

# 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>

<sup>1</sup> School of Computer Science and Technology, Shandong University, Qingdao, China

<sup>2</sup> School of Information and Technology, Qingdao University of Science and Technology, Qingdao, China

## Abstract

Surface reconstruction from noisy, non-uniformly, and unoriented point clouds is a fascinating yet difficult problem in computer vision and computer graphics. In this paper, we propose *Neural-IMLS*, a novel approach that learning noise-resistant signed distance function (SDF) for reconstruction. Instead of explicitly learning priors with the ground-truth signed distance values, our method learns the SDF from raw point clouds directly 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. Finally, a watertight and smooth 2-manifold triangle mesh is yielded by running Marching Cubes. We conduct extensive experiments on various benchmarks to demonstrate the performance of Neural-IMLS, especially for point clouds with noise.

## Introduction

The recent innovation in 3D acquisition technology makes 3D data easily available. The raw data may be with various artifacts, mostly noisy, non-uniformly, and unoriented. One of the fundamental research tasks is to reconstruct a watertight and accurate shape from the defective data.

In general, reconstruction methods can be divided into explicit reconstruction methods and implicit ones. Explicit methods (Edelsbrunner and Mücke 1994; Bernardini et al. 1999) aim at directly connecting a point cloud to a triangles 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 weak in generating a watertight 2-manifold surface, especially when the input points are insufficient and noisy. Instead, the implicit approaches fit an implicit function (e.g., signed distance function (SDF)) (Kazhdan, Bolitho, and Hoppe 2006) from oriented point clouds, followed by extracting the iso-surface using volumetric methods such as Marching Cubes (Lorensen and Cline 1987). However, the methods in this category need additional information, such as oriented normals, to prevent trivial solutions. As a result, the quality of oriented normal vectors has a decisive influence on reconstruction results.

In recent years, many learning-based algorithms have been proposed for surface reconstruction by learning SDF. Given an image or point clouds, most of them regress a

global or local latent code and then decode an implicit function by the neural network, followed by extracting the final mesh using volumetric methods (Mescheder et al. 2019; Chen and Zhang 2019; Jiang et al. 2020; Erler et al. 2020). Despite owning the ability to process partial scans based on data-priors, they are very time-consuming at the inference stage since there are many parameters in encoder modules.

Instead, other methods (Atzmon and Lipman 2020; Gropp et al. 2020; Sitzmann et al. 2020; Atzmon and Lipman 2021; Baorui et al. 2021) that learn an implicit function from raw point clouds directly. Most of them consider predicting sign distance fields from unsigned distance fields by leveraging additional constraints. In this way, they have the ability to learn the sign information, which is the key to reconstruction. Nevertheless, these approaches cannot handle noisy scans since they are weak in eliminating the influence of noisy points.

In this paper, we propose *Neural-IMLS*, a novel method to learn SDF directly from noisy, non-uniformly, and unoriented clouds. Rather than supervising signed distance values only, we suggest supervising signed distance value and its gradients 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, i.e., for a query point  $q$  and its neighboring points in the point clouds, we hope that the predicted by our network is as close as the IMLS based SDF at  $q$ . When the couple of SDFs coincide, we get a learned IMLS predictor that can report the reliable SDF even if the input point clouds contain noise. We conduct extensive experiments on several datasets and provide ablations studies. The experiments demonstrate the performance of our method, especially for point clouds that contain noise.

## Related Work

Surface reconstruction is a fundamental problem in computer vision and computer graphics. Numerous reconstruction algorithms have been proposed (You et al. 2020). In this section, we briefly review the implicit-based surface reconstruction methods, including traditional methods and learning-based methods.

**Traditional Implicit Methods** Most traditional implicit methods require the input points to be equipped with ori-

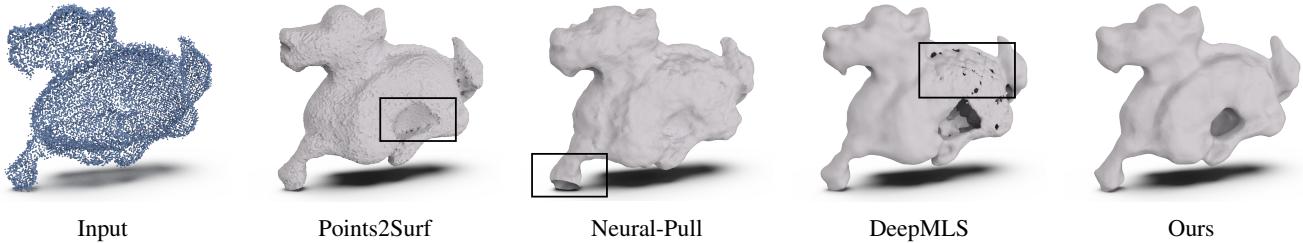


Figure 1: We present *Neural-IMLS*, a novel approach that learns reliable SDF from point clouds directly in a self-supervised fashion. Unlike other learning-based methods (Points2Surf (Erler et al. 2020), Neural-Pull (Baorui et al. 2021), and DeepMLS (Shi-Lin Liu et al. 2021)), it has the ability to reconstruct smooth and watertight surfaces with noisy input.

