Neural-Pull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces

Baorui Ma [*1] **Zhizhong Han** [*2] **Yu-Shen Liu** [1] **Matthias Zwicker** [3]

Abstract

Reconstructing continuous surfaces from 3D point clouds is a fundamental operation in 3D geometry processing. Several recent state-of-the-art methods address this problem using neural networks to learn signed distance functions (SDFs). In this paper, we introduce *Neural-Pull*, a new approach that is simple and leads to high quality SDFs. Specifically, we train a neural network to pull query 3D locations to their closest points on the surface using the predicted signed distance values and the gradient at the query locations, both of which are computed by the network itself. The pulling operation moves each query location with a stride given by the distance predicted by the network. Based on the sign of the distance, this may move the query location along or against the direction of the gradient of the SDF. This is a differentiable operation that allows us to update the signed distance value and the gradient simultaneously during training. Our outperforming results under widely used benchmarks demonstrate that we can learn SDFs more accurately and flexibly for surface reconstruction and single image reconstruction than the state-of-the-art methods. Our code and data are available at `https://github.com/mabaorui/NeuralPull`.

1. Introduction

Signed Distance Functions (SDFs) have been an important 3D shape representation for deep learning based 3D shape analysis (Park et al., 2019; Mescheder et al., 2019; Mildenhall et al., 2020; Michalkiewicz et al., 2019; Saito et al., 2019; Rematas et al., 2021; Sitzmann et al., 2020; Ost et al., 2020; Takikawa et al., 2021; Martel et al., 2021; Oechsle et al., 2021; Azinovic et al., 2021; Dupont et al., 2021), due to their advantages over other representations in representing high resolution shapes with arbitrary topology. Given ground truth signed distance values, it is intuitive to learn an SDF by training a deep neural network to regress signed distance values for query 3D locations, where an image (Michalkiewicz et al., 2019; Park et al., 2019) or a point cloud (Jia & Kyan, 2020; Erler et al., 2020) representing the shape can serve as a condition which is an additional input of the network. It has also been shown how to learn SDFs from multiple 2D images rather than 3D information using differentiable renderers (Liu et al., 2020; Jiang et al., 2020b; Zakharov et al., 2020; Wu & Sun, 2020). In this paper, we address the problem of learning SDFs from raw point clouds and propose a new method that outperforms the state-of-the-art on widely used benchmarks.

Current solutions (Gropp et al., 2020; Chibane et al., 2020b; Atzmon & Lipman, 2020a;b) aim to estimate unsigned distance fields by leveraging additional constraints. The rationale behind these solutions is that an unsigned distance field can be directly learned from the distances between a set of query 3D locations and their nearest neighbors on the 3D point clouds, while the signs of these distances require more information to infer, such as geometric regularization (Gropp et al., 2020), sign agnostic learning (Atzmon & Lipman, 2020a;b), or analytical gradients (Chibane et al., 2020b).

In this paper, we propose a method to learn SDFs directly from raw point clouds without requiring ground truth signed distance values. Our method learns the SDF from a point cloud, or from multiple point clouds with conditions by training a neural network to learn to pull the surrounding 3D space onto the surface represented by the point cloud. Hence we call our method *Neural-Pull*. Specifically, given a 3D query location as input to the network, we ask the network to pull it to its closest point on the surface using the predicted signed distance value and the gradient at the query location, both of which are calculated by the network itself. The pulling operation is differentiable, and depending on the sign of the predicted distance, it moves the query location along or against the direction of the gradient with a stride given by the signed distance. Since our training

[*] Equal contribution [1] School of Software, BNRist, Tsinghua University, Beijing 100084, P. R. China [2] Department of Computer Science, Wayne State University, Detroit, USA [3] Department of Computer Science, University of Maryland, College Park, USA. Correspondence to: Yu-Shen Liu <liuyushen@tsinghua.edu.cn>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

objective involves both the signed distance and its gradient, it leads to highly effective learning. Our experiments using widely used benchmarks show that Neural-Pull can learn SDFs more accurately and flexibly when representing 3D shapes in different applications than previous state-of-the-art methods. Our contributions are listed below.

i) We introduce Neural-Pull, a novel approach to learn SDFs directly from raw 3D point clouds without ground truth signed distance values.

ii) We introduce the idea to effectively learn SDFs by updating the predicted signed distance values and the gradient simultaneously in order to pull surrounding 3D space onto the surface.

iii) We significantly improve the state-of-the-art accuracy in surface reconstruction and single image reconstruction under various benchmarks.

2. Related Work

Deep learning models have been playing an important role in different 3D computer vision applications (Han et al., 2019c;a; Wen et al., 2020b; Han et al., 2020c; 2019d; Liu et al., 2019c; Hu et al., 2020; Wen et al., 2020b; Groueix et al., 2018; Tretschk et al., 2020; Bednarik et al., 2020; Han et al., 2019e; Tancik et al., 2020; Han et al., 2019b; 2020a; Badki et al., 2020; Mi et al., 2020; Han et al., 2020b; Wen et al., 2021b;a; Jiang et al., 2020b; Wen et al., 2020a; Liu et al., 2021). In the following, we will briefly review work related to learning implicit functions for 3D shapes in different ways.

Learning from 3D Ground Truth Globally. Some techniques aim to learn implicit functions that represent conditional mappings from a 3D location to a binary occupancy value (Mescheder et al., 2019; Chen & Zhang, 2019) or a signed distance value (Michalkiewicz et al., 2019; Park et al., 2019). Early work requires the ground truth occupancy values or signed distance values as 3D supervision. For single image reconstruction, a single image (Wang et al., 2019b; Saito et al., 2019; Chibane et al., 2020a; Littwin & Wolf, 2019; Genova et al., 2019; Han et al., 2020c) or a learnable latent code (Park et al., 2019) can be a condition to provide information about a specified shape. For surface reconstruction (Williams et al., 2019; Liu et al.; Mi et al., 2020; Genova et al., 2019), we can leverage a point cloud as a condition to learn an implicit function which further produces a surface (Jia & Kyan, 2020; Erler et al., 2020).

Learning from 3D Ground Truth Locally. To improve the performance of learning implicit functions, a local strategy was also explored that focuses on more local shape information. Jiang et al. (Jiang et al., 2020a) introduced the local implicit grid to improve the scalability and generality. Similarly, PatchNet (Tretschk et al., 2020) was proposed to learn a patch-based surface representation to get more generalizable models. With a grid of independent latent codes, deep local shapes (Chabra et al., 2020) was proposed to represent 3D shapes without prohibitive memory requirements. Using locally interpolated features, convolutional occupancy networks (Songyou Peng, 2020) learn occupancy network for 3D scene reconstruction. Other local deep implicit functions (Genova et al., 2020) are learned by inferring the space decomposition and local deep implicit function learning from a 3D mesh or posed depth images.

Learning from 2D Supervision. We can also learn implicit functions from 2D supervision, such as multiple images. Vincent et al. (Sitzmann et al., 2019) learned a mapping from world coordinates to a feature representation of local scene properties, which reduces the computational cost on sampling points for implicit surface learning. Inspired by ray marching rendering, different differentiable renderers (Liu et al., 2020; Jiang et al., 2020b; Zakharov et al., 2020) were introduced to render signed distance functions into images. In addition, ray-based field probing (Liu et al., 2019b) or aggregating detection points on rays (Wu & Sun, 2020) were employed to mine supervision for 3D occupancy fields. With the implicit differentiation, Niemeyer et al. (Niemeyer et al., 2020) analytically derived in a differentiable rendering formulation for implicit shape and texture representations. For view synthesis, radiance fields were learned first, and then rendered using the differentiable volume rendering to calculate the loss (Mildenhall et al., 2020).

Learning from Point Clouds. Without ground truth signed distance values or occupancy values, learning implicit functions directly from raw point clouds is more challenging. Current methods learn signed or unsigned distance fields with additional constraints, such as geometric regularization (Gropp et al., 2020), sign agnostic learning with a specially designed loss function (Atzmon & Lipman, 2020a) or constraints on gradients (Atzmon & Lipman, 2020b), and analytical gradients (Chibane et al., 2020b). A recent cocurrent work (Chibane et al., 2020b) learns to predict unsigned distances and infers the surface by pulling sampled queries to the surface. While our method directly learns SDFs which can be used to directly predict 3D shapes during testing, especially for applications without knowing point clouds during inference, such as single image reconstruction.

