CMG-Net: Robust Normal Estimation for Point Clouds via Chamfer Normal Distance and Multi-scale Geometry

Yingrui Wu[1,2*], **Mingyang Zhao**[3*], **Keqiang Li**[4], **Weize Quan**[1,2], **Tianqi Yu**[5], **Jianfeng Yang**[5], **Xiaohong Jia**[6,2], **Dong-Ming Yan**[1,2 †]

[1]MAIS, Institute of Automation, Chinese Academy of Sciences, [2]University of Chinese Academy of Sciences
[3]Hong Kong Institute of Science & Innovation, Chinese Academy of Sciences, [4]SenseTime Research
[5]School of Electronic and Information Engineering, Soochow University, [6]AMSS, Chinese Academy of Sciences
wuyingrui2023@ia.ac.cn, {migyangz, likeq98, qweizework, yandongming}@gmail.com
{tqyu, jfyang}@suda.edu.cn, xhjia@amss.ac.cn

Abstract

This work presents an accurate and robust method for estimating normals from point clouds. In contrast to predecessor approaches that minimize the deviations between the annotated and the predicted normals directly, leading to direction inconsistency, we first propose a new metric termed *Chamfer Normal Distance* to address this issue. This not only mitigates the challenge but also facilitates network training and substantially enhances the network robustness against noise. Subsequently, we devise an innovative architecture that encompasses *Multi-scale Local Feature Aggregation* and *Hierarchical Geometric Information Fusion*. This design empowers the network to capture intricate geometric details more effectively and alleviate the ambiguity in scale selection. Extensive experiments demonstrate that our method achieves the state-of-the-art performance on both synthetic and real-world datasets, particularly in scenarios contaminated by noise. Our implementation is available at https://github.com/YingruiWoo/CMG-Net_Pytorch.

1 Introduction

Normal estimation is a fundamentally important task in the field of point cloud analysis, which enjoys a wide variety of applications in 3D vision and robotics, such as surface reconstruction (Fleishman, Cohen-Or, and Silva 2005; Kazhdan, Bolitho, and Hoppe 2006), denoising (Lu et al. 2020b) and semantic segmentation (Grilli, Menna, and Remondino 2017; Che and Olsen 2018). In recent years, many powerful methods have been developed to enhance the performance of normal estimation. However, these approaches involving both traditional and learning-based ones often suffer from *heavy noise* and struggle to attain high-quality results for point clouds with *complex geometries*.

Traditional methods (Hoppe et al. 1992; Levin 1998; Cazals and Pouget 2005) typically encompass fitting local planes or polynomial surfaces and inferring normal vectors from the fitted outcomes. Although straightforward, these approaches are vulnerable to noise and encounter challenges
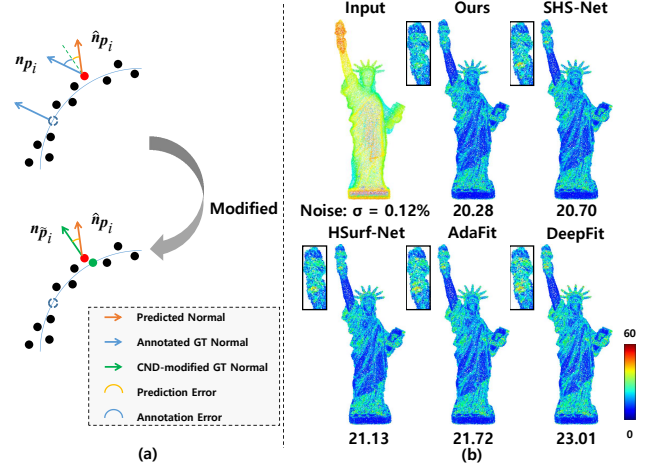
Figure 1: (a) Comparison between the annotated and the proposed CND-modified normals, where the latter is more consistent with the underlying surface geometry. (b) Our method outperforms competitors with higher robustness to noise and intricate shape details (indicated by the heat map).

when attempting to generalize to complex shapes. Furthermore, their performance hinges significantly on the meticulous tuning of parameters.

In comparison with traditional approaches, learning-based proposals (Guerrero et al. 2018; Ben-Shabat et al. 2019; Hashimoto and Saito 2019; Zhou et al. 2020; Wang and Prisacariu 2020; Lenssen, Osendorfer, and Masci 2020; Ben-Shabat et al. 2020; Cao et al. 2021; Zhu et al. 2021; Zhou et al. 2022b; Zhang et al. 2022; Li et al. 2022a,b; Du et al. 2023; Li et al. 2023a) have better generalization and less dependency on parameter tuning. There are two types of learning-based normal estimators comprising *deep surface fitting* and *regression*. The former predicts pointwise weights of the input point cloud patch and derives a polynomial surface by *weighted least-squares* (WLS) fitting. However, due to the fixed order of polynomial functions, deep surface fitting usually grapples with overfitting or underfitting when dealing with various surfaces. On the con-

trary, regression-based methods adopt *Multi-Layer Perception* (MLP) to extract features of the input patch and directly regress normal vector from these features. Benefiting from their strong feature extraction capability, recent regression-based methods have advanced normal estimation for clean point clouds. However, they have not made substantial headway in improving normal estimation on point clouds that are affected by noise.

To address noisy normal estimation issues, in this paper, we first analyze the normal estimation deviation produced by noisy point clouds, and then point out the *inconsistency* between the annotated (ground-truth) normal and the input patch, as illustrated in Fig. 1(a). We find that this direction inconsistency indeed significantly affects the network training and the output evaluation, also degrading the downstream tasks such as surface reconstruction. The reason is that when point coordinates change significantly due to noisy influence, their neighborhood geometry and normals change accordingly, while the annotated normals are fixed. To deal with this problem, instead of using the conventional *Root of Mean Squared Error* (RMSE), we propose a more reasonable metric for normal estimation termed *Chamfer Normal Distance* (CND), which replaces the original annotated normal vector with the normal of the closest point locating on the potential clean point cloud. Moreover, we minimize the loss function modified by CND to reduce the disturbance of inconsistency deviations during training. We show that our newly defined loss function achieves much higher normal estimation accuracy than competitors on a set of benchmark datasets.

Furthermore, we develop a novel network framework named CMG-Net, integrating the CND-modified loss with multi-scale geometric structures, for more stable and robust normal estimation. Unlike previous approaches that merely capture a single scale of local or global features, CMG-Net performs a multi-scale feature extraction followed by integration through an attention layer. This greatly facilitates the network capability to capture intricate geometric details and address the ambiguity of the optimal scale selection. Moreover, we further combine the local and global features from various scales together in the hierarchical architecture to increase the multi-scale information for network inference.

We conduct extensive experiments to validate the developed method and compare it with the *state-of-the-art* (SOTA) approaches on various benckmark datasets including *PCPNet* (Guerrero et al. 2018) and the *indoor SceneNN* dataset (Hua, Tran, and Yeung 2018). Results demonstrate that our method outperforms baselines by a large margin, especially on point clouds with noise, and those with intricate geometric details and various distribution density. To summarize, our main technical contributions are threefold as follows:

- We propose a new method that integrates the CND metric for robust normal estimation, which solves the direction inconsistency problem effectively and significantly boosts network training and inference.

- We design a novel network that incorporates multi-scale feature extraction along with hierarchical inference com-

bined with intricate geometry information fusion, which is capable of capturing intricate geometric details and addressing the challenge of scale selection ambiguity.

- We perform comprehensive experiments to demonstrate the enhancements brought by our proposed method, thereby pushing the boundaries of SOTA performance, especially on noisy normal estimation scenarios.

2 Related Work

2.1 Traditional Methods

Principal Component Analysis (PCA) (Hoppe et al. 1992) stands as the most widely adopted point cloud normal estimation method, which fits a plane to the input surface patch. Subsequent variants involving Moving Least Squares (MLS) (Levin 1998), truncated Taylor expansion fitting (n-jet) (Cazals and Pouget 2005), local spherical surface fitting (Guennebaud and Gross 2007) and multi-scale kernel (Aroudj et al. 2017) are proposed to reduce the noisy influence through selecting larger patches and employing more intricate energy functions. Nevertheless, these approaches typically tend to oversmooth sharp features and geometric details. To circumvent these issues, Voronoi diagram (Amenta and Bern 1998; Alliez et al. 2007; Mérigot, Ovsjanikov, and Guibas 2010), Hough transform (Boulch and Marlet 2012), and plane voting (Zhang et al. 2018) are deployed in normal estimation. However, these techniques depend on manual parameter tuning heavily, which hinders their practical applications.

2.2 Learning-based Methods

With the powerful development of neural network, learning-based normal estimation achieves better performance and less dependence of parameter tuning than traditional approaches. They can be generally divided into two categories: *deep Surface fitting* and *regression-based* approaches.

Deep surface fitting methods. These methods typically employ a deep neural network to predict point-wise weights and then fit a polynomial surface to input patches using WLS such as IterNet (Lenssen, Osendorfer, and Masci 2020) and DeepFit (Ben-Shabat et al. 2020). Analogously, Zhang *et al.* (2022) adopted the predicted weights as the guiding geometric information. AdaFit (Zhu et al. 2021) proposed a novel layer to aggregate features from multiple global scales and then predicted point-wise offset to improve the normal estimation accuracy. To learn richer geometric features, GraphFit (Li et al. 2022a) combined graph convolutional layers with adaptive modules, while Du *et al.* (2023) analyzed the approximation error of these methods and suggested two fundamental design principles to further improve the estimation accuracy. However, due to the constant order of the objective polynomial functions, deep surface fitting methods typically suffer from overfitting and underfitting.

