

# Neural-IMLS: Learning Implicit Moving Least-Squares for Surface Reconstruction from Unoriented Point Clouds

Zixiong Wang<sup>1</sup>, Pengfei Wang<sup>1</sup>, Qiujie Dong<sup>1</sup>, Junjie Gao<sup>1</sup>, Shuangmin Chen<sup>2</sup>, Shiqing Xin<sup>1</sup>  
 Changhe Tu<sup>1</sup>, Wenping Wang<sup>3</sup>

<sup>1</sup> Shandong University <sup>2</sup>Qingdao University of Science and Technology <sup>3</sup>Texas A&M University

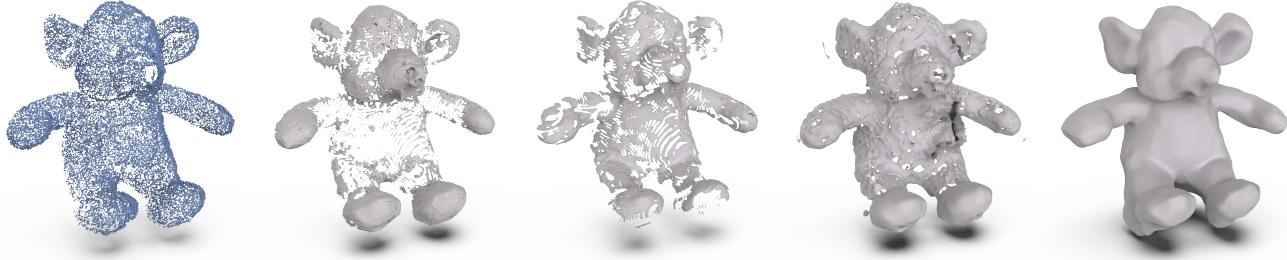


Figure 1. We present Neural-IMLS, a novel approach that directly learns a faithful signed distance function (SDF) from point clouds in a self-supervised fashion. Compared with other learning-based methods (from left to right: real scan data, Points2Surf [14], Neural-Pull [4], DeepMLS [31], and our result), our approach has the ability to reconstruct surfaces more accurately for noisy input without knowledge of the ground-truth SDF. The input data is the point cloud of model 105 in the DTU-MVS dataset [17], which is noisy and sparse (no points available on the backside).

## Abstract

*Surface reconstruction from noisy, non-uniform, and unoriented point clouds is a fascinating yet challenging problem in computer vision and graphics. With the innovations of 3D scanning technology, it is highly desired to directly transform raw scan data, typically with severe noise, into a manifold triangle mesh. Existing learning-based approaches aim at learning an implicit function whose zero-level surface encodes the underlying shape. However, most of them cannot obtain desirable results for noisy and sparse point clouds, limiting use in practice.*

*In this paper, we introduce Neural-IMLS, a novel approach that learns the noise-resistant signed distance function (SDF) directly from unoriented raw point clouds. Instead of explicitly learning priors with the ground-truth signed distance values, our method learns the underlying SDF from raw point clouds in a self-supervised fashion by minimizing the loss between a couple of SDFs, one obtained by the implicit moving least-square function (IMLS) and the other by our neural network, where the gradients of our predictor define the tangent bundle that facilitates the computation of IMLS. We prove that when the couple of SDFs coincide, our neural network can predict a signed implicit*

*function whose zero level-set serves as a good approximation to the underlying surface.*

*We conduct extensive experiments on various benchmarks, including synthetic scans and real-world scans, to exhibit the ability to reconstruct faithful shapes from various inputs, especially for point clouds with noise or gaps.*

## 1. Introduction

Surface reconstruction from unstructured point clouds remains a hot research topic since it feeds many downstream tasks in computer vision and graphics applications. While the recent innovation in 3D acquisition technology makes 3D data readily available, the raw data may be noisy, non-uniform, and unoriented. It is much desired that even with various artifacts, the reconstructed surface has high reliability, i.e., preserving as-much-as-possible shape variation information about the underlying surface. However, it is a challenging task if the input point cloud is unoriented, noisy, and sparse. Just as the Shannon sampling theorem states, it is not possible to completely reconstruct a band-limited signal if its sampling frequency is below a fixed threshold value.

Methods	Independence of normals	Independence of ground-truth SDF	Noise-resistance	Speed
SPR [21]	✗	✓	✗	✓
DeepSDF [28]	✓	✗	✗	✗
Points2Surf [14]	✓	✗	✓	✓
IGR [15]	✓	✓	✗	✓
SIREN [32]	✗	✓	✗	✓
Neural-Pull [4]	✓	✓	✗	✓
DeepMLS [31]	✓	✗	✓	✓
Ours	✓	✓	✓	✓

Table 1. Comparison of the main characteristics between the leading implicit reconstruction methods and our method. We only consider the speed of the inference stage. ✓ and ✗ denote YES and NO, respectively. ✗ indicates an “intermediate” level between YES and NO.

In general, reconstruction methods can be divided into explicit reconstruction (or interpolation-based methods) methods and implicit ones. Explicit methods [1, 6, 10, 13] aim at directly connecting a point cloud to a triangle mesh by taking each point as a vertex. For most explicit algorithms, the Delaunay triangulation or Voronoi diagram tool is utilized to guarantee high triangulation quality. However, the approaches in this category are sensitive to noise and hard to generate a manifold surface, especially when the input points are insufficient and noisy. Typical implicit approaches fit an implicit function (e.g., signed distance function (SDF)) [19, 27] from oriented point clouds, followed by extracting the isosurface using volumetric methods such as Marching Cubes [24]. It is worth noting that oriented normals help infer the variation of the implicit function in the region close to the point clouds and thus have a decisive influence on the final reconstruction result. However, estimating reliable oriented normals for low-quality point clouds is a highly non-trivial task.