3. Method

Problem Statement. We employ a neural network to learn SDFs that represent 3D shapes. An SDF $\boldsymbol{f}$ predicts a signed distance value $s \in \mathbb{R}$ for a query 3D location $\boldsymbol{q} = [x, y, z]$. Optionally, we provide an additional condition $\boldsymbol{c}$ as input, such that $\boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}) = s$. Given ground truth signed distances as supervision, current methods (Michalkiewicz et al., 2019;

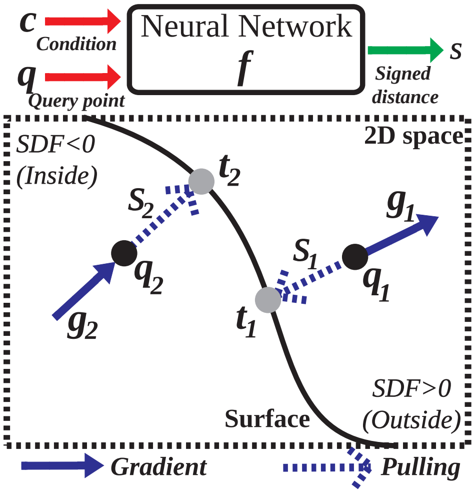

Figure 1. Demonstration of pulling surrounding 2D space to a surface, where gradients $\boldsymbol{g}_i$ and signed distance value s are from neural network $\boldsymbol{f}$.

Park et al., 2019) can employ a neural network to learn $\boldsymbol{f}$ as a regression problem. Different from these method, we aim to learn SDF $\boldsymbol{f}$ in a 3D space directly from 3D point cloud $\boldsymbol{P} = \{\boldsymbol{p}_j, j \in [1, J]\}$.

Overview. We introduce Neural-Pull as a neural network to learn how to pull a 3D space onto the surface represented by the point cloud $\boldsymbol{P}$. Rather than leveraging unsigned distances as previous methods (Gropp et al., 2020; Chibane et al., 2020b; Atzmon & Lipman, 2020a;b), Neural-Pull trains an SDF $\boldsymbol{f}$ to predict signed distances using the point cloud $\boldsymbol{P}$ and the gradient within the network itself to represent 3D shapes. Neural-pull tries to learn to pull a query location $\boldsymbol{q}_i$ which is randomly sampled around the surface to its nearest neighbor $\boldsymbol{t}_i$ on the surface, where the query locations form a set $\boldsymbol{Q} = \{\boldsymbol{q}_i, i \in [1, I]\}$. The pulling operation pulls the query location $\boldsymbol{q}_i$ with a stride of signed distance s_i, along or against the direction of the gradient $\boldsymbol{g}_i$ at $\boldsymbol{q}_i$, obtained within the network.

We demonstrate our idea using a 2D surface in Fig. 1, where the 2D surface splits the space into inside and outside of the shape. We train a neural network to employ the predicted signed distances s_1 (or s_2) to pull the query location $\boldsymbol{q}_1$ (or $\boldsymbol{q}_2$) to its nearest neighbor $\boldsymbol{t}_1$ (or $\boldsymbol{t}_2$) against (or along) the gradient $\boldsymbol{g}_1$ (or $\boldsymbol{g}_2$) at the query location $\boldsymbol{q}_1$ (or $\boldsymbol{q}_2$).

Pulling Query Points. We pull a 3D query location $\boldsymbol{q}_i$ to its nearest neighbor $\boldsymbol{t}_i$ on the surface using the predicted signed distance s_i and the gradient $\boldsymbol{g}_i$ at $\boldsymbol{q}_i$ within the network. The gradient $\boldsymbol{g}_i$ is a vector whose components are the partial derivatives of $\boldsymbol{f}$ at $\boldsymbol{q}_i$, such that $\boldsymbol{g}_i = [\partial \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)/\partial x, \partial \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)/\partial y, \partial \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)/\partial z]$, which is also denoted as $\nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)$, where $\boldsymbol{c}$ is a condition. It is the direction of the fastest signed distance increase in 3D space. Therefore, we can leverage this property to move a query location along or against the direction of gradient $\boldsymbol{g}_i$ to its nearest point on the surface. We leverage the following equation to pull query locations $\boldsymbol{q}_i$,

$$\boldsymbol{t}'_i = \boldsymbol{q}_i - \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i) \times \nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)/||\nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)||_2, \quad (1)$$

where $\boldsymbol{t}'_i$ is the pulled query location $\boldsymbol{q}_i$ after pulling, $\boldsymbol{c}$ is the condition to represent ground truth point cloud $\boldsymbol{P}$, and $\nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)/||\nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)||_2$ is the direction of gradient $\nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)$. Since $\boldsymbol{f}$ is a continuously differentiable function, we can obtain $\nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)$ in the back-propagation process of training $\boldsymbol{f}$. As Fig. 1 demonstrates, for query locations inside of the shape $\boldsymbol{P}$, if the sign of the signed distance value is negative, and the network will move the query location $\boldsymbol{q}_i$ along the direction of gradient to $\boldsymbol{t}'_i$ on $\boldsymbol{P}$ using $\boldsymbol{t}'_i = \boldsymbol{q}_i + |\boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)| \times \nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)/||\nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)||_2$. Instead, the network will move query locations outside of $\boldsymbol{P}$ against the direction of gradient due to the positive signed distance value, using $\boldsymbol{t}'_i = \boldsymbol{q}_i - |\boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)| \times \nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)/||\nabla \boldsymbol{f}(\boldsymbol{c}, \boldsymbol{q}_i)||_2$.

Query Locations Sampling. We randomly sample query locations around each point $\boldsymbol{p}_j$ of the ground truth point cloud $\boldsymbol{P}$. For each point $\boldsymbol{p}_j \in \boldsymbol{P}$, we establish an isotropic Gaussian function $\mathcal{N}(\boldsymbol{p}_j, \sigma^2)$ to form a distribution, according to which we randomly sample 25 query locations, where σ^2 is the parameter to control how far away from the surface we can sample query locations. Here, we employ an adaptive way to set σ^2 as the square distance between $\boldsymbol{p}_j$ and its 50-th nearest neighbor, which reflects location density around $\boldsymbol{p}_j$. The sampled query locations can cover the area around the surface represented by the point cloud $\boldsymbol{P}$, both inside and outside of the shape. Our preliminary results show that sampling near the surface will improve the learning accuracy, since it is hard for the network to predict accurate signed distance and gradient to move a query location that is far from surface to its nearest neighbor on the surface. We will elaborate on the details of leveraging these query locations sampled around $\boldsymbol{P}$ during training later.

Loss Function. Neural-pull aims to train a network to learn to pull a query location $\boldsymbol{q}_i$ to its nearest neighbor $\boldsymbol{t}_i$ on the point cloud $\boldsymbol{P}$. So, we leverage a square error to minimize the distance between the pulled query location $\boldsymbol{t}'_i$ obtained in Eq. 1 and the nearest neighbor $\boldsymbol{t}_i$ among $\boldsymbol{p}_j$ on $\boldsymbol{P}$ below,

$$d(\{\boldsymbol{t}'_i\}, \{\boldsymbol{t}_i\}) = \frac{1}{I} \sum_{i \in [1,I]} ||\boldsymbol{t}'_i - \boldsymbol{t}_i||_2^2, \quad (2)$$

Convergence to SDF. One question that is not answered yet is that why the learned function $\boldsymbol{f}$ can converge to a signed distance field. Obviously, Eq. 1 is also satisfied in an unsigned distance field. We illustrate the difference between

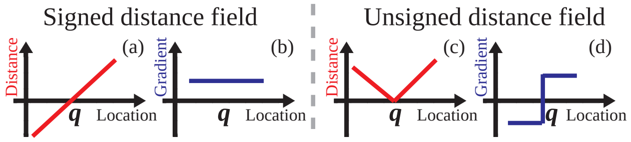

Figure 2. The illustration of the difference between signed distance field and unsigned distance field in terms of distance sign in (a), (c) and gradient in (b), (d).

