shape boundaries and higher reconstruction accuracies.
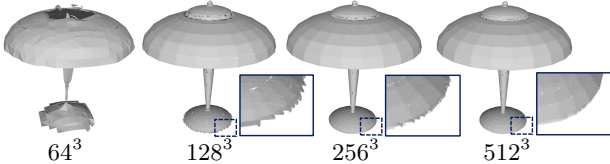

64^3 128^3 256^3 512^3

Figure 9. Reconstruction results of a shape using different resolutions. From the left to right, the CD ($\times 10^{-5}$) values for these shapes are: 14.49, 2.52, 2.21 and 2.12; the EMD ($\times 10^2$) values are: 3.42, 0.336, 0.267 and 0.227 .

Timing cost for field-to-mesh conversion. We quantitatively evaluate the timing cost for field-to-mesh conversion in different output sampling densities. For the octree depth of 6 (64^3), 7 (128^3), 8 (256^3) and 9 (512^3), the average field-to-mesh conversion times of 3PSDF for a single shape are 0.006s, 0.11s, 0.54s, and 3.72s respectively. In contrast, the conversion times for NDF [10] given the comparable number of sampling points are: 2.1s, 15mins, 3hrs, 34hrs, using the provided post-processing setting (radius=0.005) by NDF. The experiments are conducted on a machine with a 48-Core AMD EPYC CPU and 64GB memory.

Different sampling strategies. We further study the impact of different sampling strategies on the performance of

	Random	Uniform	Octree
CD ($\times 10^{-4}$) ↓	7.43	2.16	**1.08**
EMD ($\times 10^3$) ↓	3.28	1.55	**1.12**

Table 4. Reconstruction accuracy using different sampling strategies.

3PSDF; we evaluate on the task of shape reconstruction from point cloud on the unprocessed car data. Three strategies are used to generate sampling points: 1) randomly draw samples in the space; 2) uniform sampling which generates adjacent points in equal distance; 3) octree-based sampling that uses the corner points of leaf octree cells as training samples. We use around 18 million sampling points for all strategies. Table 4 shows that the octree-based sampling yields the best result. Compared to the other methods, octree-based sampling is able to densely sample points with inside/outside labels, generating a more balanced training set containing all the 3 labels. We use octree-based sampling for all of our experiments unless otherwise stated.

Limitation. 3PSDF has difficulty in reconstructing multi-layer surfaces that are very close to each other, especially when the resolution is low. This is because 3PSDF requires denser sampling rate compared to SDF in order to insert a null layer in between to prevent artifact surface. Besides, given the enhanced representational ability of 3PSDF, it requires more informative features to learn and longer time to train; for example, the network converges much faster and achieves better geometry given point clouds as input, compared to single images.

5. Conclusions and Discussions

We introduce 3PSDF, a learnable implicit distance function to represent 3D shapes with arbitrary topologies. Different from the widely used implicit representations like SDF that can only encode watertight shapes, 3PSDF can faithfully represent various shapes with both open and close surfaces. The key insight of the 3PSDF is the introduction of the NULL sign to additionally indicate the inexistence of surface. We further formulate a classification-based learning paradigm to effectively learn this representation. As a result, the representational power of the distance function is significantly enhanced. Extensive evaluations demonstrate that 3PSDF is a versatile implicit representation that accommodates various 3D reconstruction tasks.

Future work. We have shown in the supplemental that 3PSDF can be learned via an alternative method that combines binary classification and regression. Compared to 3-way classification, such a method has the potential to generate smoother surface with fewer sampling points at training time. However, it is not as robust as the 3-way counterpart as it requires the results of the two branches align well in order to prevent holes and artifacts. It would be an interesting future avenue to investigate how to resolve this issue.

References

[1] Adobe. https://www.mixamo.com/, 2017. 2, 4, 5, 11

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 6, 16

[3] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. 2

[4] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 2, 4, 5, 6, 7, 11, 13

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 4, 6, 7, 13

[6] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 1, 2, 4

[7] 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. 6

[8] 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. 2

[9] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 6, 13, 16

[10] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020. 1, 2, 5, 6, 7, 8, 13, 16, 17

[11] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2

[12] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 7, 13

[13] Lei Chu, Hao Pan, Yang Liu, and Wenping Wang. Repairing man-made meshes via visual driven global optimization with minimum intrusion. *ACM Transactions on Graphics (TOG)*, 38(6):1–18, 2019. 5

[14] Angela Dai and Matthias Nießner. Scan2mesh: From unstructured range scans to 3D meshes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5574–5583, 2019. 2

[15] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J. Guibas. Curriculum DeepSDF. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–67, 2020. 2

[16] Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Cao Li, Zengqi Xun, Chengyue Sun, Yiyun Fei, Yu Zheng, Ying Li, et al. 3d-front: 3d furnished rooms with layouts and semantics. *arXiv preprint arXiv:2011.09127*, 2020. 2, 4, 6

