Implicit Filtering for Learning Neural Signed Distance Functions from 3D Point Clouds

Shengtao Li[1,2], Ge Gao[1,2]([✉]), Yudong Liu[1,2], Ming Gu[1,2], and Yu-Shen Liu[2]

[1] Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China
[2] School of Software, Tsinghua University, Beijing, China
`list21@mails.tsinghua.edu.cn, gaoge@tsinghua.edu.cn,`
`liuyd23@mails.tsinghua.edu.cn, guming@tsinghua.edu.cn,`
`liuyushen@tsinghua.edu.cn`

Abstract. Neural signed distance functions (SDFs) have shown powerful ability in fitting the shape geometry. However, inferring continuous signed distance fields from discrete unoriented point clouds still remains a challenge. The neural network typically fits the shape with a rough surface and omits fine-grained geometric details such as shape edges and corners. In this paper, we propose a novel non-linear implicit filter to smooth the implicit field while preserving high-frequency geometry details. Our novelty lies in that we can filter the surface (zero level set) by the neighbor input points with gradients of the signed distance field. By moving the input raw point clouds along the gradient, our proposed implicit filtering can be extended to non-zero level sets to keep the promise consistency between different level sets, which consequently results in a better regularization of the zero level set. We conduct comprehensive experiments in surface reconstruction from objects and complex scene point clouds, the numerical and visual comparisons demonstrate our improvements over the state-of-the-art methods under the widely used benchmarks. Project page: `https://list17.github.io/ImplicitFilter`.

Keywords: Implicit filtering · Signed distance functions · Point cloud reconstruction

1 Introduction

Reconstructing surfaces from 3D point clouds is an important task in 3D computer vision. Recently signed distance functions (SDFs) learned by neural networks have been a widely used strategy for representing high-fidelity 3D geometry. These methods train the neural networks to predict the signed distance for every position in the space by signed distances from ground truth or inferred from the raw 3D point cloud. With the learned signed distance field, we can obtain the surface by running the marching cubes algorithm [27] to extract the zero level set.

Without signed distance ground truth, inferring the correct gradient and distance for each query point could be hard. Since the gradient of the neural

| (a) Input | (b) SAP | (c) DIGS | (c) NeuralPull | (e) **Ours** |
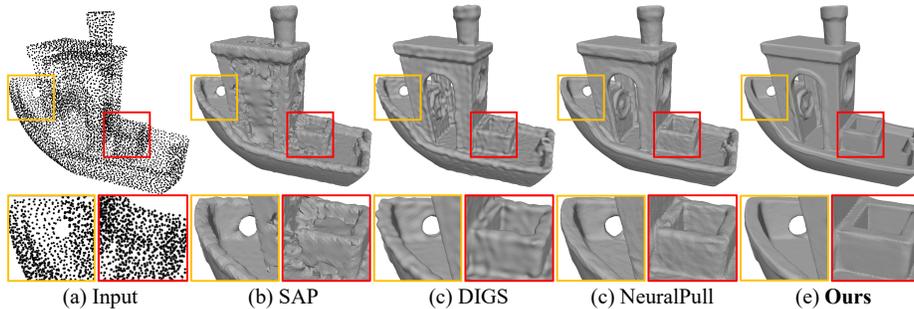
Fig. 1: Visualization of the comparisons on FAMOUS dataset [12]. Our implicit filter can improve the reconstruction by removing the noise and keeping the geometric details compared with other methods.

network also indicates the direction in which the signed distance field changes, recent works [1, 2, 4, 14, 29, 38] typically add constraints on the network gradient to learn a stable field. In terms of the rate at which the field is changing, the eikonal term [1,2,5,38] is widely used to ensure the norm of the gradient to be one everywhere. For the gradient direction constraint, some methods [4, 10] use the direction from the query point to the nearest point on the surface as guidance. Leveraging the continuity of the neural network and the gradient constraint, all these methods could reconstruct discrete points. However, the continuity cannot guarantee the prediction is correct everywhere. Therefore, reconstructed surfaces of previous methods usually contain noise and ignore geometry details when there are not enough points to guide the reconstruction, as shown in Fig. 1.

The above issue arises from the fact that these methods overlook the geometric information within the neighborhood but only focus on adding constraints on individual points to optimize the network. To resolve this issue, we introduce the bilateral filter for implicit fields that reduces surface noise while preserving the high-frequency geometric characteristics of the shape. Our designed implicit filter takes into account both the position of point clouds and the gradient of learned implicit fields. Based on the assumption of all input points lying on the surface, we can filter noise points on the zero level set by minimizing the weighted projection distance to gradients of the neighbor input points. Moreover, by moving the input points along the gradient of the field to other level sets, we can easily extend the filter to the whole field. This helps constrain the signed distance field near the surface and achieve better consistency through different level sets. To evaluate the effectiveness of our proposed implicit filtering, we validate it under widely used benchmarks including object and scene reconstructions. Our contributions are listed below.

– We introduce the implicit filter on SDFs to smooth the surface while preserving geometry details for learning better neural networks to represent shapes or scenes.

– We improve the implicit filter by extending it to non-zero level sets of signed distance fields. This regularization of the field aligns different level sets and provides better consistency within the whole SDF field.
– Both object and scene reconstruction experiments validate our implicit filter, demonstrating its effectiveness and ability to produce high-fidelity reconstruction results, surpassing the previous state-of-the-art methods.

