Neural-IMLS: Self-Supervised Implicit Moving Least-Squares Network for Surface Reconstruction

Zixiong Wang ⊙, Pengfei Wang ⊙, Peng-Shuai Wang ⊙, Qiujie Dong ⊙, Junjie Gao ⊙, Shuangmin Chen ⊙, Shiqing Xin ⊙, Changhe Tu ⊙, and Wenping Wang ⊙, Fellow, IEEE

Abstract—Surface reconstruction is a challenging task when input point clouds, especially real scans, are noisy and lack normals. Observing that the Multilayer Perceptron (MLP) and the implicit moving least-square function (IMLS) provide a dual representation of the underlying surface, we introduce Neural-IMLS, a novel approach that directly learns a noise-resistant signed distance function (SDF) from unoriented raw point clouds in a self-supervised manner. In particular, IMLS regularizes MLP by providing estimated SDFs near the surface and helps enhance its ability to represent geometric details and sharp features, while MLP approximates IMLS by providing estimated normals. We prove that at convergence, our neural network produces a faithful SDF whose zero-level set approximates the underlying surface due to the mutual learning mechanism between the MLP and the IMLS. Extensive experiments on various benchmarks, including synthetic and real scans, show that Neural-IMLS can reconstruct faithful shapes even with noise and missing parts. The source code can be found at https://github.com/bearprin/Neural-IMLS.

Index Terms—Implicit moving least squares, self-supervised neural network, surface reconstruction, implicit neural representations.

I. INTRODUCTION

SURFACE reconstruction from an unstructured point cloud remains an active research topic since it is essential in many downstream computer vision and graphics applications, such as games, rendering, and animation. The target surface is assumed to be 2-manifold and watertight in most scenarios. Because of this, the signed distance function (SDF) \( f \) serves as a popular representation of the surface reconstruction problem. Given a point cloud \( P \), the SDF \( f \) to be reconstructed has to satisfy two basic conditions including (1) all the points are nearly sitting on the zero-value level-set surface, i.e., \( f(p_i) ≈ 0 \) for each point \( p_i \) in \( P \), and (2) the gradients of \( f \) are approximated as unit vector almost everywhere.

Due to the fact that the raw data inevitably contains noise and sometimes lacks normals, it is notoriously hard yet fascinating to reconstruct a high-fidelity surface in the presence of severe noise. The difficulties are two-fold. On the one hand, one has to sacrifice the first condition to deal with noise while utilizing the positional clues of the noisy points as much as possible, which necessitates a careful balance. On the other hand, it is non-trivial to recover the real surface variations, especially when severe noise exists. The theme of this paper is to study the surface reconstruction problem assuming that the input point cloud is noisy and lacks normals, without any supervising data.

Most traditional implicit approaches fit an implicit function by leveraging oriented normals [1], [2], [3], [4], followed by extracting the zero-level surface (e.g., Marching Cubes [5]). It’s observed that the oriented normals have a decisive influence on inferring the variation of the final reconstructed surface. They cannot work except that the normals are given. To our knowledge, there are several traditional surface reconstruction approaches for handling unoriented point clouds, such as VIPSS [6] and iPSR [7], but these approaches lack robust noise resistance.

Many supervised learning-based algorithms have been proposed for implicit surface reconstruction in recent years [8]. Most of them learn an implicit function by fitting data samples supervised with the ground-truth SDFs or Occupancy Fields [9], [10], [11], [12], [13], [14], [15]. Despite the fact that most existing methods focus on leveraging small receptive fields to incorporate local data priors [12], [13], [15], [16] for improved generalization across various categories, the performance of learning-based methods may deteriorate if the test point cloud distribution is not present in the training data [17]. There are some optimization-based methods [18], [19], [20], [21], [22], [23], [24], [25], [26] that do not require knowledge of ground-truth field values. When severe noise exists, however, they lack

Manuscript received 8 November 2022; revised 1 June 2023; accepted 1 June 2023. Date of publication 8 June 2023; date of current version 1 July 2024. This work was supported in part by the National Key R&D Program of China under Grant 2022YFB3303200, in part by the National Natural Science Foundation of China under Grants 62002190, 62272277, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2020MF036. Recommended for acceptance by Amir Vaxman. (Corresponding author: Shiqing Xin.)

Zixiong Wang, Pengfei Wang, Qiujie Dong, Junjie Gao, Shiqing Xin, and Changhe Tu are with the School of Computer Science and Technology, Shandong University, Jinan 251600, China (e-mail: zixiong_wang@outlook.com; 8144756@qq.com; qiujie.jay.dong@gmail.com; gjjsdnu@163.com; xinshiqing@sdu.edu.cn; chtu@sdu.edu.cn).

Peng-Shuai Wang is with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100871, China (e-mail: wangps@hotmail.com).

Shuangmin Chen is with the School of Information and Technology, Qingdao University of Science and Technology, Shandong 266101, China (e-mail: csmq@qq.com).

Wenping Wang is with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: wenping@tamu.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2023.3284233, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2023.3284233
Taking the unoriented and noisy points (the white points) as the input, \textit{Neural-IMLS} learns a noise-resistant signed distance function (SDF) whose zero level set (the black lines) reports the reconstructed surface.

Our network can directly predict a high-fidelity surface (right) from a noisy input point cloud (left). We also visualize the inward and outward offset surfaces by extracting the iso-value level sets at different values.

We have made a couple of interesting observations. Firstly, the Implicit Moving Least-Square (IMLS) function \cite{1, 2} can accurately reconstruct the underlying surface by estimating SDFs near the surface if the point normals are accurate. Secondly, MLPs fit the SDF of the input cloud on a global basis and can robustly provide the normal vector field through automatic differentiation. Based on these observations, we propose \textit{Neural-IMLS}, a novel method for learning the SDF directly from poor-quality point clouds (noisy, irregular, and without normals; see Figs. 1 and 2). Our method learns the underlying SDF 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 the MLP. The MLP and the IMLS are taken as a dual representation of the underlying surface since they complement each other: IMLS regularizes MLP by providing estimated SDFs near the surface and helps enhance its ability to represent geometric details and sharp features, while MLP regularizes IMLS by providing estimated normals. Through this process, the two SDFs, provided by the MLP and IMLS respectively, mutually enhance each other during neural network optimization. We have also developed a novel loss function that considers both the difference between the two SDFs and the spatial coherence of their gradients. The rationale of our approach is based on the principle of mutual learning \cite{27} (see Fig. 3). After optimization convergence, the neural network is capable of accurately encoding the underlying shape through the learned SDF.

We have conducted extensive experiments on a wide range of shapes, including synthetic scans and real scans. The experimental results (see Section IV) demonstrate that our approach can learn more accurate SDFs than other leading optimization-based techniques, particularly on noisy scans. The main contributions of this paper include:

- A self-supervised neural network for reconstructing a 2-manifold surface from a noisy and unoriented input point cloud. The key insight is that the MLP and the IMLS can be taken as a dual representation of the underlying surface. Our network is provably convergent.
- A novel loss that considers not only the spatial coherence of the gradients but also the difference between a pair of distance fields, one reported by the IMLS and the other reported by the neural network.
- Our approach combines the noise-resistance of IMLS with the global expressiveness of MLP. We have conducted extensive comparative experiments and ablation studies to validate the effectiveness of our approach on various benchmarks, including both synthetic and real-scan data. The code and data are available at https://github.com/bearprin/Neural-IMLS.

II. RELATED WORK

Surface reconstruction is a fundamental research topic in computer graphics and computer vision. Numerous reconstruction algorithms \cite{28} have been proposed in the last three decades. In this section, we mainly review the implicit surface reconstruction methods, including traditional and learning-based methods.