[17] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, pages 484–499. Springer, 2016. 2

[18] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9785–9795, 2019. 1, 2

[19] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 2

[20] Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. Deep volumetric video from very sparse multi-view performance capture. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 336–354, 2018. 1, 3

[21] Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. Robust inside-outside segmentation using generalized winding numbers. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013. 5

[22] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018. 6

[23] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2

[24] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[25] William E. Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Transactions on Graphics (TOG)*, 21(4):163–169, 1987. 2

[26] 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. 1, 2, 4, 6, 13, 16

[27] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4480–4490, 2019. 2

[28] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9964–9973, 2019. 2

[29] 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. 1, 2, 3, 5, 13

[30] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2

[31] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1, 2

[33] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3D point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3859–3868, 2019. 2

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[35] Jia-Heng Tang, Weikai Chen, Jie Yang, Bo Wang, Songrun Liu, Bo Yang, and Lin Gao. Octfield: Hierarchical implicit functions for 3d modeling. In *The Thirty-Fifth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2

[36] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017. 1, 2

[37] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 4

[38] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3d point clouds via graph convolution. In *International Conference on Learning Representations*, 2019. 2

[39] Rahul Venkatesh, Tejan Karmali, Sarthak Sharma, Aurobrata Ghosh, László A Jeni, R Venkatesh Babu, and Maneesh Singh. Deep implicit surface point prediction networks. *arXiv preprint arXiv:2106.05779*, 2021. 1, 2

[40] 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. 1, 2

[41] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3D mesh models from single RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 1, 2

[42] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017. 1, 2, 4

[43] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3, 5, 6, 7, 13, 14

[44] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4541–4550, 2019. 2

[45] Fang Zhao, Wenhao Wang, Shengcai Liao, and Ling Shao. Learning anchored unsigned distance functions with gradient direction alignment for single-view garment reconstruction. *arXiv preprint arXiv:2108.08478*, 2021. 2

In this supplemental material, we discuss an alternative framework of learning 3PSDF (Section A), use 3PSDF to model functions or manifolds (Section B), provide additional implementation details (Section C), network structure for each experiment (Section D), comparison between our proposed 3PSDF and TSDF (Section E), and more results (Section F).

A. Alternative Learning Framework

In addition to 3-way classification, 3PSDF can be learned using an alternative framework that combines binary classification and regression. Specifically, the binary classification branch learns to classify the space into nan and non-nan regions, where the non-nan region forms a valid narrow band for extracting surface as demonstrated in Figure 2(b) as shown in the main paper. The regression branch strives to regress a continuous SDF in the narrow-band region as generated by the classification branch. Formally, we formulate this alternative framework as follows:

$$\Phi_C(\mathbf{p}, \mathbf{x}) : \mathbb{R}^3 \times \mathcal{X} \mapsto [0, 1], \qquad (6)$$

$$\Psi_R(\mathbf{p}, \mathbf{x}) = SDF(\mathbf{p}). \qquad (7)$$

In particular, the classification branch Φ_C consumes a 3D query point $\mathbf{p}$ and its corresponding observation $\mathbf{x}$ and predicts the probability of the query point locating in the non-nan region; the regression branch Φ_R directly infers the signed distance of $\mathbf{p}$ as defined in Equation (3) in the main paper.
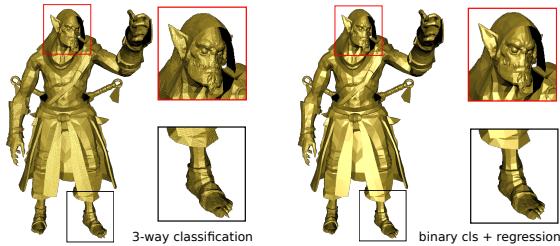

Figure 10. Comparisons of two ways of learning 3PSDF. Quantitative comparisons of shape reconstruction, Mixamo: 0.32:0.31(CD); 0.944:0.950(F-score); MGN: 0.07:0.07(CD); 0.991:0.993(F-score). Note all numbers are reported in format of (3-way cls. : bin. cls.+reg.).

Surface extraction. The framework based on binary classification and regression requires training of two branches, which can be implemented either using two heads of a backbone network or two independent networks. Once the networks are trained, the sampling points that are classified as nan points by the classification branch are assigned with nan value. The rest points are assigned with continuous SDF

distance using the predictions of the regression branch. The resulting 3PSDF field can be directly converted into mesh using the Marching Cubes (MC) algorithm with the iso-value set to 0. Same as 3-way classification, after MC computation, we only need to remove all the nan vertices and faces generated by the null cubes. The remaining vertices and faces serve as the meshing result.

A.1. Comparisons with 3-way Classification

We provide in-depth comparisons between the two candidate learning frameworks: binary classification + regression (BR) v.s. 3-way classification (3C) in this section. Specifically, we evaluate both methods in the task of shape reconstruction and point cloud completion.