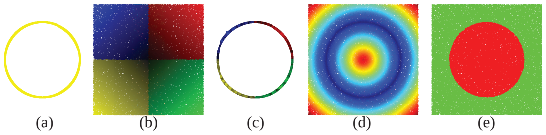

Figure 3. Optimization visualization on a 2D case.

signed distance field and unsigned distance field near a 2D surface in Fig. 2, where the location is shown in 1D. For a query point $\boldsymbol{q}$ on the surface, the signed distance field near $\boldsymbol{q}$, $\boldsymbol{q}+\triangle\boldsymbol{q}$, is changing the sign of distance when going across the surface in Fig. 2(a), while keeping the gradient the same in Fig. 2(b). In contrast, the direction of the gradient for the region of $\boldsymbol{q}+\triangle\boldsymbol{q}$ in an unsigned distance field is changing in Fig. 2(d) since the unsigned distance is increasing in both sides of the surface in Fig. 2(c). According to this difference, we have the following theorem indicating that a continuous function implemented by MLP can automatically converge to an SDF using our loss.

Theorem 1. *A continuous function f implemented by MLP which is trained to minimize Eq. 2 can converge to a signed distance function if Eq. 3 is satisfied at any point* $\mathbf{p}$ *on the surface ($f(\boldsymbol{p})=0$), where* $\mathbf{N}$ *is the norm of* $\mathbf{p}$, $\|\triangle t\|<\mu$ *and μ indicates a small range.*

$$f(\boldsymbol{p}-\boldsymbol{N}\triangle t)=-f(\boldsymbol{p}+\boldsymbol{N}\triangle t). \tag{3}$$

Proof: Since f is a continuous function representing SDF, if $\nabla f(\boldsymbol{p})\neq\boldsymbol{0}$, we have $\boldsymbol{N}=\nabla f(\boldsymbol{p})/||\nabla f(\boldsymbol{p})||_2$. Assume $\triangle\boldsymbol{p}=\boldsymbol{N}\triangle t$, using the definition of gradient, we have

$$\lim_{\triangle\boldsymbol{p}\to\boldsymbol{0}}(f(\boldsymbol{p}+\triangle\boldsymbol{p})-f(\boldsymbol{p}))/\triangle\boldsymbol{p}=\boldsymbol{N}*||\nabla f(\boldsymbol{p})||_2. \tag{4}$$

We can rewrite the equation above by removing lim into

$$(f(\boldsymbol{p}+\triangle\boldsymbol{p})-f(\boldsymbol{p}))/\triangle\boldsymbol{p}=\boldsymbol{N}*||\nabla f(\boldsymbol{p})||_2+\alpha, \tag{5}$$

where α is infinitesimal when $\triangle\boldsymbol{p}\to\boldsymbol{0}$. We can further have $f(\boldsymbol{p}+\triangle\boldsymbol{p})-f(\boldsymbol{p})=(\boldsymbol{N}*||\nabla f(\boldsymbol{p})||_2+\alpha)*\triangle\boldsymbol{p}\neq 0$ by multiplying $\triangle\boldsymbol{p}$ on both sides, since $\triangle\boldsymbol{p}$ approaches $\boldsymbol{0}$ but never equals to $\boldsymbol{0}$. Similarly, we also have $f(\boldsymbol{p}-\triangle\boldsymbol{p})-f(\boldsymbol{p})=-(\boldsymbol{N}*||\nabla f(\boldsymbol{p})||_2+\alpha)*\triangle\boldsymbol{p}$. Since $f(\boldsymbol{p})=0$, we have

$$f(\boldsymbol{p}-\triangle\boldsymbol{p})=-f(\boldsymbol{p}+\triangle\boldsymbol{p}). \tag{6}$$

We can further replace $\triangle\boldsymbol{p}$ into $\boldsymbol{N}\triangle t$ to get Eq. 3 proofed.

Next, we will further proof our loss can significantly penalize $\nabla f(\boldsymbol{p})=\boldsymbol{0}$. Assume $\nabla f(\boldsymbol{p})=\boldsymbol{0}$, so $\lim_{\triangle\boldsymbol{p}\to\boldsymbol{0}}(f(\boldsymbol{p}+\triangle\boldsymbol{p})-f(\boldsymbol{p}))/\triangle\boldsymbol{p}=\boldsymbol{0}$. Since $f(\boldsymbol{p})=\boldsymbol{0}$, $f(\boldsymbol{p}+\triangle\boldsymbol{p})$ is higher order infinitesimal of $\triangle\boldsymbol{p}$. If we pull $\boldsymbol{p}+\triangle\boldsymbol{p}$ to $\boldsymbol{p}$, our loss is $||\boldsymbol{p}-(\boldsymbol{p}+\triangle\boldsymbol{p}-f(\boldsymbol{p}+\triangle\boldsymbol{p})\times\nabla f(\boldsymbol{p}+\triangle\boldsymbol{p})/||\nabla f(\boldsymbol{p}+\triangle\boldsymbol{p})||_2)||_2^2$, which can be rewritten into $||\triangle\boldsymbol{p}-f(\boldsymbol{p}+\triangle\boldsymbol{p})\times\nabla f(\boldsymbol{p}+\triangle\boldsymbol{p})/||\nabla f(\boldsymbol{p}+\triangle\boldsymbol{p})||_2||_2^2$. However, this equation can not be $\boldsymbol{0}$ since $f(\boldsymbol{p}+\triangle\boldsymbol{p})\times\nabla f(\boldsymbol{p}+\triangle\boldsymbol{p})/||\nabla f(\boldsymbol{p}+\triangle\boldsymbol{p})||_2$ is still higher order infinitesimal of $\triangle\boldsymbol{p}$. So, $\nabla f(\boldsymbol{p})\neq\boldsymbol{0}$.

Optimization Visualization. We demonstrate the optimization using a 2D case in Fig. 3. We learn a circle $\boldsymbol{P}$ in Fig. 3 (a) using query locations $\boldsymbol{q}_i$ sampled in Fig. 3 (b), where the color of $\boldsymbol{q}_i$ is used to track the pulled query locations $\boldsymbol{t}_i'$ in Fig. 3 (c). The consistent color indicates that our loss can correctly pull the queries onto the surface. Additionally, we visualize the unsigned distances of the learned signed distance field in Fig. 3 (d) and their signs in Fig. 3 (e). Fig. 3 justifies the effectiveness of our method.

Training. We randomly sample $J=2\times10^4$ points $\boldsymbol{p}_j$ from point clouds formed by 1×10^5 points released by OccNet (Mescheder et al., 2019) as the ground truth point cloud $\boldsymbol{P}$ for each shape, where $j\in[1,J]$. As mentioned, we sample 25 3D query locations $\boldsymbol{q}_i$ around each point $\boldsymbol{p}_j$ to form the corresponding query location set $\boldsymbol{Q}$, such that $i\in[1,I]$ and $I=5\times10^5$. During training, we randomly select 5000 query locations from $\boldsymbol{Q}$ as a batch to train the network. We try two different ways to select the 5000 query locations. One way is to randomly select from $\boldsymbol{Q}$, the other is to uniformly sample 5000 points on the ground truth point cloud $\boldsymbol{P}$, and then select one query location around each sampled point, where the second way can better cover the whole shape in each batch. Our preliminary results show that both of the two ways achieve good learning performance.

We employ a neural network similar to OccNet (Mescheder et al., 2019) to learn the signed distance function (more details can be found in our supplemental material). We use the Adam optimizer with an initial learning rate of 0.0001, and train the model in 2500 epochs. Moreover, we initialize the parameters in our network using the geometric network initialization (GNI) (Atzmon & Lipman, 2020a) to approximate the signed distance function of a sphere, where the sign of the signed distance inside of the shape is negative and positive outside.