A. Traditional Implicit Methods

Most traditional implicit methods require the input points to be equipped with orienting normals. Some methods \cite{6, 29, 30} leverage the radial basis function (RBF) to represent the underlying signed distance function (SDF) as a weighted combination of a group of radial basis kernels. Poisson reconstruction \cite{3} and its variants \cite{4, 31, 32} seek a continuous occupancy field that can be formulated as the solution to Poisson’s equation. Additionally, Implicit Moving Least-Square (IMLS) methods \cite{1, 2, 33, 34} approximate the underlying SDF by blending local smooth planes.
In general, traditional implicit reconstruction approaches require orienting normals as input, and the quality of the reconstruction is heavily dependent on the reliability of these normal vectors. However, estimating orienting normal vectors can be challenging, particularly for point clouds with significant defects such as noise, missing regions, and thin plates. It should be noted that while there are a few traditional algorithms [7] that can operate without normals, they lack noise resistance. In contrast, our method is capable of directly reconstructing reliable results from noisy input without normals.

B. Learning Implicit Function With Ground Truth Supervision

Recently, several learning-based methods have been proposed to learn implicit functions from a large dataset with ground-truth SDFs or occupancy values [8]. Early works tend to encode each shape into a fixed-length latent code and then recover the underlying surface by a decoding operation [9], [10]. While these methods can encode the overall shape for a group of similar models, they cannot deal with unseen shapes whose geometric and topological structures differ greatly from the training data. Therefore, some local feature-based methods have been proposed to address this issue. LIG [12] splits the input point cloud into point patches and learns the patch-wise geometric features across various shapes. IF-Net [35] first voxelized the input point clouds and then leveraged 3D CNN to extract the multi-scale features seeking for high-quality reconstruction for articulated humans. Peng et al. [11] proposed a convolutional operator for aggregating local and global information. In the decoding phase, this method leverages grid features with 3D-U-Net to interpolate features for query points. Points2surf [13] uses global features to predict signs, and local features to predict distances, respectively. It leverages a mechanism called sign propagation to reduce the computational cost of the inference phase. However, the sign propagation yields inaccurate results when the input point cloud does not have a high-quality distribution (as pointed out in [36]), causing unwanted holes in the reconstructed mesh. By combining the benefits of the implicit approaches with the point set methods, DeepMLS [14] can generate a set of MLS projection points that help predict the implicit approximation surface. Unfortunately, DeepMLS cannot learn reliable parameters (e.g., radius) that facilitate the computation of IMLS, especially for sparse and noisy point clouds. It is likely to produce unwanted ghost geometries. POCO [15] proposes a kind of attention mechanism convolution to compute latent vectors at each input point, which is scalable to scenes of arbitrary size. Different from above, NDC [37] directly leverages the network to predict cell edge intersections in favor of mesh extraction from the unoriented point cloud to enable reconstruction. However, NDC tends to yield unwanted holes in the output, and is tough to fill it even with the post-processing to repair. Recently, some methods [38], [39] are proposed to partition and encode the given point cloud by an octree, and then decode the field from the feature code of octree nodes. The methods mentioned above, whether global or local, need supervision with the ground-truth SDF or occupancy function. With access to ground truth SDFs, learning-based methods benefit from perfect scalar fields with consistent gradients and can be generalized to various categories. However, they struggle to generalize learned priors to unseen cases or point distributions not present in training sets with large geometric variations [17]. In contrast, our method optimizes each tested shape individually and is capable of better fitting test shapes through optimization.

C. Learning Implicit Function From Raw Data

There are some approaches aiming at learning an implicit function from raw point clouds without knowledge of ground truth SDF or occupancy. These methods require optimizing one network per object with additional time for each tested shape. They generally introduce some principles to constrain the predicted implicit function (e.g., gradients being a unit vector). SAL/SALD [19], [21] performs sign agnostic regression to get a signed version from the unsigned distance function. IGR [18] introduces a geometric regularization term to encourage the predicted implicit function to have unit gradients, which can work with or without normals. Ma et al. [22] trained a neural network to predict the signed distance as well as the gradients so that a query point can be pulled onto the underlying surface. Next, they also proposed PredictableContextPrior [24] to combine local context prior and predictive context prior...
by an additional fine-tuning mechanism. SAP [40] introduces a differentiable version of Poisson Surface Reconstruction to repeatedly minimize the Chamfer distance between an explicit mesh and the input point cloud. Lipman [41] observed that there is a connection between the occupancy field and the SDF, and thus introduced a loss to learn a density function whose log transform approaches the SDF, which is based on the theory of phase transitions. DiGS [23] considers a kind of soft constraint about the divergence of the gradients of the distance field in the loss function and can help produce more reliable results in situations where normals are not available. In summary, most existing methods struggle to produce high-quality reconstructed results in the presence of severe noise, as it is challenging to balance the need to respect positional clues while eliminating noise. In contrast, our method utilizes mutual regularization between IMLS and MLP to effectively suppress the influence of noise and produce a high-fidelity surface.

III. METHOD

First, we will state problems and review the definition of the IMLS surface in Sections III-A and III-B, respectively. We then elaborate on the mutual-regularization mechanism of our method in Section III-D.

A. Problem Statement and Motivation

Given an input point cloud \( P = \{ p_i \}_{i \in I} \subset \mathbb{R}^3 \), we aim to learn a signed distance function \( f: \mathbb{R}^3 \rightarrow \mathbb{R} \), from which we can reconstruct a 2-manifold surface \( S \) by extracting the zero-value level set, i.e.,

\[
S = \{ q \mid f(q) = 0, q \in \mathbb{R}^3 \}.
\]

We consider the reconstruction problem from a different perspective. Denote \( \mathcal{R} \) as a surface reconstruction solver, which takes the points \( P = \{ p_i \}_{i \in I} \) and the normals \( N = \{ n_i \}_{i \in I} \) as the input and is able to reconstruct the “ideal” surface \( S \) when \( P \) is dense enough and \( N \) is clean. However, in real scenarios, the input point cloud is noisy and unoriented. We consider the following iterative process:

\[
(N^{(0)}; P) \xrightarrow{\mathcal{R}} S^{(0)} \rightarrow (N^{(1)}; P) \xrightarrow{\mathcal{R}} S^{(1)} \rightarrow \ldots \rightarrow (N^{(\infty)}; P) \xrightarrow{\mathcal{R}} S^{(\infty)},
\]

where \( N^{(0)} \) denotes the initial orienting normals, \( S^{(i)} \) denotes the underlying surface determined by applying the reconstruction solver \( \mathcal{R} \) on the pair \((N^{(i)}, P)\), and \( S^{(i)} \) can help update the setting of normals in the iteration. We propose utilizing IMLS to drive the evolution of the implicit representation of \( S^{(i)} \). IMLS is robust to noise in the point cloud and can accurately reconstruct geometric details, especially in regions near the surface. We hope that both the normals and the underlying surface can progressively change toward the “ideal” configuration during mutual updates between the IMLS and the neural network.

B. IMLS Surface

Each point \( p_i \in P \) contributes to the query point \( q \) a signed distance \( \langle q - p_i, n_i \rangle \), where \( n_i \) is the unit normal vector at \( p_i \). The final distance at \( q \) is defined as a weighted average \( \theta \) (1):

\[
f_{\text{IMLS}}^{P}(q; P, N) = \sum_{p_i \in P} \frac{\theta(||q - p_i||) \cdot \langle q - p_i, n_i \rangle}{\sum_{p_i \in P} \theta(||q - p_i||)},
\]

where \( \theta(||q - p_i||) \) denotes the weighting scheme and generally gives the nearer points in \( P \) a bigger influence. By default, \( \theta \) is set as the Gaussian function \( \theta(x) = \exp(-x^2/\sigma_{\text{IMLS}}^2) \), where \( \sigma_{\text{IMLS}} \) is the support radius and theoretically depends on the local feature size (LFS). In practice, we set it empirically (check Section IV-A for the details). The IMLS has various forms, among which the IMLS scheme is the simplest case of (3).