Shape reconstruction. We use the same experiment settings with that of the main paper for evaluating the two candidate frameworks. Both methods are validated using two datasets that contain non-watertight open surfaces: MGN [4] and Mixamo [1].

We show both the qualitative and quantitative comparisons in Figure 10. While the two methods are trained using the same data, the BR framework can generate smoother reconstruction compared to that of 3C method, thanks to its continuous SDF output. This is also reflected in the quantitative measurements, where BR can achieve comparable or even better results.

	Chamfer-L_2				Chamfer-L_2	
	3K	300			10K	3K
BR	0.312	1.025		BR	0.095	0.314
3C	**0.112**	**0.595**		3C	**0.071**	**0.258**

Table 5. Left: results of point cloud completion for closed watertight cars from 3000 and 300 points. Right: results of point cloud completion for unprocessed cars from 10000 and 3000 points. Chamfer distance is reported in $\times 10^{-4}$.

Reconstruction from point cloud. We also validate the performance of both candidate frameworks in the task of surface reconstruction from sparse point cloud. Specifically, we evaluate their performance on reconstructing both closed and open surface with the same setting as that of the main paper. We show in Figure 11 that in the BR framework, though both the classification and regression branches can generate reasonable reconstructions, the final merged results still exhibit incompleteness. We further demonstrate the cause of incomplete reconstructions in the overlaid visualization of the two branches (Figure 12). Since the results of the two branches are not perfectly aligned due to the different natures of their tasks, the classification branch would mistakenly remove part of the regressed surfaces generated by the regression branch. This

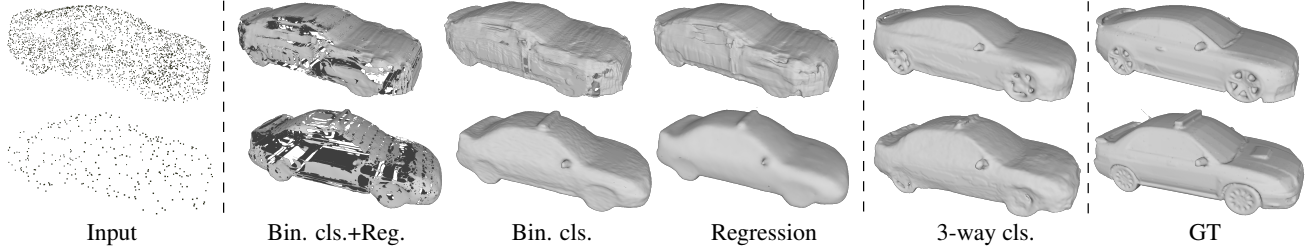| Input | Bin. cls.+Reg. | Bin. cls. | Regression | 3-way cls. | GT |

Figure 11. Comparisons of point cloud completion trained on watertight shapes by using two candidate learning frameworks of 3PSDF: binary classification (bin. cls.)+regression (reg.) and 3-way classification (cls.). For the results of BR, we also show the results generated from the two branches.
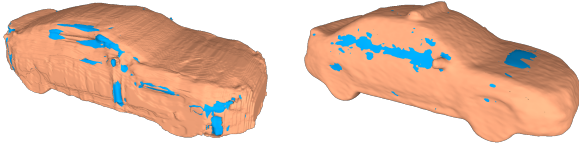

Figure 12. We overlay the reconstruction results of the classification and regression branches under the BR framework as shown in Figure 11. The classification results are highlighted in orange while the regression results are marked with blue. The misalignment of the two branches' results leads to the incomplete reconstruction in Figure 11.

could render holes and discontinuity in the results of the BR method. In comparisons, the 3C method does not suffer from such a problem as it only requires a single branch to generate the final reconstruction. This is also reflected in the quantitative measurements in Table 5.

Discussion. We have evaluated the performance of both candidate frameworks in two different tasks. In the applications where the binary classification and regression branches are well aligned, e.g. the shape reconstruction task, the BR method can lead to higher-quality results with smoother surface compared to the 3C approach. However, for more challenging scenarios, e.g. point cloud completion, where the two branches of BR framework may produce slightly deviated reconstructions, the final reconstruction may be incomplete despite that the two branches have obtained faithful reconstructions. In contrast, the 3C framework is robust over all kinds of task without the need of worrying about the misalignment issue. It would be an interesting future avenue to investigate how to resolve the misalignment problem of the BR method while enjoying its smooth nature.

B. Modeling Functions and Manifolds using 3PSDF

Following NDF, we train 3PSDF on 1 million points sampled from 1000 functions, which are either linear, parabola or sinusoids. Figure 13 shows the fitting results of 3PSDF to a variety of functions and manifolds. In Figure 13, red dots are points labeled as "inside" while cyan ones as "outside". "Nan" points are omitted for clear demonstration. As shown in the results, 3PSDF can faithfully model various functions and manifolds, which further validate that it is a versatile representation.
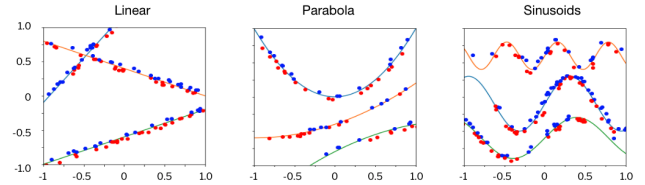

Figure 13. Function and manifold fitting using 3PSDF.

