Learning Normals of Noisy Points by Local Gradient-Aware Surface Filtering

Anonymous ICCV submission

Paper ID 8136

Abstract

Estimating normals for noisy point clouds is a persistent challenge in 3D geometry processing, particularly for end-to-end oriented normal estimation. Existing methods generally address relatively clean data and rely on supervised priors to fit local surfaces within specific neighborhoods. In this paper, we propose a novel approach for learning normals from noisy point clouds through local gradient-aware surface filtering. Our method projects noisy points onto the underlying surface by utilizing normals and distances derived from an implicit function constrained by local gradients. We start by introducing a distance measurement operator for global surface fitting on noisy data, which integrates projected distances along normals. Following this, we develop an implicit field-based filtering approach for surface point construction, adding projection constraints on these points during filtering. To address issues of over-smoothing and gradient degradation, we further incorporate local gradient consistency constraints, as well as local gradient orientation and aggregation. Comprehensive experiments on normal estimation, surface reconstruction, and point cloud denoising demonstrate the state-of-the-art performance of our method. The source code and trained models will be made publicly available.

1. Introduction

Point clouds are indispensable in 3D computer vision and play a foundational role in applications such as virtual reality, autonomous driving, and robotic perception. Surface normal estimation, as a fundamental task in 3D point cloud analysis, is critical for understanding object geometry and supporting downstream tasks like surface reconstruction [22, 23] and segmentation [24]. However, real-world point clouds are often contaminated with noise, leading to distorted representations that hinder accurate normal estimation. Traditional methods [2, 29, 30, 32, 33, 62] that rely on supervised learning require extensive labeled data and struggle with noisy, unstructured data, making it challenging to obtain reliable normals from corrupted point clouds.

To address these limitations, we propose a novel approach that leverages local gradient-aware surface filtering for estimating oriented normals in noisy point clouds. Inspired by recent advancements in neural implicit representations, we adopt techniques from implicit function learning to bridge the gap between raw point clouds captured by 3D sensors and the smooth, continuous surfaces required for inferring accurate normals. Unlike existing methods [1, 35, 38, 60] that focus solely on individual point constraints, often resulting in over-smoothing or gradient degradation, our method can recover high-quality 3D geometry from noisy observations by introducing specialized loss functions with local gradient constraints.

To learn the surface representations, we introduce a distance measurement operator that enables global surface fitting from noisy data by incorporating projected distances along normals. We propose implicit field-based filtering to project points onto the underlying surface based on normals and distances derived from an implicit function, which is defined through signed distance fields and local gradient constraints. To properly guide the projection during the filtering, we incorporate the constraints of local gradient consistency, orientation and aggregation to preserve high-frequency geometric details in noisy data. The surface filtering effectively reduces noise while maintaining the shape's intricate details, allowing us to achieve a refined and noise-resilient surface representation. To demonstrate the effectiveness of our method, we evaluate it on three key tasks in point cloud processing: normal estimation, surface reconstruction, and point cloud denoising. Experimental results show that our approach significantly improves performance on noisy data, highlighting its robustness and suitability for practical 3D vision applications. In summary, our main contributions include:

- We propose a new paradigm for surface fitting from noisy point clouds by conducting filtering using normals and distances derived from an implicit function.
- We introduce the local gradient consistency constraints, local gradient orientation and aggregation to enhance the surface filtering for learning normals.
- We report the state-of-the-art performance of our method

across three tasks in point cloud processing.

2. Related Work

Normal Estimation. The classical approaches for normal estimation include Principal Component Analysis (PCA) [18] and its refinements [20, 42], which remain popular in many geometric processing tasks. Other methods [7, 15, 27], by introducing new representations of complex surfaces, estimate normals over larger neighborhoods. However, these techniques often struggle with noisy data and tend to oversmooth geometry when more neighboring points are incorporated. More recent methods [3, 13, 16, 29, 34, 48, 50, 56, 57] leverage neural networks trained on large, labeled datasets to regress normals for point clouds. Additionally, other approaches [2, 6, 26, 28, 55, 59, 62] focus on predicting pointwise weights through neural networks, with normals subsequently calculated using traditional surface fitting techniques. However, these methods typically produce unoriented normals and require supervised training with ground truth data for accurate normal predictions.

Normal Orientation. For normal orientation, classical methods like Minimum Spanning Tree (MST) [18] and its improved variants [21, 25, 41, 45, 51] rely on propagating orientations through measuring the similarity between neighboring points. Later, some approaches [10, 49, 52] employ volumetric representation techniques to enhance robustness across diverse data, though they often require manual tuning of hyperparameters for different data types. More recently, researchers have developed deep learning methods [17, 30, 32, 33, 46] that directly regress oriented normals from point clouds in a data-driven manner. While these learning-based methods generally outperform traditional data-independent approaches, they often rely heavily on costly labeled training data and struggle with accurately orienting normals in noisy point clouds. In contrast, our proposed method can learn oriented normals directly from a single noisy point cloud without any labeled data.

Learning Implicit Function from Raw Point Clouds. Unlike traditional approaches that train neural networks using supervised signals such as signed distances or occupancy labels, recent works [1, 8, 35, 38–40, 43, 58, 60] have proposed methods to directly learn implicit functions from raw point clouds in an unsupervised manner. These methods train neural networks to overfit individual point clouds to infer implicit functions without relying on learned priors. Leveraging gradient constraints [1, 8, 38], designed priors [39, 40], implicit geometric regularization [14], or differentiable Poisson solvers [43], these techniques can generalize across varying point cloud sizes and accommodate limited input data. In this work, we build on the neural network's approximation ability and incorporate new techniques for learning signed distance fields. By applying surface filtering, we aim to recover geometric details based on implicit field information and accurately infer normals from noisy point clouds.

3. Method

Preliminary. Implicit representation approaches usually denote surfaces as the level sets of implicit function, *i.e.*, $\boldsymbol{S}_d = \{\boldsymbol{x} \in \mathbb{R}^3 \mid f_\theta(\boldsymbol{x}) = d\}$, where $f_\theta \colon \mathbb{R}^3 \to \mathbb{R}$ is implemented as a neural network with parameter θ. The implicit function can be learned by overfitting the neural network on individual point clouds. If the function f_θ is correctly defined by a signed distance field inferred from points, the normal of a point $\boldsymbol{p}$ in this implicit field can be obtained by $\boldsymbol{n_p} = \nabla f_\theta(\boldsymbol{p}) / \|\nabla f_\theta(\boldsymbol{p})\|$, where $\|\cdot\|$ means the Euclidean L^2-norm and $\nabla f_\theta(\boldsymbol{p})$ denotes the gradient at $\boldsymbol{p}$. Specifically, the zero level set $f_\theta(\boldsymbol{x}) = 0$ is usually extracted as the object or scene surface $\boldsymbol{S}$. Random points on a level set have specific signed distances, such as $f_\theta(\boldsymbol{x}) < 0$ for outside and $f_\theta(\boldsymbol{x}) > 0$ for inside. The gradients on a specific iso-surface should have uniform orientations. In this work, we aim to apply surface filtering to project noisy points onto the underlying surface defined by the zero level set without supervision of ground truth labels or clean points. We use the signed distances and normals of noisy points to define the projection path and incorporate rules of the local field.

3.1. Surface Fitting and Filtering

We perform surface fitting and point filtering by learning an implicit field from a given noisy point cloud $\boldsymbol{P} = \{\boldsymbol{p}_i | \boldsymbol{p}_i \in \mathbb{R}^3\}_{i=1}^N$. From the perspective of implicit function learning, we aim to construct a signed distance field that minimizes the signed distance of all points to a zero level set, defined as follows:

$$\arg\min_{f_\theta} \frac{1}{N} \sum_{i=1}^{N} |f_\theta(\boldsymbol{p}_i)|^2. \tag{1}$$

The underlying surface can be fitted by finding the zero level set of the implicit function f_θ. However, directly fitting this surface would force it to pass through all noisy points, resulting in a zero signed distance for each point and thus obtaining a solution to the above equation that fails to accurately represent the desired surface.