Regression-based methods. This type casts the normal estimation problem as a regression process and predicts the point cloud normals via the network straightforward. For instance, HoughCNN (Boulch and Marlet 2016) transformed

point clouds into a Hough space and then utilized Convolutional Neural Networks (CNN) to directly infer normal vectors, whereas Lu *et al.* (2020a) projected point clouds into a height map by computing distances between scatter points and the fitted plane. However, these approaches sacrifice the 3D geometry unavoidably when executing in 2D spaces. PCPNet (Guerrero et al. 2018) directly adopted the unstructured point clouds as input and then used the Point-Net (Qi et al. 2017a) to capture multi-scale features instead. Hashimoto *et al.* (2019) combined PointNet with 3D-CNN to extract local and spatial features, and NestiNet (Ben-Shabat et al. 2019) employed mixture-of-experts framework to determine the optimal normal estimation scale. To provide more information of the input patch, Refine-Net (Zhou et al. 2022a) additionally calculated the initial normals and the height map. Recent work involve HSurf-Net (Li et al. 2022b) and SHS-Net (Li et al. 2023a) first transformed point clouds into a hyper space through local and global feature extractions and then performed plane fitting in the constructed space. NeAF (Li et al. 2023b) inferred an angle field around the ground truth normal to make it learn more information of the input patch. Benefiting from the strong feature extraction abilities of the network architectures, recent regression-induced approaches demonstrate promising results on clean point clouds. However, they have yet made significant progress in normal estimation on noisy point clouds, which are often emerged in practical scenarios.

Aiming at improving the robustness to noise, we identify a crucial inconsistency between the annotated normal and the neighborhood geometry of the noisy point and introduce CND to address this problem. Besides, compared with the recent regression methods, we propose a network that combines various geometric information extraction with a hierarchical architecture to make the complex information capture more effectively.

3 Rethinking Noisy Normal Estimation

3.1 Direction Inconsistency

Previous learning-based approaches directly minimize the deviations between the predicted normals and the annotated ones for training and evaluation. This is reasonable for noise-free scenarios, however, for the noisy point clouds, due to the noise-caused relative coordinate changes, the annotated normals indeed are inconsistent with the neighborhood geometry of the query points. As presented in Fig. 2(a), given a set of noisy point clouds $\mathcal{P}$, suppose the ground truth position locating on the surface of the noisy point $\boldsymbol{p}_i$ is $\tilde{\boldsymbol{p}}_i$. The annotated normal of $\boldsymbol{p}_i$ is $\boldsymbol{n}_{\boldsymbol{p}_i} \in \mathbb{R}^3$, which is the same as the one of the point before adding noise, and the normal of $\tilde{\boldsymbol{p}}_i$ is $\boldsymbol{n}_{\tilde{\boldsymbol{p}}_i} \in \mathbb{R}^3$. If we optimize the typically defined normal estimation loss $\|\boldsymbol{n}_{\boldsymbol{p}_i} - \hat{\boldsymbol{n}}_{\boldsymbol{p}_i}\|_2^2$ as predecessors, where $\hat{\boldsymbol{n}}_{\boldsymbol{p}_i}$ is the predicted normal, this will unavoidably lead to inconsistency between the annotated normal $\boldsymbol{n}_{\boldsymbol{p}_i}$ and the input patch $\boldsymbol{P}_i$. What's worse, this inconsistency greatly decreases the quality of the training data and thus lowers down the estimation ability of the network on noisy point clouds.

Moreover, this inconsistency also degrades downstream tasks such as denoising and 3D reconstruction. For instance,
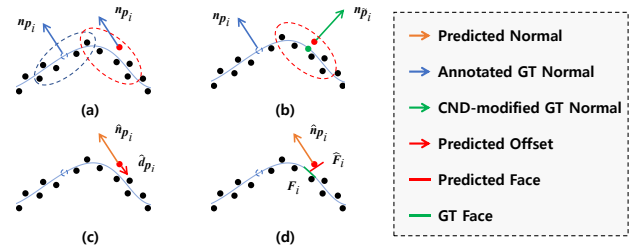

Figure 2: (a) The annotated normal $\boldsymbol{n}_{\boldsymbol{p}_i}$ of noisy point $\boldsymbol{p}_i$ determined before noisy disturbance indeed is inconsistent with the input patch (dashed red ellipse). (b) The direction of the normal $\boldsymbol{n}_{\tilde{\boldsymbol{p}}_i}$ of the nearest clean point $\tilde{\boldsymbol{p}}_i$ is more consistent with the input patch. (c) The predicted offset $\hat{\boldsymbol{d}}_{\boldsymbol{p}_i}$ cannot drag $\boldsymbol{p}_i$ to the noise-free underlying surface. (d) This inconsistency also arises for surface reconstruction assignments.

Fig. 2(c) shows the denosing principle for point clouds. If we utilize the predicted normal vector $\hat{\boldsymbol{n}}_{\boldsymbol{p}_i}$, which closely resembles the annotated normal vector $\boldsymbol{n}_{\boldsymbol{p}_i}$ (indicating a highly accurate estimation), then the introduced offset $\hat{\boldsymbol{d}}_{\boldsymbol{p}_i}$ will not align or bring $\boldsymbol{p}_i$ closer to the noise-free underlying surface. Anonymously, in the context of reconstruction tasks, as shown in Fig. 2(d), the regenerated mesh face $\hat{\boldsymbol{F}}_i$ in relation to the normal vector $\hat{\boldsymbol{n}}_{\boldsymbol{p}_i}$ significantly deviates from the authentic mesh fact $\boldsymbol{F}_i$.

3.2 Scale Ambiguity

Another challenge in current normal estimation approaches is the ambiguity regarding the optimal scale in both local and global feature extraction. Concerning local structures, using large scales typically improves robustness against noise but can lead to oversmoothing of shape details and sharp features. Conversely, small scales can preserve geometric details but are relatively sensitive to noise. When it comes to global features, large scales include more structure information from the underlying surface but may also incorporate irrelevant points, thus degrading the geometry information of the input patch. On the other hand, small scales reduce irrelevant points but are less robust to noise. Previous works have struggled to effectively extract and combine multi-scale local and global features, making them highly dependent on scale selection and resulting in unsatisfactory performance on both noisy point clouds and complex shape details.

4 Proposed Method

To solve the aforementioned issues, we propose a novel normal estimation approach that is robust against noise and less sensitive to scale selection. Concrete technical contributions are presented in the following.

4.1 Chamfer Normal Distance

To bridge the direction inconsistency between the annotated normal and the predicted one of the input patch, instead of using the conventional metric $\|\boldsymbol{n}_{\boldsymbol{p}_i} - \hat{\boldsymbol{n}}_{\boldsymbol{p}_i}\|_2^2$, inspired from
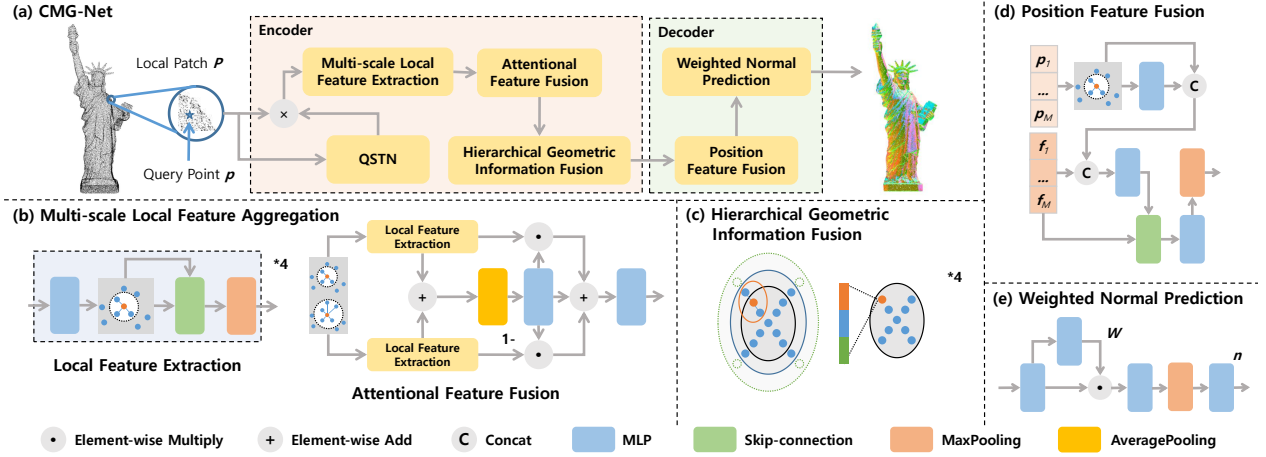
Figure 3: Architecture of the proposed method. (a) Overall structure of CMG-Net. (b) Multi-scale Local Feature Aggregation. (c) Hierarchical Geometric Information Fusion. (d) Position Feature Fusion. (e) Weighted Normal Prediction.

the *Chamfer Distance* (CD)

$$\mathrm{C}(\mathcal{P}, \hat{\mathcal{P}}) = \frac{1}{N_1} \sum_{\boldsymbol{p}_i \in \mathcal{P}} \min_{\hat{\boldsymbol{p}}_j \in \hat{\mathcal{P}}} (\|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_j\|_2^2) + \frac{1}{N_2} \sum_{\hat{\boldsymbol{p}}_j \in \hat{\mathcal{P}}} \min_{\boldsymbol{p}_i \in \mathcal{P}} (\|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_j\|_2^2), \quad (1)$$

where N_1 and N_2 represent the cardinalities of the point cloud $\mathcal{P}$ and $\hat{\mathcal{P}}$, we formulate the *Chamfer Normal Distance* (CND) as

$$\mathrm{CND}(\mathcal{P}, \tilde{\mathcal{P}}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \arccos^2 < \boldsymbol{n}_{\tilde{\boldsymbol{p}}_i}, \hat{\boldsymbol{n}}_{\boldsymbol{p}_i} >}, \quad (2)$$

where $< \cdot, \cdot >$ represents the inner product of two vectors and $\tilde{\boldsymbol{p}}_i$ is the closest point of $\boldsymbol{p}_i$ in the noise-free point cloud $\tilde{\mathcal{P}}$. In contrast to previous approaches that relied on annotated normal correspondence, our proposed CND manner assures consistency with the underlying geometric structure of the input patch (Fig. 2(b)). The CND metric not only faithfully captures the prediction errors in noisy point clouds, but also eliminates the direction inconsistency during network training, thus substantially improving the network robustness and facilitating the subsequent assignments.