2   Related Work

With the rapid development of deep learning, neural networks have shown great potential in surface reconstruction from 3D point clouds. In the following, we briefly review methods related to implicit learning for 3D shapes and reconstructions from point clouds.

Implicit Learning from 3D Supervision. The most commonly used strategy to train the neural network is to learn priors in a data-driven manner. These methods require signed distances or occupancy labels as 3D supervision to learn global priors [6, 12, 26, 31, 32] or local priors [7, 17, 18, 22, 35, 40, 41, 44, 45]. With large-scale training datasets, the neural network can perform well with similar shapes, but may not generalize well to unseen cases with large geometric variations. These models often have limited inputs that can be difficult to scale for varying sizes of point clouds.

Implicit Learning from Raw Point Clouds. Different from the supervised methods, we can learn implicit functions by overfitting neural networks on single point clouds globally or locally to learn SDFs [1–4, 10, 21, 28, 30, 34]. These unsupervised methods rely on neural networks to infer implicit functions without learning any priors. Therefore, apart from the guidance of original input point clouds, we also need constraints on the direction [3, 4, 10, 21] or the norm [1, 2, 30] of the gradients, specially designed priors [3, 28], or differentiable poisson solver [34] to infer SDFs. This unsupervised approach heavily depends on the fitting capability and continuity of neural networks. However, these SDFs lack accuracy because there is no reliable guidance available for each query point across the entire space when working with discrete point clouds. Therefore, deducing the correct geometry for free space becomes particularly crucial. Our implicit filtering enhances SDFs by inferring the geometric details through the implicit field information of neighbor points.

Feature Preserving Point Cloud Reconstruction. Early works [16, 23, 33] reconstruct point clouds with sharp features usually by point cloud consolidation. The key idea of these methods is to enhance the quality of point clouds with sharp features. One popular category is the local projection operation (LOP) [25] and its variants [15, 16, 23, 36]. The projection operator provides a stable and easily generalizable method for point cloud filtering, which is also the foundation of our implicit filter. The difference lies in that we do not need any normal or other priors and our filtering can be directly applied to implicit fields to extract high-fidelity meshes. Some other learning-based methods [47, 48] try to consolidate point clouds with edge points in a data-driven manner. Although

capable of generating high-quality point clouds, these methods still require a proper reconstruction method [13] to inherit the details in meshes.

With the advancement of deep learning in point cloud reconstruction, some approaches [5, 24, 38, 42] also explored employing neural networks to reconstruct high-precision models. FFN [39], SIREN [38], and IDF [43] introduce high-frequency features into the neural network in different ways to preserve the geometric details of the reconstructed shape. DIGS [5] and EPI [42] smooth the surface by using the divergence as guidance to alleviate the implicit surface roughness. Compared with these methods, we first introduce local geometric features through filtering to optimize the implicit field, so that we can achieve higher accuracy.

3   Method

Neural SDFs overview. This section will briefly describe the concepts we used in our implicit filtering. We focus on the SDF $f : \boldsymbol{R}^3 \to \boldsymbol{R}$ inferred from the point cloud $\boldsymbol{P} = \{\boldsymbol{p}_i | \boldsymbol{p}_i \in \boldsymbol{R}^3\}_{i=1}^N$ without ground truth signed distances and normals. f predicts a signed distance $s \in \boldsymbol{R}$ for an arbitrary query point $\boldsymbol{q}$, as formulated by $s = f_\theta(\boldsymbol{q})$, where θ denotes the parameters of the neural network.

The level set $\mathcal{S}_d$ of SDF is defined as a set of continuous query points with the same signed distance d, formulated as $\mathcal{S}_d = \{\boldsymbol{q} | f_\theta(\boldsymbol{q}) = d\}$. The goal of our implicit filtering is to smooth each level set with geometry details. Then we can extract the zero level set as a mesh by running the marching cubes algorithm [27].

Level set bilateral filtering. Filtering for 2D images replaces the intensity of each pixel with the weighted intensity values from nearby pixels. Different from images, the resolution of implicit fields is infinite and we need to find the neighborhood on each level set for filtering. By minimizing the following loss function,

$$L_{dist} = \frac{1}{N} \sum_{i=1}^N |f_\theta(\boldsymbol{p}_i)|, \tag{1}$$

we can approximate that all points in $\boldsymbol{P}$ are located on level set $\mathcal{S}_0$, which makes it feasible to find neighbor points on $\mathcal{S}_0$. For a given point $\bar{\boldsymbol{p}}$ on $\mathcal{S}_0$, one simple strategy of filtering is to average positions of neighbor points $\mathcal{N}(\bar{\boldsymbol{p}}, \mathcal{S}_0) \subset \boldsymbol{P}$ on $\mathcal{S}_0$ by a Gaussian filter based on relative positions as follows:

Fig. 2: By minimizing the weighted projection distance, our filter can preserve the sharp feature but the average method leads to a wrong result.