From the perspective of data fitting, we typically solve an optimization problem to obtain a surface whose distance to all data points is minimized. The surface $\boldsymbol{S}$ to be solved is continuous, and we use its discretization to approximate it. Let $\hat{\boldsymbol{P}} = \{\hat{\boldsymbol{p}}_i | \hat{\boldsymbol{p}}_i \in \mathbb{R}^3\}_{i=1}^{N'}$, $N' > N$ denote the discretization of the clean surface, *i.e.*, the point set $\hat{\boldsymbol{P}}$ lies on the surface. In the implicit field space, points $\hat{\boldsymbol{p}}_i$ should be located on the zero level set, while noisy points $\boldsymbol{p}_i$ may be on a non-zero level set. By conducting surface fitting using

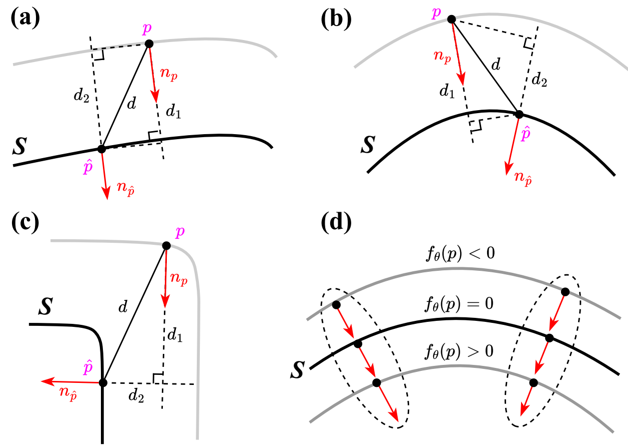

Figure 1. We minimize the distances from noisy points $\boldsymbol{p}$ to discrete points $\hat{\boldsymbol{p}}$ of the underlying surface for implicit surface fitting and filtering. To this end, (a-c) we adopt three distance measures d, d_1 and d_2, and use their sum to handle various cases. (d) Meanwhile, we enforce local gradient consistency between adjacent level sets where the noisy points are located. Red arrows indicate normals (*i.e.*, gradients).

the points in $\boldsymbol{P}$, we can solve for the surface points by

$$\arg\min_{\hat{\boldsymbol{P}}} \frac{1}{N}\sum_{i=1}^{N} \|\hat{\boldsymbol{p}}_i - \boldsymbol{p}_i\|, \tag{2}$$

where each point $\hat{\boldsymbol{p}}_i$ is selected for a corresponding point $\boldsymbol{p}_i$ based on certain criteria, such as nearest neighbor searching. However, this distance measure is inadequate because $\hat{\boldsymbol{p}}_i$ and $\boldsymbol{p}_i$ do not always have a one-to-one correspondence due to the discretization of these points and the interference of noise. As a result, this approach cannot accurately measure the distance and often yields an over-smoothed geometry or even fails.

To comprehensively measure the distance error between two points from multiple perspectives, we employ two projection distance measurements using the normals $\boldsymbol{n}_{\hat{\boldsymbol{p}}_i}$ and $\boldsymbol{n}_{\boldsymbol{p}_i}$ at points $\hat{\boldsymbol{p}}_i$ and $\boldsymbol{p}_i$, respectively. Specifically, these two projection distances are calculated as $d_1 = |(\hat{\boldsymbol{p}}_i - \boldsymbol{p}_i)\boldsymbol{n}_{\boldsymbol{p}_i}^\top|$ and $d_2 = |(\hat{\boldsymbol{p}}_i - \boldsymbol{p}_i)\boldsymbol{n}_{\hat{\boldsymbol{p}}_i}^\top|$, as illustrated in Fig. 1(a-c). The key insight behind these distance measurements is that if $\hat{\boldsymbol{p}}_i$ is the true corresponding surface point of $\boldsymbol{p}_i$, then all three distance errors should be minimized. Taking these projection distances into account, our distance measurement operator for surface fitting from noisy points is defined as

$$\mathcal{D}(\hat{\boldsymbol{p}}_i, \boldsymbol{p}_i) = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\boldsymbol{p}}_i - \boldsymbol{p}_i\| + |(\hat{\boldsymbol{p}}_i - \boldsymbol{p}_i)\boldsymbol{n}_{\boldsymbol{p}_i}^\top| + |(\hat{\boldsymbol{p}}_i - \boldsymbol{p}_i)\boldsymbol{n}_{\hat{\boldsymbol{p}}_i}^\top|. \tag{3}$$

An ideal distance measure is shown in Fig. 1(d), where the level surface is parallel, and the points are correctly matched, resulting in the three distance errors being equal.

Next, we introduce the process to obtain the surface points $\hat{\boldsymbol{p}}$ and the corresponding point normals $\boldsymbol{n}$. To determine the point $\hat{\boldsymbol{p}}$, we first define a new point set $\boldsymbol{Q}$, which is generated from the raw point set $\boldsymbol{P}$. This set, $\boldsymbol{Q} = \{\boldsymbol{q}_i \mid \boldsymbol{q}_i \in \mathbb{R}^3\}_{i=1}^N$, is also randomly distributed around the underlying surface. Since the gradient indicates the direction in which the signed distance from the surface increases most rapidly, moving a point along or against the gradient (depending on the sign of f_θ) will allow it to reach its nearest position on the surface. We thus adopt a point translation operation [31, 38] to project a query point $\boldsymbol{q}$ to a new position $\boldsymbol{q}'$, where $\boldsymbol{q}' = \boldsymbol{q} - f_\theta(\boldsymbol{q}) \cdot \boldsymbol{n}_{\boldsymbol{q}}$. If the implicit function is properly learned, it should provide the correct signed distance f_θ and gradient ∇f_θ to move the point $\boldsymbol{q}$ to its nearest location on the underlying surface. We then obtain the surface points set $\boldsymbol{Q}' = \{\boldsymbol{q}_i' \mid \boldsymbol{q}_i' = \boldsymbol{q}_i - f_\theta(\boldsymbol{q}_i) \cdot \boldsymbol{n}_{\boldsymbol{q}_i},\ \boldsymbol{q}_i \in \boldsymbol{Q}\}_{i=1}^N$. Using the raw points in $\boldsymbol{P}$ and their nearest points in $\boldsymbol{Q}'$, we fit a surface by applying the distance measure operator in Eq. (3), and the loss function is formulated as

$$\mathcal{L}_d = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{q}_i' - \boldsymbol{p}_i\| + |(\boldsymbol{q}_i' - \boldsymbol{p}_i)\boldsymbol{n}_{\boldsymbol{p}_i}^\top| + |(\boldsymbol{q}_i' - \boldsymbol{p}_i)\boldsymbol{n}_{\boldsymbol{q}_i'}^\top|. \tag{4}$$

For constructing the noisy point set $\boldsymbol{Q}$, we employ a Gaussian based sampling strategy [1, 5]. Specifically, we first obtain uniformly sampled points $\boldsymbol{p}$ from $\boldsymbol{P}$, then add Gaussian noise $\mathcal{N}(\boldsymbol{p}, \sigma^2)$ to each $\boldsymbol{p}$, where the standard deviation parameter σ is adaptively set based on the distance from $\boldsymbol{p}$ to its ξ-th nearest neighbor. In our surface fitting and filtering, we include $\boldsymbol{P}$, which together with $\boldsymbol{Q}$, to provide more useful information from the raw data.