Since we hope that the zero-value level set of \( f_{\text{IMLS}}^{P} \) defines the target surface, \( f_{\text{IMLS}}^{P} \) should report an as-accurate-as-possible distance value for points nearby the underlying surface. To be more specific, we constrain the query point \( q \) to be lying in the thin shell of a width \( 2r \). Let \( B_r(q) \) be the open ball centered at \( q \). By ignoring the contribution of those points outside \( B_r(q) \) (note that \( \theta \) decreases with the increasing distance between \( q \) and \( p_i \)), (3) can be reduced to:

\[
f_{\text{IMLS}}^{P}(q; B_r(q), N) = \frac{\sum_{p_i \in B_r(q)} \theta(||q - p_i||) \cdot \langle q - p_i, n_i \rangle}{\sum_{p_i \in B_r(q)} \theta(||q - p_i||)}.
\]

(4) has two aspects of benefits. First, it requires less computational overhead than (3). Second, it encourages \( f_{\text{IMLS}}^{P} \) to be estimated based on local information.

C. Insight

As pointed out in [1], the IMLS has a noise-resistance ability. It can predict a noise-resistant distance field if the normals are available and faithful. The predicted results are a good approximation to the signed distance function of the original surface. Inspired by this, we use the neural network \( f_{\text{MLP}}^{P} \) to encode the implicit surface while allowing the IMLS to help tune \( f_{\text{MLP}}^{P} \) such that its gradient is consistent with the SDF reported by the IMLS. During regularization, \( f_{\text{MLP}}^{P} \) learns about the distance values given by the IMLS, and the IMLS is promoted by the gradients of \( f_{\text{MLP}}^{P} \), which is the following iterative process:

\[
(N^{(0)}; P) \xrightarrow{\text{IMLS}} f_{\text{MLP}}^{P(0)} \rightarrow (N^{(1)}; P) \xrightarrow{\text{IMLS}} f_{\text{MLP}}^{P(1)} \rightarrow \ldots \rightarrow (N^{(\infty)}; P) \xrightarrow{\text{IMLS}} f_{\text{MLP}}^{\infty}.
\]

In this way, we define a smart mechanism for tuning the neural network \( f_{\text{MLP}}^{P} \) to the “ideal” configuration using the IMLS.

Remark: One could use the IMLS purely to define a similar iterative process by using the gradients of the IMLS distance field to update the normals of the point cloud. However, this optimization is easily trapped by local minima where the normals are far from desired. Given that MLP has global expressiveness and naturally supports gradient computation, our intention is to combine IMLS and MLP to predict a more faithful SDF. We
conduct an ablation study on the benefits of our method over a pure IMLS or MLP scheme in Section IV-D.

D. Neural-IMLS

We propose to use the IMLS to regularize the distance values reported by the MLP while using the MLP to regularize the normals of the data points for running the IMLS. We hope that the neural network can output faithful SDF whose zero-level set well approximates the underlying surface, benefiting from the mutual learning mechanism.

Naïve Loss: Let \( Q \) be a set of pre-computed query points around \( P \), \( f_{\text{MLP}} \) and \( f_{\text{IMLS}} \) be the two SDFs obtained by the MLP and the IMLS. To promote the IMLS by the learned gradients from the neural network, we need to feed the gradients of \( f_{\text{IMLS}} \) into the IMLS iterative scheme:

\[
f_{\text{IMLS}}^r(q; B_r(q), \nabla f_{\text{MLP}}) = \sum_{p_i \in B_r(q)} \omega(q, p_i) \left( q - p_i \right) \frac{\nabla f_{\text{MLP}}(p_i)}{\| \nabla f_{\text{MLP}}(p_i) \|}, \tag{6}
\]

where \( \nabla f_{\text{MLP}} \) is computed based on the back-propagation of the neural network. In fact, \( \frac{\nabla f_{\text{MLP}}(p_i)}{\| \nabla f_{\text{MLP}}(p_i) \|} \) can be taken as the new normal vector at the point \( p_i \).

We thus define a loss by measuring the squared difference between \( f_{\text{MLP}} \) and \( f_{\text{IMLS}}^r \) as follows:

\[
\arg\min_{\vartheta} \frac{1}{|Q|} \sum_{q \in Q} \left\| f_{\text{IMLS}}^r(q_j) - f_{\text{MLP}}(q_j; \vartheta) \right\|^2, \tag{7}
\]

where \( |Q| \) is the number of query points in \( Q \) and \( \vartheta \) is the parameters of the network to optimize.

Spatial Coherence Loss: However, we observe that the reconstruction error near sharp features or thin geometric regions is relative high with this loss term. The reason is that the IMLS defined in (6) only considers the spatial similarity of neighboring points when doing the weighted average. To combat this issue, we introduce an influence function to measure the difference between \( \nabla f_{\text{IMLS}}(p) \) and \( \nabla f_{\text{MLP}}(q) \) for a pair of close points \( p \) and \( q \).

\[
\omega(q, p) = \theta (\| q - p \|) \psi(\nabla f_{\text{MLP}}(q), \nabla f_{\text{MLP}}(p)) = \exp \left( - \frac{\left\| \frac{\nabla f_{\text{MLP}}(q)}{\| \nabla f_{\text{MLP}}(q) \|} - \frac{\nabla f_{\text{MLP}}(p)}{\| \nabla f_{\text{MLP}}(p) \|} \right\|^2}{\sigma_{\text{coherence}}} \right), \tag{8}
\]

where the hyperparameter \( \sigma_{\text{coherence}} \) is used to tune how the spatial coherence depends on the similarity between gradients. We combine the two weighting schemes \( \theta \) and \( \psi \), and obtain a novel weighting scheme:

\[
\omega(q, p) = \theta (\| q - p \|) \psi(\nabla f_{\text{MLP}}(q), \nabla f_{\text{MLP}}(p)). \tag{9}
\]

By plugging the combined weighting scheme into (6), we have an improved version of IMLS:

\[
f_{\text{IMLS}}^{\omega}(q; B_r(q), \nabla f_{\text{MLP}}) = \sum_{p_i \in B_r(q)} \omega(q, p_i) \left( q - p_i \right) \frac{\nabla f_{\text{MLP}}(p_i)}{\| \nabla f_{\text{MLP}}(p_i) \|}, \tag{10}
\]

Like (7), we re-define a loss by measuring the squared difference between \( f_{\text{MLP}} \) and \( f_{\text{IMLS}}^{\omega} \). Our final loss and only using the loss function shown below:

\[
\arg\min_{\vartheta} \frac{1}{|Q|} \sum_{q \in Q} \left\| f_{\text{IMLS}}^{\omega}(q_j) - f_{\text{MLP}}(q_j; \vartheta) \right\|^2. \tag{11}
\]

By abuse of notation, in the following sections, we also use \( f_{\text{IMLS}} \) to represent the \( \omega \)-weighted IMLS iterative scheme.

Network Architecture and Training Details: We use the same architecture as IGR [18]: an MLP with 8 layers and 512 neurons in each layer (totally 1.8 M parameters) and a single skip connection concatenating an input point as the input of the fourth layer. We use the SoftPlus Activation Function (\( \beta = 1000 \)) and the geometric initialization (GNI) proposed by [19] to initialize the network to approximate the SDF of the unit sphere. In each training iteration, we sample a batch of query points \( q_j \in Q \) and further sample their neighboring points in \( B_r(q_j) \). We then forward the MLP for these points to get the predicted gradients with automatic differentiation and evaluate \( f_{\text{IMLS}}(q_j) \) according to (10). The parameters of the MLP are updated with the Adam optimizer according to the loss function defined in (11). \( f_{\text{IMLS}} \) serves as pseudo-labels and does not attach to the computational graph for gradients.

To help readers understand the iterative process, we visualize a 2D example in Fig. 4. See Appendix A, available online for more details about the training and sampling strategy.

E. Proof

In the following, we come to analyze the convergence.