$$\bar{\boldsymbol{p}}_{\text{average}} = \frac{\sum_{\boldsymbol{p}_j \in \mathcal{N}(\bar{\boldsymbol{p}}, \mathcal{S}_0)} \boldsymbol{p}_j \phi(||\bar{\boldsymbol{p}} - \boldsymbol{p}_j||)}{\sum_{\boldsymbol{p}_j \in \mathcal{N}(\bar{\boldsymbol{p}}, \mathcal{S}_0)} \phi(||\bar{\boldsymbol{p}} - \boldsymbol{p}_j||)}, \tag{2}$$

where the Gaussian function ϕ is defined as $\phi(||\bar{\boldsymbol{p}} - \boldsymbol{p}_j||) = \exp\left(-\frac{||\bar{\boldsymbol{p}} - \boldsymbol{p}_j||^2}{\sigma_p^2}\right)$.
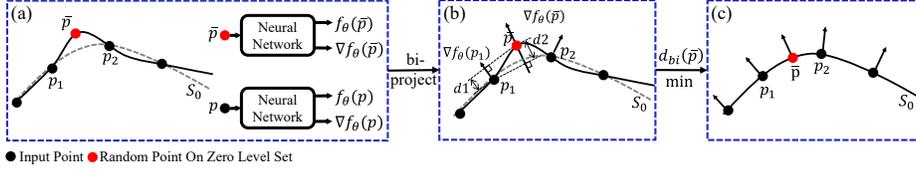
Fig. 3: Overview of filtering the zero level set. (a) We assume all input points lying on the surface and compute gradients as normals. (b) Calculating bidirectional projection distances $d1 = |\boldsymbol{n}_{p_j}^T(\bar{\boldsymbol{p}} - \boldsymbol{p}_j)|$, $d2 = |\boldsymbol{n}_{\bar{p}}^T(\bar{\boldsymbol{p}} - \boldsymbol{p}_j)|$ and the weights in Eq. (4). (c) By minimizing Eq. (4), we can remove the noise on the zero level set. The gradient ∇f_θ in this figure defaults to be regularized.

However, as depicted in Fig. 2, it is evident that this weighted mean position yields excessively smooth surfaces, causing sharp features and details to be further obscured. To keep the geometric details, our filtering operator suggests measuring the projection distance to the gradient of neighbor points as shown in Fig. 2 and Fig. 3(b). When calculating weights, it is vital to account for both the impact of relative positions and the gradient similarity. Following the principles of bilateral filtering, to compute the filtered point for $\bar{\boldsymbol{p}}$, we simply need to minimize the following distance equation:

$$d(\bar{\boldsymbol{p}}) = \frac{\sum_{\boldsymbol{p}_j \in \mathcal{N}(\bar{\boldsymbol{p}},\mathcal{S}_0)} |\boldsymbol{n}_{p_j}^T(\bar{\boldsymbol{p}} - \boldsymbol{p}_j)|\phi(||\bar{\boldsymbol{p}} - \boldsymbol{p}_j||)\psi(\boldsymbol{n}_{\bar{p}}, \boldsymbol{n}_{p_j})}{\sum_{\boldsymbol{p}_j \in \mathcal{N}(\bar{\boldsymbol{p}},\mathcal{S}_0)} \phi(||\bar{\boldsymbol{p}} - \boldsymbol{p}_j||)\psi(\boldsymbol{n}_{\bar{p}}, \boldsymbol{n}_{p_j})}, \tag{3}$$

where the gradient $\boldsymbol{n}_{\bar{p}}$, $\boldsymbol{n}_{p_j}$ and the Gaussian function ψ are defined as $\boldsymbol{n}_{\bar{p}} = \frac{\nabla f_\theta(\bar{\boldsymbol{p}})}{||\nabla f_\theta(\bar{\boldsymbol{p}})||}, \boldsymbol{n}_{p_j} = \frac{\nabla f_\theta(\boldsymbol{p}_j)}{||\nabla f_\theta(\boldsymbol{p}_j)||}, \psi(\boldsymbol{n}_{\bar{p}}, \boldsymbol{n}_{p_j}) = \exp\left(-\frac{1-\boldsymbol{n}_{\bar{p}}^T \boldsymbol{n}_{p_j}}{1-\cos(\sigma_n)}\right).$

In addition to projection to the gradient $\boldsymbol{n}_{p_j}$, we observe that the projection distance to $\boldsymbol{n}_{\bar{p}}$ can assist in learning a more stable gradient for point $\bar{p}$ which is also adopted in EAR [16]. Taking into account the bidirectional projection, our final bilateral filtering operator can be formulated as follows:

$$d_{bi}(\bar{\boldsymbol{p}}) = \frac{\sum\limits_{\boldsymbol{p}_j \in \mathcal{N}(\bar{\boldsymbol{p}},\mathcal{S}_0)} \left(|\boldsymbol{n}_{p_j}^T(\bar{\boldsymbol{p}} - \boldsymbol{p}_j)| + |\boldsymbol{n}_{\bar{p}}^T(\bar{\boldsymbol{p}} - \boldsymbol{p}_j)|\right)\phi(||\bar{\boldsymbol{p}} - \boldsymbol{p}_j||)\psi(\boldsymbol{n}_{\bar{p}}, \boldsymbol{n}_{p_j})}{\sum\limits_{\boldsymbol{p}_j \in \mathcal{N}(\bar{\boldsymbol{p}},\mathcal{S}_0)} \phi(||\bar{\boldsymbol{p}} - \boldsymbol{p}_j||)\psi(\boldsymbol{n}_{\bar{p}}, \boldsymbol{n}_{p_j})}. \tag{4}$$

Although similar filtering methods have been widely studied in applications such as point cloud denoising and resampling [16, 48], there are two critical problems when applying these methods in implicit fields:

1. Filtering the zero level set needs to sample points on the level set $\mathcal{S}_0$, which necessitates the resolution of the equation $f_\theta = 0$, or the utilization of the marching cubes algorithm [27]. Both methods pose challenges in achieving fast and uniform point sampling. For the randomly sampled point $\boldsymbol{q}$ on non-zero level set $\mathcal{S}_{f_\theta(\boldsymbol{q})}$, we can also not filter this level set since there are no neighbor points on $\mathcal{S}_{f_\theta(\boldsymbol{q})}$.
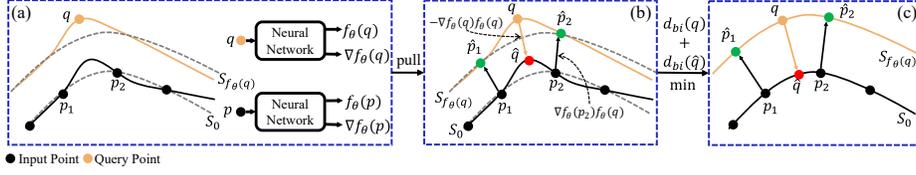
Fig. 4: Overview of sampling points. (a) Sampling query points near the surface. (b) Pulling the query point to the zero level set and input points to the level set where the query point is located. (c) Applying the filter on each level set. The gradient ∇f_θ in this figure defaults to be regularized.

2. The normals utilized in our filtering are derived from the gradients of the neural network f_θ. While the network typically offers reliable gradients, we may find that $\nabla f_\theta = 0$ is also the optimal solution to the minimum value of Eqs. (3) and (4). This degenerate solution is unexpected, as it implies a scenario where there is no surface when the gradient is zero everywhere.

We will focus on addressing the two issues in the subsequent sections.

Sampling points for filtering. Inspired by NeuralPull [4], we can pull a query point to the zero level set by the gradient of the neural network f_θ. For a given query point $\boldsymbol{q}$ as input, the pulled location $\hat{\boldsymbol{q}}$ can be formulated as follows:

$$\hat{\boldsymbol{q}} = \boldsymbol{q} - f_\theta(\boldsymbol{q})\nabla f_\theta(\boldsymbol{q})/||\nabla f_\theta(\boldsymbol{q})||. \tag{5}$$

The point $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$ lie respectively on level set $\mathcal{S}_{f_\theta(q)}$ and $\mathcal{S}_0$ as illustrate in Fig. 4(b). By adopting the sampling strategy in NeuralPull, we can generate samples $\boldsymbol{Q} = \{\boldsymbol{q}_i | \boldsymbol{q}_i \in \boldsymbol{R}^3\}_{i=1}^M$ on different level sets near the surface and pull them to $\mathcal{S}_0$ by Eq. (5), to obtain $\hat{\boldsymbol{Q}} = \{\hat{\boldsymbol{q}}_i | \hat{\boldsymbol{q}}_i = \boldsymbol{q}_i - f_\theta(\boldsymbol{q}_i)\nabla f_\theta(\boldsymbol{q}_i)/||\nabla f_\theta(\boldsymbol{q}_i)||, \boldsymbol{q}_i \in \boldsymbol{Q}\}_{i=1}^M$. Hence, we can filter the zero level set by minimizing Eq. (4) across all pulled query points $\hat{\boldsymbol{Q}}$, which is equivalent to optimizing the following loss:

$$L_{zero} = \sum\nolimits_{\hat{\boldsymbol{q}} \in \hat{\boldsymbol{Q}}} d_{bi}(\hat{\boldsymbol{q}}), \tag{6}$$

where for each $\hat{\boldsymbol{q}} \in \hat{\boldsymbol{Q}}$, $\mathcal{N}(\hat{\boldsymbol{q}}, \mathcal{S}_0)$ denotes finding the neighbors of $\hat{\boldsymbol{q}}$ within the input points $\boldsymbol{P}$, since $\boldsymbol{P}$ is assumed to be located on $\mathcal{S}_0$.

This filtering mechanism can be easily extended to non-zero level sets in a similar inverse manner. To be more specific, as for level set $S_{f_\theta(\boldsymbol{q})}$, the neighbor points for query point $\boldsymbol{q} \in \boldsymbol{Q}$ are required. These points should lie on the level set $S_{f_\theta(\boldsymbol{q})}$ same as $\boldsymbol{q}$, allowing us to filter the level set $S_{f_\theta(\boldsymbol{q})}$ using the same filter as described in Eq. (4).

However, obtaining $\mathcal{N}(\boldsymbol{q}, S_{f_\theta(\boldsymbol{q})})$ in $\boldsymbol{P}$ is not feasible, since all input points $\boldsymbol{P}$ are situated on the zero level set instead of the $S_{f_\theta(\boldsymbol{q})}$ level set. To address this issue, we propose a technique for identifying neighbors of $\boldsymbol{q}$ on level set $S_{f_\theta(\boldsymbol{q})}$, by projecting the input points $\boldsymbol{P}$ inversely onto the specific level set $S_{f_\theta(\boldsymbol{q})}$ based

on the gradient, as depicted in Fig. 4(b). The projected neighbor points can be represented as in Eq. (7). Filtering across multiple level sets helps to enhance the performance of our method by optimizing the consistency between different level sets within the SDF field, We further showcase this evidence in the ablation study detailed in Section Sec. 4.4.

$$\mathcal{N}(\boldsymbol{q}, S_{f_\theta(q)}) = \{\hat{\boldsymbol{p}}|\hat{\boldsymbol{p}} = \boldsymbol{p} + f_\theta(\boldsymbol{q})\frac{\nabla f_\theta(\boldsymbol{p})}{||\nabla f_\theta(\boldsymbol{p})||}, \boldsymbol{p} \in \mathcal{N}(\hat{\boldsymbol{q}}, \mathcal{S}_0))\}. \tag{7}$$