In recent years, many learning-based algorithms have been proposed for implicit surface reconstruction. Most of them learn an implicit function by fitting data samples supervised with values of the ground-truth implicit function [8, 9, 14, 18, 25, 28, 29, 31]. Unfortunately, it is notoriously hard to get the ground-truth signed distance function for a real-world raw scan with severe noise. This motivates researchers to develop some methods [2–4, 15, 32] that learn an implicit function from raw point clouds directly (without knowledge of the ground-truth implicit function). SIREN [32] and IGR [15] need to input the normal vectors to get more convincing results, while Neural-Pull [4] learns signed distances by pulling a query point to the surface.

These approaches tend to set the input points on the zero-level surface and thus cannot easily eliminate the influence of noise. Instead, SAL [2] directly predicts the signed distance function using a sign agnostic loss that is defined as the mean absolute difference between the point cloud-based unsigned distance (from the query point to the point cloud) and the absolute value of the predicted signed distance. It

is able to yield a smooth surface due to the property of planar reproduction (see Figure 3). However, the distances, computed in this way, are wildly inaccurate around the zero-level surface. The predictor of SAL fails to help increase the accuracy for estimating the distance from the query point to the point cloud. As a result, SAL reports an overly-smooth surface as the output.

In this paper, we propose *Neural-IMLS*, a novel method to learn SDF directly from noisy, non-uniform, and unoriented clouds. We suggest supervising the function values as well as the gradients of the signed distance function simultaneously in a self-supervised fashion by minimizing the loss between the couple of SDFs, one obtained by the implicit moving least-square function (IMLS) and the other by our network. During each iteration, the signed distance reported by the IMLS function is computed based on the values and gradients of the up-to-date implicit function predicted by our network, making the couple of SDFs mutually corrected. When the optimization converges, they coincide and yield a learned IMLS predictor that accurately characterizes the underlying shape. In other words, we use IMLS as the smart mechanism to enforce the consistent pairing between the values and the gradients of the learned implicit function. It can be seen from Figure 3 that our algorithm is able to achieve a better trade-off between accuracy and noise-resistance. We show the main features of the leading techniques and our method in Table 1, which shows that our approach has a great potential in dealing with noisy scans.

We conduct extensive experiments on a large variety of shapes, including synthetic scans and real scans obtained by stereo image sensing technology. The experimental results demonstrate that our approach can learn more faithful SDFs than other leading techniques, especially when the input point cloud contains noise, gaps, and other defects.

## 2. Related Work

Surface reconstruction is a fundamental research problem in computer vision and graphics. Numerous reconstruction algorithms have been proposed [5, 33] in the last

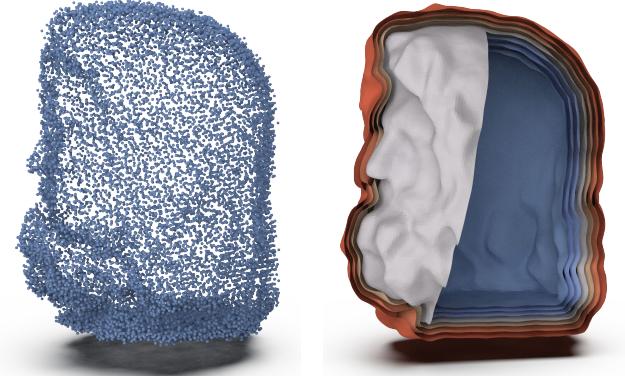


Figure 2. The level sets of  $f(\mathbf{q})$  (right) trained by our method. The warm color represents positive level sets, and the cold color represents negative level sets. The gray surface is produced by our method, which shows that our method has the ability to get faithfully reconstructed results for noisy point clouds.

three decades. In this section, we briefly review the implicit surface reconstruction methods, including traditional and learning-based methods.

## 2.1. Traditional Implicit Methods

Most traditional implicit methods require the input points to be equipped with oriented normals. The earliest implicit method [16] computes the projection distance from a query point to the tangent plane of the closest point. While this method is straightforward, it produces noisy output for non-uniform point clouds since the closest point maybe not be located on the underlying surface. Radial basis functions (RBF) are a well-known technique for data interpolation. Some methods [7, 12] represent the implicit function as a combination of a group of radial basis kernels and then enforce the value of the function to be zero at each data point. In order to avoid trivial solutions, they need to enforce  $f(\mathbf{p}_i + \epsilon \mathbf{n}_i) = \epsilon$  and  $f(\mathbf{p}_i - \epsilon \mathbf{n}_i) = -\epsilon$  at the same time. Poisson reconstruction [19] and its variants [20, 21] seek an indicator function based on the Poisson equation. The indicator function is able to encode the probability in which a point is located inside. Besides, implicit moving least methods (IMLS) [23, 27, 30] construct a low-degree polynomial function (typically a linear function) at input points and then generate a global approximation by a weighted average. To summarize, the traditional implicit reconstruction approaches need to take oriented normals as the input, and the reconstruction quality relies heavily on the reliability of normal vectors. However, estimating normal vectors is a highly non-trivial task, especially for point clouds with severe noise.