4. Experiments and Analysis

4.1. Surface Reconstruction

Details. We employ Neural-Pull to reconstruct 3D surfaces from point clouds. Given a point cloud $\boldsymbol{P}$, we do not leverage any condition $\boldsymbol{c}$ in Fig. 1 and overfit the neural network to the shape by minimizing the loss in Eq. 2, where we remove the network for extracting the feature of the condition. Hence our method does not require any training procedure under the training set, which differentiates our method from the previous ones (Atzmon & Lipman, 2020a; Chibane et al., 2020b; Liu et al.; Erler et al., 2020). After overfitting on each shape, our neural network learns an SDF for the shape. Then, we use the marching cubes (Lorensen & Cline, 1987) algorithm to reconstruct the mesh surface.

Dataset and Metric. For fair comparison with other methods, we leverage three widely used benchmarks to evaluate our method in surface reconstruction.

Table 1. Reconstruction comparison in terms of L2-CD (×100).

Dataset	DSDF	ATLAS	PSR	Points2Surf	IGR	Ours
ABC	8.41	4.69	2.49	1.80	0.51	**0.48**
FAMOUS	10.08	4.69	1.67	1.41	1.65	**0.22**
Mean	9.25	4.69	2.08	1.61	1.08	**0.35**

The first benchmark is the ABC dataset (Koch et al., 2019) which contains a large number and variety of CAD models. We use a subset of this dataset, released by Points2Surf (Erler et al., 2020) with the same train/test splitting. The second benchmark is FAMOUS which is also released by Points2Surf (Erler et al., 2020). The FAMOUS dataset is formed by 22 diverse well-known meshes.The last one is a subset of ShapeNet (Chang et al., 2015) the same train/test splitting released by MeshingPoint (MeshP) (Liu et al.).

To comprehensively evaluate our method with the state-of-the-art methods, we leverage different metrics for fair comparison. Following Points2Surf (Erler et al., 2020), we leverage the L2-Chamfer distance (L2-CD) to evaluate the reconstruction error between our reconstruction and the 1×10^4 ground truth points under the ABC and FAMOUS datasets, where we also randomly sample 1×10^4 points on our reconstructed mesh. Besides the L1-Chamfer distance (L1-CD), we also follow MeshP (Liu et al.) to leverage L2-CD, Normal Consistency (NC) (Mescheder et al., 2019), and F-score (Tatarchenko et al., 2019) to evaluate the reconstruction error, where we compare the 1×10^5 points sampled on the reconstructed shape with the 1×10^5 ground truth points released by OccNet (Mescheder et al., 2019). Note that L2-CD leverages the L2 norm to evaluate the distance between each pair of points, while L1-CD leverages the L1 norm.

Table 2. Surface reconstruction comparison in terms of L2-CD (×100).

Class	PSR	DMC	BPA	ATLAS	DMC	DSDF	DGP	MeshP	NUD	SALD	Ours
Display	0.273	0.269	0.093	1.094	0.662	0.317	0.293	0.069	0.077	-	**0.039**
Lamp	0.227	0.244	0.060	1.988	3.377	0.955	0.167	**0.053**	0.075	0.071	0.080
Airplane	0.217	0.171	0.059	1.011	2.205	1.043	0.200	0.049	0.076	0.054	**0.008**
Cabinet	0.363	0.373	0.292	1.661	0.766	0.921	0.237	0.112	0.041	-	**0.026**
Vessel	0.254	0.228	0.078	0.997	2.487	1.254	0.199	0.061	0.079	-	**0.022**
Table	0.383	0.375	0.120	1.311	1.128	0.660	0.333	0.076	0.067	0.066	**0.060**
Chair	0.293	0.283	0.099	1.575	1.047	0.483	0.219	0.071	0.063	0.061	**0.054**
Sofa	0.276	0.266	0.124	1.307	0.763	0.496	0.174	0.080	0.071	0.058	**0.012**
Mean	0.286	0.276	0.116	1.368	1.554	0.766	0.228	0.071	0.069	0.062	**0.038**

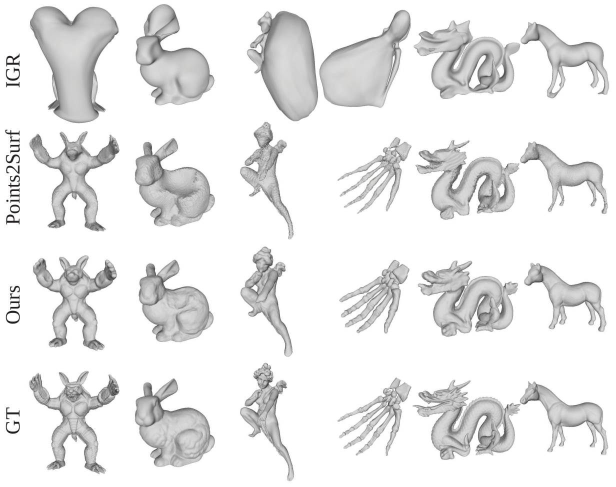

Figure 4. Comparison under FAMOUS in surface reconstruction.

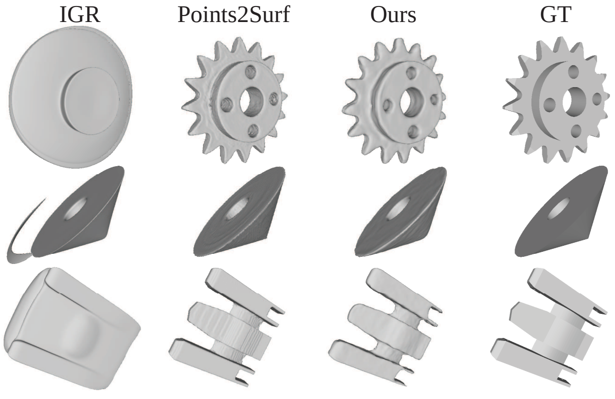

Figure 5. Comparison under ABC in surface reconstruction.

Comparison. We compare our method with state-of-the-art classic and data-driven surface reconstruction methods under the FAMOUS and ABC datasets, including DeepSDF (DSDF) (Park et al., 2019), AtlasNet (ATLAS) (Groueix et al., 2018), Screened Poisson Surface Reconstruction (PSR) (Kazhdan & Hoppe, 2013), Points2Surf (Erler et al., 2020), and IGR (Gropp et al., 2020). We report the results of DSDF, ATLAS, PSR and Points2Surf from the paper of Points2Surf (Erler et al., 2020), while reproducing the result-

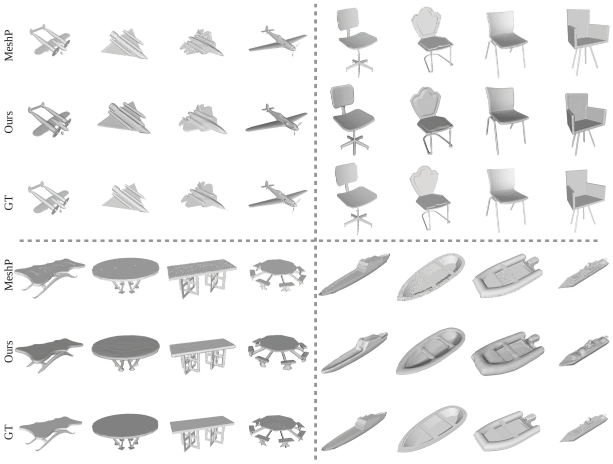

Figure 6. Comparison under ShapeNet in surface reconstruction.

s of IGR using the official code. The L2-CD comparison in Table 1 shows that our method can significantly increase the surface reconstruction accuracy under each dataset due to better inference of the surface learned in the pulling process.

We visually compare our method with IGR (Gropp et al., 2020) and Points2Surf (Erler et al., 2020) under the FAMOUS and ABC dataset in Fig. 4 and Fig. 5, respectively. We train IGR using its released code with the same settings as ours, and generate surface reconstruction using the trained parameters released by Points2Surf. The comparison in Fig. 4 demonstrates that our method can reveal geometry details in higher accuracy than other methods. Moreover, the comparison in Fig. 5 shows that our method can reconstruct a smoother plane than Points2Surf, but Points2Surf is good at reconstructing sharp edges.