Based on the above analysis, we can filter the level sets $S_{f_\theta(\boldsymbol{q})}$ by minimizing Eq. (4) over all sample points $\boldsymbol{Q}$ through Eq. (7), equivalent to optimizing the following loss:

$$L_{field} = \sum\nolimits_{\boldsymbol{q} \in \boldsymbol{Q}} d_{bi}(\boldsymbol{q}). \tag{8}$$

It is worth noting that for a fixed query point $\boldsymbol{q}$, the pulled query point $\hat{\boldsymbol{q}}$ dynamically changes when training the neural network, which results in a time-consuming process to repeatedly conduct neighbor searching for $\hat{\boldsymbol{q}}$. To handle this matter, we substitute the $\mathcal{N}(\hat{\boldsymbol{q}}, \mathcal{S}_0)$ with $\mathcal{N}(NN(\boldsymbol{q}), \mathcal{S}_0)$, where $NN(\boldsymbol{q})$ denotes the nearest point of $\boldsymbol{q}$
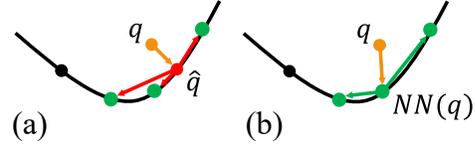
Fig. 5: (a) Searching neighbors directly for $\hat{\boldsymbol{q}}$. (b) Searching neighbors for $NN(\boldsymbol{q})$ instead of $\hat{\boldsymbol{q}}$.

within the point cloud $\boldsymbol{P}$ as shown in Fig. 5. While this substitution may introduce a slight bias for training, it also ensures the neighbor points are close to $\hat{\boldsymbol{q}}$, therefore this trade-off between efficiency and accuracy is reasonable.

Gradient constraint. The other problem of implicit filtering is gradient degeneration. Overfitting the neural network requires the SDF to be geometrically initialized. We can consider the initialized implicit field as the noisy field and apply our filter directly to train the network from the beginning to fit the raw point cloud by removing the 'noise'. However, if the denoise target is too complex, gradient degeneration will occur during the training process. Therefore, we need to add a constraint to the gradient of the SDF.

There are two ways for training the neural network to pull query points onto the surface based on NeuralPull [4] and CAP-UDF [49]. One is minimizing the distance between the pulled point $\hat{\boldsymbol{q}}$ and the nearest point $NN(\boldsymbol{q})$ as formulated below:

$$L_{pull} = \frac{1}{M} \sum_{i \in [1,M]} ||\hat{\boldsymbol{q}}_i - NN(\boldsymbol{q}_i)||_2. \tag{9}$$

The other is minimizing the Chamfer distance between moved query points and the raw point cloud:

$$L_{CD} = \frac{1}{M} \sum_{i \in [1,M]} \min_{j \in [1,N]} ||\hat{\boldsymbol{q}}_i - \boldsymbol{p}_j||_2 + \frac{1}{N} \sum_{j \in [1,N]} \min_{i \in [1,M]} ||\boldsymbol{p}_j - \hat{\boldsymbol{q}}_i||_2. \tag{10}$$

A stable SDF can be trained by the losses above since they are trying to move the query points to be in the same distribution with the point cloud, which can provide the constraint for our implicit filter. Here we choose L_{CD} since the filtered points are likely not the nearest points and L_{CD} is a more relaxed constraint.

Loss function. Finally, our loss function is formulated as:

$$L = L_{zero} + \alpha_1 L_{field} + \alpha_2 L_{dist} + \alpha_3 L_{CD}, \tag{11}$$

where α_1, α_2, and α_3 is the balance weights for our implicit filtering loss.

Implementation details. We employ a neural network similar to OccNet [31] and the geometric network initialization proposed in SAL [1] with a smaller radius the same as GridPull [10] to learn the SDF. We use the strategy in NeuralPull [4] to sample queries around each point p in P. We set the weight α_3 to 10 to constrain the learned SDF and α_1 and α_2 to 1. The parameters σ_n, σ_p are set to $15°, \max_{p_j \in \mathcal{N}(\bar{p}, \mathcal{S}_{f_\theta(\bar{p})})}(\|\bar{p} - p_j\|)$ respectively.

4   Experiments

We conducted experiments to assess the performance of our implicit filter for surface reconstruction from raw point clouds. The results are presented for general shapes in Sec. 4.1, real scanned raw data including 3D objects in Sec. 4.2, and complex scenes in Sec. 4.3. Additionally, ablation experiments were carried out to validate the theory and explore the impact of various parameters in Sec. 4.4.

4.1   Surface Reconstruction for Shapes

Datasets and metrics.
For surface reconstruction of general shapes from raw point clouds, we conduct evaluations on three widely used datasets including a subset of ShapeNet [8], ABC [20], and FAMOUS [12]. We use the same setting with NeuralPull [4] for the dataset ShapeNet. For datasets ABC and FA-MOUS, we use the train/test

Table 1: Comparisons on ABC and Famous datasets. The threshold of F-score (F-S.) is 0.01.

