HSDF: Hybrid Sign and Distance Field for Modeling Surfaces with Arbitrary Topologies

Anonymous Author(s)
Affiliation
Address
email

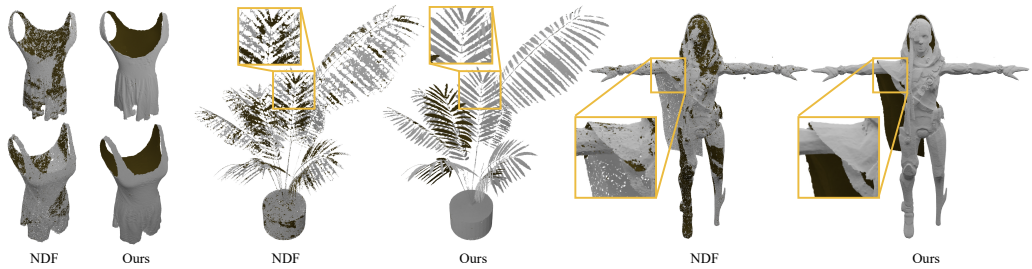


Figure 1: We reconstruct three groups of representative objects with both open and closed surfaces using NDF [13] (left) and our proposed HSDF (right). Compared with the SOTA NDF [13], our method achieves higher surface reconstruction quality and more consistent surface normals. All the results are reconstructed with an equivalent resolution.

Abstract

Neural implicit function based on signed distance field (SDF) has achieved impressive progress in reconstructing 3D models with high fidelity. However, such approaches can only represent closed surfaces. Recent works based on unsigned distance function (UDF) are proposed to handle both watertight and open surfaces. Nonetheless, as UDF is signless, its direct output is limited to point cloud, which imposes an additional challenge on extracting high-quality meshes from discrete points. To address this challenge, we present a novel neural implicit representation coded HSDF, that is a hybrid of signed and unsigned distance fields. In particular, HSDF is able to represent arbitrary topologies containing both closed and open surfaces while being compatible with existing iso-surface extraction techniques for easy field-to-mesh conversion. In addition to predicting a UDF, we propose to learn an additional sign field. Unlike traditional SDF, HSDF is able to locate the surface of interest before level surface extraction by generating surface points following NDF [13]. We are then able to obtain open surfaces via an adaptive meshing approach that only instantiates regions containing surface into a polygon mesh. We also propose HSDF-Net, a dedicated learning framework that factorizes the learning of HSDF into two easier sub-problems. Experiments and evaluations show that HSDF outperforms the state-of-the-art techniques both qualitatively and quantitatively.

1 Introduction

Recent advances in neural implicit representation [10, 35, 38, 43] have set a new state of the art in 3D modeling and reconstruction by breaking the previous barrier in resolution and topology. However, as such approaches rely on the signed distance function (SDF) which divides the space into inside and

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

outside of the object, they are limited to representing closed shapes. To lift the limitation, methods based on unsigned distance function (UDF) [13, 47, 45] are proposed such that a much broader class of shapes containing open surfaces can be effectively represented and learned via deep neural networks.

However, as UDF is signless, directly applying the iso-surface extracting technique, e.g. the Marching Cubes algorithm, would convert all open surfaces into closed mesh. To generate open structures, these approaches have to convert the resulting UDF field into discrete points and then apply the Ball-Pivoting algorithm [4] (BPA) to obtain the meshing result. Nonetheless, the BPA technique is prone to introduce self-intersections and disconnected surface patches with inconsistent normals (see Figure 1). In addition, BPA is highly sensitive to the input parameters and often requires per-shape parameter tuning in order to generate a complete meshing result. This hinders UDF-based approaches from being practically used in real-world applications as mesh remains the prominent standard for modeling and rendering in both industry and academia.

To address the above issue, we present a novel learnable implicit representation, named *Hybrid Sign and Distance Function (HSDF)*, that can faithfully represent complex geometry containing both closed and open surfaces, while being compatible with off-the-shelf iso-surface extraction methods, e.g. the Marching Cubes algorithm, for easy and high-quality field-to-mesh conversion. The key idea of HSDF is to integrate the advantages of both SDF and UDF while avoiding their adversarial impacts. We empirically find that the learning of UDF is quite robust and can generalize well to novel data. Therefore, to inherit the benefit of UDF and overcome its limitation, we propose to learn an additional sign field in addition to UDF via a sign predictor. Unlike traditional SDF, HSDF is able to locate the surface of interest before performing level surface extraction. We achieve this by generating surface points via the gradient field of unsigned distance function following NDF [13]. Hence, we are able to create local SDFs by multiplying the UDF with the sign field and cast complex shapes containing both closed and open surfaces by incorporating an adaptive meshing algorithm that only instantiates the regions containing surface points into polygon mesh.

As HSDF may be discontinuous at regions far from the surface due to the sudden change of signs, we propose to factorize the learning of HSDF into two easier sub-tasks, each of which learns a continuous function. Specially, we propose HSDF-Net that learns the unsigned distance and sign field individually using the same input in a separate manner. A fusion framework is also proposed to faithfully fuse the predicted distance and sign field at the inference time. Finally, we introduce an adaptive masked Marching Cubes algorithm for converting HSDF into high-quality mesh at flexible resolution efficiently. Extensive experiments show that HSDF can outperform the previous state-of-the-art methods in both qualitative and quantitative measurements. We summarize the key contributions as follows:

- We introduce HSDF, a novel neural implicit field that can faithfully represent complex shapes with closed and open surfaces while being compatible with existing level surface extraction techniques.
- We propose a dedicated learning framework that separates the decoding of sign and distance in accordance with the definition of HSDF.
- We propose an adaptive masked Marching Cubes algorithm to generate meshes from HSDF efficiently.