Based on the surface points $\boldsymbol{q}_i' \in \boldsymbol{Q}'$ and their corresponding noisy observations $\boldsymbol{q}_i \in \boldsymbol{Q}$ and $\boldsymbol{p}_i \in \boldsymbol{P}$, we can define the implicit function learning process using Eq. (1) as follows:

$$\mathcal{L}_{sd} = \frac{1}{N}\sum_{i=1}^{N} |f_\theta(\boldsymbol{q}_i')|^2 + |f_\theta(\boldsymbol{q}_i)|^2 + |f_\theta(\boldsymbol{p}_i)|^2 . \tag{5}$$

Since the surface points in $\boldsymbol{Q}'$ are located on the zero level

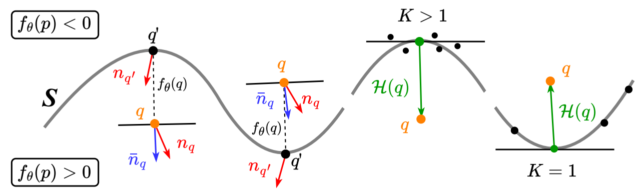

Figure 2. *Left*: computation of $f_\theta(\boldsymbol{q}) \cdot \bar{\boldsymbol{n}}_{\boldsymbol{q}}$ and $\bar{\boldsymbol{n}}_{\boldsymbol{q}} = (\boldsymbol{n}_{\boldsymbol{q}} + \boldsymbol{n}_{\boldsymbol{q}'})/||\boldsymbol{n}_{\boldsymbol{q}} + \boldsymbol{n}_{\boldsymbol{q}'}||$. Gradients point to the positive side of the signed distance field. *Right*: computation of $\mathcal{H}(\boldsymbol{q})$ for specific noise and density using different neighborhood scales K.

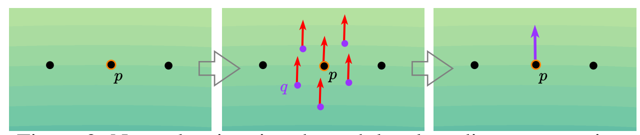

Figure 3. Normal estimation through local gradient aggregation.

set, their signed distances should ideally approach zero. To enforce this, we empirically assign a larger weight to their signed distance (*e.g.*, ten times greater) compared to the noisy observations in $\boldsymbol{Q}$ and $\boldsymbol{P}$, which are normally distributed near the underlying surface and their average signed distances should be zero.

Although $\mathcal{L}_d$ and $\mathcal{L}_{sd}$ can guide global surface fitting and filtering, our ablation studies reveal that using these terms alone fails to capture accurate implicit surfaces and point normals in noisy point clouds. One issue with this approach is that it neglects local geometric details, leading to over-smoothed and noise-sensitive surfaces. Another significant issue is gradient degradation, which disrupts surface fitting and often accompanies noisy data and complex geometries. For Eq. (4), we observe that $\nabla f_\theta = 0$ can be an optimal solution, minimizing the function. This degradation reduces the objective to the original formulation in Eq. (2), which implies no valid level set learned by the network, resulting in an inaccurate local distance field and disordered iso-surfaces. The solution we propose next incorporates local gradient consistency constraints, gradient aggregation between level sets, and local gradient orientation within a level set, effectively addressing these issues.

3.2. Local Gradient Consistency of Inter-Level

Inspired by the strategy employed in [31], which constrains directional consistency in a multi-step moving process. We hope to make the projection $\boldsymbol{Q} \to \boldsymbol{Q}'$ bridge the geometric relationship between noisy points and their corresponding surface points, enhancing the surface filtering accuracy. We constrain the local gradients of neighboring level sets to have similar directions, as illustrated in Fig. 1(d). Specifically, we enforce similarity in gradient direction between the initial noisy point $\boldsymbol{q}_i$ and its projected point $\boldsymbol{q}_i'$. Recognizing that local gradients between distant level sets may vary, we account for the signed distance in the constraint. Thus, the confidence-weighted direction distance, used to evaluate gradient consistency between points on neighboring level sets, is formulated as

$$\mathcal{L}_n = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \boldsymbol{n}_{\boldsymbol{q}_i'}\,\boldsymbol{n}_{\boldsymbol{q}_i}^{\top}\right)\cdot w_i \ , \tag{6}$$

where $w_i = \exp(-\rho \cdot |f_\theta(\boldsymbol{q}_i)|)$ is an adaptive weight that emphasizes points near the underlying surface based on the predicted distance. Ablation experiments show that this loss can not only reduce noise impact but also guide the network to generate valid gradients and surface points on level sets.

3.3. Local Gradient Orientation of Intra-Level

We also focus on the orientation of local gradients at each level set and examine the generation of surface points $\boldsymbol{Q}'$ from the raw data. From the previous equation $\boldsymbol{q}' = \boldsymbol{q} - f_\theta(\boldsymbol{q})\cdot\boldsymbol{n}_{\boldsymbol{q}}$, we see that the term $f_\theta(\boldsymbol{q})\cdot\boldsymbol{n}_{\boldsymbol{q}}$ mainly determines the position of the generated surface points. If the surface point $\boldsymbol{q}'$ is known, this term should be as close as possible to $\mathcal{H}(\boldsymbol{q}) = \boldsymbol{q} - \boldsymbol{q}'$, which we measure by

$$\mathcal{L}_v = \|f_\theta(\boldsymbol{q})\cdot\boldsymbol{n}_{\boldsymbol{q}} - \mathcal{H}(\boldsymbol{q})\| \,. \tag{7}$$

Since we only have noisy inputs, we need a robust strategy to approximate $\mathcal{H}(\boldsymbol{q})$. Traditional least squares methods typically use plane fitting within a local neighborhood:

$$\mathcal{H}(\boldsymbol{q}) = \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{q} - \boldsymbol{p}_k), \quad \boldsymbol{p}_k \in \mathbb{K}_K(\boldsymbol{q}, \boldsymbol{P}) \ , \tag{8}$$

where $\mathbb{K}_K(\boldsymbol{q}, \boldsymbol{P})$ denotes the set of K nearest points to $\boldsymbol{q}$ in $\boldsymbol{P}$. Here, $\mathcal{H}(\boldsymbol{q})$ is the oriented vector from the averaged position $\bar{\boldsymbol{p}} = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{p}_k$ to the query point $\boldsymbol{q}$. However, this fixed neighborhood approach is not robust against varying noise levels, density variations and different geometric structures. In this work, we allow the network model to learn an adaptive neighborhood size to better approximate the surface point by considering multiple scales instead of relying on a fixed neighborhood. The multi-scale approximation of the surface point $\boldsymbol{q}'$ is formulated as

$$\mathcal{L}_v = \sum_{j=1}^{N_K}\|f_\theta(\boldsymbol{q})\cdot\boldsymbol{n}_{\boldsymbol{q}} - \mathcal{H}_j(\boldsymbol{q})\| \ , \tag{9}$$

where $\mathcal{H}_j(\boldsymbol{q})$ is computed using a specific size selected from a scale set $\{K_j\}_{j=1}^{N_K}$, as shown in Fig. 2 for specific noise and density. This formulation reduces the impact of inaccuracies in $\boldsymbol{P}$ by utilizing multi-scale local neighbors, thereby inferring the possible correct position of $\boldsymbol{q}'$ from multiple corrupted observations of the same local region.

Learning from multiple corrupt observations enhances performance on noisy data, but there remains room for improvement often overlooked by previous works [31, 58]. These approaches directly use $\boldsymbol{n}_{\boldsymbol{q}}$ as the normal of $\boldsymbol{q}$, neglecting the inaccuracies in gradients introduced by the local approximation in Eq. (8). We take this a step further by rethinking the generation of $\boldsymbol{q}'$ from $\boldsymbol{q}$ and examining the geometric relationship of their gradients in the implicit field. For each query point $\boldsymbol{q}_i \in \boldsymbol{Q}$, we solve its normal as the sum of the normals at the two endpoints of the projection path, *i.e.*, $\bar{\boldsymbol{n}}_{\boldsymbol{q}_i} = (\boldsymbol{n}_{\boldsymbol{q}_i} + \boldsymbol{n}_{\boldsymbol{q}_i'})/||\boldsymbol{n}_{\boldsymbol{q}_i} + \boldsymbol{n}_{\boldsymbol{q}_i'}||$, as illustrated in Fig. 2. The objective function then becomes the aggregation of errors at each neighborhood size scale:

$$\mathcal{L}_v = \sum_{j=1}^{N_K}\frac{1}{N}\sum_{i=1}^{N}\left\|f_\theta(\boldsymbol{q}_i)\cdot\frac{\boldsymbol{n}_{\boldsymbol{q}_i} + \boldsymbol{n}_{\boldsymbol{q}_i'}}{||\boldsymbol{n}_{\boldsymbol{q}_i} + \boldsymbol{n}_{\boldsymbol{q}_i'}||} - \mathcal{H}_j(\boldsymbol{q}_i)\right\| . \tag{10}$$