ent normals. The earliest implicit method (Hoppe et al. 1992) 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 with non-uniform point clouds since the closest point may be far from the query points. Radial basis functions (RBF) are a well-known method for data interpolation. Some methods (Carr et al. 2001; Dinh, Turk, and Slabaugh 2002) define fixed radial basis kernels at each point and then compute linear coefficients to combine all kernels to interpolate the results at the input points. To avoid the trivial solution, these methods need to get additional query points equipped with signed values generated by moving existing points along normals. At the same time how to select good additional points is not an easy problem. The Poisson methods (Kazhdan, Bolitho, and Hoppe 2006) and its variants (Kazhdan and Hoppe 2013; Kazhdan et al. 2020) seek an indicator function instead of SDF. This function outputs a binary value to indicate the sign of query points. While they are based on smooth priors, they can not produce geometrically correct surfaces with noisy point clouds. Another kind of method is implicit moving least methods. These methods (Shen, O’Brien, and Shewchuk 2004; Kolluri 2008; Öztireli, Guennebaud, and Gross 2009) construct different bases (e.g., polynomial) at each point sample and then get a global function by weighing them together. This category of methods is tough to reconstruct sharp features from point clouds.

**Learning Implicit Function with Data-Priors** Several learning-based methods have been proposed to learn implicit functions from a large dataset with ground-truth SDF or occupancy values recently. Early works tend to encode each shape to a latent code and then decode corresponding implicit functions to reconstruct the surface (Park et al. 2019; Mescheder et al. 2019; Chen and Zhang 2019). Since these shapes encode one shape globally, we called those global-based methods. These methods can not generalize the unseen shapes whose geometric and topological are different from training data. To address this problem, some local-based methods have been proposed. Points2surf (Erler et al. 2020) decomposes the problem into a global and local function that uses global features to indicate signs and local features to query distances. While this method has the ability to denoise, it can not produce smooth surfaces. Jiang et al. (Jiang et al. 2020) split shape to overlap local patches and

then reconstruct shape to improve the generality of the network. The convolutional operator allows the convolutional occupancy networks (Peng et al. 2020) to aggregate local and global information. Therefore, it has ability to reconstruct scenes not only single objects. Local-based methods have better performance than global-based methods since they can capture more geometric details by learned local priors. However, they are still time-consuming as the global-based methods since the numerous parameters in encoder modules.

**Learning Implicit Function from Raw Point clouds** It is challenging to learn implicit functions from raw point clouds directly without ground-truth SDF. Existed methods usually add more constraints to reduce the difficulty of learning signed fields. IGR (Gropp et al. 2020) introduced the geometric regularization term to formula this problem as an Eikonal partial differential equation solving problem. SAL (Atzmon and Lipman 2020) introduced a specially designed loss function for getting sign information from pre-computing unsigned distance fields. After that, SALD (Atzmon and Lipman 2021) leveraged an additional gradient constraint for SAL, making reconstructed results have more geometric details. Neural-Pull (Baorui et al. 2021) trains neural networks by pulling query points to their closest points on the surface using the predicted signed distance and the gradient of the query points. Unlike previous methods, Sitzmann et al. (Sitzmann et al. 2020) proposed a new activation function called Sine which can capture high-frequency information compared with ReLU. While such methods can generate nice reconstruction results with high-quality point clouds, they are weak in eliminating the influence of noisy points.

## Method

**Problem Statement** Given a point clouds  $P = \{p_i\}_{i \in I} \in \mathbb{R}^3$  without normals  $N = \{n_i\}_{i \in I} \in \mathbb{R}^3$ , we aim to learn an signed distance function  $f: \mathbb{R}^3 \rightarrow \mathbb{R}$  from it to reconstruct watertight surface  $S$ , who is the zero level-set of  $f$ :

$$S = \{f(\mathbf{q}) = 0\}, \quad (1)$$

where  $\mathbf{q} \in \mathbb{R}^3$  is an arbitrary query point.