## 2.2. Learning Implicit Function with Data-Priors

Recently, several learning-based methods have been proposed to learn implicit functions from a large dataset with ground-truth SDFs or occupancy values. Early works tend to encode each shape into a latent code and then decode corresponding implicit functions to reconstruct the surface [9, 25, 28]. These methods can encode the overall shape for a group of similar models. They cannot deal with unseen shapes whose geometric and topological structures are much different from training data. Therefore, some local feature-based methods have been proposed to address this issue. Points2surf [14] uses global features to predict signs and local features to predict distances. It includes a step of sign propagation. When the input point cloud is noisy, sign propagation becomes inaccurate, leading to unwanted holes in the reconstructed mesh. By simultaneously utilizing the advantages of the implicit approaches and the point set methods, DeepMLS [31] can efficiently generate MLS points and the implicit approximation surfaces. Unfortunately, DeepMLS cannot provide reliable distances/gradients to the IMLS function, especially for sparse and noisy point clouds, and thus possibly end with a non-watertight surface with many holes. In [29], a convolutional operator is proposed for aggregating local and global information. It can reconstruct more complex point clouds like indoor scenes. Most of the approaches mentioned above, whether global approaches or local ones, need supervision with the ground-truth representation. However, it seems impossible to extend them to real scans since the raw data lacks ground-truth implicit/explicit representation.

## 2.3. Learning Implicit Function from Raw Scans

It is challenging to learn an implicit function from raw point clouds directly without knowledge of ground-truth SDF. Existing methods use some constraints (e.g., the gradient norm is being 1) to make the predicted implicit function own the properties of SDF. For example, IGR [15] introduces a geometric regularization term to encourage the predicted implicit function to have unit gradients, as the Eikonal equation asserts. Neural-Pull [4] trains neural networks by pulling query points to their closest points on the surface using the predicted signed distance and the gradients of the query points. SAL [2] performs sign agnostic regression to get a signed version of the unsigned distance function. After that, SALD [3] improves the sign agnostic loss by leveraging the additional gradient constraints and thus has a better ability in preserving geometric features. Recently, Sitzmann et al. [32] proposed a new activation function called Sine which can capture high-frequency information compared with ReLU [26]. While these methods can generate impressive reconstruction results with high-quality point clouds, most of them cannot produce faithfully reconstructed results when the noise level becomes severe.

### 3. Method

We give the problem statement and review the definition of the IMLS surface in Sections 3.1 and 3.2, and then we elaborate the mechanism of our network, including the loss function in Section 3.3 and the proof on the convergence in Section 3.4.

#### 3.1. Problem Statement and Motivation

Given an input point cloud  $\mathbf{P} = \{\mathbf{p}_i\}_{i \in I} \in \mathbb{R}^3$  without normals  $\mathbf{N} = \{\mathbf{n}_i\}_{i \in I} \in \mathbb{R}^3$ , we aim to learn a signed distance function  $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ , from which we shall reconstruct a surface  $S$  that is the zero level-set of  $f$ :

$$S = \{f(\mathbf{q}) = 0, \quad \mathbf{q} \in \mathbb{R}^3\}. \quad (1)$$

Existing works use two types of supervision mechanisms to train a continuous  $f$ . The first kinds of approaches formulate this problem as a regression problem [14, 18, 28, 31] and train the neural network with ground-truth signed distance values, and predict signed distance values by decoding latent codes. Generally, they need a huge amount of data and a high timing cost to get converged. The second kinds of approaches leverage additional constraints [2–4, 15, 32] to deal with raw data, but they are weak in eliminating the influence of the noisy points and require the input point clouds equipped with normals  $\mathbf{N}$  to get desirable results in general. Instead, our method does not need the normals at all and learns in a self-supervised fashion. The supervision signal comes from the IMLS function. When the optimization converges, the distances and gradients of the implicit function coincide, yielding a learned IMLS predictor that accurately characterizes the underlying shape. In its nature, our algorithm uses IMLS as the smart mechanism to enforce the consistency between the values and the gradients of the learned implicit function such that it is able to achieve a better trade-off between accuracy and noise-resistance.

#### 3.2. IMLS Surface and Our Insight

The IMLS surface [23] is defined as the zero level set of the IMLS function. Specifically, for the query point  $\mathbf{q}$ , the reconstructed value is calculated as a weighted least squares measure biased towards the region around the query point. It assumes that each point  $\mathbf{p}_i$  has a tangent plane whose normal vector  $\mathbf{n}_i$  is either specified by the user or estimated by approximation. In this paper,  $\mathbf{n}_i$  shall be automatically inferred by our neural network. The formulation of the IMLS function is shown as below:

$$g(\mathbf{q}) = \frac{\sum_{\mathbf{p}_i \in \mathcal{P}} \theta(\|\mathbf{q} - \mathbf{p}_i\|) \cdot \langle \mathbf{q} - \mathbf{p}_i, \mathbf{n}_i \rangle}{\sum_{\mathbf{p}_i \in \mathcal{P}} \theta(\|\mathbf{q} - \mathbf{p}_i\|)}, \quad (2)$$

where  $\|\cdot\|$  is the L2-norm operator and  $\theta(x)$  is a smooth weighting function. We use the Gaussian function  $\theta(x) =$