Similarly, we compare the state-of-the-art classic and data-driven methods under the ShapeNet subset, including P-SR (Kazhdan & Hoppe, 2013), Ball-Pivoting algorithm (B-PA) (Bernardini et al., 1999), ATLAS (Groueix et al., 2018), Deep Geometric Prior (DGP) (Williams et al., 2019), Deep Marching Cube (DMC) (Liao et al., 2018), DeepSDF (DSD-F) (Park et al., 2019), MeshP (Liu et al.), Neural Unsigned Distance (NUD) (Chibane et al., 2020b), SALD (Atzmon & Lipman, 2020b), Local SDF (GRID) (Jiang et al., 2020a), and IMNET (Chen & Zhang, 2019). We conduct the numerical comparison in terms of different metrics including L2-CD in Table 2, normal consistency in Table 3, and F-score with a threshold of μ in Table 4, 2μ in Table 5. We report the results of PSR, MC, BPA, ATLAS, DMC, DSDF, DGP, MeshP from the paper of MeshP (Liu et al.), while reporting the results of SALD, GRID, IMNET from their original papers and reproducing the results of NUD using the same experimental settings. The comparison shown in Table 2, 3, 4, 5 demonstrates that our method can reconstruct more accurate surfaces in terms of CD and F-Score, where we set the threshold μ as 0.002 in the F-Score calculation. Although our normal consistency results are comparable to MeshP, MeshP directly does the meshing without learning an implicit function, which requires dense and clean point clouds to guarantee the performance.

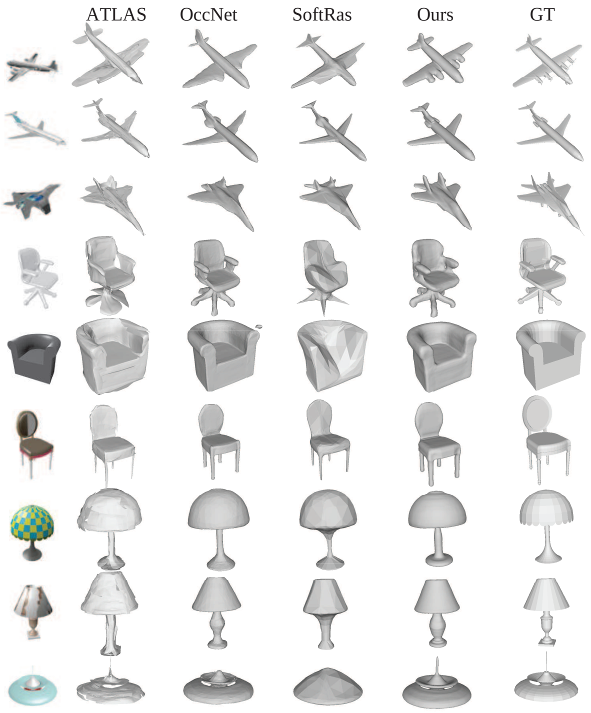

Figure 7. Comparison in single image reconstruction.

We visually compare our method with the-state-of-the-art MeshP (Liu et al.) under Airplane, Chair, Table and Vessel classes in Fig. 6. We use the parameters trained by MeshP. The comparison shows that our method can reconstruct more complete surfaces, especially for thin structures or sharp corners, which achieves much higher accuracy.

In addition, we also report our L1-CD results by comparing with 3D-R2 (Choy et al., 2016), PSGN (Fan et al., 2017), DMC (Liao et al., 2018), Occupancy Network (OccNet) (Mescheder et al., 2019), SSRNet (Mi et al., 2020) and DDT (Luo et al., 2020) under the ShapeNet subset in Table 6. The comparison shows that our method achieves the best performance.

4.2. Single Image Reconstruction

Details. We further employ Neural-Pull to reconstruct 3D shapes from 2D images. We regard the 2D image as a condition, which corresponds to a 3D shape represented as a point

Table 3. Surface reconstruction comparison in terms of normal consistency.

Class	PSR	DMC	BPA	ATLAS	DMC	DSDF	MeshP	GRID	IMNET	Ours
Display	0.889	0.842	0.952	0.828	0.882	0.932	**0.974**	0.926	0.574	0.964
Lamp	0.876	0.872	0.951	0.593	0.725	0.864	**0.963**	0.882	0.592	0.930
Airplane	0.848	0.835	0.926	0.737	0.716	0.872	**0.955**	0.817	0.550	0.947
Cabinet	0.880	0.827	0.836	0.682	0.845	0.872	**0.957**	0.948	0.700	0.930
Vessel	0.861	0.831	0.917	0.671	0.706	0.841	**0.953**	0.847	0.574	0.941
Table	0.833	0.809	0.919	0.783	0.831	0.901	**0.962**	0.936	0.702	0.908
Chair	0.850	0.818	0.938	0.638	0.794	0.886	**0.962**	0.920	0.820	0.937
Sofa	0.892	0.851	0.940	0.633	0.850	0.906	**0.971**	0.944	0.818	0.951
Mean	0.866	0.836	0.923	0.695	0.794	0.884	**0.962**	0.903	0.666	0.939

Table 4. Surface reconstruction comparison in terms of F-score with a threshold of μ.

Class	PSR	DMC	BPA	ATLAS	DMC	DSDF	DGP	MeshP	NUD	GRID	IMNET	Ours
Display	0.468	0.495	0.834	0.071	0.108	0.632	0.417	0.903	0.903	0.551	0.601	**0.989**
Lamp	0.455	0.518	0.826	0.029	0.047	0.268	0.405	0.855	0.888	0.624	0.836	**0.891**
Airplane	0.415	0.442	0.788	0.070	0.050	0.350	0.249	0.844	0.872	0.564	0.698	**0.996**
Cabinet	0.392	0.392	0.553	0.077	0.154	0.573	0.513	0.860	0.950	0.733	0.343	**0.980**
Vessel	0.415	0.466	0.789	0.058	0.055	0.323	0.387	0.862	0.883	0.467	0.147	**0.985**
Table	0.233	0.287	0.772	0.080	0.095	0.577	0.307	0.880	0.908	0.844	0.425	**0.922**
Chair	0.382	0.433	0.802	0.050	0.088	0.447	0.481	0.875	0.913	0.710	0.181	**0.954**
Sofa	0.499	0.535	0.786	0.058	0.129	0.577	0.638	0.895	0.945	0.822	0.199	**0.968**
Mean	0.407	0.446	0.769	0.062	0.091	0.468	0.425	0.872	0.908	0.664	0.429	**0.961**

cloud $\boldsymbol{P}$. During training, we leverage a condition and a set of query locations $\boldsymbol{Q}$ to minimize the loss in Eq. 2. During testing, we reconstruct a 3D shape from an input image with a given condition using marching cube (Lorensen & Cline, 1987). We leverage the 2D encoder used by SoftRas (Liu et al., 2019a) to infer the 2D image conditions.

Dataset and Metric. We use the ShapeNet subset released by Choy et al (Choy et al., 2016) to evaluate the performance in single image reconstruction, where the dataset also contains rendered RGB images in 13 shape classes and a train/test split. After getting the reconstructed meshes, we first leverage the L1-CD and Normal Consistency (NC) to evaluate the reconstruction error between the reconstructed shapes and the 1×10^5 ground truth points released by OccNet (Mescheder et al., 2019), where we uniformly sample 1×10^5 points on the reconstructed shapes. To evaluate our method in a multi-scale way, we also uniformly sample 2048 points on both of reconstructed shapes and 1×10^5 point ground truth to evaluate reconstruction error using Earth Mover Distance (EMD).

Table 5. Surface reconstruction comparison in terms of F-score with a threshold of 2μ.