Methods	ABC			FAMOUS		
	CD_{L2}	CD_{L1}	F-S.	CD_{L2}	CD_{L1}	F-S.
P2S [12]	0.298	0.015	0.598	0.012	0.008	0.752
IGR [14]	2.675	0.063	0.448	1.474	0.044	0.573
NP [4]	0.095	0.011	0.673	0.100	0.012	0.746
PCP [3]	0.252	0.023	0.373	0.037	0.014	0.435
SIREN [38]	0.022	0.012	0.493	0.025	0.012	0.561
DIGS [5]	0.021	0.010	0.667	0.015	0.008	0.772
Ours	**0.011**	**0.009**	**0.691**	**0.008**	**0.007**	**0.778**

splitting released by Points2Surf [12] and we sample points directly from the mesh in the ABC dataset without other mesh preprocessing to keep the sharp features.

| (a) IGR | (b) P2S | (c) SIREN | (d) DIGS | (e) NP | (f) **Ours** | (g) GT |
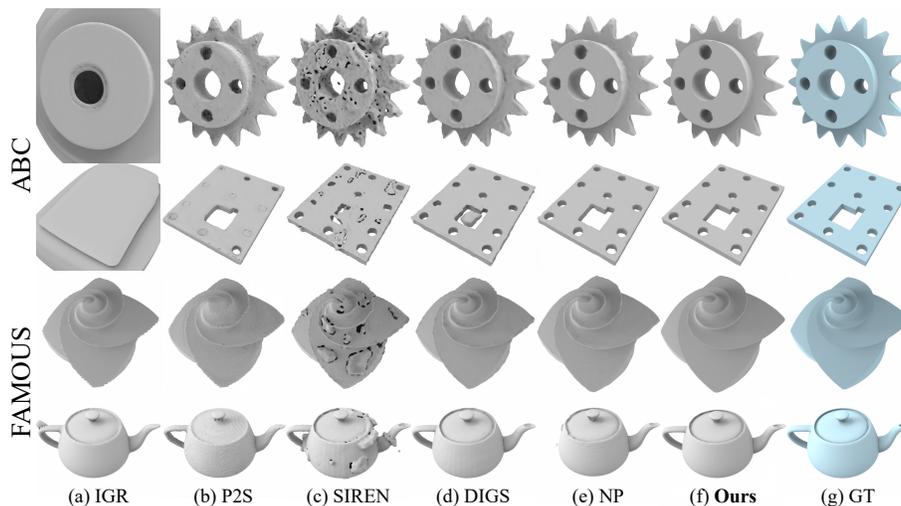
Fig. 6: Visual comparisons of surface reconstruction on ABC and FAMOUS datasets. Our method can reconstruct objects with sharp edges and less noise compared with other methods.

For evaluating the performance, we follow NeuralPull to sample 1×10^5 points from the reconstructed surfaces and the ground truth meshes on the ShapeNet dataset and sample 1×10^4 on the ABC and FAMOUS datasets. For the evaluation metrics, we use L1 and L2 Chamfer distance (CD_{L1} and CD_{L2}) to measure the error. Moreover, we adopt normal consistency (NC) and F-score to evaluate the accuracy of the reconstructed surface, the threshold is the same with NeuralPull.

Comparisons. To evaluate the validity of our implicit filter, we compare our method with a variety of methods including SPSR [19], Points2Surf (P2S) [12], IGR [14], NeuralPull (NP) [4], LPI [9], PCP [3], GridPull (GP) [10], SIREN [38], DIGS [5]. The quantitative results on ABC and FAMOUS datasets are shown in Tab. 1, and selectively visualized in Fig. 6. Our model reaches state-of-the-art performance on both datasets, accomplishing the goal of eliminating noise on each level set while preserving the geometric details.
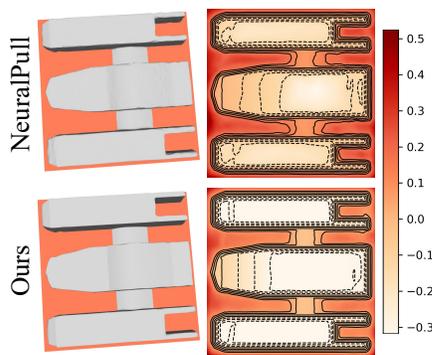
Fig. 7: Visualization of level sets on a cross section.

To more intuitively validate the efficacy of our filtering, we visualize the level sets on a cross section in Fig. 7. We also report the results on

Table 2: Comparisons on ShapeNet dataset.

	SPSR [19]	NP [4]	LPI [9]	PCP [3]	GP [10]	Ours
$CD_{L2} \times 100$	0.286	0.038	0.0171	0.0136	0.0086	**0.0032**
NC	0.866	0.939	0.9596	0.9590	0.9723	**0.9779**
F-Score (0.002)	0.407	0.961	0.9912	0.9871	0.9896	**0.9976**
F-Score (0.004)	0.618	0.976	0.9957	0.9899	0.9923	**0.9985**

(a) Input   (b) SPSR   (c) PCP   (d) GridPull   (e) NeuralPull   (f) **Ours**   (g) GT
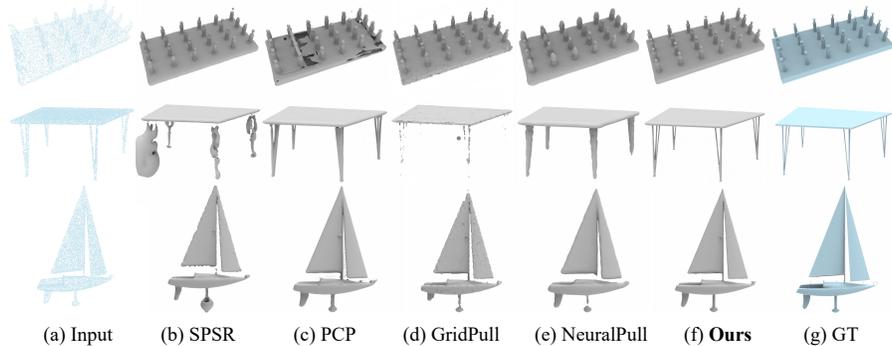
Fig. 8: Visual comparisons of surface reconstruction on ShapeNet dataset.