Recently works (Park et al. 2019; Jiang et al. 2020; Erler et al. 2020) formula this problem as a regression problem, which train with ground-truth signed distance values

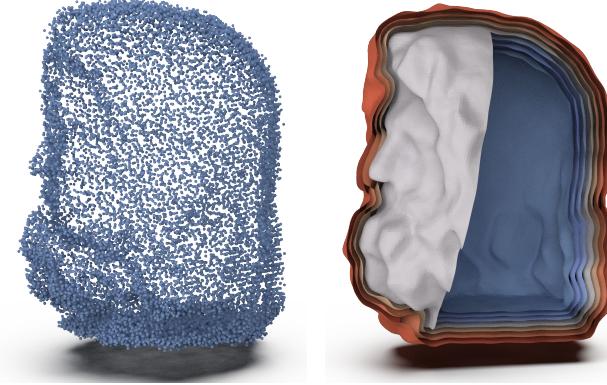


Figure 2: The level sets of  $f(\mathbf{q})$  (right) trained by our method. The warm color represents positive level sets, and the cold color represents negative level sets. The white color represents the IMLS surface. Our method has the ability to learn smooth SDF directly from noisy point clouds.

and predict signed distance values by decoding latent codes. In this way, the neural network needs much data and time for converging. Besides, since the encoder has many parameters, it is very time-consuming in inference procedure, especially with high-resolution output. Instead, our method learns  $f$  directly from raw point clouds in a self-supervised fashion by the supervision signal provided from the IMLS function.

**IMLS Surface** The IMLS surface (Kolluri 2008) is defined as the zero level set of the IMLS function. Specifically, this method constructs a polynomial basis for each point  $\mathbf{p}_i$  in input point clouds. The basis considering the projection distance between query points  $\mathbf{q}$  and tangent plane on each point  $\mathbf{p}_i$ , which described by the location of  $\mathbf{p}_i$  with corresponding normal  $\mathbf{n}_i$ . The final signed distance value of query point  $\mathbf{q}$  has been computed by the weighted average of all bases together. The formulation of the IMLS function has been shown below:

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

where  $\|\cdot\|$  is L2-norm,  $\theta(x)$  is a smooth weight function and we used Gaussian functions  $\theta(x) = \exp(-x^2/\sigma_s^2)$  here.  $\sigma_s$  is the kernel size of Gaussian function and we will talk about it later.

In (Kolluri 2008), the author has proved that the IMLS surface reconstructing from the IMLS function is a good approximation of the original surface and is geometrically and topologically correct under  $\epsilon$ -sampling condition. At the same time, the IMLS function is also a good approximation of the SDF of the original surface.

Since  $\theta$  is a fast-decreasing smooth weight function, the point  $\mathbf{p}_i$  who is far away from query point  $\mathbf{q}$  have almost no influence for query point  $\mathbf{q}$ . Thus we can only consider local neighbors of query point  $\mathbf{q}$ .

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

where  $\mathcal{P}$  is the local patch of point clouds  $\mathcal{P}$  who has points inside the ball centered at query point  $\mathbf{q}$  with radius  $r$ .

It is worth noting that since the IMLS surface is the approximation of the original surface, the points defined by the IMLS surface are not equal with input point clouds which makes this method can reconstruct smooth surfaces from noisy point clouds. We define a self-supervised formulation based on that to learn high-quality SDF directly from raw point clouds.

**Neural-IMLS** 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 a query point  $\mathbf{q}$ , the signed distance value  $f(\mathbf{q})$  is predicted by the network. Besides, the gradient  $\nabla f(\mathbf{q})$  at  $\mathbf{q}$  can be elegantly computed analytically with back-propagation. Since we only considering the direction of  $\nabla f(\mathbf{q})$ , we normalize that as unit vector  $\nabla f(\mathbf{q}) / \|\nabla f(\mathbf{q})\|$ . The same operation can be done for the points  $\mathbf{p}_i$  in points clouds. Therefore, we can compute the IMLS function by leveraging the gradient as the normal:

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

Here, we set the default number  $|\mathcal{P}|$  of  $\mathcal{P}$  is 200 and padding the patch if there are not enough points in  $\mathcal{P}$  ( $\mathcal{P} < 200$ ) while do random subsampling with over points ( $\mathcal{P} > 200$ ). The support radius  $\sigma_s$  is defined as  $\sigma_s = \sqrt{d_{\mathcal{P}} / |\mathcal{P}|}$  by the advice from (Huang et al. 2009), where  $d_{\mathcal{P}}$  is the length of the bounding box of the patch  $\mathcal{P}$ .

To supervise the network as IMLS function  $\mathbf{g}(\mathbf{q})$ , we leverage a square error between  $\mathbf{g}(\mathbf{q})$  and  $f(\mathbf{q})$ :

$$L = \frac{1}{|\mathcal{Q}|} \sum_{\mathbf{q}_j \in \mathcal{Q}} \|\mathbf{g}(\mathbf{q}_j) - f(\mathbf{q}_j)\|^2, \quad (5)$$

where  $|\mathcal{Q}|$  is the number of the set of query points  $\mathcal{Q}$ . We generate  $\mathcal{Q}$  by adopting the sampling category from Neural-Pull (Baorui et al. 2021).