4.2 CMG-Net

To capture more fruitful multi-scale structure information and solve the scale ambiguity issue simultaneously, we develop a network combining various geometric information extraction with a hierarchical architecture termed CMG-Net. Given a patch $\boldsymbol{P} = \{\boldsymbol{p}_i \in \mathbb{R}^3\}_{i=1}^N$ centralized at a query point $\boldsymbol{p}$, as shown in Fig. 3(a), CMG-Net first normalizes the input points and rotates $\boldsymbol{P}$ by PCA and QSTN (Qi et al. 2017a; Du et al. 2023) to initialize the normal vectors. Then, we group the local features by *k-nearest neighbors* (*k*-NN) with different scales and aggregate them together. Besides, we design a hierarchical structure with intricate geometry information fusion, followed by the decoding of the embedding features. Our loss function modified by CND enables the network jumping out of the annotation inconsistency.

Multi-scale Local Feature Aggregation. Previous methods group the local features by k-NN and capture the geometric information by MLP and maxpooling (Li et al. 2022b). However, this manner often suffers from scale ambiguity and results in unsatisfactory robustness against noise. To solve this issue, as presented in Fig. 3(b), we construct graphs by k-NN with small and large scales and employ the skip-connection and maxpooling to capture the local structures. The *Local Feature Extraction* (LFE) can be formulated as

$$\boldsymbol{f}_i^{n+1} = \mathrm{MaxPool} \left\{ \phi_1 \left(\varphi_1 \left(\boldsymbol{f}_i^n \right), \varphi_1 \left(\boldsymbol{f}_{i,j}^n \right), \varphi_1 \left(\boldsymbol{f}_i^n - \boldsymbol{f}_{i,j}^n \right) \right) \right\}_{j=1}^{s_l}, \quad (3)$$

where $\boldsymbol{f}_{i,j}^n$ is the neighbor feature of the feature $\boldsymbol{f}_i^n$, φ_1 is the MLP layer, ϕ_1 is the skip-connection layer, and s_l represents the scale of k-NN with $l = 1, 2$ in default. Moreover, we use an *Attentional Feature Fusion* (AFF) architecture to aggregate the features which can benefit both the small and large scales. The AFF can be formulated as

$$M \left(\boldsymbol{f}_i^{s1}, \boldsymbol{f}_i^{s2} \right) = \mathrm{sigmoid} \left(\varphi_2 \left(\mathrm{AvgPool} \left\{ \boldsymbol{f}_i^{s1} + \boldsymbol{f}_i^{s2} \right\}_{i=1}^N \right) \right), \quad (4)$$

$$\boldsymbol{f}_i = \varphi_3 \left(\boldsymbol{f}_i^{s1} \cdot M \left(\boldsymbol{f}_i^{s1}, \boldsymbol{f}_i^{s2} \right) + \boldsymbol{f}_i^{s2} \cdot \left(1 - M \left(\boldsymbol{f}_i^{s1}, \boldsymbol{f}_i^{s2} \right) \right) \right), \quad (5)$$

where $\boldsymbol{f}_i^{s1}$ abd $\boldsymbol{f}_i^{s2}$ are the local structures with different scales of feature $\boldsymbol{f}_i$, φ_2 and φ_3 are the MLP layers, N represents the cardinality of the input point cloud patch.

Hierarchical Geometric Information Fusion. Recent approaches have proven the effectiveness of multi-scale global feature extraction (Qi et al. 2017b; Li et al. 2022b; Qin et al. 2022), however, large scale global information and local structures may be lost after point cloud downsampling. To alleviate this problem, as shown in Fig. 3(c), we propose a *hierarchical architecture* that combines the multi-scale global features with the local structures. During the Hierarchical Geometric Information Fusion, the global feature $\boldsymbol{G}_{N_h}$ of current scale N_h can be formulated as

$$\boldsymbol{G}_{N_h} = \varphi_5 \left(\mathrm{MaxPool} \left\{ \varphi_4 \left(\boldsymbol{f}_i^{N_h} \right) \right\}_{i=1}^{N_h} \right), \quad (6)$$

Table 1: Quantitative comparisons in terms of RMSE and CND on the PCPNet dataset. **Bold** values indicate the best estimator.

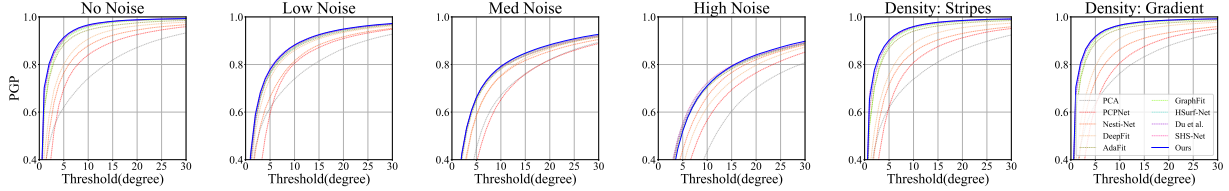| Method	RMSE							CND						
	Noise (σ)				Density		Ave.	Noise (σ)				Density		Ave.
	None	0.12%	0.6%	1.2%	Stripes	Gradient		None	0.12%	0.6%	1.2%	Stripes	Gradient	
PCA (Hoppe et al. 1992)	12.28	12.86	18.40	27.61	13.63	12.79	16.26	12.28	12.78	16.41	24.46	13.63	12.79	15.39
n-jet (Cazals and Pouget 2005)	12.32	12.82	18.34	27.77	13.36	13.09	16.29	12.32	12.77	16.36	24.67	13.36	13.09	15.43
PCPNet (Guerrero et al. 2018)	9.62	11.36	18.89	23.32	11.15	11.69	14.34	9.62	11.23	17.28	20.16	11.15	11.69	13.52
Nesti-Net (Ben-Shabat et al. 2019)	8.43	10.72	17.56	22.63	10.20	10.66	13.37	8.43	10.57	15.00	18.16	10.20	10.66	12.17
DeepFit (Ben-Shabat et al. 2020)	6.51	9.21	16.73	23.12	7.93	7.31	11.80	6.51	8.98	13.98	19.00	7.93	7.31	10.62
AdaFit (Zhu et al. 2021)	5.21	9.05	16.44	21.94	6.01	5.90	10.76	5.21	8.79	13.55	17.31	6.01	5.90	9.46
GraphFit (Li et al. 2022a)	4.49	8.69	16.04	21.64	5.40	5.20	10.24	4.49	8.43	13.00	16.93	5.40	5.20	8.91
HSurf-Net (Li et al. 2022b)	4.17	8.78	16.25	21.61	4.98	4.86	10.11	4.17	8.52	13.23	16.72	4.98	4.86	8.75
Du *et al.* (Du et al. 2023)	4.11	8.66	**16.02**	21.57	4.89	4.83	10.01	4.11	8.43	13.10	17.08	4.89	4.83	8.74
SHS-Net (Li et al. 2023a)	3.95	8.55	16.13	**21.53**	4.91	4.67	9.96	3.95	8.29	13.13	16.60	4.91	4.67	8.59
Ours	**3.86**	**8.45**	16.08	21.89	**4.85**	**4.45**	**9.93**	**3.86**	**8.13**	**12.55**	**16.23**	**4.85**	**4.45**	**8.35**

Figure 4: AUC on the PCPNet dataset. X and Y axes are the angle threshold and the percentage of good point (PGP) normals.

where φ_4 and φ_5 are MLP layers. Meanwhile, the local structures $\boldsymbol{g}_i^{N_h+1}$ are captured by

$$\boldsymbol{g}_i^{N_h+1} = \mathrm{MaxPool}\left\{\varphi_6\left(\boldsymbol{g}_{i,j}^{N_h}\right)\right\}_{j=1}^{s} + \boldsymbol{g}_i^{N_h}, i = 1, ..., N_{h+1}, \quad (7)$$

where $\boldsymbol{g}_{i,j}^{N_h}$ is the neighborhood feature of point $\boldsymbol{p}_i$ in the scope of the scale N_{h+1}, s is the scale of the neighborhood features, and φ_6 represents the MLP layer. Then, we downsample the patch by decreasing the patch size. Moreover, we integrate the global features of the current scale and the last scale with the local structures by

$$\boldsymbol{f}_i^{N_h+1} = \varphi_7\left(\boldsymbol{G}_{N_h}, \boldsymbol{G}_{N_{h-1}}, \boldsymbol{g}_i^{N_h+1}\right) + \boldsymbol{f}_i^{N_h}, i = 1, ..., N_{h+1}, \quad (8)$$

where φ_7 is the MLP layer, and $N_{h+1} \leq N_h \leq N_{h-1}$.