2 Related Works

3D neural shape representation is a very fundamental and popular research topic in computer vision and graphics. In this section, we will review some recent advances on neural shape representations from two aspects: Explicit & Implicit Representation. Some survey papers [25, 6, 1, 54] have comprehensively summarized the development of 3D shape representation.

Learning on 3D Explicit Representations. For the learning on 3D shapes, various 3D representations are applied to the learning-based approaches, such as point clouds, voxels, and polygonal meshes. The voxel represents the concrete 3D shapes with regular domain which is directly an extension of 2D pixel grids, and it is widely used in 3D deep learning. Many previous works [33, 16, 53] investigated this representation by extending the 2D deep learning operators to the 3D domain. However, 3D voxel representation suffers from low-resolution, high-computation, and memory-consumption

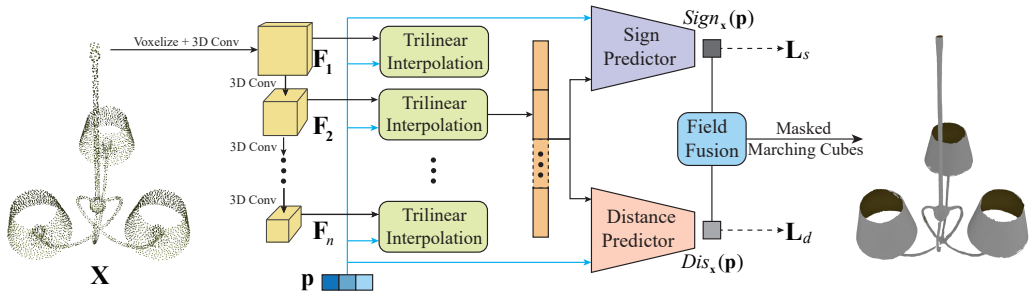


Figure 2: HSDF-Net architecture. The input sparse point cloud is voxelized and encoded in a multi-scale manner into a shape code. Next, the distance predictor takes the shape code and query point $\mathbf{p}$ as input to predict an unsigned distance $Dis(\mathbf{p})$. The sign predictor takes the same input and predicts a signed value $Sign(\mathbf{p})$. The field fusion module (Sec. 3.3) is proposed to fuse $Dis(\mathbf{p})$ and $Sign(\mathbf{p})$. The rightmost lamp example is reconstructed from test data using the proposed adaptive masked Marching Cubes algorithm (Sec. 3.4).

drawbacks. Further, Wang et al.tried their best to address the above drawbacks by introducing the octree-based deep learning methods [49, 51, 50, 31]. The point cloud is flexible and easier to be captured by some portable 3D scanners (*e.g.* Microsoft Kinect), but the main challenge is that the fidelity is too low to represent high-quality shape geometry and topology. Due to its irregular and disordered structure, the traditional deep learning methods cannot directly be applied, Qi et al.firstly introduced PointNet [40] and PointNet++ [41] for 3D classification and segmentation by pooling operation which is order-independent. Wang et al. [52] constructed a sub-graph by KNN and applied the graph neural network on the point cloud to learn the local geometric features for shape analysis [8]. The predefined connections of the vertices set to limit the flexibility and topology. Some works [9, 29, 23, 42, 30, 24] exploited the topology of polygonal meshes for shape analysis. There is a recent trend [22, 48, 26, 32, 37, 20, 56] of studying the deformable meshes to represent the detailed geometry.

Learning on 3D Implicit Representations. Implicit function is a powerful tool capable of modeling arbitrary geometric details and topological structures. With the help of deep neural networks, a complex shape can be implicitly represented by classifying the query point in/out-side a shape (Binary Occupancy) [34, 39, 15, 21, 44, 18, 11] or predicting the continuous signed distance (SDF) to the shape surface [10, 35, 38, 27, 43], where the sign of SDF determines the query point in/out-side of a shape. The binary occupancy and signed distance function can only represent the watertight closed surface, since it is essential that partition the set of query points inside or outside of the shape. For the open surfaces (*e.g.* cloth, thin-shell models), Atzmon et al. [2] developed a deep learning approach – Sign Agnostic Learning (SAL) for raw data with any inside/outside labeling, but the prediction of SAL is still SDF for the close surface reconstruction. Furthermore, the concurrent works NDF [14] and DUDE [47] proposed neural-based methods that predict the *unsigned* distance field to represent arbitrary surfaces without any closed surface data. NDF adopts the multi-scale encoding techniques from IF-Net [12] to enhance the surface details. DUDE introduces a disentangled shape representation that utilizes the unsigned distance field (UDF) and normal vector field (nVF), which learns high fidelity representations. Due to the gradient vanishing of UDF close to the surface, UDF fails to predict high-quality estimation of high-ordered geometry attributes (*e.g.* normal, tangent plane), CSPNet [46] represents such surfaces by a class of implicit representation – closest-surface-point (CSP), which achieves local geometric attributes accurately and efficiently.

Recent concurrent works such as GIFS [57] and DeepCurrents [36] also focus on modeling open surfaces. DeepCurrents [36] combines the explicit boundary curves and the implicit field. And the final open surfaces are obtained by solving a minimal surface problem, which is not as flexible as pure implicit field to model arbitrary shapes. Another work GIFS [57] models the relationships between every two points, which is not as efficient as modelling the relationships between points and surfaces. And the reconstructed surface normals will be different from the ground truth meshes because point-surface relationships are missing.

In contrast, HSDF can efficiently model arbitrary shapes with both open and closed surfaces and achieves high-quality reconstruction with more consistent surface normals similar to the ground truth using the proposed adaptive masked Marching Cubes algorithm.

3 Method

In this section, we introduce the proposed shape representation and our HSDF-Net with an adaptive masked Marching Cubes algorithm into high fidelity single-layer meshes. A diagram of our system architecture is shown in Fig. 2. We first define the proposed implicit shape representation (Sec. 3.1). Then we introduce the model structure of HSDF-Net and explain our learning strategy (Sec. 3.2). Finally, we illustrate the approach to fuse the hybrid distance and sign together to form a HSDF (Sec. 3.3) and the adaptive masked Marching Cubes algorithm (Sec. 3.4).