Unfortunately, while this formulation can work well in simple shapes, it can not correctly reconstruct shapes with thin structures (see Figure 8). The thin structures are tough 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 gradient coverage. Specifically, we compare the L2-norm differences of the gradient between query point  $\mathbf{q}$  and its neighbors  $\mathbf{p}_i$ . The term is defined as

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

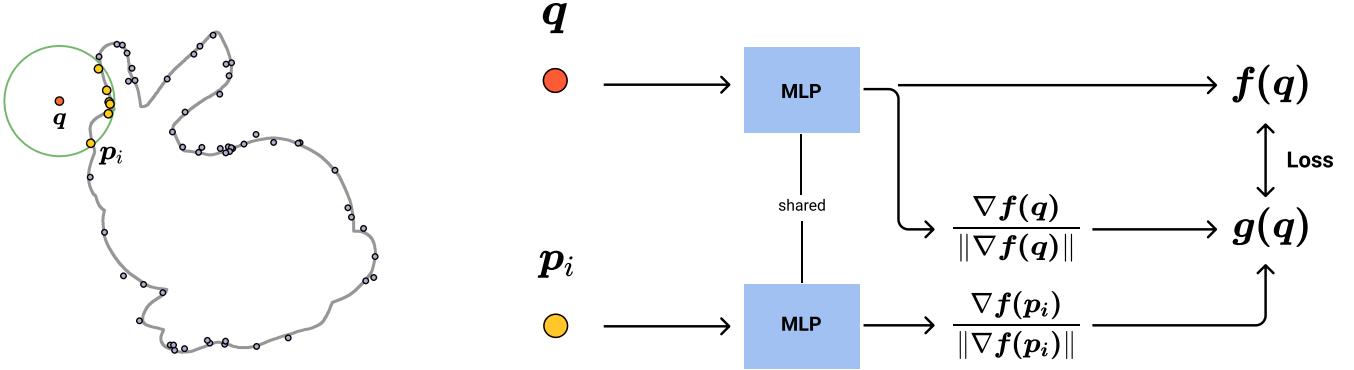


Figure 3: The self-supervised pipeline of our method. For an query point  $q$  (red), we search its nearest neighbors  $p_i$  (yellow) who are inside the ball (green) centered at query point  $q$  with radius  $r$  from input point clouds. The supervision signal is provided from the IMLS function  $g(q)$  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  $g(q)$ .

where the parameter  $\sigma_r$  scales the similarity of neighboring normals and we set that as a constant 0.03. Now we can denote both weight function together:

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

Based on that, our IMLS function becomes:

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

We still use the loss function described by the equation 5 to constraint neural network  $f(\mathbf{q})$  as close as IMLS function  $g(\mathbf{q})$ . When the couple of SDFs coincide, the learned function converges to a signed distance function.

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	<b>0.007</b>	0	0.550
Ours	<b>0.927</b>	<b>0.007</b>	<b>0</b>	<b>0</b>

Table 1: Quantitative evaluation of difficulty method on the FAMOUS no-noise dataset.

## Experiments

**Training Details** We set the number of input point clouds  $I = 2 \times 10^4$  and padding the  $I$  same with the strategy for patch  $\mathcal{P}$ . We sample 25 query points for each point in  $\mathcal{P}$  which means the numbers of query points  $\mathcal{Q}$  is  $5 \times 10^5$ . The default patch radius  $r$  is set as  $0.05d_{\mathcal{P}}$  where  $d_{\mathcal{P}}$  is the bounding box length of point clouds  $\mathcal{P}$ .

**Network configuration** We overfit the neural network on a single point cloud. We use the MLP architecture to represent the shape, including 7 hidden layers and each hidden layer has 512 hidden units with ReLU (Nair and Hinton 2010) activation function. The network predicts the signed distance value for input query points. Furthermore, we use

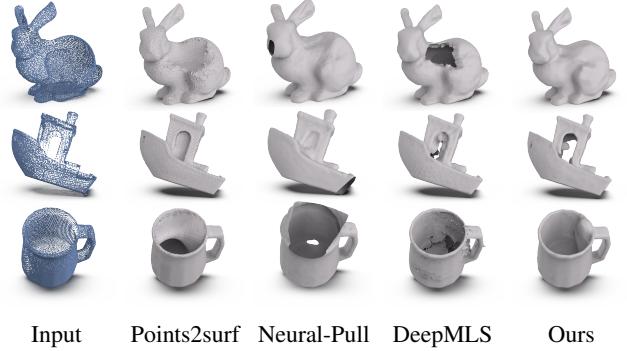


Figure 4: Qualitative results of 3D point clouds reconstruction on the FAMOUS no-noise dataset.

the initial method proposed by (Atzmon and Lipman 2020) called geometric network initialization to approximate the initial network as the signed distance function of a sphere.