Our method learns to identify underlying surface points by using local plane fitting of multi-scale neighbors and replacing $\boldsymbol{n}_{\boldsymbol{q}}$ with the averaged normal $\bar{\boldsymbol{n}}_{\boldsymbol{q}}$. In this way, we can

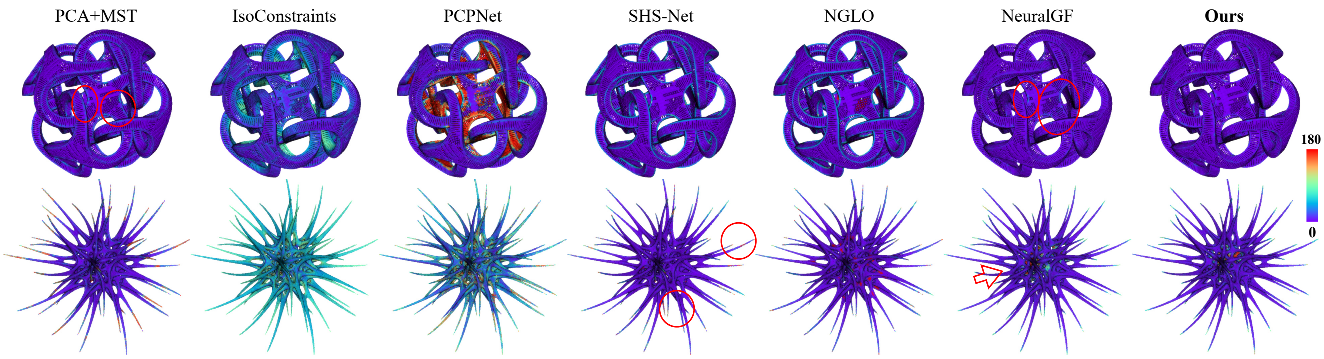

Figure 4. Visual comparison of oriented normals on two point clouds with complex geometry. Colors indicate normal errors.

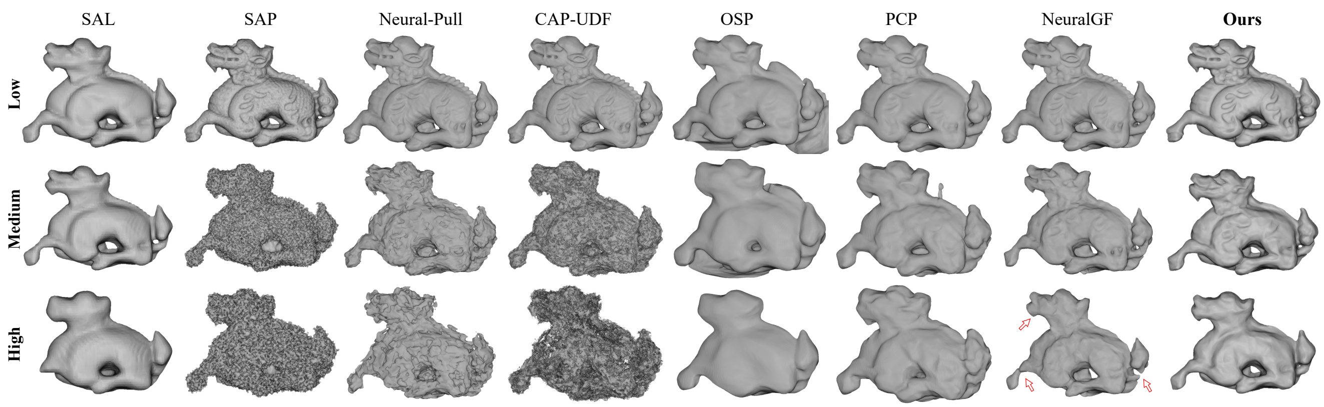

Figure 5. Comparison with implicit function methods. As the noise increases (from low to high), our method becomes more advantageous.

effectively reduce noise-induced errors and avoid potential zero values in $\boldsymbol{n_q}$ due to gradient degradation.

Final Loss. Our final loss function of learning normals from noisy points is defined as

$$\mathcal{L} = \mathcal{L}_{sd} + \lambda_1 \mathcal{L}_d + \lambda_2 \mathcal{L}_n + \mathcal{L}_v \ , \tag{11}$$

where the weight factors λ_1 and λ_2 are first set empirically and then fine-tuned based on experimental results.

3.4. Local Gradient Aggregation of Inter-Level

We train the network model to overfit on a given single point cloud $\boldsymbol{P}$. To infer the normal of a point $\boldsymbol{p} \in \boldsymbol{P}$ using the learned model, we first use the combination of $\boldsymbol{P}$ and $\boldsymbol{Q}$ as the input and derive the gradients of all points. The additional set $\boldsymbol{Q}$ is used to fully explore the possible true spatial positions of noisy points $\boldsymbol{P}$ corresponding to the underlying surface. We then search for the κ nearest neighbors of $\boldsymbol{p}$ in the point set $\{\boldsymbol{P}, \boldsymbol{Q}\}$, as shown in Fig. 3, and the final normal of $\boldsymbol{p}$ is calculated as the weighted average of the normal at $\boldsymbol{p}$ and its neighboring point normals $\bar{\boldsymbol{n}}'_i$, as follows:

$$\boldsymbol{n_p} = \frac{1}{\kappa + 1} \left(\bar{\boldsymbol{n}}_{\boldsymbol{p}} + \sum_{i=1}^{\kappa} \bar{\boldsymbol{n}}'_i \cdot \mu_i \right) , \tag{12}$$

where $\mu_i = \exp\left(-\delta_i - \phi(\bar{\boldsymbol{n}}'_i, \bar{\boldsymbol{n}}_{\boldsymbol{p}})\right)$, and δ_i is the Euclidean distance between point $\boldsymbol{p}$ and its neighboring points. $\phi(\bar{\boldsymbol{n}}'_i, \bar{\boldsymbol{n}}_{\boldsymbol{p}}) = \left((1 - \bar{\boldsymbol{n}}'_i\, \bar{\boldsymbol{n}}_{\boldsymbol{p}}^{\top})/(1 - \cos\vartheta)\right)^2$, where ϑ is a given angle. Given the nearest neighbors, the term μ adaptively assigns higher weight to neighboring points that are closer to point $\boldsymbol{p}$ or have a small normal angle with it. Ablation experiments show that this strategy is effective in improving the robustness of the algorithm in various cases.

4. Experiments

Implementation. We employ a simple neural network similar to that used in [1, 31, 38]. It consists of eight linear layers and includes a skip connection. We also use the geometric network initialization from [1]. In all evaluation experiments, the network structure and loss function components remain consistent. The neighborhood scale set $\{K_j\}_{j=1}^{N_\mathcal{K}}$ is chosen as $\{1, \mathcal{K}/2, \mathcal{K}\}$, with $N_\mathcal{K} = 3$, $\mathcal{K} = 8$, and we set the parameters $\lambda_2 = 0.01$, $\rho = 60$, $\kappa = 8$, and $\vartheta = \pi/12$. The hyperparameters ξ and λ_1 are adjusted based on specific datasets. The number of points N used in each training iteration is set to 5000. As in [16, 29, 32], we evaluate the normal estimation results using Root Mean Squared Error (RMSE) and Percentage of Good Points (PGP). *More results are provided in the supplementary material.*

4.1. Normal Estimation

Comparison of Oriented Normal. Our approach requires no training labels and learns solely from raw data. We com-

ICCV
#8136

Table 1. Oriented RMSE on PCPNet and FamousShape datasets. We achieve better performance even compared with supervised methods.