3.1 Shape Representation

We define the target surface as $\mathcal{S}$. Our proposed formulation takes a vectorized shape code $\mathbf{z} \in \mathcal{Z}$ and a point $\mathbf{p} \in \mathbb{R}^3$ as input, and predicts a hybrid sign and distance function $\mathrm{HSDF}(\mathbf{p}, \mathbf{z}) : \mathbb{R}^3 \times \mathcal{Z} \mapsto \mathbb{R}$. If we define $\mathbf{p}'$ as the closest point of $\mathbf{p}$ on the surface $\mathcal{S}$ and the surface normal of $\mathbf{p}'$ as $\mathbf{v}$. Then unsigned distance can be computed as $\mathrm{UDF}(\mathbf{p}, \mathbf{z}) = \|\mathbf{p} - \mathbf{p}'\|_2$ and the sign is calculated as $\mathrm{SIGN}(\mathbf{p}, \mathbf{z}) = \mathrm{sign}(\mathbf{v} \cdot (\mathbf{p} - \mathbf{p}'))$, where $\mathrm{sign}(\cdot)$ returns the sign of the input. Specifically, $\mathrm{SIGN}(\mathbf{p}, \mathbf{z})$ is positive if $\mathbf{v}$ aligns with $\mathbf{p} - \mathbf{p}'$ and negative otherwise. The overall HSDF expression can be defined as:

$$\begin{aligned} \mathrm{HSDF}(\mathbf{p}, \mathbf{z}) &= \mathrm{SIGN}(\mathbf{p}, \mathbf{z}) \cdot \mathrm{UDF}(\mathbf{p}, \mathbf{z}) \\ &= \mathrm{sign}(\mathbf{v} \cdot (\mathbf{p} - \mathbf{p}')) \cdot \|\mathbf{p} - \mathbf{p}'\|_2 \\ \text{s.t. } \mathbf{p}' &= \underset{\mathbf{p}_s \in \mathcal{S}}{\operatorname{argmin}} \|\mathbf{p} - \mathbf{p}_s\|_2 \end{aligned} \tag{1}$$

By this definition, we are able to locate the surface of interest by searching for the points on the surface following the paradigm of [13] on leveraging the gradient of $\mathrm{UDF}(\cdot)$. In addition, the introduced sign field SIGN can convert the UDF into a signed one, enabling the use of the iso-surface extraction technique for easy field-to-mesh conversion. By incorporating an adaptive masked Marching Cubes algorithm (detailed in Section 3.4) that only instantiates the regions containing surface points into mesh, we are able to model complex topology with open surfaces.

3.2 HSDF-Net: Model Architecture

For regions that are far from the surface, the HSDF representation may be discontinuous as the sign field may undergo a sudden change. To ease the learning of HSDF, we factorize the learning into two sub-problems: learning of the unsigned distance and the sign field. As depicted in Fig. 2, we propose HSDF-Net that contains three components: a shared volume encoder, a distance predictor, and a sign predictor. The shape encoder extracts features from input sparse point clouds in a multi-scale manner, which follows IF-Net [12]. Then we feed encoded multi-scale shape code and query point $\mathbf{p}$ into the sign and the distance predictors in Fig. 2 to get a sign prediction $Sign(\mathbf{p})$ and a distance prediction $Dis(\mathbf{p})$. Next, we will introduce the formulation of our network components respectively in detail.

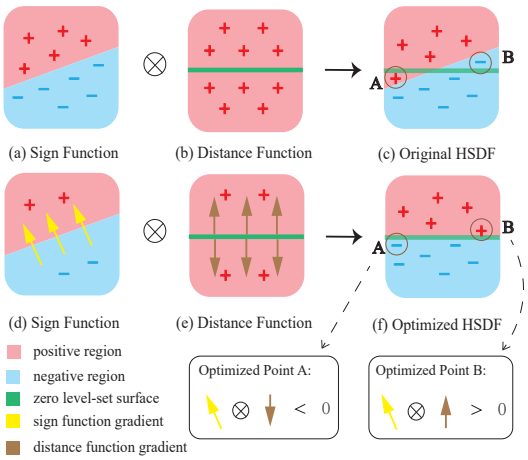


Figure 3: 2D illustration of sign and distance fusion. Assume green line is the target surface. **Row 1:** By multiplying sign and distance point-wise, we can obtain an original HSDF, where some points (e.g. A and B) may be wrongly signed. **Row 2:** By using gradients of sign function (*i.e.* yellow vectors) and gradients of distance function (*i.e.* brown vectors) to optimize the original fused HSDF, HSDF of points A, B can be effectively rectified (*i.e.* green line).

Volume Encoder: Our goal is to reconstruct an arbitrary surface $\mathcal{S}$ from a sparse point cloud $\mathbf{X} \in \mathcal{X}$. The input point cloud is first voxelized and encoded by 3D CNNs into multi-scale grid features $\mathbf{F}_1, .., \mathbf{F}_n, \mathbf{F}_k \in \mathcal{F}_k^{K \times K \times K}$, where K is the grid size and $\mathcal{F}_k \in \mathbb{R}^C$ is a feature with C channels. If we denote the result of trilinear interpolation of grid feature $\mathbf{F}_i$ at position $\mathbf{p}$ as

$\mathbf{T}_i$, then $\Psi_{\mathbf{x}}(\mathbf{p}) = (\mathbf{T}_1, \ldots, \mathbf{T}_n)$ can represent the concatenated shape code for input $\mathbf{X}$ at point $\mathbf{p}$ and $\Psi_{\mathbf{x}}(\mathbf{p}) : \mathbb{R}^3 \mapsto \mathcal{F}_1 \times \ldots \times \mathcal{F}_n$ defines the encoder function, see IF-Net [12].