Decoder. Note that the point coordinates are important basic attributes for point cloud processing and the spatial relationship between them such as distance can guide the inference process of the network (Zhao et al. 2021; Zhang et al. 2022). To explore this idea, we introduce two modules including *Position Feature Fusion* (PFF) and *Weighted Normal Prediction* (WNP) into the decoder part. As shown in Fig. 3(d), during the PFF, we embed the neighborhood coordinates of each point and fuse them with the extracted feature by skip-connections, which can be formulated as

$$\boldsymbol{F}_i = \mathrm{MaxPool}\left\{\phi_2\left(\boldsymbol{f}_i, \boldsymbol{p}_{i,j} - \boldsymbol{p}_i, \varphi_8\left(\boldsymbol{p}_{i,j} - \boldsymbol{p}_i\right)\right)\right\}_{j=1}^{s}, \quad (9)$$

where $\boldsymbol{p}_{i,j}$ is the neighbor coordinate of the point $\boldsymbol{p}_i$, $\boldsymbol{f}_i$ is the extracted feature of $\boldsymbol{p}_i$, s represents the neighborhood scale, φ_8 is the MLP layer and ϕ_2 is the skip-connection. As shown in Fig. 3(e), we predict weights based on the geometry information of each point and use the weighted features to predict the normal vector of the query point:

$$\boldsymbol{n} = \varphi_{11}\left(\mathrm{MaxPool}\left\{\varphi_{10}\left(\boldsymbol{F}_i \cdot \mathrm{softmax}_M\left(\varphi_9\left(\boldsymbol{F}_i\right)\right)\right)\right\}_{i=1}^{M}\right), \quad (10)$$

where φ_9, φ_{10} and φ_{11} are the MLP layers, and the normalized $\boldsymbol{n}$ is the finally predicted unit normal vector.

Loss function. To bridge the gap between the annotated normal and the noise-caused neighborhood geometry variation of the query point, we reformulate the sine loss by CND, namely, taking the normal $\boldsymbol{n}_{\tilde{\boldsymbol{p}}}$ of the nearest neighbor point $\tilde{\boldsymbol{p}}$ in the corresponding noise-free point cloud $\tilde{\mathcal{P}}$ as the ground truth

$$\mathcal{L}_1 = \|\boldsymbol{n}_{\tilde{\boldsymbol{p}}} \times \hat{\boldsymbol{n}}_{\boldsymbol{p}}\|. \quad (11)$$

Meanwhile, we use the transformation regularization loss and the z-direction transformation loss to constrain the output rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ of the QSTN (Du et al. 2023)

$$\mathcal{L}_2 = \left\|\boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^{\mathrm{T}}\right\|^2, \quad (12)$$

$$\mathcal{L}_3 = \|\boldsymbol{n}_{\tilde{\boldsymbol{p}}}\boldsymbol{R} \times \boldsymbol{z}\|, \quad (13)$$

where $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ represents the identity matrix, $\boldsymbol{z} = (0, 0, 1)$. Additionally, to make full use of the spatial relationships between data points, we adopt the weight loss similar to Zhang *et al.* (2022)

$$\mathcal{L}_4 = \frac{1}{M}\sum_{i=1}^{M}(w_i - \hat{w}_i)^2, \quad (14)$$

where $\hat{w}$ are the predicted weights for each data point, M represents the cardinality of the downsampled patch, $w_i = \exp(-(\boldsymbol{p}_i \cdot \boldsymbol{n}_{\tilde{\boldsymbol{p}}})^2/\delta^2)$ and $\delta = \max\left(0.05^2, 0.3\sum_{i=1}^{M}(\boldsymbol{p}_i \cdot \boldsymbol{n}_{\tilde{\boldsymbol{p}}})^2/M\right)$, where $\boldsymbol{p}_i$ is the point in the downsampled patch. Therefore, our final loss function is defined as

$$\mathcal{L} = \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_2 + \lambda_3\mathcal{L}_3 + \lambda_4\mathcal{L}_4, \quad (15)$$

where $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, $\lambda_3 = 0.5$, and $\lambda_4 = 1$ are weighting factors.

Table 2: Quantitative comparisons of CND on the PCPNet dataset with gradually increased noise.

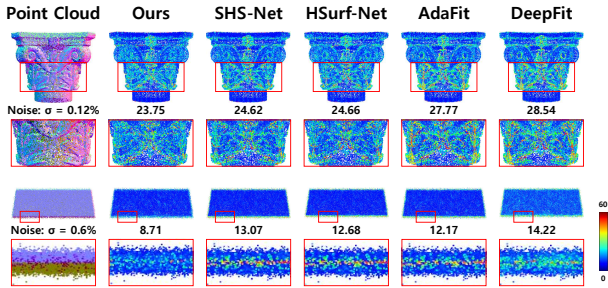| Method	Noise (σ)					Ave.
	0.125%	0.25%	0.5%	0.75%	1.25%	
PCA	14.46	15.09	17.75	21.40	31.81	20.10
n-jet	14.40	14.98	17.67	21.50	32.08	20.12
PCPNet	13.42	15.52	18.12	20.02	23.92	18.20
Nesti-Net	13.34	14.33	16.63	18.34	22.31	16.99
DeepFit	11.70	12.83	15.62	17.64	24.01	16.36
AdaFit	11.42	12.95	15.52	17.02	21.74	15.73
GraphFit	11.01	**12.40**	15.05	16.66	20.56	15.14
HSurf-Net	11.04	12.67	15.33	16.79	20.67	15.30
Du *et al.*	10.97	12.48	15.17	16.77	21.04	15.29
SHS-Net	10.90	12.66	15.18	16.59	20.89	15.24
Ours	**10.60**	12.56	**14.89**	**16.31**	**19.62**	**14.79**

Figure 5: Comparisons on the PCPNet datsts (Noise: $\sigma = 0.12\%, 0.6\%$). We use the heat map to visualize the CND error.

5 Experimental Results

Datasets. As predecessor approaches, we first adopt the synthetic dataset PCPNet (Guerrero et al. 2018) for comparison, in which we follow the same experimental setups including train-test split, adding noise, and changing distribution density on test data. To test the generalization capability of our method, we then evaluate and compare the models trained on the PCPNet on the real-world indoor SceneNN dataset (Hua, Tran, and Yeung 2018).

Implementation details. We set the input patch size $N = 700$ and the downsampling factors $\rho = \{2/3, 2/3, 2/3, 1\}$. The scales of k-NN in the LFE are equivalent to 16 and 32, and $s = \{32, 32, 16, 16\}$ in the Hierarchical Geometric Information Fusion. The number of the neighbor points during the PPF is 16. We adopt the AdamW (Loshchilov and Hutter 2017) optimizer with initial learning rate 5×10^{-4} for training. The learning rate is decayed by a cosine function. Our model is trained with a 64 batch size on an NVIDIA A100 GPU in 900 epochs. More implementation details are reported in *Supplementary Materials (SM)*.

Evaluation. We adopt the proposed CND metric to assess the normal estimation results and compare it with the RMSE. Moreover, we use the *Area Under the Curve* (AUC) metric to analyze the error distribution of the predicted normals. AUC is attained by the *Percentage of Good Points* (PGP) metric, which measures the percentage of normal vectors with errors below different angle thresholds.

Table 3: Statistical CND results on the SceneNN dataset.

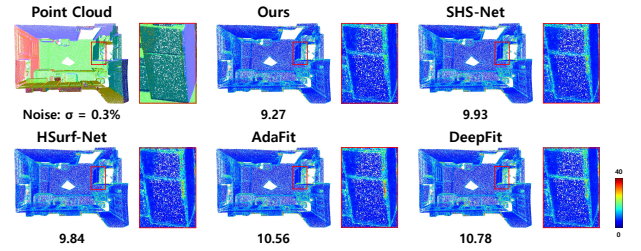| Method	Ours	SHS-Net	Du *et al.*	HSurf-Net	GrapFit	AdaFit	DeepFit
Clean	6.92	7.20	6.97	**6.73**	7.38	7.55	9.46
Noise	**10.82**	11.30	10.94	11.30	11.38	11.82	12.27
Ave.	**8.87**	9.25	8.96	9.02	9.38	9.97	10.86

Figure 6: Qualitative comparisons on the SceneNN datasets (Noise: $\sigma = 0.3\%$).

5.1 Results on Synthetic Data

PCPNet. Table 1 reports the statistical results of all compared approaches on the PCPNet dataset, measured in terms of both RMSE and CND metrics. As observed, our method achieves the overall highest normal estimation accuracy across different scenarios, particularly in scenarios with noise. In comparison to RMSE, the CND metric allows for more accurate and faithful prediction evaluations while mitigating the annotation inconsistency. Additionally, the AUC results of the CND metric are illustrated in Fig. 4, where our method still showcases the superior performance, suggesting its remarkable stability across different angular thresholds. Qualitative comparison results are presented in Fig. 5. Notably, our method exhibits the smallest errors in regions characterized by noise and intricate geometry.

Robustness to noise. Subsequently, we specifically employ five representative models from the PCPNet dataset to assess the robustness against noise. We introduce varying levels of noise to these data which encompass one CAD model and four scanned point clouds. The quantitative outcomes displayed in Table 2 indicate that our method exhibits superior performance compared to competitors, particularly in scenarios contaminated by high levels of noise.

5.2 Generalization to Real-world Data

Next, we investigate the generalization capability using the real-world indoor SceneNN dataset. Results in Table 3 suggest that our method has the highest normal estimation accuracy in an average sense. The qualitative results presented in Fig. 6 exhibit our superiority. It is noticeable that our method successfully preserves more geometric details, such as the handle of the refrigerators. Additionally, more results on different real-word datasets can be found in *SM*.