Theorem 1: Suppose that the sample points are sufficiently dense, (e.g., satisfying the \( \epsilon \)-sampling condition [11]), our network \( f_{\text{MLP}} \) converges to a signed distance function near the surface.

Proof: Under the assumption that the sample points are sufficiently dense, we can choose a sufficiently small \( r \). For a query point \( q \) that is sufficiently close to the underlying surface, we suppose that \( p_j \) is the nearest point to \( q \). Recall that we have an assumption on the spatial coherence of \( \nabla f_{\text{MLP}} \)’s gradient. (10) can be thus rewritten as

\[
f_{\text{IMLS}}^{\omega}(q; B_r(q), \nabla f_{\text{MLP}}) = \frac{\sum_{p_i \in B_r(q)} \omega(q, p_i) \left( q - p_i \right) \frac{\nabla f_{\text{MLP}}(p_i)}{\| \nabla f_{\text{MLP}}(p_i) \|}}{\sum_{p_i \in B_r(q)} \omega(q, p_i) \| \nabla f_{\text{MLP}}(p_i) \|}.
\]

Since \( \sum_{p_i \in B_r(q)} \omega(q, p_i) \) approaches \( p_j \) if \( r \) is sufficiently small (all \( p_i \)’s in the neighborhood are much close to \( p_j \), the above...
We must point out that in real situations, the points \( f \), \( f \) fits the \( q \) \( B \) empirically following \( = \) as an invalid query point and ignore it. If \( \sigma \) is the diagonal length of the \( f \), \( f \) is set to 0.3 by default. If \( \sigma \) is empty, \( f \) on three datasets: to 50 points. We have a parameter \( \sigma \), \( \sigma \) are close to each other for points in \( \sigma \), \( \sigma \) distributes to coincide with \( \sigma \) and \( \sigma \) (middle) and ours (bottom) on how the learned SDF changes with the increasing number of iterations \( \sigma \) is measured w.r.t \( \sigma \) for a given query point \( \sigma \).

Therefore, we immediately have

\[
\mathbf{f}_{\text{IMLS}}(\sigma) = 0, \quad \| \mathbf{\nabla}_q \mathbf{f}_{\text{IMLS}} \| = \left\| \frac{\mathbf{\nabla} \mathbf{f}_{\text{MLP}}(p_j)}{\mathbf{\nabla} \mathbf{f}_{\text{MLP}}(p_j)} \right\| = 1,
\]

which implies that the IMLS surface tends to interpolate the given point cloud and approximates a signed distance field (i.e., the gradients have a unit norm). At the same time, the neural function \( \mathbf{f}_{\text{MLP}} \) tends to coincide with \( \mathbf{f}_{\text{IMLS}} \) with the decreasing of the loss.

Remark: We must point out that in real situations, the points are noisy and sparse, it requires a number of iterations to consistently transform \( \mathbf{f}_{\text{IMLS}} \) or \( \mathbf{f}_{\text{MLP}} \) into a real signed distance function.

It’s also worth noting that the difference between \( \mathbf{f}_{\text{IMLS}} \) and \( \mathbf{f}_{\text{MLP}} \) is measured w.r.t \( \sigma \), a pre-computed sample set nearby \( P \). Because of this, the values of \( \mathbf{f}_{\text{IMLS}} \) and \( \mathbf{f}_{\text{MLP}} \) are close to each other for points in \( \sigma \), rather than all points in \( \mathbb{R}^3 \). Their distance values are not necessarily identical to each other at points far from the input point set \( P \). An interesting fact is that \( \mathbf{f}_{\text{IMLS}} \) fits the point cloud mainly on local information, but \( \mathbf{f}_{\text{MLP}} \) fits the point cloud in a more global manner since its basis functions of \( \mathbf{f}_{\text{MLP}} \) do not vanish even when \( q \) is located infinitely far away. Therefore, when the input point cloud has missing parts, \( \mathbf{f}_{\text{MLP}} \) still can complete completing a reasonable shape, but \( \mathbf{f}_{\text{IMLS}} \) cannot.

IV. Experiments

A. Setup

Dataset: Surface reconstruction tasks can be divided into object-level and scene-level based on the size of the reconstruction target. For object-level surface reconstruction, we evaluate our method on five datasets: Surface Reconstruction Benchmark (SRB) \[46\], ShapeNet \[42\], ABC \[47\], FAMOUS \[13\], and Thingi10K \[48\]. We also test our method on a real-scan object-level benchmark from \[28\]. For scene-level surface reconstruction, we conduct experiments on the synthetic indoor scene dataset proposed by \[11\]. All output meshes are extracted with the improved Marching Cubes \[5\]. The resolution for running Marching Cubes is set to \( 512^3 \) on three datasets: SRB, real-scanned object-level benchmark, and synthetic indoor scene dataset. For other datasets, including FAMOUS, ABC, and Thingi10 K, we use a resolution of \( 256^3 \), following Points2Surf \[13\]. The resolution of Marching Cubes for the ShapeNet dataset is also set to \( 256^3 \).

Parameters: We have a parameter \( r \) to define the ball for finding the point subset in \( P \) for a given query point \( q \). The IMLS algorithm should use different local size parameters to accommodate different noise levels, with larger parameters used to resist larger noise levels. We set the default search radius \( r \) to be 0.01 \( d_P \), where \( d_P \) is the diagonal length of the bounding box of the input point cloud \( P \). If \( B_r(q) \) is empty, we deem \( q \) as an invalid query point and ignore it. If \( |B_r(q)| \) is less than 50, we augment it to 50 points with a padding technique. Otherwise, we downsample \( B_r(q) \) to 50 points. We set \( \sigma_{\text{IMLS}} = \sqrt{d_{B_{r}(q)}} / |B_r(q)| \) empirically following \[49\], where \( d_{B_{r}(q)} \) is the diagonal length of the bounding box of \( B_r(q) \). It’s worth noting that \( d_{B_{r}(q)} \) depends on the distribution of the points in \( B_r(q) \), and thus cannot be directly determined by \( r \). Additionally, \( \sigma_{\text{coherence}} \) is set to 0.3 by default.
We implement our method with PyTorch [50] on an RTX3090 and use Adam [51] as the optimizer with a learning rate of $5 \times 10^{-5}$ for training the network. We group query points in batches of size 100 and set the maximum epoch value of our trainer to 200. We sample $I = 25$ query points around each point $p_i$ following a Gaussian distribution, using the same sampling strategy as IGR [18]. The standard deviation of the Gaussian distribution is aligned with the distance from $p_i$ to the $k = 50$-th nearest neighbor.

**Metrics:** As various approaches use different evaluation metrics, we tune our experimental settings to ensure a fair comparison. For the SRB dataset, we follow DiGS [23] to evaluate related approaches with Chamfer Distance (CD) and Hausdorff Distance (HD), using $10^6$ test points. For ShapeNet and the synthetic indoor scene dataset, we use three indicators: Chamfer Distance (CD), normal consistency (NC), and F-Score (FS) with threshold values of 0.01 and 0.02 respectively, using $10^5$ test points. For the ABC, FAMOUS, and Thingi10K datasets, we make comparisons based on Chamfer Distance, using the same setting as Points2Surf [13] and $10^4$ test points. For the real-scan benchmark, we follow [28] to report Chamfer Distance (CD), F-Score (FS), normal consistency (NC), and Neural Feature Similarity (NFS). The threshold of the FS indicator is increased to 0.5 for real-scan data and the number of test points is set to $2 \times 10^5$.

**B. Object-Level Reconstruction**

1) **Surface Reconstruction Benchmark (SRB):** The SRB contains five shapes with complex geometry and topology, with noisy simulated scans provided by Williams et al. [25]. The challenges of reconstruction lie in the high genus, rich details, missing data, and varying feature sizes. We report average scores for Chamfer distance and Hausdorff distance in Table I (see Appendix A, available online for details).