C. More Implementation Details

C.1. Reconstruction from Sparse Point Cloud

We use octree-based sampling to generate the ground-truth data for our approach. The sampling points are the corner points of the leaf cells generated by octree decomposition. In particular, we use depth of 6 for generating training data on pre-processed ShapeNet car category. For raw, unprocessed ShapeNet car, MGN, and 3D-Front, we use depth of 8, 7, and 9 respectively for training data generation. We train separate models for different numbers of input points. All models are trained using the same set of hyperparameters. For all experiments, we use the Adam optimizer with parameters $lr = 1e^{-4}$, $betas = (0.9, 0.999)$, $eps = 1e^{-8}$, $weight\_decay = 0$.

For MGN dataset, we split the data into train and test set with 9:1 ratio. For 3D-Front dataset, we extract 100 living rooms, 10 of which is used for testing and the rest is used for training. For NDF, we generate 1 million points for all

experiments except the scene reconstruction task where we generate a more dense point containing 3 million points. The meshing results of NDF are obtained by running the script (including Ball Pivoting algorithm (BPA) and post-processing operations) provided by the authors in Mesh-Lab. All the results are reported using the test data. For the ShapeNet car dataset, we use the common train and test split by [43].

C.2. Single-view Reconstruction on MGN

We evaluate and compare the representation capability of 3PSDF, DISN [43] and OccNet [26] on MGN dataset [4] for single-view 3D reconstruction. Each garment model in MGN dataset is rendered into an 256×256 RGB image from a front-view textured mesh. All the meshes and images are aligned with the same camera settings and normalized.

For 3PSDF, open surface models in MGN dataset are directly sampled with Octree-based subdivision at a resolution of 128^3, resulting in a mean sampling points of 300k across all models. The training batch size is set to 8 and the number of sampling points is 10k per sample. We use Adam optimizer with initial learning rate of 3e-4 and exponentially decayed to 0.99 at every 10k steps. For DISN and OccNet, models in MGN dataset are first converted to watertight form and then sampled with the default strategies used in the original papers. Each watertight model is sampled with 300k points, equivalent to that in 3PSDF. All the other training hyperparameters are set to default values.

MGN dataset is split into training and testing datasets with 9:1 ratio, and all 3 networks are evaluated at 20k epoches.

C.3. Single-view Reconstruction on ShapeNet

We use 17803 shapes from 5 categories of ShapeNet [5] for evaluation, including Airplane, Car, Lamp, Chair and Boat. We use the same image renderings (24 views per shape) and train/test split as Choy et al. [12].

We directly use the raw mesh of ShapeNet to generate the ground truth to train 3PSDF, while the competitive methods are trained using pre-processed watertight meshes. The ground truth 3PSDF values are sampled with resolution 128^3 and the results are evaluated using resolution 256^3. The images are all scaled to the resolution of 224×224. We first train the network for 30 epochs with learning rate 1e-4, and then finetune the network for 80 epochs using learning rate 5e-5. The batch size is set to 8 and the number of sampled points is 20k for one shape in each iteration during training. The reconstruction results are post-processed with simple hole filling and smoothing.

D. Network Structure

D.1. Network Architecture for Shape Reconstruction

Figure 14 shows the detailed network structure for the experiment of shape reconstruction. In particular, the network follows the design of the auto-decoder [29] which does not requires an encoder for learning the shape priors of training data. The input to the decoder contains: 1) a 512-dimensional per-object latent code, that is learned during training, and 2) a point feature obtained after applying point feature extractor to the 3D coordinate of the query point.
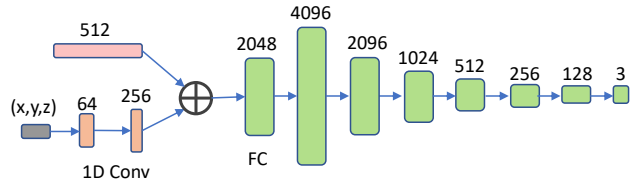

Figure 14. Network structure for shape reconstruction.

The point feature extractor is implemented using 1D convolutional operator. The concatenation of the latent code and the point feature is then fed into the decoder which consists of multiple fully connected layers. The output layer of the decoder predicts the per-class probability for the 3 categories defined by 3PSDF.

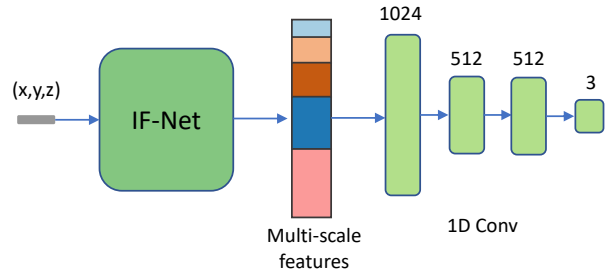### D.2. Network Architecture for Reconstruction from Point Cloud

Figure 15. Network structure for reconstruction from point cloud.