Methods	PCPNet Dataset							FamousShape Dataset						
	Noise				Density		Average	Noise				Density		Average
	None	Low	Medium	High	Stripe	Gradient		None	Low	Medium	High	Stripe	Gradient	
Supervised														
AdaFit [62]+MST [18]	27.67	43.69	48.83	54.39	36.18	40.46	41.87	43.12	39.33	62.28	60.27	45.57	42.00	48.76
AdaFit [62]+SNO [45]	26.41	24.17	40.31	48.76	27.74	31.56	33.16	27.55	37.60	69.56	62.77	27.86	29.19	42.42
AdaFit [62]+ODP [41]	26.37	24.86	35.44	51.88	26.45	20.57	30.93	41.75	39.19	44.31	72.91	45.09	42.37	47.60
HSurf-Net [29]+MST [18]	29.82	44.49	50.47	55.47	40.54	43.15	43.99	54.02	42.67	68.37	65.91	52.52	53.96	56.24
HSurf-Net [29]+SNO [45]	30.34	32.34	44.08	51.71	33.46	40.49	38.74	41.62	41.06	67.41	62.04	45.59	43.83	50.26
HSurf-Net [29]+ODP [41]	26.91	24.85	35.87	51.75	26.91	20.16	31.07	43.77	43.74	46.91	72.70	45.09	43.98	49.37
PCPNet [16]	33.34	34.22	40.54	44.46	37.95	35.44	37.66	40.51	41.09	46.67	54.36	40.54	44.26	44.57
DPGO [46]	23.79	25.19	35.66	43.89	28.99	29.33	31.14	-	-	-	-	-	-	-
SHS-Net [32, 33]	10.28	13.23	25.40	35.51	16.40	17.92	19.79	21.63	25.96	41.14	52.67	26.39	28.97	32.79
NGLO [30]	12.52	12.97	25.94	33.25	16.81	**9.47**	18.49	**13.22**	18.66	39.70	51.96	31.32	**11.30**	27.69
Unsupervised														
PCA [18]+MST [18]	19.05	30.20	31.76	39.64	27.11	23.38	28.52	35.88	41.67	38.09	60.16	31.69	35.40	40.48
PCA [18]+SNO [45]	18.55	21.61	30.94	39.54	23.00	25.46	26.52	32.25	39.39	41.80	61.91	36.69	35.82	41.31
PCA [18]+ODP [41]	28.96	25.86	34.91	51.52	28.70	23.00	32.16	30.47	31.29	41.65	84.00	39.41	30.72	42.92
LRR [53]+MST [18]	43.48	47.58	38.58	44.08	48.45	46.77	44.82	56.24	57.38	45.73	64.63	66.35	56.65	57.83
LRR [53]+SNO [45]	44.87	43.45	33.46	45.40	46.96	37.73	41.98	59.78	60.18	45.02	71.37	62.78	59.90	59.84
LRR [53]+ODP [41]	28.65	25.83	36.11	53.89	26.41	23.72	32.44	39.97	42.17	48.29	88.68	44.92	47.56	51.93
IsoConstraints [49]	24.42	26.52	87.30	94.99	28.69	32.02	48.99	38.23	41.59	83.11	93.07	42.47	49.68	58.03
NeuralGF [31]	10.60	18.30	24.76	33.45	12.27	12.85	18.70	16.57	19.28	36.22	50.27	17.23	17.38	26.16
Ours	**9.71**	**11.99**	**24.39**	**32.74**	**11.30**	11.84	**17.00**	13.71	**18.40**	**34.97**	**49.25**	**14.35**	13.76	**24.07**

Table 2. Comparison of oriented PGP-70° (%) on the PCPNet and FamousShape datasets under medium noise.

Dataset	HSurf-Net +ODP	Iso-Cons.	PCPNet	NGLO	SHS-Net	NeuralGF	Ours
PCPNet	90.28	58.54	89.47	94.59	94.59	94.67	**94.96**
Famous.	87.57	57.83	87.19	91.10	90.90	92.87	**93.14**

Table 3. Oriented normal RMSE on sparse point clouds. IsoConstraints [49] and GCNO [52] are traditional methods.

	HSurf-Net +ODP	Iso-Cons.	GCNO	PCPNet	NGLO	SHS-Net	NeuralGF	Ours
3K	63.88	40.01	33.40	53.13	32.65	37.31	25.54	**24.03**
5K	62.51	37.45	41.24	48.48	28.34	32.64	24.35	**21.80**

Table 4. Oriented RMSE on the SceneNN and ScanNet datasets.

Dataset	HSurf-Net +ODP	PCPNet	NGLO	SHS-Net	NeuralGF	Ours
SceneNN						
Clean	51.85	70.70	48.52	78.71	47.80	**44.82**
Noisy	50.24	70.82	45.42	77.60	48.69	**40.66**
Average	51.05	70.76	46.97	78.16	48.24	**42.74**
ScanNet	49.34	68.10	39.40	74.36	39.10	**37.09**

pare our method with both supervised and unsupervised methods, including end-to-end and two-stage pipeline approaches. In Table 1, we report quantitative evaluation results on the PCPNet [16] and FamousShape [32] datasets. Our method achieves superior performance across most data categories (in terms of noise levels and density variations) and delivers the best average results, even compared to supervised approaches. Quantitative comparisons of PGP at a threshold of 70° are reported in Table 2, indicating that our method provides more accurate normals for a higher proportion of points.

In Table 3, we present evaluation results on sparse point cloud data. These point sets are sparse versions of the FamousShape dataset [32], each containing only 3000 and 5000 points. The quantitative comparison results demonstrate that our method achieves the lowest error on these sparse point sets. We further evaluate our approach on real-world scanned datasets to assess its generalization capability. Table 4 provides quantitative results on the SceneNN [19] and ScanNet [11] datasets, where our method outperforms baselines, demonstrating a stronger ability to handle real-world data.

Comparison of Unoriented Normal. In this evaluation, we use our oriented normals to compare with baselines but ignore normal orientations, which are often challenging to determine. In Table 5, we report quantitative results on the PCPNet and FamousShape datasets. Most existing methods for unoriented normal estimation rely on supervised training with ground truth normals. We evaluate both supervised and unsupervised methods, and our approach outperforms in most data categories across both datasets, achieving the highest average results among unsupervised methods. Notably, CAP-UDF [58, 60] performs well on noise-free point clouds but struggles with noisy data. In contrast, our method demonstrates a significant advantage on noisy data. The quantitative comparisons of PGP at a threshold of

Table 5. Unoriented RMSE on PCPNet and FamousShape datasets. We achieve better performance compared with unsupervised methods.