![Image](image_url)

**TABLE I**
**SURFACE RECONSTRUCTION BENCHMARK**

| Method   | CD↑  | HD↓  | rel. CD | rel. HD |
|----------|------|------|---------|---------|
| SAL [19] | 0.36 | 7.47 | 0.18    | 4.32    |
| IGR [18] | 1.38 | 16.33| 1.20    | 13.18   |
| IGR+FF [18] | 0.96 | 11.06| 0.78    | 6.33    |
| SIREN [20] | 0.42 | 7.67 | 0.78    | 4.52    |
| PHASE+FF [41] | 0.22 | 4.96 | 0.04    | 1.81    |
| SAP [40] | 0.20 | 4.60 | 0.02    | 1.45    |
| iPsr [7] | 0.21 | 5.01 | 0.03    | 1.86    |
| DiGS [23] | 0.19 | 3.52 | 0.01    | 0.37    |
| PCP [24] | 0.54 | 7.11 | 0.36    | 3.96    |
| Ours     | 0.18 | 3.15 | 0.00    | 0.00    |

We report chamfer distance (CD), hausdorff distance (HD), and their mean deviation relative to the best-performing method (rel. CD and rel. HD). All methods do not require ground-truth supervision, and we evaluate them without normals.

Using Fourier features [58] in a mode free of normals, named IGR+FF and PHASE+FF respectively, Table I shows that our method outperforms state-of-the-art methods.

We visualize qualitative results in Fig. 5. As extensions of Poisson reconstruction, SAP and iPsr suffer from shallow gap issues that wrongly close gaps. The resulting surfaces produced by PCP have topological errors and miss geometric details. DiGS is comparable to ours on the dataset except that it is weak in reconstructing sharp or neat structures (see the first row of Fig. 5).

2) **ShapeNet:** The ShapeNet [42] dataset contains CAD models with diverse shapes. We use 13 categories of shapes in ShapeNet, with 20 shapes per category (totaling 260 shapes) for experiments, using the same preprocessing as [44] to get watertight meshes with [59]. For each mesh, we randomly sample 10 k non-uniformly distributed points and add Gaussian noise with a standard deviation of 0.005. The methods included for comparison are RIMLS [2], Screen Poisson Surface Reconstruction (SPR) [4], Neural Splines (NSP) [44], SAL [19], Neural-Pull [22], DiGS [23], and Latent Partition Implicit (LPI) [45]. RIMLS, SPR, and NSP require oriented normals, so we use Dipole [43] to equip the points with normals. As pointed out in [43], this approach for computing normals is robust to thin structures and noisy inputs. We also compare with supervised methods NDC [37] and POCO [15].

To better deal with noise, we use a larger local-size parameter $r = 0.03d_p$. In this situation, the standard number of points in $B_r$ is set to 100. Statistics are reported in Table II and visual results are provided in Fig. 6. Although Dipole [43] is strong in estimating reliable oriented normals, RIMLS, SPR, and NSP still cannot produce high-quality results as it is difficult to infer reliable normals when the input shape contains thin structures and the corresponding point cloud is noisy. In the case of the supervised method, it has been observed that employing NDC alone cannot produce watertight results even with additional post-processing. Additionally, the low NC indicates that it is influenced by noise. POCO achieves the best performance with its learning-based prior. SAL is somewhat noise-resistant but produces over-smooth results and has trouble capturing thin structures. Noise affects Neural-Pull, LPI, and DiGS, causing unwanted surface changes and unnecessary topological holes.
Among them, DiGS has the lowest NC scores. In contrast, our method outperforms the above optimization-based methods in recovering geometric details with the help of mutual regulation between MLP and IMLS. We also achieve better performance than the supervised method NDC. It is important to note that the POCO method uses ShapeNet as both its training and test set.

3) ABC, FAMOUS, and Thingi10K: In this section, we conduct experiments under the three datasets that are preprocessed into points by Points2Surf [13]. ABC, Thingi10K have 100 shapes and FAMOUS has 22 shapes. It applies BlenSor [60] to simulate real scanners to produce ray-direction noise. Each shape is scanned by 10 virtual cameras with variable Gaussian noise and variable point densities. Let $\sigma$ be the standard deviation of scanned points. We use four different noise-level settings: $\sigma = 0$ (no noise, no-n), $\sigma = 0.01 L$ (medium noise, med-n), $\sigma \in [0, 0.05 L]$ (variable noise, var-n), and $\sigma = 0.05 L$ (maximum noise, max-n), where $L$ is the length of the longest bounding box edge. Moreover, there are “sparser” (5 scans with $\sigma = 0.01 L$) and “dense” (30 scans with $\sigma = 0.01 L$) variants to test the robustness of the point density. We adjust the local-size parameter to adapt to different noise levels: $r = 0.01 d_p$, $|B_r(q_j)| = 50$ (no-n) $r = 0.03d_p$, $|B_r(q_j)| = 100$ (sparse, dense, med-n), $0.1d_p$, $|B_r(q_j)| = 200$ (var-n, max-n).

We compare our method with state-of-the-art data-driven surface reconstruction methods, including DeepSDF [9], AtlasNet [53], Points2Surf [13], and POCO [15]. We also include RIMLS [2] and Screened Poisson Surface Reconstruction (SPR) [4] for comparison. Note that they required oriented normals, and thus we use PCP-Net [52] for estimating normal vectors to facilitate the execution of the methods. We also compare our method with the optimization-based methods, including SAL [19], Neural-Pull [22], and DiGS [23]. Table III reports the quantitative results under all variants of the dataset. See Appendix B.4, available online for qualitative results. Both quantitative and qualitative results show that our method is superior to the other optimization-based methods across different noise-level datasets. Further, our method performs better than the RIMLS methods with normal estimation. Similarly, SAL produces over-smooth surfaces, possibly with topological errors. Neural-Pull and DiGS are sensitive to noise, and thus the presence of noise makes them produce non-smooth or even topologically incorrect results. In fact, both of them assume that the input points are accurately on the 0-isosurface, making those noisy points pull the surface toward an unwanted configuration.

We must point out that our scores about Neural-Pull [22] are different from the scores in their original publication due to different formulations of Chamfer Distance; Check Appendix B.4, available online for details.

Remark: Compared with the two latest supervision methods, i.e., Points2Surf and POCO, our method can produce comparable or even better results for most datasets. For example, quantitative statistics on the ABC no-noise dataset show that our method outperforms Points2Surf and POCO; See Table III. However, we also note that on those max-noise datasets, our strategy cannot currently outperform the supervision methods.

4) Real Scans: We also present an experiment on the real-scan benchmark proposed by [28]. This dataset contains 20 objects that are commonly seen in life. The scan quality varies with different types of material. We take the point clouds scanned by the consumer-grade depth camera SHINING Einscan SE as input and the point clouds scanned by a high-precision camera OKIO 5 M as the ground truth. The input point clouds have various artifacts, such as misaligned scans, noise, and missing parts. Fig. 7 visualizes the reconstruction results. The quantitative statistics are available in Table IV. RIMLS and DeepMLS, in spite of being based on the IMLS, struggle to separate the points on the opposite sides of a thin structure and address misaligned scans. Our method, by contrast, can better manifest thin structures even in the presence of noise and misalignment. Furthermore, our method has the best NFS...
| Method          | no-n. | ABC max-n. | FAMOUS max-n. | Thing10k max-n. |
|-----------------|-------|------------|---------------|-----------------|
| RIMLS* [2]      | 2.12  | 4.16       | 8.65          | 1.49            |
| SPR* [4]        | 2.49  | 3.29       | 3.85          | 1.78            |
| DeepSDF+ [9]    | 8.41  | 12.51      | 11.34         | 9.16            |
| AtlassNet+ [53] | 4.69  | 4.04       | 4.47          | 5.29            |
| Points2Surf+ [13]| 1.80  | 2.14       | 2.76          | 1.41            |
| POCO* [15]      | 1.70  | 2.01       | 2.50          | 1.35            |
| SAL [19]        | 3.44  | 4.77       | 7.10          | 2.85            |
| Neural-Pull [22]| 3.62  | 6.33       | 6.36          | 1.86            |
| DGS [23]        | 2.39  | 2.99       | 5.14          | 1.59            |
| Ours            | 1.69  | 2.49       | 5.09          | 1.41            |

Chamfer distance × 100 on ABC, famous, and thing10 k test sets with variable gaussian noise (σ uniformly picked in [0, 0.05L], L largest box length), as prepared by [13]: ‘no-N.’ (no noise), ‘var-N.’ (variable noise, σ in [0, 0.05L]), ‘med-N.’ ( σ = 0.01L), ‘max-N.’ ( σ = 0.05L), ‘sparse’ (5 scans with σ = 0.01L), ‘dense’ (30 scans with σ = 0.01L). The methods marked with ‘*’ require point normals estimated by PCPNet [52] and the methods marked with ‘+’ are supervision based.