We show the detailed network structure for reconstruction from point cloud in Figure 15. To ensure fair comparison, we use the identical network with NDF [10], which is based on IF-Net [9], for extracting the features from the input point cloud. The extracted multi-scale point features are then fed into the decoder. The decoder is implemented using four 1D convolution layers, where the last layer predicts the per-class probability for 3PSDF.
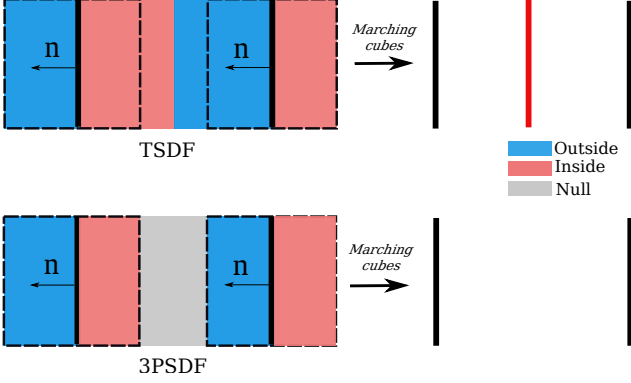
Figure 16. Comparison between TSDF and the proposed 3PSDF. For reconstructing two adjacent single layers of mesh, TSDF would introduce artifacts (the red layer show on the right of first row) to the reconstruction result.

D.3. Network Architecture for SVR

Figure 17 shows the detailed network architecture for 3D reconstruction based on single-view images. The network takes a set of sampled 3D points and a single view image as input. We use several 1D convolution layers to obtain the point features and a VGG-16 (with batch normalization) architecture to encode the input image. We adopt a two-stream network architecture, where the point features are concatenated with global and local image features respectively, and then fed into two branches to predict the 3PSDF.

The global image features are obtained from an average pooling and a fully connected layer at the end of the image encoder. For the local features, we project the input 3D points to the image plane and retrieve the features on each feature map using the projected coordinates. The retrieved features on each feature map are concatenated together to obtain a local image feature vector.

The decoder has two streams with the same structure, each of which consists of a set of fully connected layers to predict the 3PSDF separately. The outputs from the two branches are summed up and passed through a Softmax layer to obtain the final prediction.

E. Comparison with TSDF

Truncated Signed Distance Field (TSDF) is widely used in obtaining reconstruction results from the volumetric range data, e.g. the RGBD stream from depth sensors. One mainstream application of TSDF is large-scale tracking and mapping in reconstructing 3D scenes. As one may have seen open surfaces, e.g. the walls in the reconstructed 3D environment, can be reconstructed using TSDF, we provide detailed comparisons here stating the difference between TSDF and 3PSDF regarding the ability of modeling surfaces with arbitrary topologies.

The motivation of introducing TSDF is to set a lower bound of reconstruction error during the fusion of different SDFs converted from the depth maps. In particular, in real-world scanning, the raw data obtained from the depth sensor is highly likely to be contaminated by the noises. In practice, the depth maps are converted into SDFs in order to fuse the per-frame observation into a more complete reconstruction in the canonical space. However, the most widely adopted way of fusing the SDFs is based on weighted summation, where the errors brought by each SDF would be accumulated and affecting the previously fused results. TSDF alleviates this issue by clipping the minimum and maximum signed distance value and hence prevents the summed TSDFs from deviating too much from the ground-truth value.

After analyzing the motivation of TSDF, we can better understand the difference between TSDF and our proposed 3PSDF. (1) Unlike 3PSDF, TSDF remains a binary-sided signed distance function which only has positive and negative signs. This could render TSDF failed to represent open surfaces without introducing artifacts in many cases. As shown in Figure 16 upper row, for two adjacent surfaces with consistent normals, the positive and negative signs would intersect with each other in the middle region where the SDFs are truncated to maximum and minimum respectively. This leads to an additional surface/artifact (the red boundary on the right) if meshing such a field using the Marching Cubes algorithm. In contrast, 3PSDF can achieve artifact-free reconstruction by inserting a NULL layer in between to prevent the formation of the additional decision boundary. (2) The way that TSDF models open surfaces is completely different from that of 3PSDF. In particular, TSDF generates open surfaces by space clipping, where only the field within a bounded volume is converted into mesh. In comparison, 3PSDF is able to model open surfaces by directly meshing the entire 3D space without requiring a clipping bounding volume.

F. More Results

Reconstruction of closed surfaces from sparse point cloud. We provide more qualitative comparison results with the state-of-the-art approaches on the task of shape reconstruction from sparse point cloud. In Figure 18, we show the reconstruction result using the models trained on preprocessed ShapeNet car data (watertight mesh with inner structure removed) provided by [43].