Method	NC $\uparrow$	L2-CD $\downarrow$	Non-MF $\downarrow$	Non-WT $\downarrow$
P2S	<b>0.920</b>	0.087	0.980	0.980
NP	0.838	0.357	0.060	0.100
DM	0.915	<b>0.019</b>	0.020	0.650
Ours	0.904	0.033	<b>0</b>	<b>0</b>

Table 2: Quantitative evaluation of difficulty method on the ABC no-noise dataset.

**Hyperparameters** We implement Neural-IMLS by PyTorch deep learning framework (Paszke et al. 2019) and train on a Linux server with an Intel Xeon Silver 4210 Processor and a GeForce RTX 2080Ti GPU. We use the Adam optimizer (Kingma and Ba 2015) with a learning rate  $1 \times 10^{-5}$  and batch size of 100. For each point cloud in our experiment, our method can converge in 300 epochs.

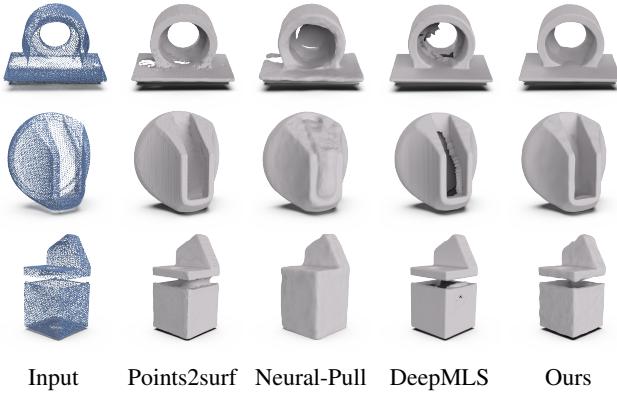


Figure 5: Qualitative results of 3D point clouds reconstruction on ABC no-noise dataset.

**Dataset** We evaluate our method on the subset of ABC dataset (Koch et al. 2019) released by Points2Surf (Erler et al. 2020). The ABC dataset contains various CAD shapes. Besides, we also use the FAMOUS dataset, which was also released by Points2Surf (Erler et al. 2020). This dataset has 22 famous shapes in computer graphics. We used variants of the above two datasets to demonstrate the performance of our method.

**Metric** After overfitting for single point clouds, we use the Marching Cubes (Lewiner et al. 2003) algorithm to extract 0-isosurface from  $128^3$  grid as the reconstruction surface. To comprehensively evaluate the reconstruction quality of our results, we introduced Normal Consistency (NC) and L2-Chamfer distance (L2-CD, we multiplied by 100) to compute reconstruction errors between reconstruction result and the ground truth with  $1 \times 10^5$  random samples. Besides, we also report the ratio of non-manifold (None-MF) shapes and non-watertight (Non-WT) shapes on each dataset. The detailed metrics formulation is provided in the supplemental material.

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

**Baseline** We compared three methods, including Points2Surf (P2S) (Erler et al. 2020), Neural-Pull (NP) (Baorui et al. 2021), and DeepMLS (DM) (Shi-Lin Liu et al. 2021). Points2Surf is a local-based method that infers distance information by the local patch while infers sign information by the global points. Neural-Pull learns SDF from raw point clouds by pulling points to the surface defined by the input scans. DeepMLS is also based on the IMLS function which outputs MLS

points, normals and radius explicitly from the network with voxels input. We reproduce the results of Points2Surf and DeepMLS by pre-trained model released by the official. Besides, we reproduce the results of Neural-Pull using the official code.

### Surface reconstruction from point clouds

We show the visual results under FAMOUS no-noise and ABC no-noise datasets in Figure 4 and Figure 5. Compared with other methods, our method can handle incomplete point clouds better and more consistently with topologically input point clouds. The results of other methods have many holes, which means they can not generate watertight shapes. Moreover, our results can reconstruct smoother surfaces. Furthermore, Table 1, Table 2 show the quantitative results under the above two datasets. We can observe that other methods generate non-watertight shapes, even non-manifold surfaces under two datasets, especially Points2Surf (All results under FAMOUS no-noise are non-manifold). Points2Surf proposed a sign propagation mechanism to reduce the query times of isosurface extraction. In this way, there are much error signed points leading to non-manifold and non-watertight surfaces since this mechanism is sensitive to the distribution of input scans. To improve efficiency at the 0-isosurface extraction stage, DeepMLS modified the Marching Cubes while the modified Marching Cubes do not have topological guarantees and generate non-manifold surfaces. There are topologically inaccurate surfaces in the reconstructed mesh of Neural-Pull since it is tough to learn high-quality SDF by simple pulling for topologically complex shapes (e.g., genus not equal 0).