5.3 Ablation Study

CND-modified loss function. To demonstrate the effectiveness and generalization of the newly introduced CND-modified loss function, we conduct experiments on the PCPNet dataset, comparing the results with and without its in-

Table 4: Network training with or without the CND-modified loss function on the PCPNet dataset.

Method	Ours		HSurf-Net		DeepFit		PCPNet	
$\mathcal{L}_{CND}$	✓		✓		✓		✓	
No Noise	3.86	3.85	4.24	4.17	6.53	6.51	**8.52**	9.62
Noise: ($\sigma = 0.12\%$)	**8.13**	8.23	**8.50**	8.52	**8.77**	8.98	**10.40**	11.23
Noise: ($\sigma = 0.6\%$)	**12.55**	12.76	**12.83**	13.22	**13.66**	13.98	**15.71**	17.28
Noise: ($\sigma = 1.2\%$)	**16.23**	16.46	**16.47**	16.71	**18.69**	19.00	**18.31**	20.16
Density: Stripes	4.85	4.65	5.18	4.98	7.95	7.93	**9.96**	11.15
Density: Gradients	**4.45**	4.51	4.94	4.86	**7.31**	7.31	**10.25**	11.69
Ave.	**8.35**	8.41	**8.69**	8.75	**10.48**	10.62	**12.19**	13.52

Table 5: Ablation studies with the (a) multi-scale local feature aggregation; (b) hierarchical architecture; (c) decoder; (d) QSTN and (e) scale selection and downsampling factor.

	Category	Noise (σ)				Density		Ave.
		None	0.12%	0.6%	1.2%	Stripes	Gradient	
(a)	w/o Local Feature Extration (LFE)	4.05	8.23	12.76	16.45	4.93	4.68	8.52
	w/ Single-scale Local Feature Extration	3.98	8.20	12.76	16.40	4.96	4.66	8.50
	w/o Attentional Feature Fusion (AFF)	3.96	8.19	12.68	16.39	4.78	4.62	8.44
(b)	w/o Hierarchical Architecture	3.88	8.45	13.80	18.93	4.87	4.50	9.07
	w/o Muli-scale Global Feature	3.87	8.27	12.56	16.24	4.94	**4.45**	8.39
	w/o Local Feature	3.98	8.46	12.68	16.21	5.00	4.60	8.49
(c)	w/o Position Feature Fusion (PFF)	3.93	8.15	12.62	16.24	4.87	4.68	8.42
	w/o Weighted Normal Prediction (WNP)	4.32	8.23	12.56	16.22	5.01	4.89	8.54
(d)	w/o QSTN	4.04	8.34	12.67	16.41	4.95	4.74	8.53
	w/o Z-direction Transformation Loss	4.02	8.18	12.62	16.32	4.98	4.72	8.47
(e)	$N = 600$	3.90	8.36	12.59	16.35	4.78	4.46	8.41
	$N = 800$	4.05	8.24	12.52	**16.17**	4.99	4.65	8.44
	$\rho = \{1/3, 1/3, 1, 1\}$	3.95	8.25	12.56	16.39	**4.76**	4.61	8.42
	$\rho = \{1/2, 1/2, 1, 1\}$	3.93	8.22	12.72	16.29	4.80	4.50	8.41
	Ours	**3.86**	**8.13**	**12.55**	16.23	4.85	**4.45**	**8.35**

corporation. We employ representative methods, including the deep surface fitting method DeepFit (Ben-Shabat et al. 2020), as well as the regression methods PCPNet (Guerrero et al. 2018), Hsurf-Net (Li et al. 2022b), and Ours. Table 4 highlights the impact of the CND component, demonstrating its significant enhancement in normal estimation accuracy for both deep surface fitting and regression methods.

Network architecture. CMG-Net comprises three key components: Multi-scale Local Feature Aggregation, Hierarchical Geometric Information Fusion, and Decoder. We delve into the functions of them on the PCPNet dataset.

(1). In the Multi-scale Local Feature Aggregation, we capture the local structure using two scales and integrate them by AFF. Table 5(a) reports the results of 1) without LFE; 2) with single-scale LFE, and 3) integrating multi-scale local features directly by MLP instead of AFF. As observed, compared with Ours, the multi-scale local features with AFF can effectively improve the network performance.

(2). To validate the effectiveness of the Hierarchical Geometric Information Fusion, we carry out experiments using the model with a fixed global scale that is equivalent to the output scale of CMG-Net. Additionally, we compare the results of the models without the global feature of the last scale and the local feature in the hierarchical architecture. Results shown in Table 5(b) demonstrate that the Hierarchical Geometric Information Fusion operation can also boost the normal estimation performance.

(3). Table 5(c) shows the ablation studies of the Decoder part, suggesting the effectiveness of PFF and WNP.

(4). We also investigate the functionality of QSTN, the input patch sizes N, and the downsampling factors ρ in Table 5(d) and Table 5(e), where quantitative results validate their usefulness in our method.
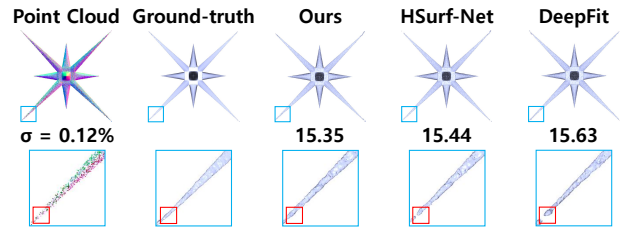

Figure 7: Comparisons on Poisson surface reconstruction.

5.4 Application of the Proposed Method

We also demonstrate the application of our method on downstream tasks. Fig. 7 presents the *Poisson surface reconstruction* (Kazhdan, Bolitho, and Hoppe 2006) results using the normal vectors predicted by competing approaches. Compared with ground-truth surfaces, our method achieves the best reconstruction quality (quantified by the *Symmetric Mean Hausdorff Distance* (SMD)($\times 10^{-4}$)), especially in shape details of noisy regions, underscoring the higher accuracy of our normal estimation. We provide more reconstruction instances and highlight the application of our newly developed method to point cloud denoising in the *SM*.

6 Limitations

While our method has demonstrated remarkable normal estimation accuracy across diverse 3D models, especially in noisy scenarios, it is not yet real-time capable and still depends on annotated training data, as is the case with previous approaches. Therefore, it is highly desirable in the future to reduce the computation time and delve into semi-supervised or unsupervised normal estimation frameworks.

7 Conclusions

We propose a novel method for robust normal estimation in unorganized point clouds, which shows superiority across various datasets and scenarios. We identify the issue of direction inconsistency in predecessor approaches and introduce the CND metric to address this concern. This not only boosts the network training and evaluation, but also greatly enhances the network robustness against noisy disturbance. Additionally, we design an innovative architecture that combines multi-scale local and global feature extraction with hierarchical information fusion to deal with scale selection ambiguity. Extensive experiments validate that our method outperforms competitors in terms of both accuracy and robustness for normal estimation. Moreover, we demonstrate its ability to generalize in real-world settings and downstream application tasks.

Acknowledgements

References

Alliez, P.; Cohen-Steiner, D.; Tong, Y.; and Desbrun, M. 2007. Voronoi-based variational reconstruction of unoriented point sets. In *Proc. Symp. Geom. Process.*, 39–48.

Amenta, N.; and Bern, M. 1998. Surface reconstruction by Voronoi filtering. In *Proc. Ann. Symp. Comput. Geom.*, 39–48.

Aroudj, S.; Seemann, P.; Langguth, F.; Guthe, S.; and Goesele, M. 2017. Visibility-consistent thin surface reconstruction using multi-scale kernels. *ACM Trans. Graph.*, 36(6): 1–13.

Ben-Shabat; et al. 2020. DeepFit: 3D surface fitting via neural network weighted least squares. In *Proc. Eur. Conf. Comput. Vis.*, 20–34.

Ben-Shabat, Y.; et al. 2019. Nesti-Net: Normal estimation for unstructured 3D point clouds using convolutional neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 10112–10120.

Boulch, A.; and Marlet, R. 2012. Fast and robust normal estimation for point clouds with sharp features. *Comput. Graph. Forum*, 31(5): 1765–1774.

Boulch, A.; and Marlet, R. 2016. Deep learning for robust normal estimation in unstructured point clouds. *Comput. Graph. Forum*, 35(5): 281–290.

Cao, J.; Zhu, H.; Bai, Y.; Zhou, J.; Pan, J.; and Su, Z. 2021. Latent tangent space representation for normal estimation. *IEEE Trans. Ind. Electron.*, 69(1): 921–929.

Cazals, F.; and Pouget, M. 2005. Estimating differential quantities using polynomial fitting of osculating jets. *Comput. Aided. Geom. Des.*, 22(2): 121–146.

Che, E.; and Olsen, M. J. 2018. Multi-scan segmentation of terrestrial laser scanning data based on normal variation analysis. *ISPRS J. PhotoGramm.*, 143: 233–248.

Dong, Z.; Liang, F.; Yang, B.; Xu, Y.; Zang, Y.; Li, J.; Wang, Y.; Dai, W.; Fan, H.; Hyyppä, J.; et al. 2020. Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. *ISPRS J. PhotoGramm.*, 163: 327–342.

Du, H.; Yan, X.; Wang, J.; Xie, D.; and Pu, S. 2023. Rethinking the approximation error in 3d surface fitting for point cloud normal estimation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 9486–9495.

Fleishman, S.; Cohen-Or, D.; and Silva, C. T. 2005. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3): 544–552.

Grilli, E.; Menna, F.; and Remondino, F. 2017. A review of point clouds segmentation and classification algorithms. *Int. arch. photogramm. remote sens. spat. inf. sci.*, 42: 339–344.