C. Scene-Level Reconstruction

To investigate whether our method has the scalability for indoor scene reconstruction, we conduct experiments on the synthetic indoor scene dataset [11]. We use the split plan of [61] that contains 50 shapes. Each scene has 30 k points and we add Gaussian noise with a standard deviation of 0.005. The methods compared include SAL [19], IGR [18] (without normal), Neural-Pull [22], PredictableContextPrior (PCP) [24], and DiGS [23]. We also compare supervised methods SA-ConvONet [61], NDC [37], and POCO [15]. Note that SA-ConvONet is not only supervised but also optimized for each input during inference. From the qualitative comparison shown in Fig. 8, we can see that our approach can still recover some holes in chairs and thin lamps while others cannot capture these small-sized geometric features. This shows that our method can handle scene-level objects. Statistics in Table V demonstrate the superiority of our approach compared to relevant optimization-based methods.

D. Ablation Study and Discussion

The Impact of Mutual Regularization: To conduct the ablation study, we compare three different strategies: (1) IMLS-self-regularization: we modify the IMLS [1] and the RIMLS [2] to develop an IMLS based self-regularization iterative scheme.
Fig. 8. Scene-level reconstruction under synthetic-scan rooms [11]. The input is 30k noisy points.

**TABLE V**

| Method       | CD ($\times 100$) | NC ↑ | FS ↑ |
|--------------|-------------------|------|------|
| SA-CONet$^+$ [61] | 1.48              | 0.89 | 0.86 |
| NDC$^-$ [37]   | 0.56              | 0.78 | 0.97 |
| POCO$^+$ [15]  | 0.48              | 0.92 | 0.99 |
| SAL [19]       | 4.33              | 0.64 | 0.32 |
| IGR [18]       | 7.28              | 0.71 | 0.44 |
| Neural-Pull [18]| 1.18              | 0.87 | 0.87 |
| PCP [24]       | 1.12              | 0.89 | 0.91 |
| DGS [25]       | 0.91              | 0.68 | 0.82 |
| Ours           | 0.91              | 0.90 | 0.93 |

Qualitative comparison for surface reconstruction from unoriented point clouds on the synthetic room indoor scene dataset provided by [11]. The methods denoted with $^+$ are supervised-based.

Fig. 9. Ablation study about mutual regularization.

(2) RBF-self-regularization: we also conduct the experiment with the global-based traditional Radial Basis Function (RBF) kernels method [29] to develop an iterative scheme. The normals are updated according to the last iteration’s results as (2) shown.

(3) MLP-self-regularization: we first leverage SAL [19] to fit the shape and then use the trained model to supervise the new MLP network. (4) Mutual-regularization: using the MLP to provide normals to the IMLS while using the IMLS to provide distances to the MLP as the reference, which is the strategy proposed in this paper. We have a couple of observations. First, all the strategies make the resulting surfaces change gradually, but require different iterations. Also, the final surfaces at the convergence differ from each other. As reported in Table VI and Fig. 9, in contrast to the other two strategies, the mutual-regularization strategy has a higher convergence rate, and the final surface is closer to the ground truth.

**TABLE VI**

| Method       | 1st iter | last iter |
|--------------|----------|-----------|
| IMLS [1]     | 10.97    | 0.45      | 0.10      |
| RIMLS [2]    | 10.84    | 0.46      | 0.10      |
| RBF [29]     | 0.73     | 0.60      | 0.75      |
| SAL [19]     | 0.61     | 0.65      | 0.82      |
| Ours         | 0.45     | 0.92      | 0.94      |

The Impact of Loss Terms: By switching off the gradient weighting scheme $\psi$ and the distance weighting scheme $\theta$ respectively, we come to observe how different the reconstructed results become. We use “w/o gradient” and “w/o distance” to denote the options. The qualitative results and the quantitative results are shown in Fig. 10 and Table VII, respectively. It shows

Fig. 10. Reconstruction results without using distance weighting term $\theta$ (left), without using gradient weighting term $\psi$ (middle), and vanilla including both (right).
TABLE VII
ABLATION STUDY OF LOSS TERM

|                     | FAMOUS no-n. | med-n. |
|---------------------|--------------|--------|
| w/o θ               | 1.83         | 2.13   |
| w/o ψ               | 2.01         | 2.17   |
| vanilla             | 1.34         | 1.62   |
| σ_{coherence} = 0.1 | 1.34         | 1.68   |
| σ_{coherence} = 0.3 | 1.34         | 1.62   |
| σ_{coherence} = 0.5 | 1.40         | 1.71   |
| σ_{coherence} = 1.0 | 1.79         | 1.79   |

We show the chamfer distance scores with different options.

Fig. 11. Visual comparison of different σ_{coherence}’s. A small σ_{coherence} tends to better manifest sharp features.

TABLE VIII
ABLATION STUDY ABOUT NETWORK COMPONENTS

|                     | FAMOUS no-n. | w/ SIREN | w/ Eikonal term | w/o GNI | vanilla |
|---------------------|--------------|----------|-----------------|--------|---------|
|                     |              | 7.16     | 1.34            | 3.48   | 1.34    |

We show the chamfer distance scores under different options.

that without ψ or θ, the overall accuracy decreases. Under the mode of “w/o gradient”, it is hard to enforce the spatial coherence of gradients, causing conspicuous artifacts around thin structures or sharp tips. Under the mode of “w/o distance”, the ability to preserve geometric details is weakened due to the fact that θ emphasizes more on the contribution of a local part.

The Impact of Parameter σ_{coherence}: As mentioned above, ψ is to enforce the spatial coherence of gradients. The parameter σ_{coherence} in ψ is used to tune the degree to which sharp features are preserved. As shown in Fig. 11, a small σ_{coherence} tends to better manifest sharp features, while a large σ_{coherence} tends to reconstruct a smoother surface. In the default setting, we set σ_{coherence} = 0.3 to balance the ability of feature preserving and noise resistance. The quantitative results are shown in Table VII.

The Impact of Network Components: We explore the possibilities of improving the performance using some known techniques. Specially, we come to discuss the influence of SIREN [20] (activation function), Eikonal term [18] (gradients constraints), and geometric network initialization [19] under the no-noise FAMOUS dataset. The qualitative results and the quantitative results are shown in Fig. 12 and Table VIII, respectively.

Fig. 12. Ablation studies of network components.

First, we replace Softplus with SIREN [20] in terms of the activation function. We use “w/ SIREN” to denote this option but cannot see any performance improvement. For example, under the option of “w/ SIREN”, it may lead to the occurrence of outlier surfaces due to its non-monotonously.

Second, the Eikonal term is enforced in many existing research works. We used an option of “w/ Eikonal term” to explicitly add such a constraint but didn’t observe any change. The reason lies in that the constraint about the gradient norm has been fully considered in our loss design.

Finally, we use an option of “w/o GNI” to initialize the normals randomly. Even so, our method can still converge to an SDF, and sometimes the SDF can also manifest the real shape, which is due to the strong mutual regularization between the MLP and the IMLS.