Class	PSR	DMC	BPA	ATLAS	DMC	DSDF	DGP	MeshP	NUD	Ours
Display	0.666	0.669	0.929	0.179	0.246	0.787	0.607	0.975	0.944	**0.991**
Lamp	0.648	0.681	0.934	0.077	0.113	0.478	0.662	**0.951**	0.945	0.924
Airplane	0.619	0.639	0.914	0.179	0.289	0.566	0.515	0.946	0.944	**0.997**
Cabinet	0.598	0.591	0.706	0.195	0.128	0.694	0.738	0.946	0.980	**0.989**
Vessel	0.633	0.647	0.906	0.153	0.120	0.509	0.648	0.956	0.945	**0.990**
Table	0.442	0.462	0.886	0.195	0.221	0.743	0.494	0.963	0.922	**0.973**
Chair	0.617	0.615	0.913	0.134	0.345	0.665	0.693	0.964	0.954	**0.969**
Sofa	0.725	0.708	0.895	0.153	0.208	0.734	0.834	0.972	0.968	**0.974**
Mean	0.618	0.626	0.885	0.158	0.209	0.647	0.649	0.959	0.950	**0.976**

Table 6. Reconstruction comparison in terms of L1-CD.

3D-R2	PSGN	DMC	OccNet	SSRNet	DDT	Ours
0.169	0.202	0.117	0.079	0.024	0.020	**0.011**

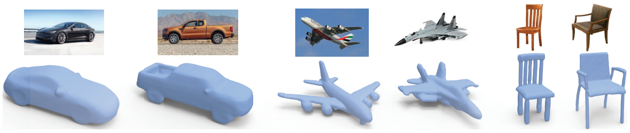

Figure 8. Reconstruction from real images

Comparison. We report numerical comparisons with 3D-R2 (Choy et al., 2016), PSGN (Fan et al., 2017), Pix2Mesh (Wang et al., 2018), ATLAS (Groueix et al., 2018), OccNet (Mescheder et al., 2019), IMNET (Chen & Zhang, 2019), 3DN (Wang et al., 2019a), DISN (Wang et al., 2019b) in Table 7. The comparison in terms of L1-CD and Normal Consistency shows that our method can significantly improve the reconstruction performance under almost all shape classes by providing more geometry details on the 3D shapes in higher resolution. The EMD comparison also shows our outperforming results over other methods under a sparse point setting. We further present a visual comparison with ATLAS (Groueix et al., 2018), OccNet (Mescheder et al., 2019) and SoftRas (Liu et al., 2019a) in Fig. 7 under Airplane, Chair and Lamp classes, which shows that we can reconstruct shapes with smoother surface in higher accuracy.

Reconstruction from Real Images. We collect some real images and reconstruct shapes using our model trained under synthetic data in Fig. 8. The high fidelity reconstructions show that our method can generalize well to real images.

4.3. Analysis

Ablation Study. We conduct ablation studies in surface reconstruction under FAMOUS dataset. First, we explore the contribution made by the geometric network initialization (GNI). We report the result without GNI as "No GNI" using the random network initialization in Table 8. The degenerated result compared to our method of "Ours" demonstrates that GNI can help the network to better understand the shape. Moreover, we highlight the strategy that we use in the query location sampling near the ground truth point clouds. We replace our sampling by randomly sampling query locations in the entire 3D space, where the number of query locations is kept the same. We report this result as "Space sampling" in Table 8, which demonstrates that it is more effective to use the query locations near the surface to probe the space for the learning. We also try to leverage an additional constraint introduced by IGR (Gropp et al., 2020) to keep the normal of gradient to be 1, but the result of "Gradient constraint" shows that the constraints bring no improvement.

The Effect of Noise. We further explore the effect of noise on the ground truth point clouds under the ABC and FA-

Table 7. Single image reconstruction comparison in terms of different metrics.

	L1-CD,10^5 points						Normal Consistency,10^5 points					EMD×100,2048 points					
	3D-R2	PSGN	Pix2Mesh	ATLAS	OccNet	Ours	3D-R2	Pix2Mesh	ATLAS	OccNet	Ours	IMNET	3DN	Pix2Mesh	ATLAS	DISN	Ours
Airplane	0.227	0.137	0.187	0.104	0.147	**0.016**	0.629	0.759	0.836	0.840	**0.858**	2.90	3.30	2.98	3.39	2.67	**1.32**
Bench	0.194	0.181	0.201	0.138	0.155	**0.016**	0.678	0.732	0.779	0.813	**0.820**	2.80	2.98	2.58	3.22	2.48	**1.37**
Cabinet	0.217	0.215	0.196	0.175	0.167	**0.018**	0.782	0.834	0.850	0.879	**0.888**	3.14	3.21	3.44	3.36	3.04	**1.62**
Car	0.213	0.169	0.180	0.141	0.159	**0.022**	0.714	0.756	0.836	0.852	**0.861**	2.73	3.28	3.43	3.72	2.67	**1.56**
Chair	0.270	0.247	0.265	0.209	0.228	**0.024**	0.663	0.746	0.791	**0.823**	0.810	3.01	4.45	3.52	3.86	2.67	**2.03**
Display	0.314	0.284	0.239	0.198	0.278	**0.020**	0.720	0.830	0.858	0.854	**0.867**	2.81	3.91	2.92	3.12	2.73	**1.64**
Lamp	0.778	0.314	0.308	0.305	0.479	**0.021**	0.560	0.666	0.694	0.731	**0.867**	5.85	3.99	5.15	5.29	4.38	**2.85**
Loudspeaker	0.318	0.316	0.285	0.245	0.300	**0.032**	0.711	0.782	0.825	0.832	**0.849**	3.80	4.47	3.56	3.75	3.47	**2.10**
Rifle	0.183	0.134	0.164	0.115	0.141	**0.019**	0.670	0.718	0.725	0.766	**0.811**	2.65	2.78	3.04	3.35	2.30	**1.41**
Sofa	0.229	0.224	0.212	0.177	0.194	**0.019**	0.731	0.820	0.840	**0.863**	0.856	2.71	3.31	2.70	3.14	2.62	**1.51**
Table	0.239	0.222	0.218	0.190	0.189	**0.025**	0.732	0.784	0.832	**0.858**	0.810	3.39	3.94	3.52	3.98	3.11	**1.99**
Telephone	0.195	0.161	0.149	0.128	0.140	**0.018**	0.817	0.907	0.923	0.935	**0.946**	2.14	2.70	2.66	3.19	2.06	**1.23**
Vessel	0.238	0.188	0.212	0.151	0.218	**0.027**	0.629	0.699	0.756	0.794	**0.827**	2.75	3.92	3.94	4.39	2.77	**1.71**
Mean	0.278	0.215	0.216	0.175	0.215	**0.021**	0.695	0.772	0.811	0.834	**0.851**	3.13	3.56	3.34	3.67	2.84	**1.72**

Table 8. Ablation studies in terms of L2-CD (×100).

No GNI	Space sampling	Gradient constraint	Ours
0.35	0.80	1.15	**0.22**

MOUS datasets in surface reconstruction. We conduct experiments using the "ABC max-noise" and "FAMOUS max-noise" with strong noise, "ABC var-noise" with varying noise strength, and "FAMOUS med-noise" with a constant noise strength, all of which are released by Points2Surf (Erler et al., 2020). We report our results under these datasets in Table 9, where we show that our method can better resist the noise than the state-of-the-art results. We also visually compare our results with noise and without noise under "FAMOUS med-noise" in Fig. 9. The slight degeneration further demonstrates our ability of learning signed distance functions from point cloud with noise.

Table 9. Comparison with noise in terms of L2-CD (×100).

Dataset	DSDF	ATLAS	PSR	Points2Surf	Ours
ABC var-noise	12.51	4.04	3.29	2.14	**0.72**
ABC max-noise	11.34	4.47	3.89	2.76	**1.24**
F-med-noise	9.89	4.54	1.80	1.51	**0.28**
F-max-noise	13.17	4.14	3.41	2.52	**0.31**
Mean	11.73	4.30	3.10	2.23	**0.64**

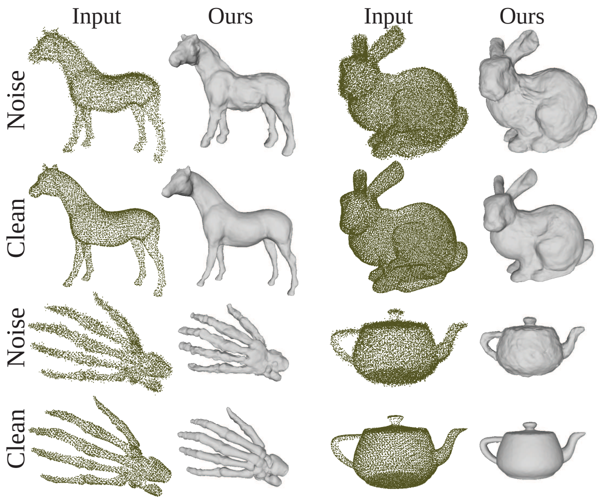

Figure 9. Demonstration of resisting noise.