Guennebaud, G.; and Gross, M. 2007. Algebraic point set surfaces. *ACM Trans. Graph.*, 26: 23–es.

Guerrero, P.; Kleiman, Y.; Ovsjanikov, M.; and Mitra, N. J. 2018. Pcpnet learning local shape properties from raw point clouds. *Comput. Graph. Forum*, 37(2): 75–85.

Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J. D.; Schindler, K.; and Pollefeys, M. 2017. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Ann. Photogram., Remote Sens. Spatial Inf. Sci.*, volume IV-1-W1, 91–98.

Hashimoto, T.; and Saito, M. 2019. Normal Estimation for Accurate 3D Mesh Reconstruction with Point Cloud Model Incorporating Spatial Structure. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 54–63.

Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; and Stuetzle, W. 1992. Surface reconstruction from unorganized points. In *Proc. Ann. Conf. Comput. Graph. Interact. Tech.*, 71–78.

Hua, B.-S.; Tran, M.-K.; and Yeung, S.-K. 2018. Pointwise Convolutional Neural Networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 984–993.

Kazhdan, M.; Bolitho, M.; and Hoppe, H. 2006. Poisson surface reconstruction. In *Proc. Symp. Geom. Process.*, volume 7.

Lenssen, J. E.; Osendorfer, C.; and Masci, J. 2020. Deep iterative surface normal estimation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 11247–11256.

Levin, D. 1998. The approximation power of moving least-squares. *Math. Comput.*, 67(224): 1517–1531.

Li, K.; Zhao, M.; Wu, H.; Yan, D.-M.; Shen, Z.; Wang, F.-Y.; and Xiong, G. 2022a. Graphfit: Learning multi-scale graph-convolutional representation for point cloud normal estimation. In *Proc. Eur. Conf. Comput. Vis.*, 651–667.

Li, Q.; Feng, H.; Shi, K.; Gao, Y.; Fang, Y.; Liu, Y.-S.; and Han, Z. 2023a. SHS-net: Learning signed hyper surfaces for oriented normal estimation of point clouds. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 13591–13600.

Li, Q.; Liu, Y.-S.; Cheng, J.-S.; Wang, C.; Fang, Y.; and Han, Z. 2022b. HSurf-Net: Normal estimation for 3D point clouds by learning hyper surfaces. In *Proc. Int. Conf. Neural Inf. Process. Syst.*, volume 35, 4218–4230.

Li, S.; Zhou, J.; Ma, B.; Liu, Y.-S.; and Han, Z. 2023b. Neaf: Learning neural angle fields for point normal estimation. In *Proc. AAAI Conf. Artif. Intell.*, volume 37, 1396–1404.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. arXiv:1711.05101.

Lu, D.; Lu, X.; Sun, Y.; and Wang, J. 2020a. Deep feature-preserving normal estimation for point cloud filtering. *Comput. Aided. Des.*, 125: 102860.

Lu, X.; Schaefer, S.; Luo, J.; Ma, L.; and He, Y. 2020b. Low rank matrix approximation for 3D geometry filtering. *IEEE Trans. Vis. Comput. Graph.*, 28(4): 1835–1847.

Mérigot, Q.; Ovsjanikov, M.; and Guibas, L. J. 2010. Voronoi-based curvature and feature estimation from point clouds. *IEEE Trans. Vis. Comput. Graph.*, 17(6): 743–756.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proc. Int. Conf. Neural Inf. Process. Syst.*, volume 30.

Qin, Z.; Yu, H.; Wang, C.; Guo, Y.; Peng, Y.; and Xu, K. 2022. Geometric transformer for fast and robust point cloud registration. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 11143–11152.

Wang, Z.; and Prisacariu, V. A. 2020. Neighbourhood-insensitive point cloud normal estimation network. In *Proc. Brit. Mach. Vis. Conf.*

Zhang, D.; Lu, X.; Qin, H.; and He, Y. 2020. Pointfilter: Point cloud filtering via encoder-decoder modeling. *IEEE Trans. Vis. Comput. Graph.*, 27(3): 2015–2027.

Zhang, J.; Cao, J.; Liu, X.; Chen, H.; Li, B.; and Liu, L. 2018. Multi-normal estimation via pair consistency voting. *IEEE Trans. Vis. Comput. Graph.*, 25(4): 1693–1706.

Zhang, J.; Cao, J.-J.; Zhu, H.-R.; Yan, D.-M.; and Liu, X.-P. 2022. Geometry Guided Deep Surface Normal Estimation. *Comput. Aided. Des.*, 142: 103119.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proc. IEEE Int. Conf. Comput. Vis.*, 16259–16268.

Zhou, H.; Chen, H.; Feng, Y.; Wang, Q.; Qin, J.; Xie, H.; Wang, F. L.; Wei, M.; and Wang, J. 2020. Geometry and learning co-supported normal estimation for unstructured point cloud. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 13238–13247.

Zhou, H.; Chen, H.; Zhang, Y.; Wei, M.; Xie, H.; Wang, J.; Lu, T.; Qin, J.; and Zhang, X.-P. 2022a. Refine-Net: Normal Refinement Neural Network for Noisy Point Clouds. *IEEE Trans. Pattern Anal. Mach. Intell.*

Zhou, J.; Jin, W.; Wang, M.; Liu, X.; Li, Z.; and Liu, Z. 2022b. Fast and accurate normal estimation for point clouds via patch stitching. *Comput. Aided. Des.*, 142: 103121.

Zhu, R.; Liu, Y.; Dong, Z.; Wang, Y.; Jiang, T.; Wang, W.; and Yang, B. 2021. AdaFit: Rethinking Learning-based Normal Estimation on Point Clouds. In *Proc. IEEE Int. Conf. Comput. Vis.*, 6118–6127.

Supplementary Materials

In this exposition, we present more implementation details, experimental comparisons, and qualitative results to support our work. Concretely, the following content comprising of

1. implementation details with respect to the local features in Hierarchical Geometric Information Fusion and the QSTN structure,

2. explanation of the difference between CND and Mean Square Angular Error (MSAE)(Lu et al. 2020a),

3. thorough comparisons with the results of normal estimation using denoising pre-processing,

4. more quantitative and qualitative results on real-word datasets and point clouds contaminated by heavier noise,

5. employment of our method to real-time application and downstream tasks containing *surface reconstruction* and *denoising*

are reported.

A More Implementation Details

In this section, we provide more implementation details of our method, especially the local features in Hierarchical Geometric Information Fusion and the QSTN structure are reported.

A.1 Local Features in Hierarchical Architecture

To capture the local structures of both geometry and semantics in the Hierarchical Geometric Information Fusion module, we leverage various local features in the odd and even hierarchical layers. The local features of the odd hierarchical layers are defined by

$$g_{i,j}^o = \text{Concat}\left(p_i, p_i - p_{i,j}, \varphi\left(p_i - p_{i,j}\right)\right), \quad (16)$$

where $p_{i,j}$ is the neighbor coordinate of the point p_i and φ represents the MLP layer. Concurrently, the even ones put more focus on semantic feature defined as

$$g_{i,j}^e = \text{Concat}\left(p_i, p_i - p_{i,j}, f_i - f_{i,j}\right), \quad (17)$$

where f_i and $f_{i,j}$ are the semantic features of p_i and its neighborhood.

In addition, to validate the effectiveness of our newly proposed feature extraction method, we conduct ablation studies on the models

1. without local features;

2. with local feature in Eq. 16 only;

3. with local feature in Eq. 17 only.

The results on the PCPNet dataset presented in Table 6 demonstrate that different local features in the odd and even hierarchical layers can significantly enhance the performance on normal estimation.

A.2 QSTN Structure

Given a point cloud patch $P = \{p_i \in \mathbb{R}^3\}_{i=1}^N$, the QSTN part (Qi et al. 2017a; Du et al. 2023) first computes the quaternion by MLP layers and then translates this quaternion into a rotation matrix $R \in \mathbb{R}^{3\times3}$, as shown in Fig. 8.

Table 6: Ablation studies on the local features in the Hierarchical Geometric Information Fusion.

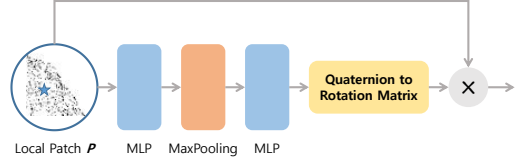| Method	Ours	(3)	(2)	(1)
No Noise	**3.86**	3.91	3.89	3.98
Noise: ($\sigma = 0.12\%$)	**8.13**	8.22	8.18	8.46
Noise: ($\sigma = 0.6\%$)	**12.55**	12.64	12.56	12.68
Noise: ($\sigma = 1.2\%$)	**16.23**	16.28	16.25	16.24
Density: Stripes	**4.85**	4.95	4.93	5.00
Density: Gradients	**4.45**	4.62	4.71	4.60
Average	**8.35**	8.44	8.42	8.49

Figure 8: Architecture of QSTN.

B More Normal Estimation Results

In this section, we further explain the difference between CND and MSAE, a metric used to evaluate the denoising results of CAD models and compare our method with the normal estimator with denoising pre-processing (Zhang et al. 2020). Additionally, more qualitative results on the point clouds contaminated by heavier noise and LiDAR datasets are provided.

B.1 Differences between CND and MSAE Metrics

As a metric for denoising, MSAE searches nearby ground-truth points' normals and picks the minimal normal error. As shown in Fig. 9, our closest distance-induced CND metric effectively addresses direction inconsistency caused by noise, distinguishing it from MSAE, which primarily focuses on normal similarity and selects the neighboring point with the minimal normal error instead of relative coordinates. As shown in Table 7, we set the neighbor points size in MSAE to 4 and train the same network by the CND-loss and MSAE-loss respectively, while our CND-loss consistently attains superior results on both metrics in the test phase.