Reconstruction of complex surfaces from sparse point cloud. In Figure 19 we provide more qualitative comparisons of shape reconstruction results of complex surfaces that contain both closed and open surfaces. All the candidate approaches, including ours, are trained on on raw, unprocessed ShapeNet car data, which contain inner structures and open surfaces. As seen in the highlighted regions
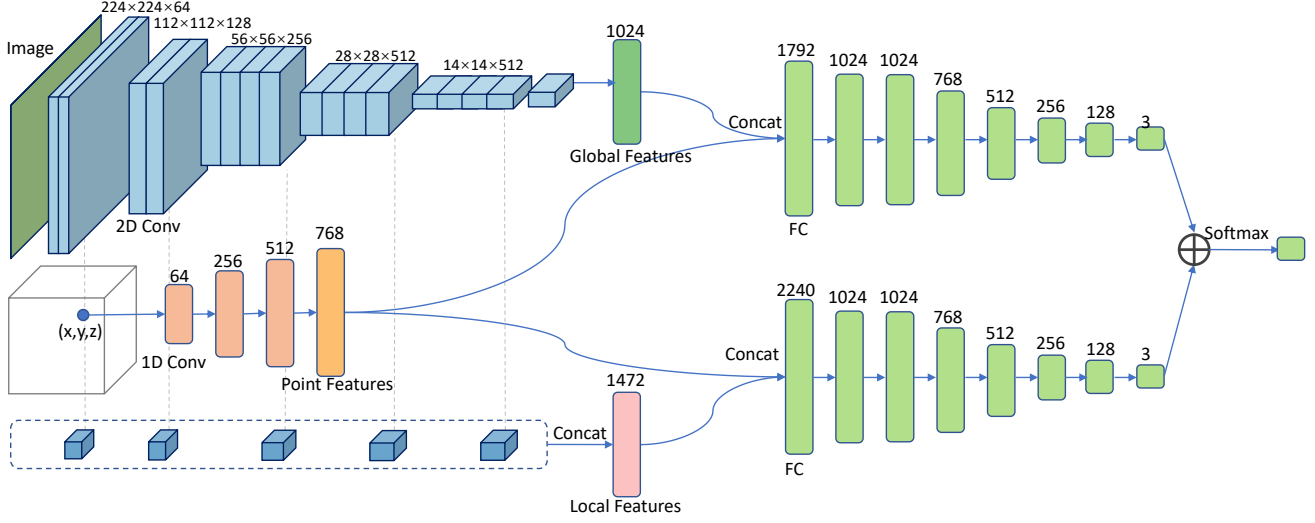
14

Figure 17. Detailed network architecture for single view reconstruction.

within the red rectangles, our approach is able to generate shapes with consistent normals even when the ground truth data may contain flipped face patches.

Reconstruction of 3D scenes from sparse point cloud. In Figure 20, we show more visual comparisons of scene reconstruction results. The input point cloud (for both main paper and supplementary material) contains 50K points. Note that we are not able to generate plausible meshing result for NDF even after experimenting with various parameters of BPA algorithm. Hence we show the raw output point cloud of NDF in the closeup figure.