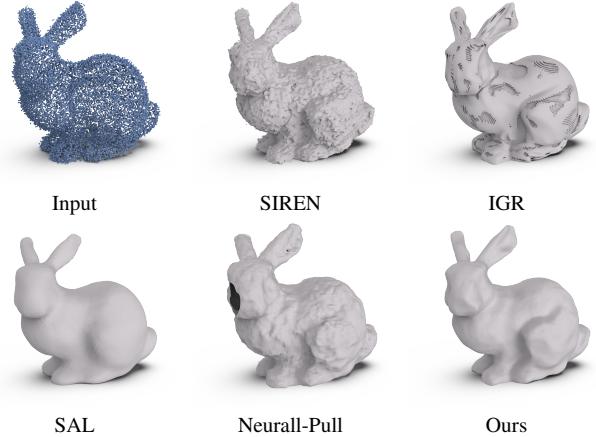


Figure 3. Reconstruction results on a noisy point cloud using the methods that learn from raw data. It is difficult to seek a good trade-off between accuracy and noise-resistance. Most of the existing solutions (SIREN [32], IGR [15], and Neurall-Pull [4]) are sensitive to noise, whereas SAL [2] tends to report an overly-smooth result, failing to preserve geometric features.

$\exp(-x^2/\sigma_s^2)$  to define the weighting scheme, where  $\sigma_s$  is the support radius of the Gaussian function.

In [23], the author has proved that the IMLS surface reconstructed from the IMLS function is a good approximation of the original surface and is geometrically and topologically correct when the input point cloud satisfies the  $\epsilon$ -sampling condition. In fact, even for a point cloud with lower sampling quality, the IMLS function still serves as a good approximation of the SDF of the underlying surface. In the implementation,  $\theta$  decreases very sharply when  $\|\mathbf{q} - \mathbf{p}_i\|$  becomes larger, making distant points (from the query point  $\mathbf{q}$ ) have little contribution to the function value at  $\mathbf{q}$ . Therefore, we consider only a small set of points in a local patch.

$$g(\mathbf{q}) = \frac{\sum_{\mathbf{p}_i \in \mathcal{P}} \theta(\|\mathbf{q} - \mathbf{p}_i\|) \cdot \langle \mathbf{q} - \mathbf{p}_i, \mathbf{n}_i \rangle}{\sum_{\mathbf{p}_i \in \mathcal{P}} \theta(\|\mathbf{q} - \mathbf{p}_i\|)}, \quad (3)$$

where  $\mathcal{P} \subset \mathbf{P}$  is the local patch of the point cloud  $\mathbf{P}$ , and  $\mathcal{P}$  is the point subset in the  $r$ -radius ball centered at the query point  $\mathbf{q}$ . (If  $\mathcal{P}$  is empty, we deem  $\mathbf{q}$  as an invalid query point.)

It is worth noting that the IMLS surface, defined by Eq. (3), is the approximation of the original surface. In fact, one can define the gradients at  $\mathbf{q}$  using the same weighting scheme as defined by Eq. (3). An interesting observation is that for an invalid normal-vector configuration, the distances estimated by IMLS are inconsistent with the gradients estimated by IMLS. This observation motivates us to use IMLS as a black box to tune the neural network such that the gradients of the learned function  $f$  can align with

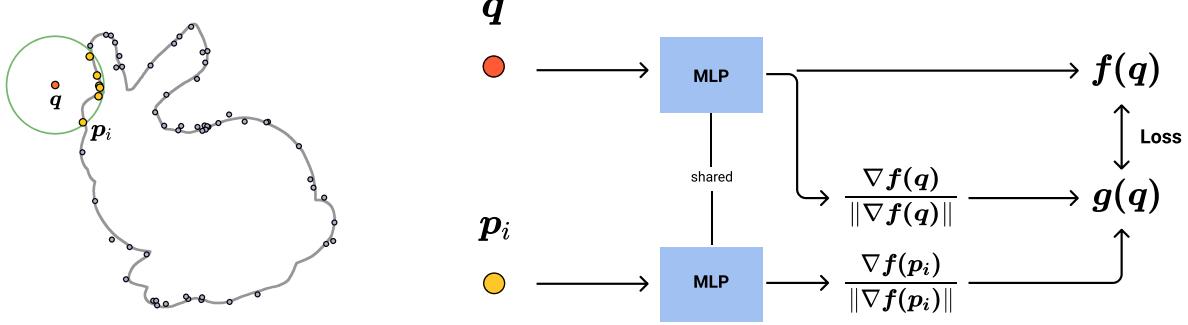


Figure 4. The self-supervised pipeline of our method. For query point  $q$  (red), we search its nearest neighbors  $p_i$  (yellow), which are inside the ball (green) centered at the query point  $q$  with radius  $r$  from input point clouds. The supervision signal is provided from the IMLS function  $g(q)$  using the gradients  $\nabla f(q), \nabla f(p_i)$  computed by back-propagation, and the neural network is updated by minimizing the error between  $f(q)$  and  $g(q)$ .

the normal vectors at the data points. Therefore, we define a self-supervised formulation to leverage this property and learn high-quality SDF directly from noisy point clouds.