Distance Predictor: The distance predictor predicts the UDF between points and the ground truth surface. We formulate this as a function $\Phi\left((\mathbf{T}_1, \ldots, \mathbf{T}_n)\right) : \mathcal{F}_1 \times \ldots \times \mathcal{F}_n \mapsto \mathbb{R}^+$, and we use fully connected layers with ReLU activation functions to guarantee $\Phi \geq 0$. In general, the distance predictor could be formulated as $Dis_{\mathbf{x}}(\mathbf{p}) = (\Phi \circ \Psi_{\mathbf{x}})(\mathbf{p}) : \mathbb{R}^3 \mapsto \mathbb{R}_0^+$.

Sign Predictor: We extend sign concept to arbitrary shapes. For example, a T-shirt model has consistent outward face normal directions, and we can easily tell the sign of a query point $\mathbf{p}$ even though a T-shirt is obviously an open surface because we will intuitively choose the closest point $\mathbf{p}'$ on the surface and use the normal direction of p' as a clue of the sign. Detailed formulation is defined above in Sec. 3.1. For cars with inner structures, we also compute the sign of a query point $\mathbf{p}$ according to its closest surface normal direction, no matter whether $\mathbf{p}$ is inside the car or close to the outside surface.

Therefore, we extend the concept of "signed distance" to a broader range of shapes like open shapes by combining "distance" and "sign" computation and prediction processes. We define $\Theta\left((\mathbf{T}_1, \ldots, \mathbf{T}_n)\right) : \mathcal{F}_1 \times \ldots \times \mathcal{F}_n \mapsto \mathbb{R}_0$ as a sign predictor. By composition we obtain our sign predictor $Sign_{\mathbf{x}}(\mathbf{p}) = (\Theta \circ \Psi_{\mathbf{x}})(\mathbf{p}) : \mathbb{R}^3 \mapsto \mathbb{R}_0$ which we treat as a regressor to the signed distance. Hence, the predicted binary sign can be computed as $sign(Sign_{\mathbf{x}}(\mathbf{p})) : \mathbb{R}^3 \mapsto [-1, 1]$ where $\mathrm{sign}(\cdot)$ returns the sign of the input. And the ground truth binary sign can also be represented as $SIGN_{\mathbf{x}}(\mathbf{p}) = sign(\mathrm{SDF}(\mathbf{p}, \mathcal{S}_{\mathbf{x}})) : \mathbb{R}^3 \mapsto [-1, 1]$.

Learning: We define the volume encoder, distance predictor and sign predictor as $\mathbf{w_e}$, $\mathbf{w_d}$ and $\mathbf{w_s}$, respectively. During distance regression training, we use L_1 mini-batch loss to jointly optimize $\mathbf{w_e} \cup \mathbf{w_d}$:

$$\mathcal{L}_d := \sum_{\mathbf{x} \in \mathcal{B}} \sum_{\mathbf{p} \in \mathcal{P}} \left| \min\left(Dis_{\mathbf{x}}(\mathbf{p}), \delta\right) - \min\left(\mathrm{UDF}\left(\mathbf{p}, \mathcal{S}_{\mathbf{x}}\right), \delta\right) \right|$$

where $\mathcal{B}$ and $\mathcal{P}$ stand for a mini-batch and a sub-sample of points. Clamping the maximal regressed distance to value $\delta > 0$ forces the model capacity to focus more on the vicinity of the surface.

During sign predictor training, we use L_1 mini-batch loss to jointly optimize $\mathbf{w_e} \cup \mathbf{w_s}$:

$$\mathcal{L}_s := \sum_{\mathbf{x} \in \mathcal{B}} \sum_{\mathbf{p} \in \mathcal{P}} \left| \min\left(Sign_{\mathbf{x}}(\mathbf{p}), \delta\right) - \min\left(\mathrm{SDF}\left(\mathbf{p}, \mathcal{S}_{\mathbf{x}}\right), \delta\right) \right|$$

Finally, we train our HSDF-Net in an end-to-end manner by optimizing the total loss: $\mathcal{L}_{total} := \mathcal{L}_d + \mathcal{L}_s$.

3.3 Sign and Distance Field Fusion

The HSDF-Net trains a shared volume encoder and two decoders respectively for the sign and distance prediction. With this learning manner, we can obtain a predicted sign function $Sign_{\mathbf{x}}(\mathbf{p})$ and a predicted distance function $Dis_{\mathbf{x}}(\mathbf{p})$ of query points $\mathbf{p}$ after training. Although $Sign_{\mathbf{x}}(\mathbf{p})$ and $Dis_{\mathbf{x}}(\mathbf{p})$ are both learned continuous function, they can be fused into a hybrid sign and distance function $\mathrm{HSDF}(\mathbf{p}, \mathcal{S})$ with our fusing algorithm illustrated in Fig. 3. In Fig. 3, we plot a local region in 2D of learned sign function $Sign_{\mathbf{x}}(\mathbf{p})$ and distance function $Dis_{\mathbf{x}}(\mathbf{p})$ where $\mathbf{X}$ and $\mathbf{p}$ is the input sparse point cloud and the query point, respectively. The first row of Fig. 3 describes a straightforward fusion strategy that multiply distance with sign pointwisely: $\mathrm{HSDF}(\mathbf{p}, \mathbf{X}) = Sign_{\mathbf{x}}(\mathbf{p}) \cdot Dis_{\mathbf{x}}(\mathbf{p})$.