ShapeNet which contains over 3000 objects in terms of CD_{L2}, NC, and F-Score with thresholds of 0.002 and 0.004 in Tab. 2. The detailed comparison for each class of ShapeNet can be found in the supplementary material. Our method outperforms previous methods over most classes. The visualization comparisons in Fig. 8 show that our method can reconstruct a smoother surface with fine details.

To validate the effect of our filter on sharp geometric features. We evaluate the edge points by the edge Chamfer distance metric used in [11]. We sample 100k points uniformly on the surface of both the reconstructed mesh and ground truth. The edge point p is calculated by finding whether there exists a point $q \in \mathcal{N}_\epsilon(p)$ satisfied $|\boldsymbol{n}_q \boldsymbol{n}_p| < \sigma$, where $\mathcal{N}_\epsilon(p)$ represents the neighbor points within distance ϵ from p. The results are shown in Tab. 3 and visualized in Fig. 9. We set $\epsilon = 0.01$ and $\sigma = 0.1$.

4.2   Surface Reconstruction for Real Scans

Dataset and metrics. For surface reconstruction of real point cloud scans, we follow VisCo [37] to evaluate our method under the Surface Reconstruction Benchmarks (SRB) [46]. We use Chamfer and Hausdorff distances (CD_{L1} and HD) between the reconstruction meshes and the ground truth. Furthermore, we report their corresponding one-sided distances ($d_{\overrightarrow{C}}$ and $d_{\overrightarrow{H}}$) between the reconstructed meshes and the input noisy point cloud.

Table 3: Edge Chamfer distance comparisons on ABC dataset, $ECD_{L2} \times 100$.

Methods	P2S [12]	IGR [14]	NP [4]	PCP [3]	SIREN [38]	DIGS [5]	Ours
ECD_{L1}	0.0496	0.0835	0.0501	0.0628	0.0695	0.0786	**0.0256**
ECD_{L2}	1.055	2.365	1.255	1.265	1.407	2.493	**0.399**

(a) IGR   (b) P2S   (c) SIREN   (d) DIGS   (e) NP   (f) **Ours**   (g) GT
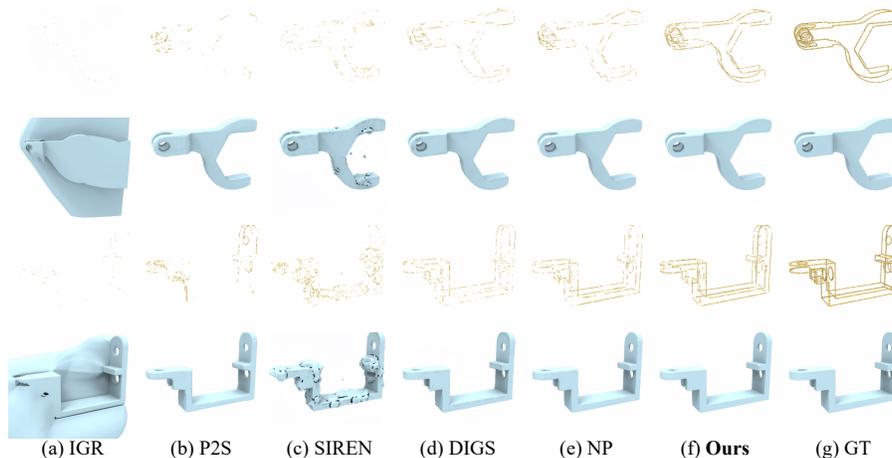
Fig. 9: Visual comparisons of edge points and reconstruction results.

Comparisons. We compare our method with state-of-the-art methods under the real scanned SRB dataset, including IGR [14], SPSR [19], Shape As Points (SAP) [34], NeuralPull (NP) [4], and GridPull (GP) [10]. The numerical comparisons are shown in Tab. 4, where we achieve the best accuracy in most cases. The visual comparisons in Fig. 10 demonstrate that our method can reconstruct a continuous and smooth surface with geometry details.

(a) Input   (b) SAP   (c) SPSR   (d) NeuralPull   (e) **Ours**
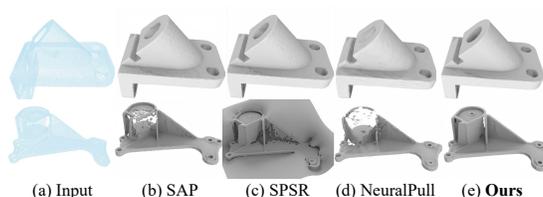
Fig. 10: Visual comparisons on SRB dataset.

4.3   Surface Reconstruction for Scenes

Dataset and metrics. To further demonstrate the advantage of our method in the surface reconstruction of real scene scans, we conduct experiments using the 3D Scene dataset. The 3D Scene dataset is a challenging real-world dataset with complex topology and noisy open surfaces. We uniformly sample 1000 points per m^2 of each scene as the input and follow PCP [3] to sample 1M points on both the reconstructed and the ground truth surfaces. We leverage L1 and L2

Table 4: Comparisons on SRB dataset.