Comparison to Existing IMLS Methods: Our neural network inherits the noise-resistance ability of the IMLS. A natural question arises: is Neural-IMLS just a variant of the original IMLS and cannot defeat the original IMLS in the noise-free situation? To answer the question, we compare the following 5 implicit functions: (1) the implicit surface obtained by IMLS [1], (2) the implicit one by RIMLS [2], (3) the implicit one by SPR [4], (4) the implicit one reported by IMLS when our neural network gets converged, i.e., f_{IMLS}, and (5) the implicit one reported by MLP when our neural network gets converged, i.e., f_{MLP}. We provide the ground truth normal for (1), (2), and (3). The qualitative results and the quantitative results are shown in Fig. 13 and Table IX, respectively. In the top row of Fig. 13, we show a teapot model whose point cloud is free of noise and complete. As can be observed, the RIMLS easily produces broken parts, while the IMLS struggles to preserve local geometric features (see the handle of the Utah teapot), and RIMLS easily produces broken parts. In the bottom row
TABLE IX
COMPARISON BETWEEN DIFFERENT IMLS METHODS

|         | IMLS | RIMLS | SPR | Func fIMLS | MLP | fMLP |
|---------|------|-------|-----|------------|-----|------|
| FAMOUS  | 2.92 | 1.46  | 1.53| 1.36       | 1.34|      |

Front View

Back View

Fig. 14. Comparing $f_{IMLS}$ and $f_{MLP}$ for scan with noise and missing parts.

TABLE X
COMPARISON BETWEEN DEEPMMLS [14] AND OURS

|         | DeepML | Ours  |
|---------|--------|-------|
| FAMOUS  | 1.68   | 1.54  |

Input $f_{IMLS}$ $f_{MLP}$

Fig. 15. Comparison between DeepML [14] and ours.

TABLE XI
ABLATION STUDY OF LOCAL SIZE

|         | $r_{small}$ | $r_{medium}$ | $r_{large}$ | $k_{small}$ | $k_{medium}$ | $k_{large}$ |
|---------|--------------|--------------|-------------|-------------|--------------|-------------|
| FAMOUS  | 1.35         | 1.92         | 2.04        | 1.63        | 1.76         | 2.01        |
| med-n   | 2.67         | 1.62         | 2.95        | 1.92        | 2.35         | 2.81        |
| max-n   | 10.63        | 6.75         | 4.71        | 9.38        | 6.99         | 4.82        |

We show the chamfer distance scores for different local-size parameters.

of Fig. 13, we show a cup model whose point cloud is free of noise but has missing parts in the inner wall (around the handle). It can be seen that IMLS, RIMLS, and SPR suffer from the data-missing defect. Additionally, from Fig. 14 we can see that $f_{MLP}$ has a stronger completion capability than $f_{IMLS}$ since the neural network, whose basis functions are globally defined, naturally owns the ability of shape completion.

Comparison to DeepMLS: DeepMLS [14] is also a learning-based reconstruction method based on IMLS [1]. Like our approach, DeepMLS can also deal with noisy data, but it needs to be trained with ground-truth SDFs. Next, we will compare our approach with DeepMLS.

For a fair comparison, we reproduce the results of DeepMLS using the officially-released pre-trained model. The test dataset is the low-noise FAMOUS, where the constant Gaussian noise strength is set to 0.005 of the maximum side length of the bounding box, which is consistent with the training data of the original DeepMLS.

Table X and Fig. 15 respectively present the quantitative results and the qualitative results. The disadvantages of DeepMLS are two-fold. First, DeepMLS cannot predict a suitable radius. Second, DeepMLS does not take the spatial coherence of gradients into account.

Impact of Local Patch Size: We also conducted an ablation study about how the local patch size influences the reconstruction results. Our tests are respectively made on the no-noise, medium-noise and max-noise FAMOUS datasets. The statistics are available in Table XI.

Generally speaking, one has to increase the radius $r$ or the number of the nearest neighboring points so as to adapt to noise. For this purpose, we set $r_{small} = 0.01d_P$, $r_{medium} = 0.03d_P$, and $r_{large} = 0.1d_P$, respectively on the three datasets. We also test the k-nearest neighbor algorithm (k-NN) with different $k$’s: $k_{small} = 100$, $k_{medium} = 300$, and $k_{large} = 500$, respectively.

The experimental observations are two-fold. On the one hand, controlling the radius is better than controlling the number of nearest neighbors since the former can deal with irregular point distributions. On the other hand, a larger $r$ tends to report a smoother surface.

Effects of Query Points: The query point plays an important role in our method. We explore its effect by merely adjusting the number of query points, such that we sample $I = \{1, 10, 25, 50, 100\}$ for each distribution, and query points range, such that the standard deviation $k = \{1, 10, 25, 50, 100\}$, under FAMOUS no-noise. We report the comparison in Table XII. On one hand, it has been observed that increasing the number of query points can enhance performance to some extent. However, the extent of improvement gradually diminishes as the number of query points continues to increase. Furthermore, it should be noted that employing more query points will inevitably lead to an increase in computational time. On the other hand, too small or too large query locations range will degenerate the surface reconstruction performance. Since it is hard to use the query locations to probe the area around the surface of the query
TABLE XII

| I  | 1  | 10 | 25 | 50 | 100 |
|----|----|----|----|----|-----|
| k  | 2.39 | 1.92 | 1.34 | 1.31 | 1.29 |

TABLE XIII

| # parameters | IGR | Neural-Pull | SIREN | DiGS | Ours |
|--------------|-----|-------------|-------|------|------|
| time [ms]    | 1.86M | 1.86M | 264.4K | 264.4K | 1.86M |
|              | 50.73 | 20.71 | 11.52 | 36.28 | 34.10 |

The comparison is made among IGR [18], Neural-Pull [22], SIREN [20], and DiGS [23] without the supervision of normals. Timing statistics are reported in milliseconds (ms).

Fig. 16. We use various approaches to generate normals, followed by feeding normals to SPR [4], to observe which kind of normals is more faithful.

location range is too small, it is also tough to make the IMLS produce reliable results.

Runnable Performance: We include IGR [18], Neural-Pull [22], SIREN [20], and DiGS [23] for comparison of the computational speed per iteration without the supervision of normals. To be fair, we set the batch size to 15 K for all the methods. Also, we use the \( r = 0.03d_p, |B_r| = 100 \) for local neighbors computation. Note that IGR, Neural-Pull, and ours leverage the same network that is 8 layers each containing 512 hidden units, and a single skip connection from the input to the middle layers, and SIREN and DiGS use the 4 hidden layers containing 256 hidden units with Sine activation. Table XIII reports the timing cost spent in a single iteration. Roughly speaking, the timing costs of Neural-Pull and ours are less than SIREN since DiGS and ours need a second-order optimization. However, ours is more computationally efficient than IGR.

V. ILLUSTRATIVE EXAMPLES

Oriented Normal Estimation: It is necessary to evaluate whether the normals produced by our approach are faithful. For this purpose, we use various approaches to generate normals, followed by feeding normals to SPR [4], to evaluate the reconstruction quality. In particular, the approaches used to generate normals include PCPNet [52], VIPSS [6], iPSR [7], and ours. According to the visual results shown in Fig. 16, PCPNet fails due to severe point insufficiency; iPSR fails due to the incorrect inference about the normals around the thin tail, and VIPSS is able to reconstruct the surface but the geometric features are not as prominent as ours. By contrast, ours is the best among the results on the super-sparse point cloud, achieving a high fidelity around the tail and the ear of the kitten model, which shows that our normals are more faithful.

Geometric Fidelity: To demonstrate the fidelity-preserving ability of our method, we conduct experiments on the noisy point cloud sampled from the implicit Bretzel. The existing optimization-based methods, e.g., PCP [24] and DiGS [23], do not resist noise.