Methods	PCPNet Dataset							FamousShape Dataset						
	Noise				Density		Average	Noise				Density		Average
	None	Low	Medium	High	Stripe	Gradient		None	Low	Medium	High	Stripe	Gradient	
Supervised														
DeepFit [2]	6.51	9.21	16.73	23.12	7.92	7.31	11.80	11.21	16.39	29.84	39.95	11.84	10.54	19.96
Zhang *et al*. [55]	5.65	9.19	16.78	22.93	6.68	6.29	11.25	9.83	16.13	29.81	39.81	9.72	9.19	19.08
AdaFit [62]	5.19	9.05	16.45	21.94	6.01	5.90	10.76	9.09	15.78	29.78	38.74	8.52	8.57	18.41
GraphFit [28]	5.21	8.96	16.12	21.71	6.30	5.86	10.69	8.91	15.73	29.37	38.67	9.10	8.62	18.40
NeAF [34]	4.20	9.25	16.35	21.74	4.89	4.88	10.22	7.67	15.67	29.75	38.76	7.22	7.47	17.76
HSurf-Net [29]	4.17	8.78	16.25	21.61	4.98	4.86	10.11	7.59	15.64	29.43	38.54	7.63	7.40	17.70
NGLO [30]	4.06	8.70	16.12	21.65	4.80	4.56	9.98	7.25	15.60	29.35	38.74	7.60	7.20	17.62
SHS-Net [32, 33]	3.95	8.55	16.13	21.53	4.91	4.67	9.96	7.41	15.34	29.33	38.56	7.74	7.28	17.61
Du *et al*. [13]	3.85	8.67	16.11	21.75	4.78	4.63	9.96	6.92	15.05	29.49	38.73	7.19	6.92	17.38
CMG-Net [48]	3.87	8.45	16.08	21.89	4.85	4.45	9.93	7.07	14.83	29.04	38.93	7.43	7.03	17.39
MSECNet [50]	3.84	8.74	16.10	21.05	4.34	4.51	9.76	6.85	15.60	29.22	38.13	6.64	6.65	17.18
Unsupervised														
CAP-UDF [60]	**7.59**	11.99	37.69	47.64	**8.26**	**7.36**	20.09	14.34	21.62	50.43	55.33	13.31	13.45	28.08
Boulch *et al*. [4]	11.80	11.68	22.42	35.15	13.71	12.38	17.86	19.00	19.60	36.71	50.41	20.20	17.84	27.29
PCV [54]	12.50	13.99	18.90	28.51	13.08	13.59	16.76	21.82	22.20	31.61	46.13	20.49	19.88	27.02
Jet [7]	12.35	12.84	18.33	27.68	13.39	13.13	16.29	20.11	20.57	31.34	45.19	18.82	18.69	25.79
PCA [18]	12.29	12.87	18.38	27.52	13.66	12.81	16.25	19.90	20.60	31.33	45.00	19.84	18.54	25.87
LRR [53]	9.63	11.31	20.53	32.53	10.42	10.02	15.74	17.68	19.32	33.89	49.84	16.73	16.33	25.63
NeuralGF [31]	7.89	9.85	18.62	24.89	9.21	9.29	13.29	13.74	16.51	31.05	40.68	13.95	13.17	21.52
Ours	7.60	**9.45**	**16.87**	**22.49**	8.52	8.55	**12.25**	**11.90**	**15.84**	**29.90**	**39.08**	**11.82**	**11.36**	**19.98**

Table 6. Comparison of unoriented PGP-20° (%) on the PCPNet and FamousShape datasets under the highest noise.

Dataset	Jet	PCA	PCV	LRR	Boulch *et al*.	CAP-UDF	NeuralGF	Ours
PCPNet	64.60	65.06	62.79	52.33	44.03	24.84	70.02	**74.65**
Famous.	27.46	28.05	25.55	19.29	17.52	12.23	37.32	**42.62**

Table 7. Surface reconstruction on the SRB dataset.

	SAP	Neural-Pull	NeuralGF	CAP-UDF	IF	Ours
CD_{L1}	0.4787	0.2845	0.2623	0.2766	0.2519	**0.2518**
F-Score	0.9383	0.9689	0.9758	0.9760	0.9782	**0.9786**

Table 8. Surface reconstruction on the 3D Scene dataset.

Metric	Stonewall			Lounge		
	CD_{L2}	CD_{L1}	NC	CD_{L2}	CD_{L1}	NC
Neural-Pull [38]	27.2995	3.0477	0.8222	0.3172	0.2350	0.8949
OSP [39]	0.7241	0.5226	0.8878	7.3628	1.6020	0.6828
SAP [43]	0.5499	0.2988	0.8599	0.1372	0.2221	0.8480
NeuralGF [31]	0.0534	0.0934	0.9469	0.1428	0.1658	0.9059
CAP-UDF [60]	0.0107	0.0795	0.9403	**0.0221**	**0.1086**	0.8903
IF [35]	0.1222	0.1998	0.9238	0.1046	0.1519	0.8979
Ours	**0.0093**	**0.0777**	**0.9527**	0.1158	0.1531	**0.9093**

20° are reported in Table 6, showing that our method delivers more accurate normals for a larger proportion of points.

4.2. Surface Reconstruction

The zero level set of our learned implicit function can be extracted as the object surface using the marching cubes algorithm [36]. We compare our surface reconstruction performance with other implicit representation methods, including SAL [1], SAP [43], Neural-Pull [38], CAP-UDF [60], OSP [39], PCP [40], IF [35], and NeuralGF [31]. As shown in Fig. 5, we present a visual comparison of reconstructed surfaces from point clouds at varying noise levels. Our method shows a clear advantage in handling noisy data, producing cleaner and more complete structures than baseline methods. For surface reconstruction from real-world data, we follow previous works [35, 40, 60] and evaluate on the SRB dataset [47] and 3D Scene dataset [61], using Chamfer distance (CD), Normal Consistency (NC), and F-Score metrics. The quantitative results reported in Table 7 and Table 8 show that our method achieves the highest accuracy on the SRB dataset and in most cases of the 3D Scene dataset.

4.3. Point Cloud Denoising

In Sec. 3.1, we use the raw point cloud $\boldsymbol{P}$ to construct a new sample set $\boldsymbol{Q}$, and obtain the corresponding surface point set $\boldsymbol{Q}'$ through filtering. For point cloud denoising, we take all points in the raw data $\boldsymbol{P}$ as input to the trained model. By applying the transformation $\boldsymbol{P}' = \{\boldsymbol{p}_i' \mid \boldsymbol{p}_i' = \boldsymbol{p}_i - f_\theta(\boldsymbol{p}_i)\cdot\boldsymbol{n}_{\boldsymbol{p}_i},\ \boldsymbol{p}_i \in \boldsymbol{P}\}_{i=1}^N$, the new generated points $\boldsymbol{P}'$ should ideally lie on the underlying clean surface. Following prior works [9, 37], we evaluate our denoising performance on the PointCleanNet dataset [44], a standard benchmark that includes two resolution levels (10K and 50K points) and three noise levels (scales of 1%, 2%, and 3% of the shape bounding sphere's radius). We also use Chamfer distance (CD) and point-to-mesh distance (P2M) as metrics to evaluate the denoised point clouds. The quantitative comparison results are reported in *the Table 1 of supplementary*

Table 9. Ablations for unoriented and oriented normal estimation on the FamousShape dataset. We decompose $\mathcal{L}_d$ into $\mathcal{L}_{ld}$ and $\mathcal{L}_{pd}$.

	Category	Unoriented Normal RMSE							Oriented Normal RMSE						
		Noise				Density		Average	Noise				Density		Average
		None	Low	Medium	High	Stripe	Gradient		None	Low	Medium	High	Stripe	Gradient	
(a)	$\mathcal{L}_{ld}$	25.19	33.08	38.42	46.76	28.80	25.51	32.96	37.51	67.44	84.01	69.70	45.80	35.95	56.73
	$\mathcal{L}_{ld}+\mathcal{L}_{pd}+\mathcal{L}_{sd}$	15.74	23.08	54.64	56.10	14.33	15.53	29.90	22.11	39.46	88.36	92.84	17.97	21.94	47.11
	$\mathcal{L}_{ld}+\mathcal{L}_{pd}+\mathcal{L}_{sd}+\mathcal{L}_v$	16.98	16.60	33.83	41.23	18.85	18.37	24.31	26.56	23.91	55.90	83.79	28.98	27.41	41.09
	$\mathcal{L}_{ld}+\mathcal{L}_{pd}+\mathcal{L}_{sd}+\mathcal{L}_n$	13.41	16.70	31.11	40.48	13.00	12.78	21.25	19.42	19.48	36.03	50.38	14.76	**13.29**	25.56
	$\mathcal{L}_{ld}+\mathcal{L}_{pd}+\mathcal{L}_n+\mathcal{L}_v$	12.00	15.87	30.00	39.30	11.95	13.14	20.38	14.66	18.89	34.96	55.00	15.12	16.45	25.85
	$\mathcal{L}_{ld}+\mathcal{L}_{sd}+\mathcal{L}_n+\mathcal{L}_v$	12.61	15.59	30.19	39.13	12.38	11.69	20.27	16.06	18.30	38.79	48.26	15.70	14.99	25.35
	$\mathcal{L}_{ld}+\mathcal{L}_n+\mathcal{L}_v$	12.28	15.63	29.88	39.26	12.18	12.19	20.24	14.98	18.25	**34.62**	50.82	14.95	21.90	25.92
	$\mathcal{L}_{pd}+\mathcal{L}_n+\mathcal{L}_v$	11.88	15.72	29.97	**39.05**	12.24	12.14	20.17	13.32	18.34	34.87	48.44	15.08	23.17	25.54
	$\mathcal{L}_{sd}+\mathcal{L}_n+\mathcal{L}_v$	12.49	**15.47**	29.94	39.19	12.49	12.11	20.28	15.06	**17.15**	35.25	**48.27**	15.44	16.62	24.63
	$\mathcal{L}_{pd}+\mathcal{L}_{sd}+\mathcal{L}_n+\mathcal{L}_v$	**11.81**	15.73	29.97	39.08	12.01	11.43	20.00	**13.31**	18.36	34.89	48.87	15.15	14.07	24.11
(b)	w/o Aggregation	12.05	16.02	30.04	39.18	11.96	11.51	20.13	13.82	18.53	35.11	49.34	14.46	13.85	24.19
(c)	$\mathcal{K}=4$	11.89	16.00	30.22	39.51	17.45	**11.17**	21.04	14.80	19.07	35.16	51.31	21.49	13.65	25.91
	$\mathcal{K}=16$	13.71	15.77	**29.86**	39.32	12.11	13.49	20.71	20.41	18.44	34.90	51.19	**14.14**	17.87	26.16
	Full	11.90	15.84	29.90	39.08	**11.82**	11.36	**19.98**	13.71	18.40	34.97	49.25	14.35	13.76	**24.07**