### 3.3. Neural-IMLS

We introduce Neural-IMLS to learn SDF by the supervision signal provided by the IMLS function. We use the gradients at data points as the oriented normals to train with the derivatives based on the observation that regression with derivatives can significantly reduce the sample complexity of problem [11]. We construct pseudo-labels based on the IMLS function, using the information from the network to supervise the network converging to a signed distance function. For query point  $q$ , the signed distance value  $f(q)$  is predicted by the network. Besides, the gradients  $\nabla f(q)$  at  $q$  can be elegantly computed analytically with back-propagation. Since we are only considering the direction of  $\nabla f(q)$ , we normalize that as the unit vector  $\nabla f(q)/\|\nabla f(q)\|$ . The same operation can be done for point  $p_i$  in the point clouds. Therefore, we can compute the IMLS function by leveraging the gradients as the normal:

$$g(q) = \frac{\sum_{p_i \in \mathcal{P}} \theta(\|q - p_i\|) \cdot \left\langle q - p_i, \frac{\nabla f(p_i)}{\|\nabla f(p_i)\|} \right\rangle}{\sum_{p_i \in \mathcal{P}} \theta(\|q - p_i\|)}. \quad (4)$$

To supervise the network as IMLS function  $g(q)$ , we leverage a squared error between  $g(q)$  and  $f(q)$ :

$$L = \frac{1}{|Q|} \sum_{q_j \in Q} \|g(q_j) - f(q_j)\|_2^2, \quad (5)$$

where  $|Q|$  is the number of the set of query points  $Q$ .

Unfortunately, although this formulation can work well for simple shapes, it cannot correctly reconstruct shapes with thin structures. The thin structures are challenging for the reconstruction task since there are ambiguities for orienting the sign of normal. For this reason, we introduce an

additional gradient similarity term to the IMLS function for helping gradients converge. Specifically, we compare the difference of the gradients between query point  $q$  and its neighbors  $p_i$ . The term is defined as

$$\psi(\nabla f(q), \nabla f(p)) = \exp\left(-\frac{\left\|\frac{\nabla f(q)}{\|\nabla f(q)\|} - \frac{\nabla f(p)}{\|\nabla f(p)\|}\right\|^2}{\sigma_r^2}\right), \quad (6)$$

where the parameter  $\sigma_r$  scales the similarity of neighboring normals. Now we can combine both weighting functions together:

$$\omega(q, p) = \theta(\|q - p\|) \psi(\nabla f(q), \nabla f(p)). \quad (7)$$

Based on that, our IMLS function becomes:

$$g(q) = \frac{\sum_{p_i \in \mathcal{P}} \omega(q, p_i) \cdot \langle q - p_i, \nabla f(p_i) \rangle}{\sum_{p_i \in \mathcal{P}} \omega(q, p_i)}. \quad (8)$$

We still use the loss function described by Equation (5) to constrain the neural network  $f(q)$  as close to IMLS function  $g(q)$  as possible. When the couple of SDFs coincide, the learned function converges to a signed distance function.

### 3.4. Convergence to Signed Distance Function

The last question is why  $f$  can converge to the signed distance function with our formulation rather than an unsigned one. As shown in [4], the gradients at  $q + \Delta q$  and  $q - \Delta q$  are quite similar ( $\Delta q$  is very small) although their distances have different signs.

**Theorem 1.** *The continuous MLP function  $f$ , trained with the supervision of Equation (5), can coverage to a signed distance function.*

*Proof.* Let  $q$  be a point on the surface ( $f(q) = 0$ ). We suppose  $\Delta q$  is small enough such that  $q + \Delta q$  is restricted in a very small neighborhood of  $q$ .

We assume that  $f$  is an unsigned distance field whose zero-level surface has a 3-dimensional measure (volume) of 0. Then we have  $\nabla f(\mathbf{q}) = \mathbf{0}$  (or equivalently  $\|\nabla f(\mathbf{q})\| = 0$ ), which further implies that  $\mathbf{g}(\mathbf{q} + \Delta\mathbf{q}) = \mathbf{0}$  for any  $\Delta\mathbf{q}$  according to Equation (8).

Therefore, our loss for the point  $\mathbf{q} + \Delta\mathbf{q}$  becomes

$$\begin{aligned} 0 &= \|\mathbf{g}(\mathbf{q} + \Delta\mathbf{q}) - f(\mathbf{q} + \Delta\mathbf{q})\|_2^2 \\ &= \|\mathbf{-f}(\mathbf{q} + \Delta\mathbf{q})\|_2^2 = f(\mathbf{q} + \Delta\mathbf{q})^2, \end{aligned}$$

which holds for any small  $\Delta\mathbf{q}$ , resulting in a zero-valued region with non-vanishing 3-dimensional measure. This contradicts our assumption that  $f$  is an unsigned distance field whose zero-level surface has a 3-dimensional measure of 0.  $\square$

## 4. Experiments

### 4.1. Experimental Setting

We overfit the neural network on a single point cloud and use the MLP architecture to represent the shape, which includes seven hidden layers with 512 hidden units per layer. Moreover, we use ReLU [26] as the activation function. The network predicts the signed distance value for each query point. Furthermore, we use GNI (geometric network initialization) to initialize  $f$ , where the initial shape is assumed to be a sphere. More details about the training configuration and the hyperparameters can be found in our supplemental material.

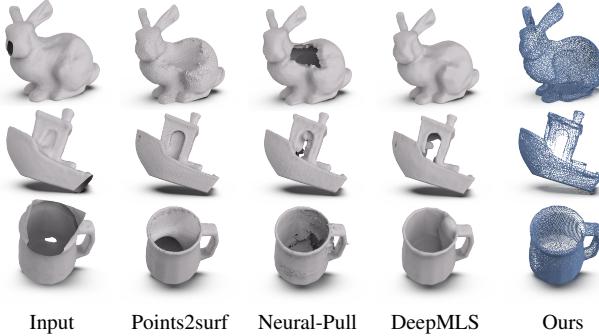


Figure 5. Qualitative comparison under the FAMOUS no-noise dataset.