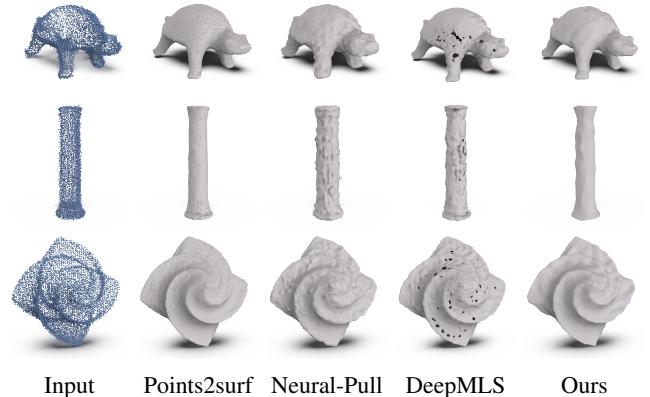


Figure 6: Qualitative results of 3D point clouds reconstruction on the FAMOUS med-noise dataset.

**The Effect of Noise** To report the denoising ability of our method, we experiment under the ABC-varnoise with varying noise strength noisy shapes and the FAMOUS-mednoise datasets with a constant strength. We tune the parameter  $r$  to  $0.1d_P$  for getting the smooth surfaces since some shapes are very noisy. The results are reported in Table 3 and Table 4. Our methods get the best performance under the FAMOUS med-noise dataset. At the same time, our results are

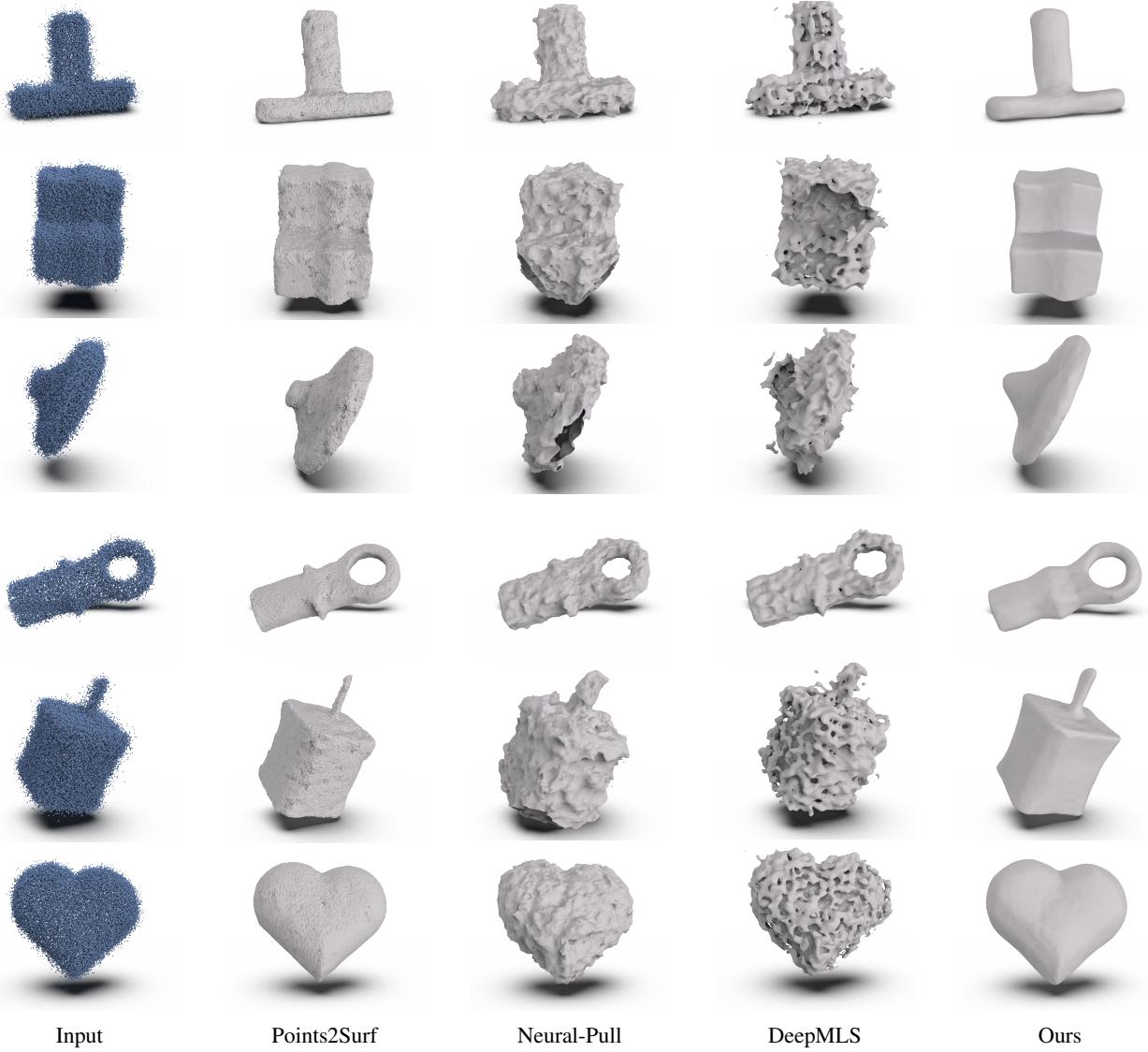


Figure 7: Qualitative results of 3D point clouds reconstruction on ABC var-noise dataset.