In the second row of Fig. 3, we design a simple but effective optimization using the gradients of

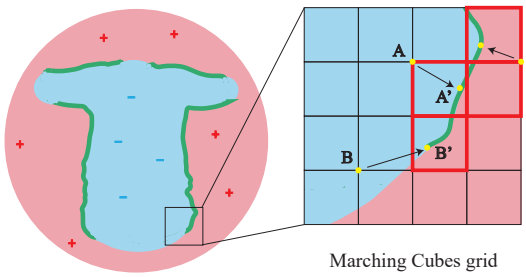


Figure 4: A 2D illustration of mesh extraction on our fused HSDF. Green contour indicates an open shape in 2D. We push Marching Cubes grid points like A and B to their closest points on the surface, i.e., A' and B', respectively, using the gradients of distance function (*i.e.* black vectors). All the boxes in red enclosing surfaces form a mask for Marching Cubes to extract the complex shapes into meshes.

the predicted sign function $Grad_s = \nabla Sign_{\mathbf{x}}(\mathbf{p})$ and the gradients of the predicted distance function $Grad_d = \nabla Dis_{\mathbf{x}}(\mathbf{p})$ to optimize the predicted signs of points like A, B. As described in Fig. 3, if $Grad_s \cdot Grad_d$ is negative as point A, then the sign will be optimized to negative and vice versa for optimizing point B. Since $Grad_s$ and $Grad_d$ can be inferred directly using the backward operation of deep neural networks, this optimization is not time-consuming.

3.4 Mesh Extraction

Our fused HSDF enables us to use efficient meshing methods like Marching Cubes. Hence, we propose an adaptive masked Marching Cubes algorithm. A 2D T-shirt example of fused HSDF is depicted on the left side in Fig. 4. Assume the green contour stands for the open surface, which disconnects at the collar, sleeves, and waist. The red/ blue region indicates where the sign is positive/ negative.

Next, we set up a Marching Cubes grid to mesh the surface $\mathcal{S}$ in green. All the grid points can be pushed to the underlying surface by the gradients of the distance function. Among them, we take points like A, B for example. We define $\mathbf{p}_A$ and $\mathbf{p}_B$ as the coordinate of A and B. Then the coordinate of the closest surface point of A (*i.e.* A') can be computed as $\mathbf{p}_{A'} = \mathbf{p}_A - Dis_{\mathbf{x}}(\mathbf{p}_A) \cdot \frac{\nabla Dis_{\mathbf{x}}(\mathbf{p}_A)}{||\nabla Dis_{\mathbf{x}}(\mathbf{p}_A)||_2}$ where $-\frac{\nabla Dis_{\mathbf{x}}(\mathbf{p}_A)}{||\nabla Dis_{\mathbf{x}}(\mathbf{p}_A)||_2}$ represent a unit vector towards the underlying surface. And we compute $\mathbf{p}_{B'}$ and all other points by the same process. After all the grid points are pushed to the surface, we can easily pick out those boxes (in red) which enclose surfaces. And these selected cubes form a mask for us to find out where the surfaces lie (works for arbitrary shapes). Finally, a simple masked Marching Cubes can be applied to our HSDF to extract any complex shapes.

4 Experiments

In this section, we validate HSDF on 3D shape reconstruction from sparse point clouds and demonstrate that HSDF can reconstruct watertight shapes on par with SOTAs. And then we show that our approach outperforms SOTA techniques on reconstructing complex shapes such as cars with inner structures in terms of quality and speed.

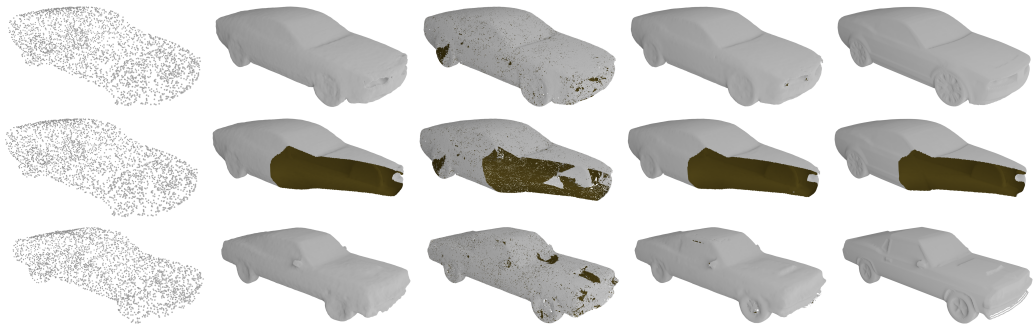
4.1 Data Preparation and Metrics

HSDF Computation. The HSDF computation follows a similar manner as mesh-to-sdf [28]. For every spatial point $\mathbf{p}$, we find the nearest point $\mathbf{p}'$ on the mesh surface and its corresponding surface normal $\mathbf{v}$. If the dot product of $\mathbf{v}$ and $\mathbf{p}' - \mathbf{p}$ is negative, we assign its sign to be positive with its distance to be $|\mathbf{p}' - \mathbf{p}|$ and vice versa. Please refer to the supplementary for further experimental details and results.

Computation for Complex Shapes. We compute HSDFs on repaired ShapeNet dataset [17] on four typical categories with complex shapes, namely cars (7,497 models) with inner structures, chairs (6,579 models), lamps (2,316 models), and ships (1,851 models) with thin open surfaces. We also attain HSDFs on 307 garments from MGN [5] because garments are also a typical category comprised of open surfaces.

Computation for Closed Shapes. In order to compare the performance on watertight shapes, we compute HSDFs on 3,094 ShapeNet [7] cars, which have been converted to closed shapes by DISN [55]. If the model only contains closed surfaces, HSDF is reduced to vanilla SDF, which also conforms to our intuition, because HSDFs are the natural extension of SDFs to represent a wider range of shapes.

Metrics. We follow the commonly used reconstruction metrics to evaluate the reconstructed meshes' quality: Chamfer distance (CD) [3] and Normal consistency (NC) [34] with changes to take face directions into account. OccNet [34] defines a normal consistency score as the mean **absolute** dot product of the normals in one mesh and the normals at the corresponding nearest neighbors in the other mesh. And we removed the absolute operation to punish those flipped mesh faces compared to their ground truth.