**Dataset.** We evaluate our method on the subset of the ABC dataset [22] and the FAMOUS dataset [14] released by Points2Surf [14]. The ABC dataset contains various CAD shapes and the FAMOUS dataset contains 22 famous shapes in computer graphics. We used variants of the above two datasets to demonstrate the performance of our method. Furthermore, we also conducted tests on real scans from the DTU-MVS dataset [17] to test our method.

**Metric.** After overfitting for a single point cloud, we use the Marching Cubes [24] algorithm to extract 0-isosurface from

Method	NC $\uparrow$	L2-CD $\downarrow$	Non-MF $\downarrow$	Non-WT $\downarrow$
P2S	0.905	0.010	1	1
NP	0.920	0.221	0.261	0.348
DM	0.925	0.007	<b>0</b>	0.550
Ours	<b>0.929</b>	<b>0.006</b>	<b>0</b>	<b>0</b>

Table 2. Quantitative comparison of reconstruction errors under the FAMOUS no-noise dataset.

the  $128^3$  grid as the reconstruction surface. To comprehensively evaluate the reconstruction quality of our results, we introduce Normal Consistency (NC) and L2-Chamfer distance (L2-CD, we multiplied by 100) to compute reconstruction errors between the reconstructed result and the ground truth with  $1 \times 10^5$  random samples. Besides, we also report the ratio of the non-manifold (None-MF) shapes and the non-watertight (Non-WT) shapes for each dataset. The detailed metrics formulation is provided in the supplemental material.

Method	NC $\uparrow$	L2-CD $\downarrow$	Non-MF $\downarrow$	Non-WT $\downarrow$
P2S	0.920	0.087	0.980	0.980
NP	0.838	0.357	0.060	0.100
DM	0.915	<b>0.020</b>	0.020	0.650
Ours	<b>0.925</b>	0.028	<b>0</b>	<b>0.060</b>

Table 3. Quantitative comparison of reconstruction errors on the ABC no-noise dataset.

**Network Efficiency.** With the default parameters, our method takes the 90s averagely for each training epoch, and the network takes 4s for inferring the function values and then performing isosurface extraction. The reasoning is very fast due to the simplicity of the network architecture (with a quite limited number of parameters).

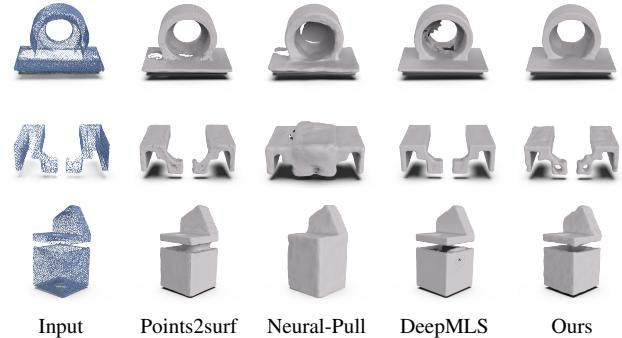


Figure 6. Qualitative comparison under the ABC no-noise dataset.

**Baselines.** To our best knowledge, existing approaches that learn from raw data cannot work well on noisy scans

Method	NC	L2-CD	Non-MF	Non-WT
P2S	0.892	0.011	1	1
NP	0.897	0.066	0.250	0.375
DM	0.904	0.013	0	0.818
Ours	<b>0.913</b>	<b>0.010</b>	<b>0</b>	<b>0</b>

Table 4. Quantitative comparison of reconstruction errors on the FAMOUS med-noise dataset.

(see Figure 3). By contrast, data priors-based methods are better in handling noisy scans. Therefore we choose famous data priors-based methods as our primary baselines, including Points2Surf (P2S) [14], Neural-Pull (NP) [4], and DeepMLS (DM) [31]. We reproduce the results of Points2Surf and DeepMLS by the pre-trained model released by the official. Besides, we use the results of noise-free point clouds of Neural-Pull released by the official and reproduce the other results using the official code.

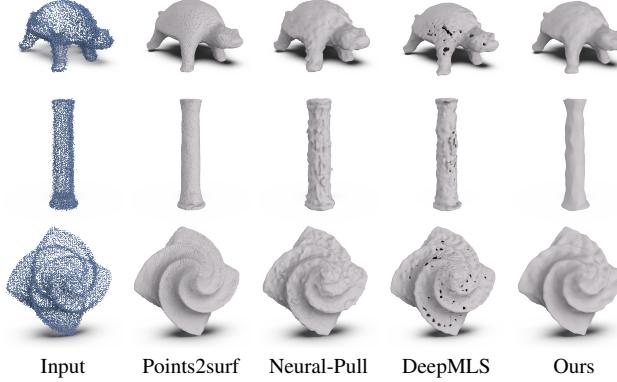


Figure 7. Qualitative results of 3D point clouds reconstruction on the FAMOUS med-noise dataset.