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 4: Quantitative evaluation of difficulty method on the ABC var-noise dataset.

still smooth and watertight compared with other methods (See Figure 6). Points2Surf handles noisy input well, but it has the same issues that their results contain many non-manifold surfaces. DeepMLS can not reconstruct convincing surfaces from very noisy scans (See the 4th columns of Figure 7) and lead non-watertight shapes in the reconstructed mesh. Neural-Pull cannot eliminate the influence of noisy points since it assumes all points are at 0-isosurface, including noisy ones. For ABC var-noise dataset, a part of our results loss geometric details since the nearest search radius  $r = 0.1d_P$  is too large for shapes with low noise or noise-free. We can tune the parameter  $r$  back to  $0.05d_P$  or other suitable configuration for those shapes to improve the accuracy of reconstruction.

**The Effect of Density** We further explore the effect of the density of input point clouds. We evaluate methods under

FAMOUS dense dataset and FAMOUS sparse dataset. The above two datasets have denser density or sparser density respectively and have constant noise length simultaneously. We report our results in Table 5 and Table 6. Obviously, the high-density point clouds yield more accurate reconstruction results while sparse point clouds leading inaccurate results. Our results achieve best under FAMOUS dense and FAMOUS sparse datasets.

Method	NC $\uparrow$	L2-CD $\downarrow$	Non-MF $\downarrow$	Non-WT $\downarrow$
P2S	<b>0.919</b>	<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>0.919</b>	0.006	<b>0</b>	<b>0</b>

Table 5: Quantitative evaluation of difficulty method on the FAMOUS dense 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	0	0.78
Ours	<b>0.898</b>	<b>0.021</b>	<b>0</b>	<b>0</b>

Table 6: Quantitative evaluation of difficulty method on the FAMOUS sparse dataset.

## Ablation Study

**Ablation study for loss terms** We conduct ablation studies for loss terms under the FAMOUS no-noise dataset. By dropping the gradient term  $\psi$  and the distance term  $\theta$  respectively, we denote them as w/o gradient and w/o distance. We report results in Table 7 and the statistics show the accuracy of the reconstruction is degrade. The gradient term has the decisive influence on reconstruction, while it can not capture geometric details without distance term. It is tough to reconstruct convincing surfaces with distance terms since they can not get reliable normals for pseudo-labels construction. Furthermore, it can not learn accurate thin structures (See Figure 8).

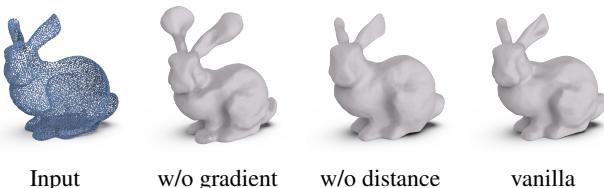


Figure 8: Visual results of Ablation studies for loss terms. It is challenging to learn accurate thin structures (the ears of bunny) without gradient term and some details are losted with only gradient term.

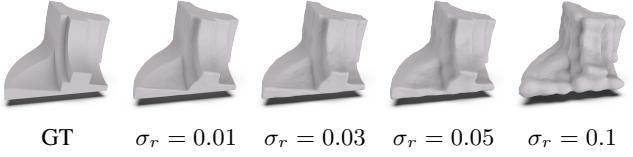


Figure 9: Visual results of the effect of the parameter  $\sigma_r$  (GT means ground-truth). Smaller values leading sharper results.

Metric	w/o distance	w/o gradient	vanilla
L2-CD	0.159	0.011	<b>0.007</b>

Table 7: Qualitative results of ablation studies for loss terms.

**The Effect of parameter  $\sigma_r$**  By changing the value of parameter  $\sigma_r$ , we found that our method can keep sharper features with smaller  $\sigma_r$  (see Figure 9), i.e., the parameter  $\sigma_r$  controls the intensity of smooth. We set it as constant 0.3 to balance the ability to keep sharp features and denoise simultaneously.

## Conclusions

In this paper, we present Neural-IMLS to learn noise-resistant SDF from raw point clouds in a self-supervised fashion. Utilizing the pseudo-label constructed by the IMLS function, our method gets high-quality SDF directly from point clouds without the ground-truth signed distance values and yield watertight and smooth surfaces. The experiments under widely used benchmarks show the power of our method.