B.2 Comparisons with Denoising Pre-Processing

To further prove the effectiveness of our proposed CND-Modified loss function, we conduct experiments on the models trained with or without CND-Modification as well as the
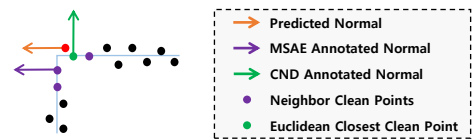

Figure 9: Differences between the CND and MSAE metrics when wrong normal predictions on noisy edge points occur.

Table 7: Test results of CND and MSAE on the PCPNet dataset.

Category	MSAE						
	None	Noise σ			Density		Averge
		0.125%	0.6%	1.2%	Stripes	Gradient	
L MSAE	4.12	6.69	9.41	12.82	5.45	5.22	7.29
L CND (ours)	**2.65**	**5.34**	**9.23**	**12.74**	**3.29**	**2.99**	**6.04**
Category	CND						
	None	Noise σ			Density		Averge
		0.125%	0.6%	1.2%	Stripes	Gradient	
L MSAE	4.66	10.62	13.37	16.31	5.63	5.48	9.35
L CND (ours)	**3.86**	**8.13**	**12.55**	**16.23**	**4.85**	**4.45**	**8.35**

Table 8: Results of the comparisons with the denosing pre-processing on the noisy part of the PCPNet Dataset.

Method	ours	w/o $\mathcal{L}_{CND}$	w/ denoising
Noise: ($\sigma = 0.12\%$)	**8.13**	8.23	13.79
Noise: ($\sigma = 0.6\%$)	**12.55**	12.76	16.28
Noise: ($\sigma = 1.2\%$)	**16.23**	16.46	17.91
Average	**12.30**	12.48	15.99

model with denoising pre-processing. All of the compared models have the same architecture. The results in Table 8 indicate that due to the destruction of geometrical information and the oversmoothing of shape details, the denoising pre-processing decreases the accuracy of normal estimation instead. In contrast, the CND modifies the annotated normal of the noisy points faithfully, and thus substantially improves the network robustness against noise without the loss of any shape details.

B.3 Qualitative Results

We present more quantitative and qualitative results on the real-word outdoor Semantic3D dataset (Hackel et al. 2017) and the LiDAR WHU-TLS dataset (Dong et al. 2020) in Fig. 10 and Fig. 11, and point clouds with heavier noise in Fig. 12 and Fig. 13 to demonstrate the better performance and generalization of our proposed method.

B.4 Ablation Studies on Real-word Dataset

As reported in Table 9, we have conducted additional ablation studies on the real indoor dataset SceneNN, to further demonstrate the generalization of each proposed component. The models are trained on the PCPNet dataset and share the same setting with the ones in Sec 5.3. Results regarding each component provide further validation of the
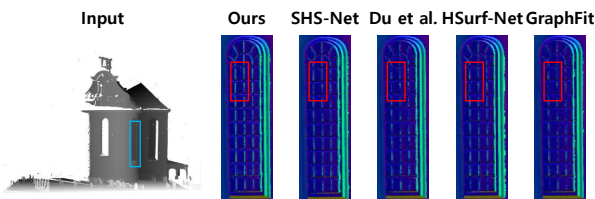

Figure 10: Qualitative comparisons on the Semantic3D dataset, where point normals are mapped to RGB colors.

Table 9: Ablation studies on the realistic dataset SceneNN.

	Category	Clean	Noise	Averge
(a)	w/o Local Feature Extration	7.58	11.18	9.38
	w/ Single-scale Local Feature Extration	7.14	11.01	9.08
	w/o Attentional Feature Fusion (AFF)	7.06	10.86	8.96
(b)	w/o Hierarchical Architecture	7.52	11.25	9.39
	w/o Muli-scale Global Feature	6.98	10.95	8.97
	w/o Local Feature	6.96	11.21	9.08
(c)	w/o Position Feature Fusion (PFF)	7.09	10.86	8.97
	w/o Weighted Normal Prediction (WNP)	7.06	11.09	9.07
(d)	w/o QSTN	7.10	11.10	9.10
	w/o Z-direction Transformation Loss	7.08	11.06	9.07
(e)	w/o CND	6.94	10.96	8.95
	ours	**6.92**	**10.82**	**8.87**

Table 10: Timings on realistic indoor dataset SceneNN with 10K points per scene.

Data (10K)	No.032	No.207	No.032-Noise	No.207-Noise	Ave.
Time (s)	4.8	3.69	3.8	3.76	4.01

technical soundness of our method, suggesting its effectiveness and robustness.

C More Applications

In this section, we provide more results of employing our method to downstream tasks. Both quantitative and qualitative results demonstrate that our method outperforms competitors, in both surface reconstruction (Kazhdan, Bolitho, and Hoppe 2006) and denoising (Lu et al. 2020b) tasks.

C.1 Real-time Application

Timings reported in Table 10 show our acceptable efficiency, which indicate that our method is not yet real-time capable, as stated in Sec. 6.

C.2 Surface Reconstruction

In Fig. 14, we show more mesh models reconstructed using normals predicted by different methods and the corresponding SMD ($\times 10^{-4}$) of reconstructed surfaces. As observed, our method consistently generates accurate reconstruction surfaces on the PCPNet dataset.

C.3 Denoising

In Fig. 15, we present the denoising results of the instances in the PCPNet dataset using normals estimated by competing approaches, along with the CD ($\times 10^{-6}$) and their corresponding reconstructed surfaces. Our method also achieves the best denosing results.
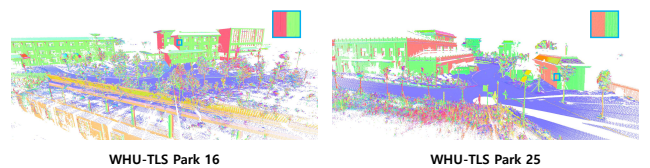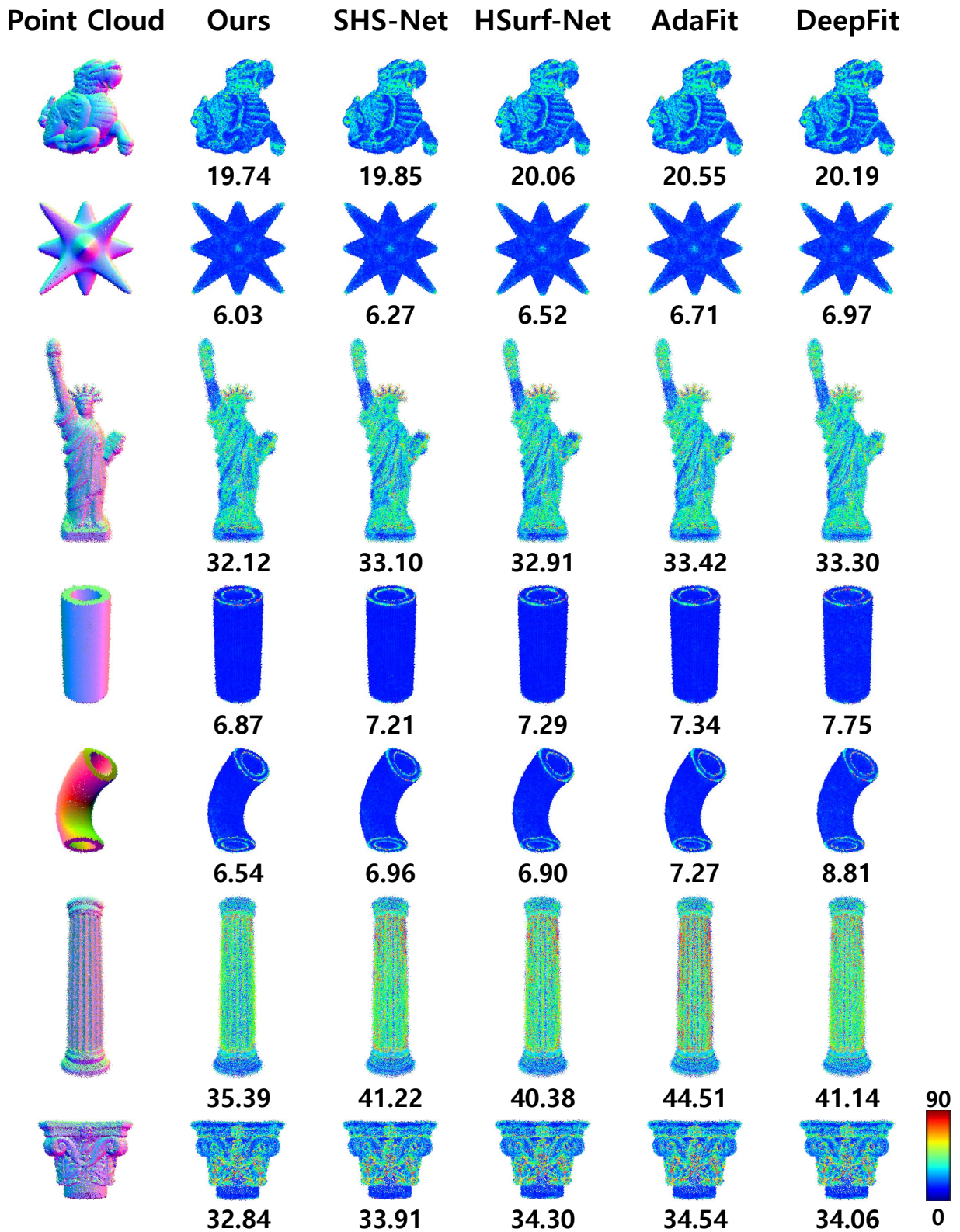

Figure 11: Qualitative results on the WHU-TLS dataset.