material. Unlike existing learning-based denoising methods, which typically require clean surface data for supervised training, our unsupervised method achieves comparable performance to these supervised baselines. Additionally, our network is lightweight, with approximately 461K parameters (14% of IterativePFN's 3.2M parameters [12]). This experiment demonstrates that our method can effectively recover underlying surfaces from noisy point clouds.

4.4. Ablation Studies

Our goal is to achieve optimal average results for both unoriented and oriented normal estimation. The ablation studies in Table 9 are discussed as follows.

(a) Loss Functions. We evaluate various combinations of the proposed loss functions from Eq. (11) to train the network model separately. For a thorough analysis, we decompose the loss in Eq. (4) as $\mathcal{L}_d = \mathcal{L}_{ld} + \mathcal{L}_{pd}$, where $\mathcal{L}_{ld} = d = \|\hat{\boldsymbol{p}} - \boldsymbol{p}\|$ and $\mathcal{L}_{pd} = d_1 + d_2$ (see Fig. 1). We observe that using only $\mathcal{L}_{ld}$ yields the poorest results, but the performance improves significantly when other loss terms are included. The introduction of $\mathcal{L}_{pd}$ and $\mathcal{L}_{sd}$ is beneficial, while the addition of $\mathcal{L}_n$ and $\mathcal{L}_v$ notably enhances performance. Some ablations yield better results in certain data categories but fail to provide consistent improvement across both unoriented and oriented normal estimation. Our method achieves the best overall performance only when all the loss functions are applied.

(b) Normal Aggregation. In Sec. 3.4, we propose a neighborhood weighted aggregation strategy to infer the normals of the raw data $\boldsymbol{P}$. Here, we use the raw data as input and infer the gradient of the implicit field at each point $\boldsymbol{p} \in \boldsymbol{P}$ as its normal. The ablation results validate the effectiveness of this inference strategy, showing improved performance in both unoriented and oriented normal estimation tasks.

(c) Neighborhood Scale. For our neighborhood scale set $\{K_j\}_{j=1}^{N_K}$, we use the base parameter $\mathcal{K}=8$ in our implementation. Here, we test different values of $\mathcal{K}$, including 4 and 16. Results indicate that while these alternative values may yield slight advantages in specific data categories, our chosen setting provides better results across both unoriented and oriented normal estimation tasks.

5. Conclusion

In this work, we presented a novel local gradient-aware surface filtering method for estimating oriented normals in noisy point clouds, overcoming the limitations of traditional approaches that often struggle with noise and require extensive labeled data. By leveraging neural implicit representations and introducing specialized loss functions with local gradient constraints, our method bridges the gap between raw, noisy data and high-quality surface representations. Our approach effectively preserves high-frequency geometric details while minimizing surface noise, yielding a refined, noise-resilient output. Experimental results across three different tasks validate the method's effectiveness and robustness, highlighting its suitability for practical 3D vision applications. Future work may further refine this framework to extend its utility across other point cloud processing tasks.

References

[1] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 1, 2, 3, 5, 7

[2] Yizhak Ben-Shabat and Stephen Gould. DeepFit: 3D surface fitting via neural network weighted least squares. In

European Conference on Computer Vision, pages 20–34. Springer, 2020. 1, 2, 7

[3] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. Nesti-Net: Normal estimation for unstructured 3D point clouds using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10112–10120, 2019. 2

[4] Alexandre Boulch and Renaud Marlet. Fast and robust normal estimation for point clouds with sharp features. In *Computer Graphics Forum*, pages 1765–1774. Wiley Online Library, 2012. 7

[5] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[6] Junjie Cao, Hairui Zhu, Yunpeng Bai, Jun Zhou, Jinshan Pan, and Zhixun Su. Latent tangent space representation for normal estimation. *IEEE Transactions on Industrial Electronics*, 69(1):921–929, 2021. 2

[7] Frédéric Cazals and Marc Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 22(2):121–146, 2005. 2, 7

[8] Chao Chen, Yu-Shen Liu, and Zhizhong Han. GridPull: Towards scalability in learning implicit representations from 3D point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18322–18334, 2023. 2

[9] Haolan Chen, Shitong Luo, Wei Hu, et al. Deep point set resampling via gradient fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2913–2930, 2022. 7

[10] Yi-Ling Chen, Bing-Yu Chen, Shang-Hong Lai, and Tomoyuki Nishita. Binary orientation trees for volume and surface reconstruction from unoriented point clouds. In *Computer Graphics Forum*, pages 2011–2019. Wiley Online Library, 2010. 2

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 6

[12] Dasith de Silva Edirimuni, Xuequan Lu, Zhiwen Shao, Gang Li, Antonio Robles-Kelly, and Ying He. IterativePFN: True iterative point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13530–13539, 2023.

[13] Hang Du, Xuejun Yan, Jingjing Wang, Di Xie, and Shiliang Pu. Rethinking the approximation error in 3D surface fitting for point cloud normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 7

[14] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020. 2

[15] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. *ACM Transactions on Graphics (TOG)*, 26(3), 2007. 2

[16] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. PCPNet: learning local shape properties from raw point clouds. In *Computer Graphics Forum*, pages 75–85. Wiley Online Library, 2018. 2, 5, 6

[17] Taisuke Hashimoto and Masaki Saito. Normal estimation for accurate 3D mesh reconstruction with point cloud model incorporating spatial structure. In *CVPR Workshops*, pages 54–63, 2019. 2

[18] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, pages 71–78, 1992. 2, 6, 7

[19] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. SceneNN: A scene meshes dataset with annotations. In *2016 Fourth International Conference on 3D Vision*, pages 92–101. IEEE, 2016. 6

[20] Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. *ACM Transactions on Graphics*, 28 (5):1–7, 2009. 2

[21] Johannes Jakob, Christoph Buchenau, and Michael Guthe. Parallel globally consistent normal orientation of raw unorganized point clouds. In *Computer Graphics Forum*, pages 163–173. Wiley Online Library, 2019. 2

[22] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3): 1–13, 2013. 1

[23] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics Symposium on Geometry Processing*, 2006. 1

[24] Ali Khaloo and David Lattanzi. Robust normal estimation and region growing segmentation of infrastructure 3D point cloud models. *Advanced Engineering Informatics*, 34:1–16, 2017. 1

[25] Sören König and Stefan Gumhold. Consistent propagation of normal orientations in point clouds. In *International Symposium on Vision, Modeling, and Visualization (VMV)*, pages 83–92, 2009. 2