(a) Input (b) IF-Net (c) NDF (d) Ours (e) GT

Figure 5: Comparison between different methods on reconstructing closed shapes (without inner structure) from point clouds. The first row and the third row are independent experiments. The second row visualizes the inner reconstruction of the example in the first row. Input point number is $3,000$ for all the results.

Dataset		MGN		Car		Chair		Ship		Lamp		Mean	
#Points		10K	3K	10K	3K	10K	3K	10K	3K	10K	3K	10K	3K
Chamfer Distance	Input	0.879	1.70	0.683	1.91	0.814	2.42	0.327	0.933	0.350	1.08	0.611	1.61
	W.GT	3.46	3.46	4.87	4.87	2.93	2.93	3.79	3.79	5.32	5.32	4.07	4.07
	NDF	**0.102**	0.158	0.137	0.325	0.122	0.291	**0.104**	0.207	**0.142**	**0.231**	**0.121**	0.242
	Ours	0.114	**0.151**	**0.124**	**0.289**	**0.101**	**0.209**	0.133	**0.194**	0.166	0.274	0.128	**0.223**
Normal Consistency	NDF	0.517	0.390	0.383	0.310	0.423	0.344	0.445	0.358	0.541	0.505	0.462	0.381
	Ours	**0.929**	**0.900**	**0.711**	**0.646**	**0.835**	**0.749**	**0.826**	**0.732**	**0.878**	**0.760**	**0.836**	**0.757**

Table 2: Comparison on Chamfer distance($\times 10^{-4}$) & Normal Consistency considering face normal direction.

4.2 Shape Fitting of Single Model

To demonstrate HSDFs can represent complex shapes with both closed and open surfaces, and can be learned by HSDF-Net, we fit models using NDF [13] and our HSDF-Net. For NDF, we extract meshes following the same process as NDF [13], namely using the Ball-Pivoting algorithm to mesh dense point clouds generated by a trained NDF.

As shown in Fig. 1, NDF results in poor quality. The holes and flipped faces cannot be easily optimized by post-processing techniques like closing holes or recomputing normals. Our approach can robustly reconstruct complex shapes with both open and closed surfaces as shown in Fig. 1, such as the dresses, plant leaves and human characters. We also compare the timing and memory consumption between two single-layer mesh prediction approaches: HSDF-Net with MCubes (ours) and NDF. As shown in Table 1, our approach outperforms NDF in time and memory.

Cost	Method	Resolution		
		64^3	128^3	256^3
Time	NDF	89s	58m	780m
	Ours	**5s**	**17s**	**95s**
Memory	NDF	17M	64M	1276M
	Ours	**1M**	**3M**	**10M**

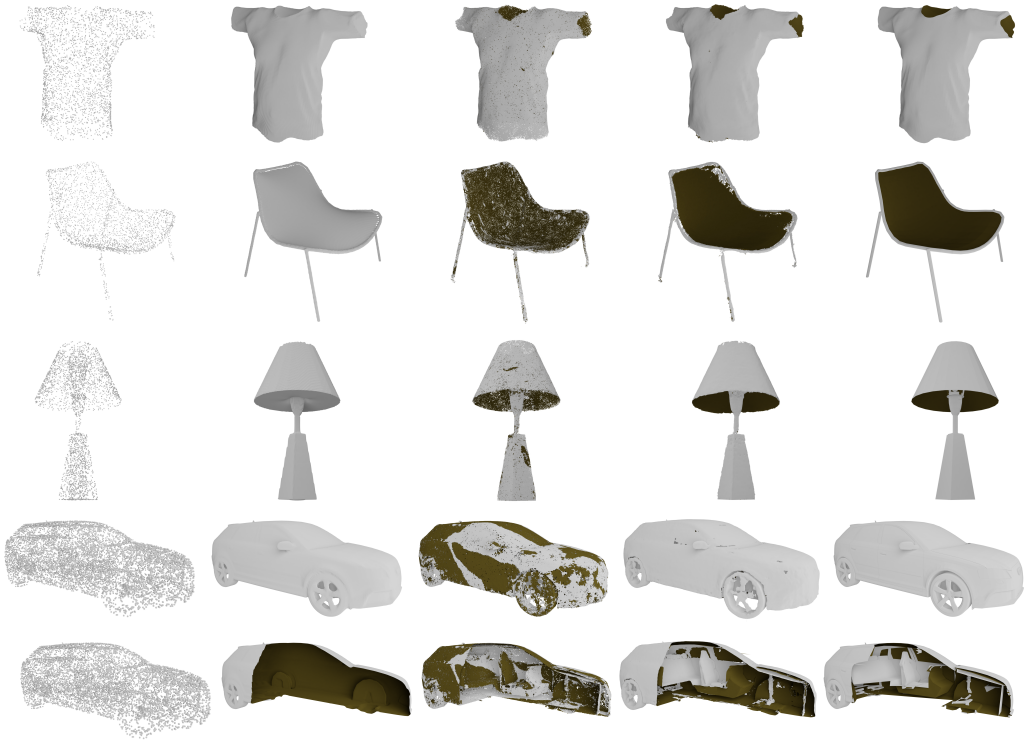
Table 1: Comparison on average meshing time and memory budget per model. Our HSDFs can reconstruct single-layer meshes much faster with a much lower memory budget.

4.3 Shape Reconstruction of Closed Surfaces

Most of the SOTA such as OccNet [34] and IF-Net [12] require watertight training data. In order to compare with these SOTA methods, we train these methods along with NDF [13] on 3094 ShapeNet [7] cars pre-processed by DISN [55]. The pre-processing transform open shapes into closed shapes and removes all the

# Points	Input	OccNet	IF-Net	NDF	Ours
300	12.5	2.53	2.76	1.51	**1.34**
3000	1.54	1.17	0.615	0.191	**0.155**

Table 3: Comparison on Watertight Cars (DISN dataset) using Chamfer-L_2 Distance($\times 10^{-4}$).