Single-view reconstruction. We include more qualitative comparison results on the test set of ShapeNet in Figure 21.
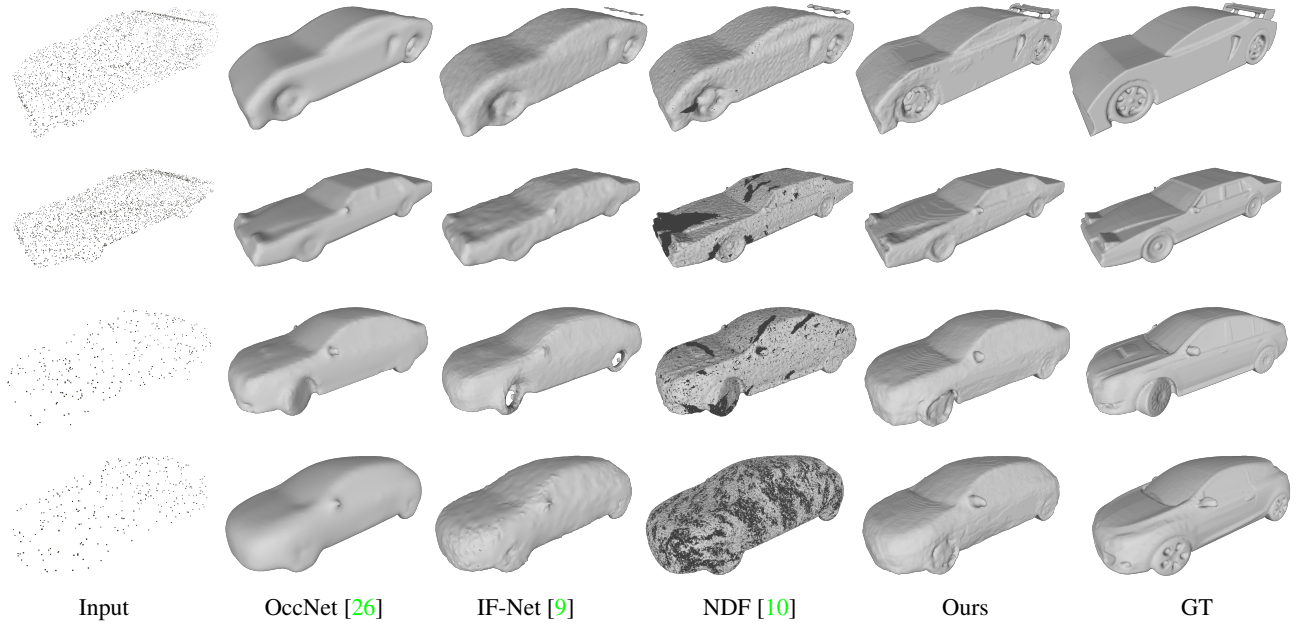
Figure 18. More shape reconstruction results trained on watertight data. We show four groups of results: the first two rows are reconstructed from 3000 points while the last two rows are generated given 300 points.
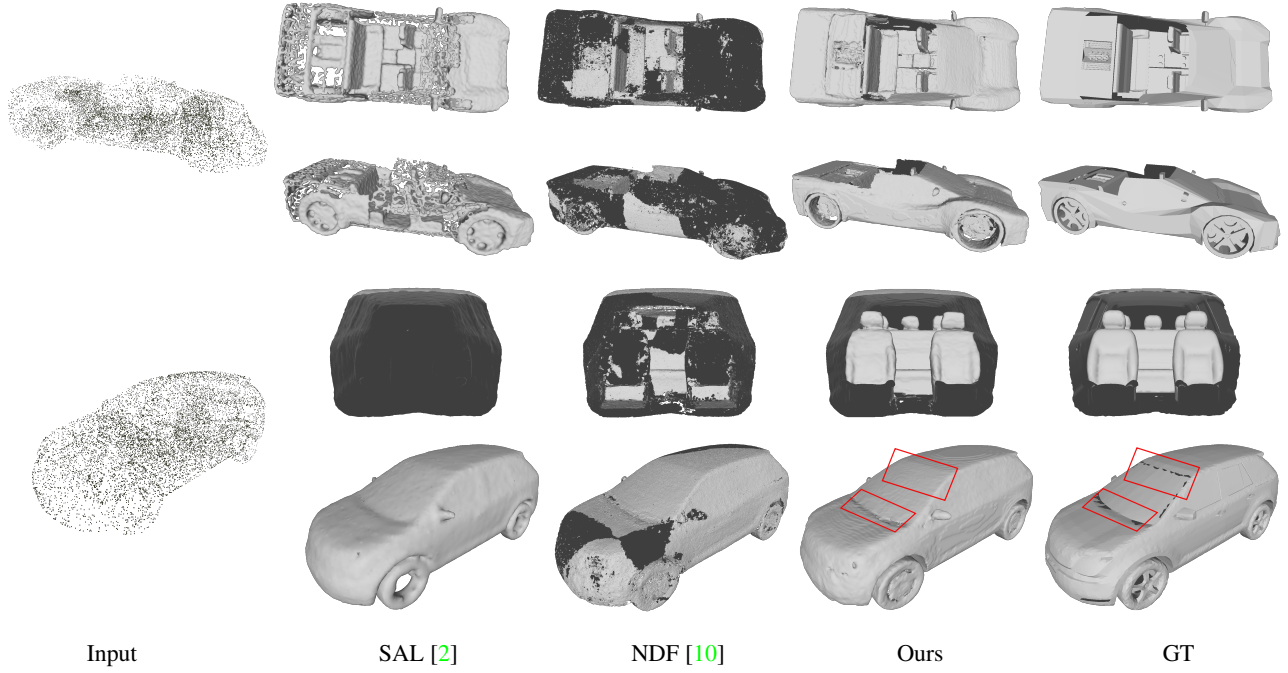

Figure 19. More shape reconstruction results trained on unprocessed, raw data. For each group of results, we show the input (10K points) on the left and two rows of corresponding results on the right. For the second group of result, we show the inner structure of reconstruction on top of an external view. The highlighted regions within the red rectangles show that our method can generat reconstruction results with consistent normals even when the ground-truth data contain flipped triangles.