#### 4.2. Surface reconstruction from point clouds

A quantitative comparison about the reconstruction accuracy on the FAMOUS no-noise dataset and the ABC-varnoise dataset is shown in Table 2 and 3. Figure 5 and 6 show the visual results. Compared with other methods, our method gets much better results for point clouds with missing parts. At the same time, the results of our method are more topologically faithful. The table comparison demonstrates that our method can reconstruct more accurate surfaces than Points2Surf. It's worth noting that all the results of Points2Surf on the FAMOUS no-noise dataset are non-manifold. The non-manifoldness may be due to the flawed sign propagation mechanism that makes many points fail to get a correct sign (inside or outside). DeepMLS learns priors of a local surface patch, which makes the learned features not rely on specific shape classes. However, it cannot

effectively complete missing parts of point clouds since this method cannot predict a valid radius (generally too small) for each MLS point. Therefore, it yields many unwanted holes or redundant fragments in the reconstructed shapes. Neural-Pull learns the SDF by pulling the sample points to the nearest point in input. Although it is straightforward to implement, this approach suffers from several issues. Similar to the early implicit method [16] which considers the tangent plane distance of the nearest point as the signed distance, it produces topologically incorrect results, especially for shapes with narrow slits (see the 3rd column of Figure 6). Furthermore, when the query points are located near the surface, the SDF predicted by it becomes inaccurate since their method moves the query points toward the data points (rather than toward the real surface) according to their formulation. Therefore, the level-set of this method is zigzag, and it has to find an appropriate value (generally not 0) to define the isosurface.

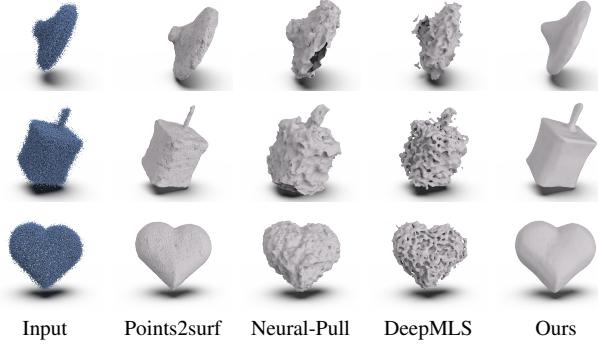


Figure 8. Qualitative comparison under the FAMOUS no-noise dataset.

Method	NC $\uparrow$	L2-CD $\downarrow$	Non-MF $\downarrow$	Non-WT $\downarrow$
P2S	0.897	<b>0.018</b>	0.99	0.99
NP	0.867	0.272	0.01	0.74
DM	0.860	0.052	0.02	0.84
Ours	<b>0.910</b>	0.056	<b>0</b>	<b>0</b>

Table 5. Quantitative comparison of reconstruction errors under the ABC var-noise dataset.

**Influence of Noise** To test the denoising ability of our method, we conducted experiments on two datasets, i.e., the ABC var-noise with varying noise strength and the FAMOUS med-noise with a constant noise strength. The statistics are reported in Table 4 and Table 5. It can be seen that our method gets the best performance on the FAMOUS med-noise dataset. Points2Surf handles noisy input well and gets the lowest L2-CD errors on the ABC-varnoise dataset since the pre-trained model comes from the same shape category (CAD shapes with varying noise

strength), but most of their results are non-manifold and non-watertight. DeepMLS cannot reconstruct convincing surfaces (see the fourth column of Figure 8) from noisy input, and its results are not watertight. From the visual results, we can observe that Neural-Pull is sensitive to noise. Neural-Pull assumes all points to be on the 0-isosurface, including noisy ones, which accounts for why it cannot eliminate the influence of noisy points. For ABC var-noise dataset, part of our results loses geometric details. In fact, we use large local size (the value of  $r$ ) to adapt severe noise, it is too large for the other shapes with low noise or noise-free in the ABC var-noise dataset. In fact, users can tune  $r$  according to the level of noise to further improve the reconstruction quality rather than use the a fixed  $r$ .

Method	NC $\uparrow$	L2-CD $\downarrow$	Non-MF $\downarrow$	Non-WT $\downarrow$
P2S	0.919	<b>0.003</b>	0.909	0.909
NP	0.898	0.055	0.250	0.375
DM	0.908	0.014	0.045	0.909
Ours	<b>0.924</b>	0.005	<b>0</b>	<b>0</b>

Table 6. Quantitative comparison of reconstruction errors under the FAMOUS dense dataset.

**The Effect of Input Point Clouds Density.** We further study how the reconstruction quality depends on the density of input point clouds. We evaluate the above-mentioned methods on the FAMOUS dense dataset and the FAMOUS sparse dataset. The above two datasets have different sampling densities. Besides, all the point clouds in these datasets have constant noise strength. We report the test results in Table 6 and Table 7. Without a doubt, the high-density point clouds yield more accurate reconstruction results, while sparse point clouds lead to inaccurate results. By contrast, our approach outperforms the others on the two datasets. When the input has missing parts, our approach can complete the region and yield a satisfactory result (except one shape in the FAMOUS sparse dataset).

Method	NC $\uparrow$	L2-CD $\downarrow$	Non-MF $\downarrow$	Non-WT $\downarrow$
P2S	0.864	0.027	1	1
NP	0.889	0.055	0.27	0.36
DM	0.891	0.023	<b>0</b>	0.78
Ours	<b>0.897</b>	<b>0.022</b>	<b>0</b>	<b>0.05</b>

Table 7. Qualitative comparison under the FAMOUS sparse dataset.