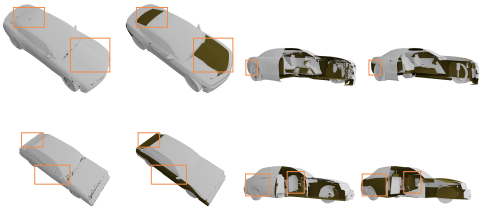
(a) Input (b) W.GT (c) NDF (d) Ours (e) GT

Figure 6: Shape reconstruction comparison of complex shapes. Dark brown area represents the back-faces. **(a)** input point clouds with 3k points for clothes, chairs and lamps; and 10k points for cars to sample complex inner structures. **(b)** the watertight version of GT pre-processed by DISN [55], which erases the inner structures and closes the open surfaces. This is the upper bound of SDF based approaches. **(c)** reconstructed meshes using BPA [4] on NDFs [13]. The results of NDF suffers because BPA is very sensitive to ball-radius, which have to be tuned per-shape [45]. **(d)** our meshes reconstructed with the proposed HSDF, which preserves the thin shapes as in **(e)** GT.

interior contents of cars like seats. In Table 3, we compare the Chamfer Distance [19] of our method against all the baselines quantitatively with 300 input points and 3,000 input points. Our method achieves the best performance. In Fig. 5, we show that our HSDFs can achieve comparable quality as UDF-based SOTA [13] and SDF-based SOTA [12]. All the results are reported using test data unseen during training time.

4.4 Shape Reconstruction of Complex Shapes

In order to demonstrate HSDFs can reconstruct meshes with better quality efficiently, we train our HSDF-Net on five representative categories containing complex shapes illustrated in Sec. 4.1. In Table 2, we compare our method quantitatively against NDF and watertight ground truth (W.GT) on the complex shapes with 3,000 and 10,000 points as input. The visual comparisons are shown in Fig. 6 where we can see thin open structures (*i.e.* the clothes, the back of the chair, and the hat of the lamp) and complex inner structures (*i.e.* the seats and windows, etc.). All these results are reported on test data unseen during training time. As depicted in Fig. 6 and Fig. 1, our HSDF is the first representation that can efficiently reconstruct high-fidelity single-layer



(a) Ours (b) GT (c) Ours (d) GT

Figure 7: Reconstructed meshes with face directions corrected. Above are two examples from test data. And they have more reasonable face directions even than their noisy ground truth meshes which have flipped faces. This shows that HSDF-Net learns a pattern of face normal directions from the prior knowledge, which can be robust to small noises.

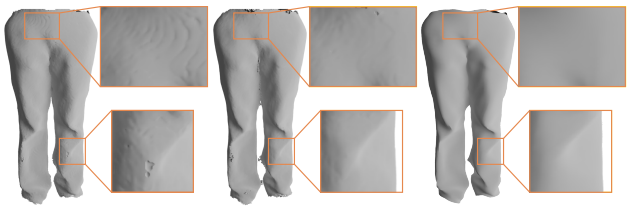
open mesh with reasonable face normal directions. These advantages are only possible because 1) HSDFs can model open shapes by hybrid distance and sign; 2) HSDF-Net can learn the prior knowledge of face normal direction in the form of signs, which results in reasonable face directions and also enables using marching cubes to accelerate meshing; 3) Our two-head training manner enables using deep neural networks (known to represent continuous functions) to learn HSDFs. Although we computed HSDFs using surface normal information in training data illustrated in Sec. 4.1, our HSDF-Net is still robust to noise like flipped faces in training data. We can even use the learned HSDF-Net to reconstruct meshes with more reasonable face directions than their ground truth with only point clouds as input, see Fig. 7.

4.5 Further Discussions

Necessity of the two-head training strategy We validate the necessity of our two-predictor training strategy for sign and distance by jointly training a single decoder to predict a signed distance instead of two decoders. All the results are from test data and reconstructed in the same condition except the number of decoders. As we can see in column (b) in Fig. 8, although the HSDFs for training are sampled in the same way as two-head training, joint training can never learn HSDFs correctly because a discontinuous function is beyond what a single DNN can express. In contrast, our results in column (c) are a plausible open surface that is similar to the ground truth.

(a) Input (b) Joint (c) two-head (d) GT

Figure 8: Ablation study on two-head training. The reconstructed meshes with a joint learning strategy are shown as column (b) where all the open structures like collars and sleeves are wrongly shaped with watertight surfaces.



(a) Direct fusing (b) Grad optimize (c) GT

Figure 9: Ablation study on sign and distance fusing strategy illustrated in Fig. 3. The gradient optimized HSDF can reconstruct meshes with better surface quality as shown in (b) compared to (a).

Necessity of HSDF fusion We validate the quality improvement by integrating HSDF fusion as described in Sec. 3.3. As shown in Fig. 9, the gradient optimization consistently removes the Moire pattern and produces smoother results than direct fusing.

Limitations The first limitation of our approach is the reconstruction accuracy of mesh boundaries. The marching cube algorithm cuts through every cube to extract the surfaces. The extraction becomes ill-posed when cubes enclose the boundary surface of an open shape, which results in non-manifold structures. We envision that increasing the resolution of boundary cubes will alleviate this problem by decreasing the boundary cubes' size. The second limitation is that we rely on consistent and clean normals of the training data to compute HSDFs. Datasets without clear normals like unprocessed ShapeNet [7] have to be repaired [17] before training.

5 Conclusions

We propose a novel hybrid signed and distance field (HSDF) and a dedicated learning model HSDF-Net as an intuitive but useful extension of SDF in previous works. Through sufficient experiments, we have shown that this natural extension leads to a substantial increase in modeling capacities for a much wider range of complex shapes. And we further validate the proposed HSDF can reconstruct arbitrary shapes containing open surfaces efficiently with high fidelity and consistent surface normals. In the future, we expect to explore more applications, such as reconstructions from 2D images to exploit our representational capacities.