Point Cloud	Ours	SHS-Net	HSurf-Net	AdaFit	DeepFit
	19.74	19.85	20.06	20.55	20.19
	6.03	6.27	6.52	6.71	6.97
	32.12	33.10	32.91	33.42	33.30
	6.87	7.21	7.29	7.34	7.75
	6.54	6.96	6.90	7.27	8.81
	35.39	41.22	40.38	44.51	41.14
	32.84	33.91	34.30	34.54	34.06

Figure 12: Comparisons on the point clouds with heavy noise ($\sigma = 0.6\%$) in the PCPNet dataset. We use the heat map to visualize the CND error. Our method achieves the highest accuracy on all models.
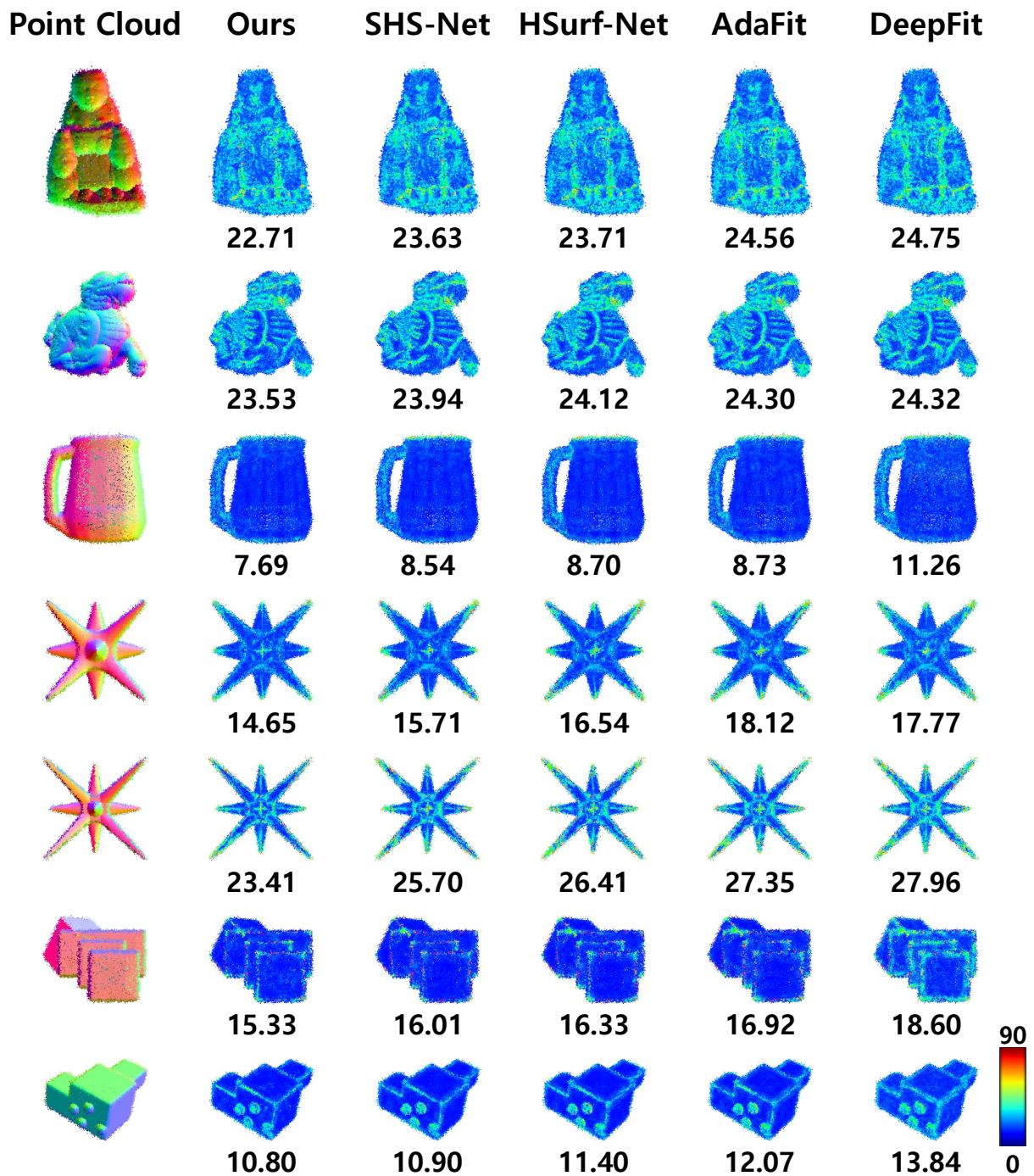
Figure 13: Comparisons on the point clouds significantly affected by heavy noise ($\sigma = 1.2\%$) in the PCPNet dataset. We use the heat map to visualize the CND error. Our method consistently achieves the highest accuracy across all models.
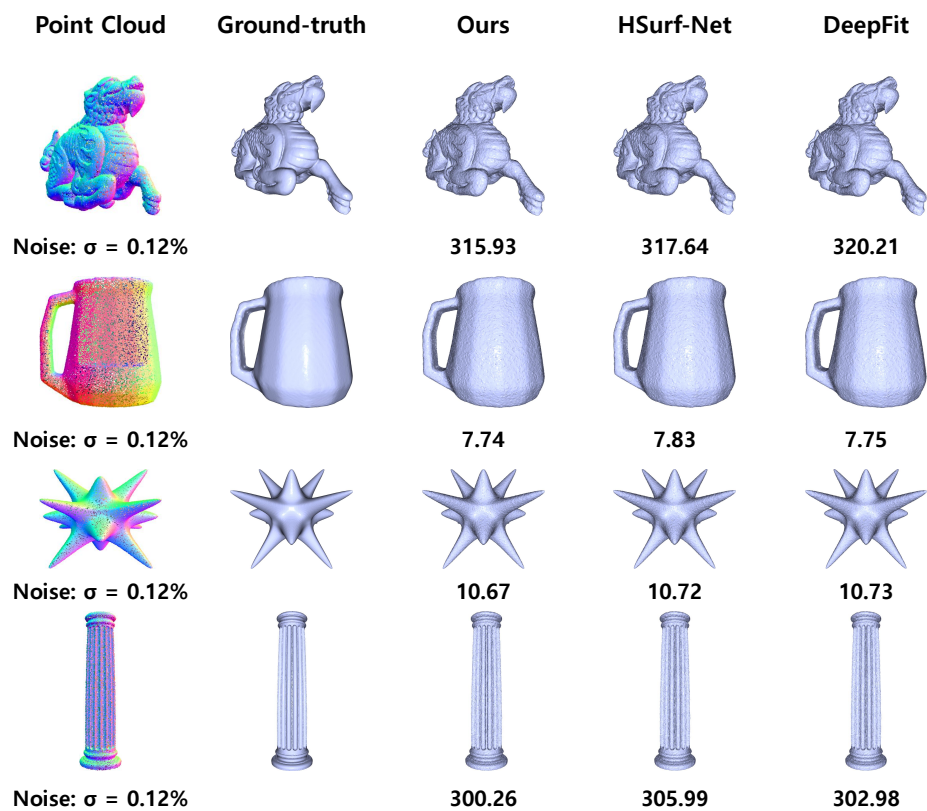
Figure 14: Comparisons on the reconstruction results. Our method achieves the best reconstruction quality.
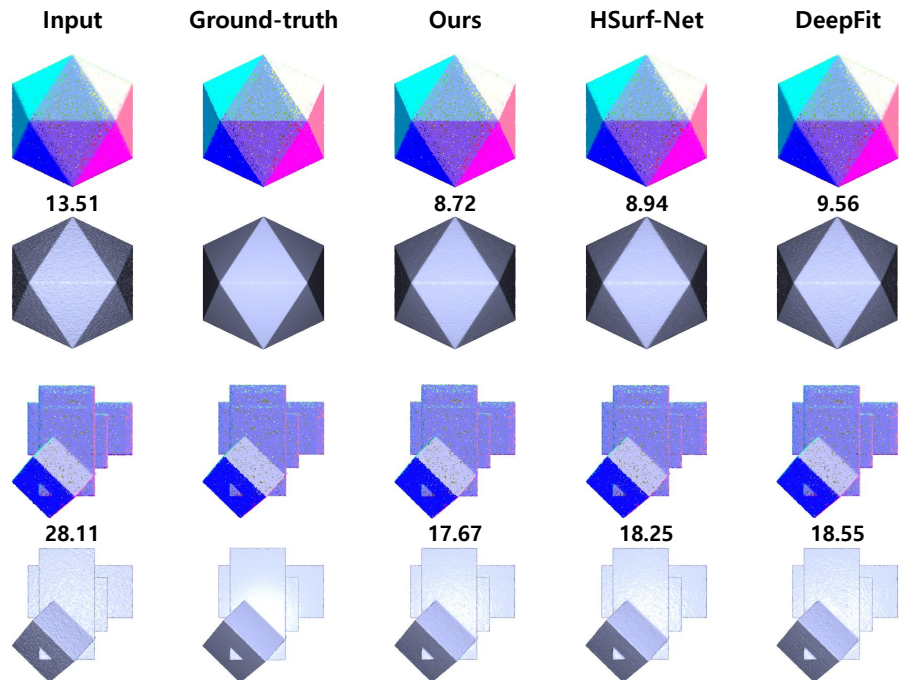

Figure 15: Comparisons on the denoising results. The first row shows the denoised point clouds while the second row shows the corresponding reconstructed suraces. Our method achieves the best denoising results along with high-quality reconstruction surfaces.