**Real Scans.** We also experimented these approaches and compared them on real-world data, as shown in Figure 1 and 9. The point clouds from the DTU-MVS dataset are highly noisy and sparse (no points available on the back-side), which is very challenging to reconstruct a complete

surface. In Figure 9, all the points at the pedestal part are missing. Our algorithm is able to yield a complete surface (including the backside) although the completion maybe a little from the real situation. The results of Points2Surf have many holes due to the issue of sign propagation. DeepMLS does not work well on the real scans and generates redundant fragments that do not exist in the original point clouds. Neural-Pull is weak in dealing with severe noise and produces a very bumpy surface as the output due to the flawed pulling strategy.

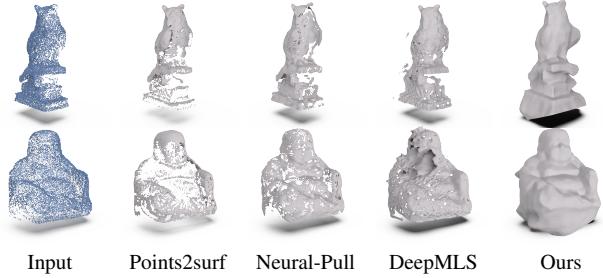


Figure 9. Real scans reconstruction with noisy and sparse point clouds (model 114, 122 from the DTU dataset [17]).

**Limitations.** A limitation of this approach lies in that our neural network, in its current form, fails to detect the noise level automatically and then adaptively tune the parameter  $r$ . In the future, we shall further explore how to invent an intelligent mechanism to tune the parameter such that the reconstruction result is noise resistant but can preserve as many geometric features as possible.

## 5. Conclusions

We introduced Neural-IMLS, a method that learns noise-resistant SDF directly from raw point clouds (no normal vectors equipped) in a self-supervised fashion. By leveraging the pseudo-label constructed by the IMLS function, our method gets high-quality SDF directly from noisy point clouds without the knowledge of the ground-truth signed distance values. Even when severe noise exists, our algorithm can still yield a faithful surface. We also prove why our neural network converges to a signed distance function rather than an unsigned one. Extensive experimental results on several benchmarks show that our method effectively eliminates the influence of noise and is very noise-resistant. Furthermore, it can still get a well-shaped surface when the input point cloud has missing parts.

In the future, it is interesting to leverage new kinds of pseudo-label to improve the ability to preserve sharp features, which is central to model CAD shapes. A possible research direction is to redefine the projection behavior such that a query point can be projected onto the feature line if the original projection point is very near to the feature line.

## References

- [1] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266, 2001. 2
- [2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3, 4
- [3] Matan Atzmon and Yaron Lipman. SALD: sign agnostic learning with derivatives. In *9th International Conference on Learning Representations, ICLR 2021*, 2021. 2, 3, 4
- [4] Ma Baorui, Han Zhizhong, Liu Yu-shen, and Zwicker Matthias. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In *International Conference on Machine Learning (ICML)*, 2021. 1, 2, 3, 4, 5, 7
- [5] Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva. A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*, volume 36, pages 301–329. Wiley Online Library, 2017. 2
- [6] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Claudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359, 1999. 2
- [7] Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, 2001. 3
- [8] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision*, pages 608–625. Springer, 2020. 2
- [9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2, 3
- [10] David Cohen-Steiner and Frank Da. A greedy delaunay-based surface reconstruction algorithm. *The visual computer*, 20(1):4–16, 2004. 2
- [11] Wojciech Marian Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4281–4290, 2017. 5
- [12] Huong Quynh Dinh, Greg Turk, and Greg Slabaugh. Reconstructing surfaces by volumetric regularization using radial basis functions. *IEEE transactions on pattern analysis and machine intelligence*, 24(10):1358–1371, 2002. 3
- [13] Herbert Edelsbrunner and Ernst P Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72, 1994. 2
- [14] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2surf learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, pages 108–124. Springer, 2020. 1, 2, 3, 4, 6, 7
- [15] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020. 2, 3, 4
- [16] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, pages 71–78, 1992. 3, 7
- [17] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 1, 6, 8
- [18] 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. 2, 4
- [19] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 2, 3
- [20] Misha Kazhdan, Ming Chuang, Szymon Rusinkiewicz, and Hugues Hoppe. Poisson surface reconstruction with envelope constraints. In *Computer Graphics Forum*, volume 39, pages 173–182. Wiley Online Library, 2020. 3
- [21] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 2, 3
- [22] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9611, 2019. 6
- [23] Ravikrishna Kolluri. Provably good moving least squares. *ACM Transactions on Algorithms (TALG)*, 4(2):1–25, 2008. 3, 4
- [24] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 2, 6
- [25] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2, 3
- [26] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icm*, 2010. 3, 6
- [27] A Cengiz Öztïreli, Gael Guennebaud, and Markus Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer graphics forum*, volume 28, pages 493–501. Wiley Online Library, 2009. 2, 3
- [28] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation.

- In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2, 3, 4
- [29] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2, 3
- [30] Chen Shen, James F O’Brien, and Jonathan R Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *ACM SIGGRAPH 2004 Papers*, pages 896–904. Association for Computing Machinery, 2004. 3
- [31] Hao-Xiang Guo Shi-Lin Liu, Hao Pan, Pengshuai Wang, Xin Tong, and Yang Liu. Deep implicit moving least-squares functions for 3d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 3, 4, 7
- [32] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 2, 3, 4
- [33] Cheng Chun You, Seng Poh Lim, Seng Chee Lim, Joi San Tan, Chen Kang Lee, and Yen Min Jasmina Khaw. A survey on surface reconstruction techniques for structured and unstructured data. In *2020 IEEE Conference on Open Systems (ICOS)*, pages 37–42, 2020. 2