[26] Jan Eric Lenssen, Christian Osendorfer, and Jonathan Masci. Deep iterative surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11247–11256, 2020. 2

[27] David Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67(224):1517–1531, 1998. 2

[28] Keqiang Li, Mingyang Zhao, Huaiyu Wu, Dong-Ming Yan, Zhen Shen, Fei-Yue Wang, and Gang Xiong. GraphFit: Learning multi-scale graph-convolutional representation for point cloud normal estimation. In *European Conference on Computer Vision*, pages 651–667. Springer, 2022. 2, 7

[29] Qing Li, Yu-Shen Liu, Jin-San Cheng, Cheng Wang, Yi Fang, and Zhizhong Han. HSurf-Net: Normal estimation

for 3D point clouds by learning hyper surfaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4218–4230. Curran Associates, Inc., 2022. 1, 2, 5, 6, 7

[30] Qing Li, Huifang Feng, Kanle Shi, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Neural gradient learning and optimization for oriented point normal estimation. In *SIGGRAPH Asia 2023 Conference Papers*, New York, NY, USA, 2023. Association for Computing Machinery. 1, 2, 6, 7

[31] Qing Li, Huifang Feng, Kanle Shi, Yue Gao, Yi Fang, Yu-Shen Liu, and Zhizhong Han. NeuralGF: Unsupervised point normal estimation by learning neural gradient function. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 66006–66019. Curran Associates, Inc., 2023. 3, 4, 5, 6, 7

[32] Qing Li, Huifang Feng, Kanle Shi, Yue Gao, Yi Fang, Yu-Shen Liu, and Zhizhong Han. SHS-Net: Learning signed hyper surfaces for oriented normal estimation of point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13591–13600, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 1, 2, 5, 6, 7

[33] Qing Li, Huifang Feng, Kanle Shi, Yue Gao, Yi Fang, Yu-Shen Liu, and Zhizhong Han. Learning signed hyper surfaces for oriented point cloud normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(12):9957–9974, 2024. 1, 2, 6, 7

[34] Shujuan Li, Junsheng Zhou, Baorui Ma, Yu-Shen Liu, and Zhizhong Han. NeAF: Learning neural angle fields for point normal estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 2, 7

[35] Shengtao Li, Ge Gao, Yudong Liu, Ming Gu, and Yu-Shen Liu. Implicit filtering for learning neural signed distance functions from 3D point clouds. In *European Conference on Computer Vision*, pages 234–251. Springer, 2025. 1, 2, 7

[36] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. 7

[37] Shitong Luo and Wei Hu. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4583–4592, 2021. 7

[38] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-Pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In *International Conference on Machine Learning*, pages 7246–7257, 2021. 1, 2, 3, 5, 7

[39] Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Reconstructing surfaces for sparse point clouds with on-surface priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6315–6325, 2022. 2, 7

[40] Baorui Ma, Yu-Shen Liu, Matthias Zwicker, and Zhizhong Han. Surface reconstruction from point clouds by learning predictive context priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6326–6337, 2022. 2, 7

[41] Gal Metzer, Rana Hanocka, Denis Zorin, Raja Giryes, Daniele Panozzo, and Daniel Cohen-Or. Orienting point clouds with dipole propagation. *ACM Transactions on Graphics*, 40(4):1–14, 2021. 2, 6

[42] Niloy J Mitra and An Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, pages 322–328, 2003. 2

[43] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. In *Advances in Neural Information Processing Systems*, pages 13032–13044, 2021. 2, 7

[44] Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J Mitra, and Maks Ovsjanikov. PointCleanNet: Learning to denoise and remove outliers from dense point clouds. In *Computer Graphics Forum*, pages 185–203. Wiley Online Library, 2020. 7

[45] Nico Schertler, Bogdan Savchynskyy, and Stefan Gumhold. Towards globally optimal normal orientations for large point clouds. In *Computer Graphics Forum*, pages 197–208. Wiley Online Library, 2017. 2, 6

[46] Shiyao Wang, Xiuping Liu, Jian Liu, Shuhua Li, and Junjie Cao. Deep patch-based global normal orientation. *Computer-Aided Design*, page 103281, 2022. 2, 6

[47] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 10130–10139, 2019. 7

[48] Yingrui Wu, Mingyang Zhao, Keqiang Li, Weize Quan, Tianqi Yu, Jianfeng Yang, Xiaohong Jia, and Dong-Ming Yan. CMG-Net: Robust normal estimation for point clouds via chamfer normal distance and multi-scale geometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6171–6179, 2024. 2, 7

[49] Dong Xiao, Zuoqiang Shi, Siyu Li, Bailin Deng, and Bin Wang. Point normal orientation and surface reconstruction by incorporating isovalue constraints to poisson equation. *Computer Aided Geometric Design*, page 102195, 2023. 2, 6

[50] Haoyi Xiu, Xin Liu, Weimin Wang, Kyoung-Sook Kim, and Masashi Matsuoka. MSECNet: Accurate and robust normal estimation for 3D point clouds by multi-scale edge conditioning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2535–2543, 2023. 2, 7

[51] Minfeng Xu, Shiqing Xin, and Changhe Tu. Towards globally optimal normal orientations for thin surfaces. *Computers & Graphics*, 75:36–43, 2018. 2

[52] Rui Xu, Zhiyang Dou, Ningna Wang, Shiqing Xin, Shuangmin Chen, Mingyan Jiang, Xiaohu Guo, Wenping Wang, and Changhe Tu. Globally consistent normal orientation for point clouds by regularizing the winding-number field. *ACM Transactions on Graphics (TOG)*, 42(4):1–15, 2023. 2, 6

[53] Jie Zhang, Junjie Cao, Xiuping Liu, Jun Wang, Jian Liu, and Xiquan Shi. Point cloud normal estimation via low-rank subspace clustering. *Computers & Graphics*, 37(6):697–706, 2013. 6, 7

[54] Jie Zhang, Junjie Cao, Xiuping Liu, He Chen, Bo Li, and Ligang Liu. Multi-normal estimation via pair consistency

voting. *IEEE Transactions on Visualization and Computer Graphics*, 25(4):1693–1706, 2018. 7

[55] Jie Zhang, Jun-Jie Cao, Hai-Rui Zhu, Dong-Ming Yan, and Xiu-Ping Liu. Geometry guided deep surface normal estimation. *Computer-Aided Design*, 142:103119, 2022. 2, 7

[56] Haoran Zhou, Honghua Chen, Yidan Feng, Qiong Wang, Jing Qin, Haoran Xie, Fu Lee Wang, Mingqiang Wei, and Jun Wang. Geometry and learning co-supported normal estimation for unstructured point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13238–13247, 2020. 2

[57] Haoran Zhou, Honghua Chen, Yingkui Zhang, Mingqiang Wei, Haoran Xie, Jun Wang, Tong Lu, Jing Qin, and Xiao-Ping Zhang. Refine-Net: Normal refinement neural network for noisy point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):946–963, 2022. 2

[58] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-aware unsigned distance functions progressively from raw point clouds. In *Advances in Neural Information Processing Systems*, pages 16481–16494, 2022. 2, 4, 6

[59] Jun Zhou, Wei Jin, Mingjie Wang, Xiuping Liu, Zhiyang Li, and Zhaobin Liu. Improvement of normal estimation for point clouds via simplifying surface fitting. *Computer-Aided Design*, page 103533, 2023. 2

[60] Junsheng Zhou, Baorui Ma, Shujuan Li, Yu-Shen Liu, Yi Fang, and Zhizhong Han. CAP-UDF: Learning unsigned distance functions progressively from raw point clouds with consistency-aware field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 2, 6, 7

[61] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics (TOG)*, 32(4):1–8, 2013. 7

[62] Runsong Zhu, Yuan Liu, Zhen Dong, Yuan Wang, Tengping Jiang, Wenping Wang, and Bisheng Yang. AdaFit: Rethinking learning-based normal estimation on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6118–6127, 2021. 1, 2, 6, 7