Table 10. Effect of I in terms of L2-CD (×100) and time.

$\times 10^6$	1	2.5	5	10
Accuracy	0.434	0.394	0.223	**0.221**
Time (s)	103	210	530	1020

The Effect of Query Location Resolution. The number of query locations is also a factor that affects the learning. We explore its effect by merely adjusting the number of query locations under FAMOUS in surface reconstruction, such that $I = \{1, 2.5, 5, 10\} \times 10^6$. We report the comparison in Table 10, where the best result is achieved with $I = 10 \times 10^6$. Moreover, we test the time used in training in one epoch for different numbers of query locations. Although the result with $I = 10 \times 10^6$ is better than the one with $I = 5 \times 10^6$ which is used in our previous experiments, it takes much more time in training.

The Effect of GT Point Cloud Resolution. We also explore how the resolution of ground truth point clouds affects the performance under the FAMOUS dataset in surface reconstruction in Table 11. We keep the number of query locations the same to $I = 5 \times 10^6$, but employ ground truth point clouds with different numbers. Results in Table 11 show that higher resolutions of the ground truth can help our method to better infer the surface, but it also takes much more time to search the nearest neighbor on the ground truth point cloud when calculating the loss, especially in real applications. Moreover, we also compare our method with DSDF, ATLAS, PSR, Points2Surf under FAMOUS sparser ("F-sparser") and FAMOUS denser ("F-denser") datasets released by Points2Surf (Erler et al., 2020). Table 12 demonstrate that our method also achieves the best.

The Effect of Query Locations Range. Finally, we discuss

Table 11. Effect of J in terms of L2-CD ($\times 100$).

$\times 10^3$	1	2.5	5	10	20	40
	0.293	0.266	0.236	0.233	0.223	**0.213**

Table 12. Resolution comparison in terms of L2-CD ($\times 100$).

Dataset	DSDF	ATLAS	PSR	Points2Surf	Ours
F-sparse	10.41	4.91	2.17	1.93	**0.84**
F-dense	9.49	4.35	1.60	1.33	**0.22**
Mean	9.60	4.66	1.98	1.62	**0.44**

the effect of the query location range. We use the parameter σ^2 to control the maximum range of query locations around each point on the ground truth point cloud. We use several σ^2 candidates, including $\{0.25\sigma^2, 0.5\sigma^2, \sigma^2, 2\sigma^2, 4\sigma^2\}$, to randomly sample the same number of query locations. We report the results under the FAMOUS dataset in surface reconstruction in Table 13. The comparison shows that a too small or too large query location range will degenerate the surface reconstruction performance. Since it is hard to use the query locations to probe the area around the surface if the query location range is too small, while it is also hard to push the network to produce the accurate direction and distance to move the query locations to the surface if the query locations are too far away from the surface.

Table 13. Effect of σ^2 in terms of L2-CD ($\times 100$).

$\times\sigma^2$	0.25	0.5	1	2	4
	0.348	0.304	**0.223**	0.243	0.271

Latent Space Visualization. We visualize the latent space learned by our network in single image reconstruction under ShapeNet subset. We randomly select two reconstructed shapes in the test set of Airplane class or Chair class, and regard their latent codes as two ends to interpolate six new latent codes between them. We leverage these interpolated latent codes to generate novel shapes by the trained point decoder. We visualize these shape interpolations under each one of Airplane and Chair classes in Fig. 10, which shows that our method can reconstruct complex shapes with arbitrary topology. Moreover, the smooth transformation from one shape to another shape demonstrates that our method can help the network to learn a semantic latent space.

Loss Visualization. We further visualize the loss curves in surface reconstruction under ABC, Famous and ShapeNet dataset in Fig. 11. We can see that our method can effectively train a network to smoothly approach to the convergence.

5. Conclusion

We introduce Neural-Pull to learn signed distance functions from 3D point clouds by learning to pull 3D space onto the surface. Without the signed distance value ground truth, we can train a network to learn an SDF by pulling a sampled query location to its nearest neighbor on the surface. We effectively pull query locations along or against the gradient within the network with a stride of the predicted signed distance values. Being able to directly predict signed distances, our method successfully increases the 3D shape representation ability during testing. Our outperforming performance in single image reconstruction and surface reconstruction shows that we can reconstruct shapes and surfaces more accurately and flexibly than the state-of-the-art methods.

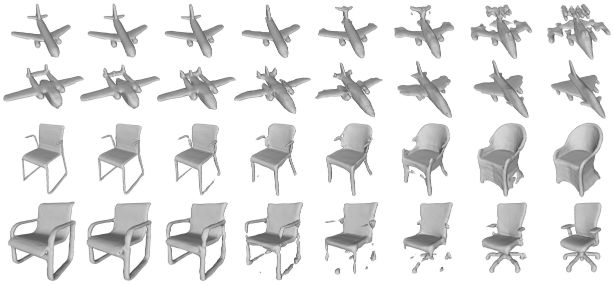

Figure 10. Interpolated shapes in single image reconstruction.

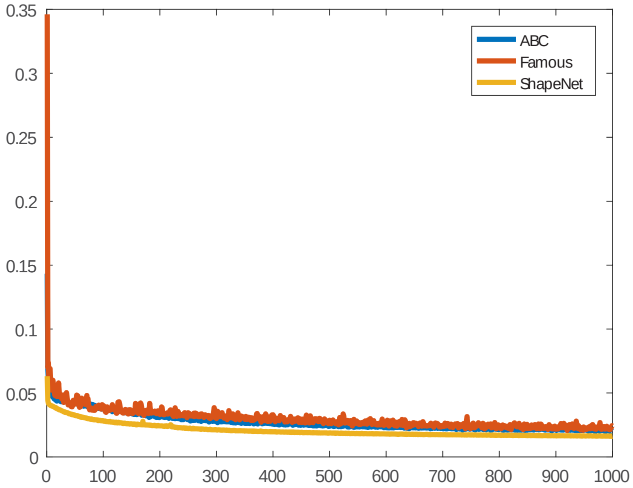

Figure 11. Training loss in surface reconstruction.

Acknowledgements

The corresponding author is Yu-Shen Liu. We thank the anonymous reviewers for reviewing our paper and providing helpful comments. This work was supported by National Key R&D Program of China (2020YFF0304100, 2018YFB0505400), the National Natural Science Foundation of China (62072268), and in part by Tsinghua-Kuaishou Institute of Future Media Data, and NSF (award 1813583).

References

Atzmon, M. and Lipman, Y. Sal: Sign agnostic learning of shapes from raw data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020a.

Atzmon, M. and Lipman, Y. SALD: Sign agnostic learning with derivatives. *arXiv*, 2006.05400, 2020b.

Azinovic, D., Martin-Brualla, R., Goldman, D. B., Nießner, M., and Thies, J. Neural RGB-D surface reconstruction. *CoRR*, abs/2104.04532, 2021.

Badki, A., Gallo, O., Kautz, J., and Sen, P. Meshlet priors for 3D mesh reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020.

Bednarik, J., Parashar, S., Gundogdu, E., and Salzmann, Mathieu andFua, P. Shape reconstruction by learning differentiable surface representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

Chabra, R., Lenssen, J. E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., and Newcombe, R. A. Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J. (eds.), *European Conference on Computer Vision*, volume 12374, pp. 608–625, 2020.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Chen, Z. and Zhang, H. Learning implicit fields for generative shape modeling. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Chibane, J., Alldieck, T., and Pons-Moll, G. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6968–6979, 2020a.

Chibane, J., Mir, A., and Pons-Moll, G. Neural unsigned distance fields for implicit function learning. *arXiv*, 2010.13938, 2020b.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *European Conference on Computer Vision*, pp. 628–644, 2016.