References

[1] Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Björn Ottersten. Deep learning advances on different 3D data representations: A survey. *arXiv preprint arXiv:1808.01462*, 1, 2018. 2

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 3

[3] 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 Raj Reddy, editor, *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, USA, August 22-25, 1977*, pages 659–663. William Kaufmann, 1977. 6

[4] 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. 2, 8

[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

[6] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 2

[7] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 6, 7, 9

[8] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3d point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14514–14523, 2021. 3

[9] Zhiqin Chen, Kangxue Yin, Matt Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-NET: Branched autoencoder for shape co-segmentation. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019. 3

[10] 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. 1, 3

[11] 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. 3

[12] 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. 3, 4, 5, 7, 8

[13] 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, 4, 7, 8

[14] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. *arXiv preprint arXiv:2010.13938*, 2020. 3

[15] 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. 3

[16] 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 *ECCV*, pages 628–644. Springer, 2016. 2

[17] Lei Chu, Hao Pan, Yang Liu, and Wenping Wang. Repairing man-made meshes via visual driven global optimization with minimum intrusion. *ACM Trans. Graph.*, 38(6):158:1–158:18, 2019. 6, 9

[18] 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. 3

[19] 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. 8

[20] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao(Richard) Zhang. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019)*, 38(6):243:1–243:15, 2019. 3

[21] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas A Funkhouser. Deep structured implicit functions. 2019. 3

[22] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. AtlasNet: A papier-mâché approach to learning 3D surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[23] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 3

[24] Shi-Min Hu, Zheng-Ning Liu, Meng-Hao Guo, Jun-Xiong Cai, Jiahui Huang, Tai-Jiang Mu, and Ralph R. Martin. Subdivision-based mesh convolution networks, 2021. 3

[25] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris. Deep learning advances in computer vision with 3D data: A survey. *ACM Computing Surveys (CSUR)*, 50(2):1–38, 2017. 2

[26] Dominic Jack, Jhony K Pontes, Sridha Sridharan, Clinton Fookes, Sareh Shirazi, Frederic Maire, and Anders Eriksson. Learning free-form deformations for 3d object reconstruction. In *Asian Conference on Computer Vision*, pages 317–333. Springer, 2018. 3

[27] 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. 3

[28] Marian Kleineberg. Calculate signed distance fields for arbitrary meshes. https://github.com/marian42/mesh_to_sdf, 2021. 6

[29] Alon Lahav and Ayellet Tal. Meshwalker: Deep mesh understanding by random walks. *ACM Transactions on Graphics (TOG)*, 39(6):1–13, 2020. 3

[30] Hsueh-Ti Derek Liu, Vladimir G Kim, Siddhartha Chaudhuri, Noam Aigerman, and Alec Jacobson. Neural subdivision. *arXiv preprint arXiv:2005.01819*, 2020. 3

[31] Qinqing Liu, Peng-Shuai Wang, Chunjiang Zhu, Blake Blumenfeld Gaines, Tan Zhu, Jinbo Bi, and Minghu Song. Octsurf: Efficient hierarchical voxel-based molecular surface representation for protein-ligand affinity prediction. *Journal of Molecular Graphics and Modelling*, 105:107865, 2021. 3

[32] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 3

[33] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 2

[34] 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. 3, 6, 7

[35] 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. 1, 3

[36] David R. Palmer, Dmitriy Smirnov, Stephanie Wang, Albert Chern, and Justin M. Solomon. Deepcurrents: Learning implicit representations of shapes with boundaries. *CoRR*, abs/2111.09383, 2021. 3

[37] 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. 3

[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. 1, 3

[39] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. 2020. 3

[40] 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. 3

[41] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 3

[42] Yi-Ling Qiao, Lin Gao, Paul Rosin, Yu-Kun Lai, Xilin Chen, et al. Learning on 3d meshes with laplacian encoding and pooling. *IEEE Transactions on Visualization and Computer Graphics*, 2020. 3

[43] 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. 1, 3

[44] 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. 3

[45] 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. 2, 8

[46] Rahul Venkatesh, Tejan Karmali, Sarthak Sharma, Aurobrata Ghosh, R Venkatesh Babu, László A Jeni, and Maneesh Singh. Deep implicit surface point prediction networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12653–12662, 2021. 3

[47] 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. 2, 3

[48] 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. 3

[49] 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. 3

[50] Peng-Shuai Wang, Yang Liu, and Xin Tong. Deep octree-based cnns with output-guided skip connections for 3d shape and scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 266–267, 2020. 3

[51] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive o-cnn: A patch-based deep representation of 3d shapes. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 3

[52] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 3

[53] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 2

[54] Yun-Peng Xiao, Yu-Kun Lai, Fang-Lue Zhang, Chunpeng Li, and Lin Gao. A survey on deep geometry learning: From a representation perspective. *Computational Visual Media*, 6(2):113–133, 2020. 2

[55] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomír Mech, and Ulrich Neumann. DISN: deep implicit surface network for high-quality single-view 3d reconstruction. 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 490–500, 2019. 6, 7, 8

[56] Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J Guibas, and Lin Gao. DSG-Net: Learning disentangled structure and geometry for 3d shape generation. *arXiv preprint arXiv:2008.05440*, 2020. 3

[57] Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. GIFS: neural implicit function for general shape representation. *CoRR*, abs/2204.07126, 2022. 3

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4.5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work focuses on the 3D shape reconstruction and modeling, which is not related to any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We will attach our code in the supplemental material.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please refer to supplementary material.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] This is not necessary in our experimental settings.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please refer to supplementary material.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.1.

(b) Did you mention the license of the assets? [Yes]

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]