Same with traditional implicit least-square reconstruction methods, the main limitation of our method is that it is challenging to reconstruct sharp features well. In the future, we would like to explore new construction of the pseudo-label to address this issue.

## References

- Atzmon, M.; and Lipman, Y. 2020. SAL: Sign Agnostic Learning of Shapes From Raw Data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Atzmon, M.; and Lipman, Y. 2021. SALD: Sign Agnostic Learning with Derivatives. In *9th International Conference on Learning Representations, ICLR 2021*.
- Baorui, M.; Zhizhong, H.; Yu-shen, L.; and Matthias, Z. 2021. Neural-Pull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces. In *International Conference on Machine Learning (ICML)*.
- Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; and Taubin, G. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4): 349–359.
- Carr, J. C.; Beatson, R. K.; Cherrie, J. B.; Mitchell, T. J.; Fright, W. R.; McCallum, B. C.; and Evans, T. R. 2001. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 67–76.

- Chen, Z.; and Zhang, H. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5939–5948.
- Dinh, H. Q.; Turk, G.; and Slabaugh, G. 2002. Reconstructing surfaces by volumetric regularization using radial basis functions. *IEEE transactions on pattern analysis and machine intelligence*, 24(10): 1358–1371.
- Edelsbrunner, H.; and Mücke, E. P. 1994. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1): 43–72.
- Erler, P.; Guerrero, P.; Ohrhallinger, S.; Mitra, N. J.; and Wimmer, M. 2020. Points2surf learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, 108–124. Springer.
- Gropp, A.; Yariv, L.; Haim, N.; Atzmon, M.; and Lipman, Y. 2020. Implicit Geometric Regularization for Learning Shapes. In *Proceedings of Machine Learning and Systems 2020*, 3569–3579.
- Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; and Stuetzle, W. 1992. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, 71–78.
- Huang, H.; Li, D.; Zhang, H.; Ascher, U.; and Cohen-Or, D. 2009. Consolidation of unorganized point clouds for surface reconstruction. *ACM transactions on graphics (TOG)*, 28(5): 1–7.
- Jiang, C.; Sud, A.; Makadia, A.; Huang, J.; Nießner, M.; Funkhouser, T.; et al. 2020. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6001–6010.
- Kazhdan, M.; Bolitho, M.; and Hoppe, H. 2006. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7.
- Kazhdan, M.; Chuang, M.; Rusinkiewicz, S.; and Hoppe, H. 2020. Poisson surface reconstruction with envelope constraints. In *Computer Graphics Forum*, volume 39, 173–182. Wiley Online Library.
- Kazhdan, M.; and Hoppe, H. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3): 1–13.
- Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.
- Koch, S.; Matveev, A.; Jiang, Z.; Williams, F.; Artemov, A.; Burnaev, E.; Alexa, M.; Zorin, D.; and Panizzo, D. 2019. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9601–9611.
- Kolluri, R. 2008. Provably good moving least squares. *ACM Transactions on Algorithms (TALG)*, 4(2): 1–25.
- Lewiner, T.; Lopes, H.; Vieira, A. W.; and Tavares, G. 2003. Efficient implementation of marching cubes’ cases with topological guarantees. *Journal of graphics tools*, 8(2): 1–15.
- Lorensen, W. E.; and Cline, H. E. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics*, 21(4): 163–169.
- Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4460–4470.
- Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Icm*.
- Öztireli, A. C.; Guennebaud, G.; and Gross, M. 2009. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer graphics forum*, volume 28, 493–501. Wiley Online Library.
- Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deep sdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 165–174.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037.
- Peng, S.; Niemeyer, M.; Mescheder, L.; Pollefeys, M.; and Geiger, A. 2020. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 523–540. Springer.
- Shen, C.; O’Brien, J. F.; and Shewchuk, J. R. 2004. Interpolating and approximating implicit surfaces from polygon soup. In *ACM SIGGRAPH 2004 Papers*, 896–904. Association for Computing Machinery.
- Shi-Lin Liu, H.-X. G.; Pan, H.; Wang, P.; Tong, X.; and Liu, Y. 2021. Deep Implicit Moving Least-Squares Functions for 3D Reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Sitzmann, V.; Martel, J. N.; Bergman, A. W.; Lindell, D. B.; and Wetzstein, G. 2020. Implicit Neural Representations with Periodic Activation Functions. In *Proc. NeurIPS*.
- You, C. C.; Lim, S. P.; Lim, S. C.; Tan, J. S.; Lee, C. K.; and Khaw, Y. M. J. 2020. A Survey on Surface Reconstruction Techniques for Structured and Unstructured Data. In *2020 IEEE Conference on Open Systems (ICOS)*, 37–42.