Fig. 17. The reconstruction results from a noisy point cloud sampled from the implicit Bretzel. The existing optimization-based methods, e.g., PCP [24] and DiGS [23], the visual comparison (see Fig. 17) shows that existing SOTA optimization-based methods cannot produce a faithful result for a noisy input. By contrast, our algorithm is superior to them in terms of the ability to reconstruct a high-fidelity surface even in the occurrence of noise.

Sparse 3D Sketch: It is interesting to make clear if our approach can transform a super-sparse 3D sketch point cloud into a meaningful shape. In VIPSS [6], the authors gave a 3D sketch point cloud of 1 k points. We visualize the reconstructed results by VIPSS [6], NDC [37], POCO [15] and ours in Fig. 18. It can be seen that the supervision methods NDC and POCO fail since their training set does not include sketch-type data. Our result is comparable to VIPSS but VIPSS seriously suffers from the number of points.

Partial Scan: By taking a partial scan as the input, we come to observe the reconstruction ability of our approach. In Fig. 19, we use an angel model [62] to compare DiGS [23] and ours. It can be seen that DiGS cannot fill the small gaps but our approach outputs a closed model. Our explanation is that our neural network aims at reconstructing a closed shape even if the input point cloud has a single layer.

Large-size Point Cloud: In order to test the ability to deal with a large-size point cloud, we use a real-scan [63] as the input, where the number of points amounts to 700 k. The approaches used for comparison include (1) SPR [4] plus Dipole [43], (2) iPSR [7], (3) NDC [37], (4) POCO [15], and (5) Ours. From Fig. 20, we can see that our approach can deal with a large-size...
benefiting from the global representation ability of the neural network, is able to deal with various kinds of defects, such as thin gaps and missing parts. Extensive experiments and ablation studies validate its effectiveness.

Our neural network, in its current form, includes a hyper-parameter $r$. We have to manually tune $r$ to adapt to different levels of noise. In the future, we shall explore the possibility of automatically learning the adaptive parameter $r$ to achieve a good balance between the noise-resistance ability and the geometric feature preserving ability. Another limitation of this approach is the run-time during inference. Our method necessitates additional optimization time for each tested shape, similar to other optimization-based methods. Consequently, it may be slower compared to pre-trained models that only require a single forward process during inference. But as learning on irregular structures becomes more mainstream and optimized (e.g., recently released packages Kaolin [64], PyTorch3D [65] and torch-points3d [66]), the runtime will improve significantly.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] R. Kolluri, “Provably good moving least squares,” ACM Trans. Algorithms, vol. 4, no. 2, pp. 1–25, 2008.
[2] A. C. Gortizrelli, G. Guennebaud, and M. Gross, “Feature preserving point set surfaces based on non-linear kernel regression,” in Computer Graphics Forum, vol. 28. Hoboken, NJ, USA: Wiley, 2009, pp. 493–501.
[3] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in Proc. 4th Eurographics Symp. Geometry Process., 2006, pp. 61–70.
[4] M. Kazhdan and H. Hoppe, “Screened poisson surface reconstruction,” ACM Trans. Graph., vol. 32, no. 3, pp. 1–13, 2013.
[5] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, “Efficient implementation of marching cubes’ cases with topological guarantees,” J. Graph. Tools, vol. 8, no. 2, pp. 1–15, 2003.
[6] Z. Huang, N. Carr, and T. Ju, “Variational implicit point set surfaces,” ACM Trans. Graph., vol. 38, no. 4, pp. 1–13, 2019.
[7] F. Hou, C. Wang, W. Wang, H. Qin, C. Qian, and Y. He, “Iterative poisson surface reconstruction (IPSR) for unoriented points,” ACM Trans. Graph., vol. 41, no. 4, pp. 1–13, 2022.
[8] Y. Xie et al., “Neural fields in visual computing and beyond,” in Computer Graphics Forum, vol. 41. Hoboken, NJ, USA: Wiley, 2022, pp. 641–676.
[9] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “DeepSDF: learning continuous signed distance functions for shape representation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 165–174.
[10] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3D reconstruction in function space,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4460–4470.
[11] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutonal occupancy networks,” in Proc. 16th Eur. Conf. Comput. Vis., Glasgow, U.K., Aug. 23–28, 2020, pp. 523–540.
[12] C. Jiang et al., “Local implicit grid representations for 3D scenes,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6001–6010.
[13] P. Ertler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Wimmer, “Points2surf learning implicit surfaces from point clouds,” in Euro Conf. Comput. Vis., 2020, pp. 108–124.
[14] H.-X. G. Shi-Lin Liu, H. Pan, P. Wang, X. Tong, and Y. Liu, “Deep implicit moving least-squares functions for 3D reconstruction,” in IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 1788–1797.
[15] A. Boulch and R. Marlet, “POCO: Point convolution for surface reconstruction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 6302–6314.
[16] J. Huang, H.-X. Chen, and S.-M. Hu, “A neural gateliner solve for accurate surface reconstruction,” ACM Trans. Graph., vol. 41, no. 6, pp. 1–6, 2022.
...
Pengfei Wang received the PhD degree in the school of computer science and technology, Shandong University. His research interests focus on computer graphics. He developed a robust algorithm for explicit surface reconstruction from a set of unoriented points.

Peng-Shuai Wang received the PhD degree from the Institute for Advanced Study, Tsinghua University, in 2018. He is currently a tenure-track assistant professor with Peking University. Before that, he was a senior Researcher with Microsoft Research Asia. His research interest includes deep learning, geometry processing, and computer graphics. His paper on Octree-based Sparse Convolutional Networks is one of the Top-5 Most Cited Papers in ACM Transactions on Graphics from 2016 to 2021.

Qiujie Dong is currently working toward the PhD degree with the school of computer science and technology, Shandong University. His research interests include computer graphics and computer vision. Currently, he focuses on geometric deep learning. His academic webpage is at https://qiujiedong.github.io/.

Junjie Gao is currently working toward the PhD degree in computer science with Shandong University. His research interests are mainly computer graphics, particularly how to make computer graphics smarter with deep learning techniques.

Shuangmin Chen received the PhD degree from Ningbo University, in 2018 and currently works as an associate professor within the School of Information and Technology with the Qingdao University of Science and Technology, China. Her research interests focus on computer graphics and computational geometry. During the past ten years, she published more than 40 research papers in famous journals/conferences.

Shiqing Xin received the PhD degree from Zhejiang University (China), in 2009. He is an associate professor with the school of computer science at Shandong University. After that, he worked as a research fellow at Nanyang Technological University (Singapore) for three years. His research interests include various geometry processing algorithms, especially geodesic computation approaches and Voronoi/power tessellation methods. During the past ten years, he published more than 70 papers in top journals and conferences, including IEEE Transactions on Visualization and Computer Graphics and ACM Transactions on Graphics. He received three best paper awards and many other academic awards.

Changhe Tu received the BSc, MEng, and PhD degrees from Shandong University, China, in 1990, 1993, and 2003, respectively. He is a professor with the School of Computer Science and Technology, Shandong University, China. His research interests are in the areas of computer graphics, 3D vision, and computer-aided geometric design. He has published more than 100 papers in international journals and conferences. Currently, he leads the CG-VIS group with Shandong University.

Wenping Wang (Fellow, IEEE) received the PhD degree in computer science from the University of Alberta, in 1992. He is the chair professor of computer science with the University of Hong Kong. His research interests include computer graphics, computer visualization, computer vision, robotics, medical image processing, and geometric computing. He is the associate editor of several premium journals, including Computer Aided Geometric Design (CAGD), Computer Graphics Forum (CGF), IEEE Transactions on Computers, and IEEE Computer Graphics and Applications, and has chaired more than 20 international conferences, including Pacific Graphics 2012, ACM Symposium on Physical and Solid Modeling (SPM) 2013, SIGGRAPH Asia 2013, and Geometry Submit 2019. He received the John Gregory Memorial Award for his contributions in geometric modeling.