16

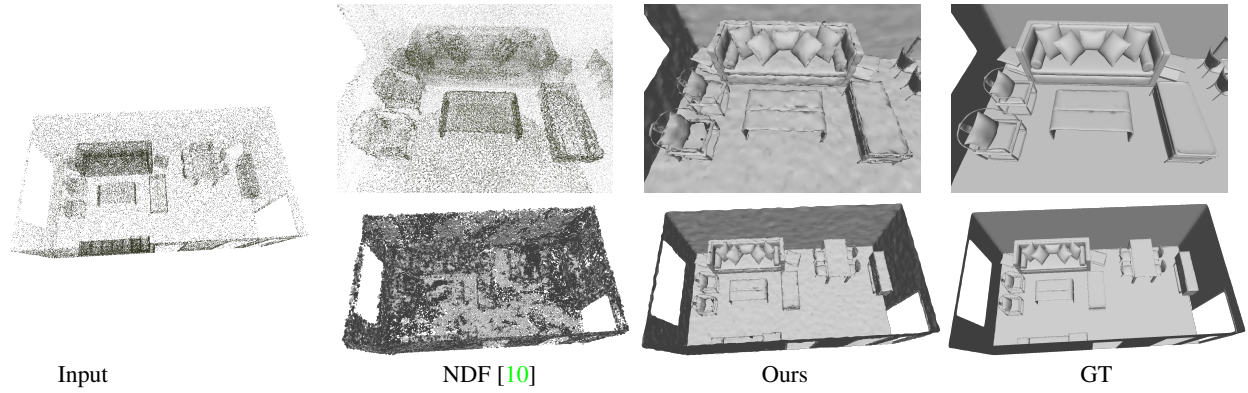Input   NDF [10]   Ours   GT

Figure 20. Scene reconstruction results from sparse point cloud. For each method, we show both the closeups (first row) and the global view (second row). NDF results contain 3 million points. Note that since we are not able to generate plausible meshing results for NDF even after experimenting with various BPA parameters, we show the output raw point cloud in the closeup of NDF. The other results are displayed in mesh form.

Input   IMNet   OccNet   DISN   Ours   GT   Input   IMNet   OccNet   DISN   Ours   GT
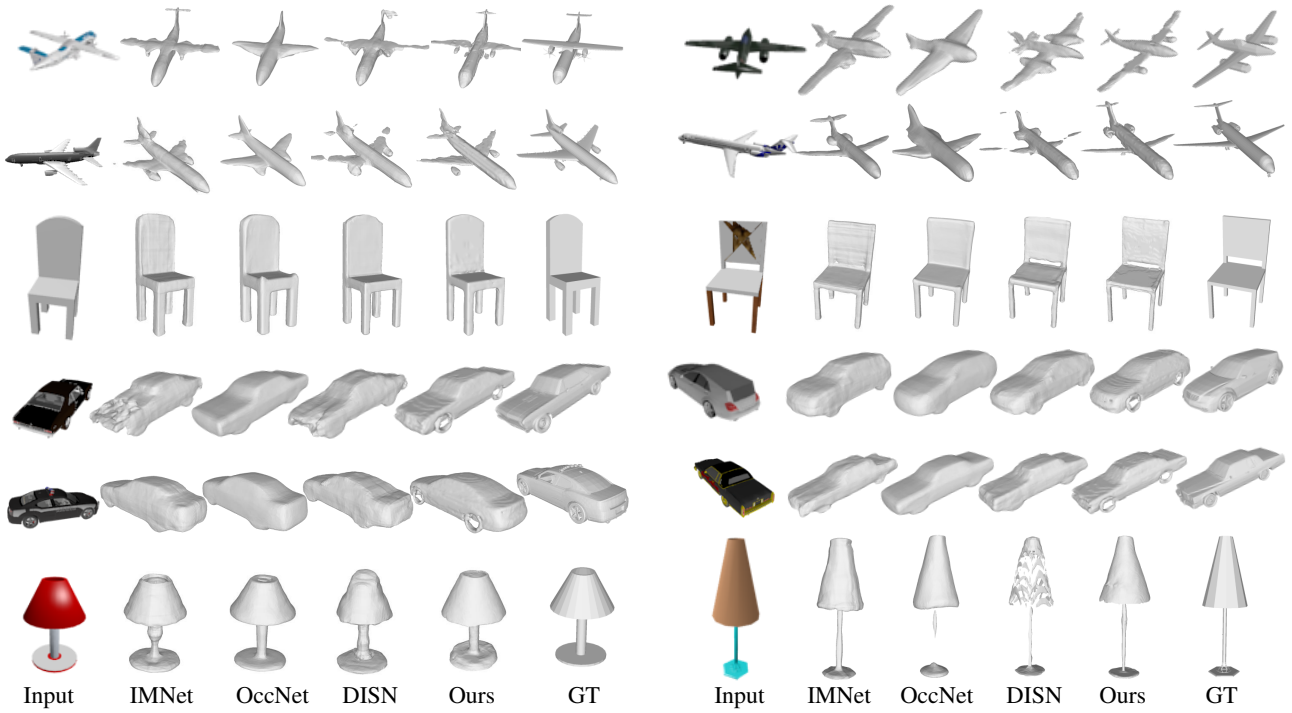
Figure 21. More qualitative comparison results with SOTA single-view reconstruction methods based on implicit functions.