Dupont, E., Teh, Y. W., and Doucet, A. Generative models as distributions of functions. *CoRR*, abs/2102.04776, 2021.

Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N. J., and Wimmer, M. Points2Surf: Learning implicit surfaces from point clouds. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. (eds.), *European Conference on Computer Vision*, 2020.

Fan, H., Su, H., and Guibas, L. J. A point set generation network for 3D object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2463–2471, 2017.

Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W. T., and Funkhouser, T. Learning shape templates with structured implicit functions. In *International Conference on Computer Vision*, 2019.

Genova, K., Cole, F., Sud, A., Sarna, A., and Funkhouser, T. Local deep implicit functions for 3d shape. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020.

Gropp, A., Yariv, L., Haim, N., Atzmon, M., and Lipman, Y. Implicit geometric regularization for learning shapes. *arXiv*, 2002.10099, 2020.

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., and Aubry, M. A papier-mch approach to learning 3D surface generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Han, Z., Lu, H., Liu, Z., Vong, C.-M., Liu, Y.-S., Zwicker, M., Han, J., and Chen, C. P. 3D2SeqViews: Aggregating sequential views for 3D global feature learning by cnn with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28(8):3986–3999, 2019a.

Han, Z., Shang, M., Liu, Y.-S., and Zwicker, M. View Inter-Prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In *AAAI*, pp. 8376–8384, 2019b.

Han, Z., Shang, M., Liu, Z., Vong, C.-M., Liu, Y.-S., Zwicker, M., Han, J., and Chen, C. P. SeqViews2SeqLabels: Learning 3D global features via aggregating sequential views by rnn with attention. *IEEE Transactions on Image Processing*, 28(2):685–672, 2019c.

Han, Z., Wang, X., Liu, Y.-S., and Zwicker, M. Multi-angle point cloud-vae:unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In *IEEE International Conference on Computer Vision*, 2019d.

Han, Z., Wang, X., Vong, C.-M., Liu, Y.-S., Zwicker, M., and Chen, C. P. 3DViewGraph: Learning global features for 3D shapes from a graph of unordered views with attention. In *IJCAI*, 2019e.

Han, Z., Chen, C., Liu, Y.-S., and Zwicker, M. ShapeCaptioner: Generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences. In *ACM International Conference on Multimedia*, 2020a.

Han, Z., Chen, C., Liu, Y.-S., and Zwicker, M. DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In *International Conference on Machine Learning*, 2020b.

Han, Z., Qiao, G., Liu, Y.-S., and Zwicker, M. SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates. In *European Conference on Computer Vision*, 2020c.

Hu, T., Han, Z., and Zwicker, M. 3D shape completion with multi-view consistent inference. In *AAAI*, 2020.

Jia, M. and Kyan, M. Learning occupancy function from point clouds for surface reconstruction. *arXiv*, 2010.11378, 2020.

Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., and Funkhouser, T. Local implicit grid representations for 3D scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020a.

Jiang, Y., Ji, D., Han, Z., and Zwicker, M. SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020b.

Kazhdan, M. M. and Hoppe, H. Screened poisson surface reconstruction. *ACM Transactions Graphics*, 32(3):29:1–29:13, 2013.

Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., and Panozzo, D. ABC: A big cad model dataset for geometric deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2019.

Liao, Y., Donné, S., and Geiger, A. Deep marching cubes: Learning explicit surface representations. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Littwin, G. and Wolf, L. Deep meta functionals for shape representation. In *IEEE International Conference on Computer Vision*, 2019.

Liu, M., Zhang, X., and Su, H. Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. In *European Conference on Computer vision*.

Liu, S., Li, T., Chen, W., and Li, H. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. *IEEE International Conference on Computer Vision*, 2019a.

Liu, S., Saito, S., Chen, W., and Li, H. Learning to infer implicit surfaces without 3D supervision. In *Advances in Neural Information Processing Systems*, 2019b.

Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., and Cui, Z. DIST: Rendering deep implicit signed distance function with differentiable sphere tracing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Liu, X., Han, Z., Liu, Y.-S., and Zwicker, M. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *AAAI*, pp. 8778–8785, 2019c.

Liu, X., Han, Z., Liu, Y.-S., and Zwicker, M. Fine-grained 3d shape classification with hierarchical part-view attention. *IEEE Transactions on Image Processing*, 30: 1744–1758, 2021.

Lorensen, W. E. and Cline, H. E. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

Luo, Y., Mi, Z., and Tao, W. Deepdt: Learning geometry from delaunay triangulation for surface reconstruction. *CoRR*, abs/2101.10353, 2020.

Martel, J. N. P., Lindell, D. B., Lin, C. Z., Chan, E. R., Monteiro, M., and Wetzstein, G. ACORN: adaptive coordinate networks for neural scene representation. *CoRR*, abs/2105.02788, 2021.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3D reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Mi, Z., Luo, Y., and Tao, W. Ssrnet: Scalable 3D surface reconstruction network. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020.

Michalkiewicz, M., Pontes, J. K., Jack, D., Baktashmotlagh, M., and Eriksson, A. P. Deep level sets: Implicit surface representations for 3D shape inference. *CoRR*, abs/1901.06802, 2019.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Oechsle, M., Peng, S., and Geiger, A. UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *CoRR*, abs/2104.10078, 2021.

Ost, J., Mannan, F., Thuerey, N., Knodt, J., and Heide, F. Neural scene graphs for dynamic scenes. *CoRR*, abs/2011.10379, 2020.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Rematas, K., Martin-Brualla, R., and Ferrari, V. Sharf: Shape-conditioned radiance fields from a single view. In *ICML*, 2021.

Saito, S., , Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. *IEEE International Conference on Computer Vision*, 2019.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019.

Sitzmann, V., Martel, J. N., Bergman, A. W., Lindell, D. B., and Wetzstein, G. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020.

Songyou Peng, Michael Niemeyer, L. M. M. P. A. G. Convolutional occupancy networks. In *European Conference on Computer Vision*, 2020.

Takikawa, T., Litalien, J., Yin, K., Kreis, K., Loop, C., Nowrouzezahrai, D., Jacobson, A., McGuire, M., and Fidler, S. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. 2021.

Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020.

Tatarchenko, M., Richter, S. R., Ranftl, R., Li, Z., Koltun, V., and Brox, T. What do single-view 3D reconstruction networks learn? In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Stoll, C., and Theobalt, C. PatchNets: Patch-Based Generalizable Deep Implicit 3D Shape Representations. *European Conference on Computer Vision*, 2020.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y. Pixel2mesh: Generating 3D mesh models from single RGB images. In *European Conference on Computer Vision*, pp. 55–71, 2018.

Wang, W., Ceylan, D., Mech, R., and Neumann, U. 3DN: 3D deformation network. In *IEEE International Conference on Computer Vision*, 2019a.

Wang, W., Xu, Q., Ceylan, D., Mech, R., and Neumann, U. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. In *NeurIPS*, 2019b.

Wen, X., Han, Z., Liu, X., and Liu, Y.-S. Point2spatialcapsule: Aggregating features and spatial relationships of local regions on point clouds using spatial-aware capsules. *IEEE Transactions on Image Processing*, 29:8855–8869, 2020a.

Wen, X., Li, T., Han, Z., and Liu, Y.-S. Point cloud completion by skip-attention network with hierarchical folding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020b.

Wen, X., Han, Z., Cao, Y.-P., Wan, P., Zheng, W., and Liu, Y.-S. Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021a.

Wen, X., Xiang, P., Han, Z., Cao, Y.-P., Wan, P., Zheng, W., and Liu, Y.-S. Pmp-net: Point cloud completion by learning multi-step point moving paths. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021b.

Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., and Panozzo, D. Deep geometric prior for surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Wu, Y. and Sun, Z. DFR: differentiable function rendering for learning 3D generation from images. *Computer Graphics Forum*, 39(5):241–252, 2020.

Zakharov, S., Kehl, W., Bhargava, A., and Gaidon, A. Autolabeling 3D objects with differentiable rendering of sdf shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.