		SPSR [19]	IGR [14]	SIREN [38]	VisCo [37]	SAP [34]	NP [4]	GP [10]	DIGS [5]	Ours
Anchor	CD_{L1}	0.60	0.22	0.32	0.21	0.12	0.122	0.093	0.063	**0.052**
	HD	14.89	4.71	8.19	3.00	2.38	3.243	1.804	1.447	**1.232**
	$d_{\vec{C}}$	0.60	0.12	0.10	0.15	0.08	0.061	0.066	0.030	**0.025**
	$d_{\vec{H}}$	14.89	1.32	2.432	1.07	0.83	3.208	0.460	0.270	**0.265**
Daratech	CD_{L1}	0.44	0.25	0.21	0.21	0.26	0.375	0.062	0.049	**0.051**
	HD	7.24	4.01	4.30	4.06	0.87	3.127	**0.648**	0.858	0.751
	$d_{\vec{C}}$	0.44	0.08	0.09	0.14	0.04	0.746	0.039	0.025	**0.028**
	$d_{\vec{H}}$	7.24	1.59	1.77	1.76	0.41	3.267	**0.293**	0.441	0.423
DC	CD_{L1}	0.27	0.17	0.15	0.15	0.07	0.157	0.066	0.042	**0.041**
	HD	3.10	2.22	2.18	2.22	1.17	3.541	1.103	**0.667**	0.815
	$d_{\vec{C}}$	0.27	0.09	0.06	0.09	0.04	0.242	0.036	0.022	**0.019**
	$d_{\vec{H}}$	3.10	2.61	2.76	2.76	**0.53**	3.523	0.539	0.729	**0.724**
Gargoyle	CD_{L1}	0.26	0.16	0.17	0.17	0.07	0.080	0.063	0.047	**0.044**
	HD	6.80	3.52	4.64	4.40	1.49	1.376	1.129	**0.971**	1.089
	$d_{\vec{C}}$	0.26	0.06	0.08	0.11	0.05	0.063	0.045	0.028	**0.022**
	$d_{\vec{H}}$	6.80	0.81	0.91	0.96	0.78	0.475	0.700	0.271	**0.246**
Lord Quas	CD_{L1}	0.20	0.12	0.17	0.12	0.05	0.064	0.047	0.031	**0.030**
	HD	4.61	1.17	0.82	1.06	0.98	0.822	0.569	**0.496**	0.554
	$d_{\vec{C}}$	0.20	0.07	0.12	0.07	0.04	0.053	0.031	0.017	**0.014**
	$d_{\vec{H}}$	4.61	0.98	0.76	0.64	0.51	0.508	0.370	**0.181**	0.230

(a) GridPull	(b) PCP	(c) NeuralPull	(d) **Ours**	(e) GT
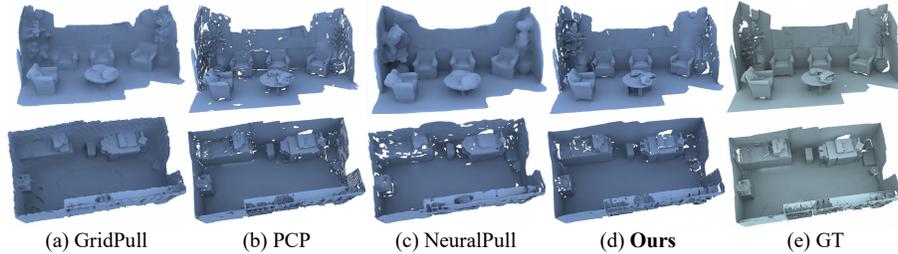
Fig. 11: Visual comparisons of surface reconstruction on 3D Scene dataset.

Chamfer distance (CD_{L1}, CD_{L2}) and normal consistency (NC) to evaluate the reconstruction quality.

Comparisons. We compare our method with the state-of-the-art methods ConvONet [35], LIG [18], DeepLS [7], NeuralPull (NP) [4], PCP [3], GridPull (GP) [10]. The numerical comparisons in Tab. 5 demonstrate our superior performance in all scenes even compared with the local-based methods. We further present visual comparisons in Fig. 11. The visualization further shows that our method can achieve smoother with high-fidelity surfaces in complex scenes. It should be noted that the surface we extract here is not the zero level set but the 0.001 level set since the scene is not watertight. For NeuralPull we use the threshold of 0.005 instead of 0.001 to extract the complete surface therefore the mesh looks thicker.

Table 5: Comparisons on 3D Scene dataset, $CD_{L2} \times 1000$.

	Burghers			Lounge			Copyroom			Stonewall			Totempole		
	CD_{L2}	CD_{L1}	NC	CD_{L2}	CD_{L1}	NC	CD_{L2}	CD_{L1}	NC	CD_{L2}	CD_{L1}	NC	CD_{L2}	CD_{L1}	NC
ConvONet [35]	27.46	0.079	0.907	9.54	0.046	0.894	10.97	0.045	0.892	20.46	0.069	0.905	2.054	0.021	0.943
LIG [18]	3.055	0.045	0.835	9.672	0.056	0.833	3.61	0.036	0.810	5.032	0.042	0.879	9.58	0.062	0.887
DeepLS [7]	0.401	0.017	0.920	6.103	0.053	0.848	0.609	0.021	0.901	0.320	0.015	0.954	0.601	0.017	0.950
GP [10]	1.367	0.028	0.873	4.684	0.053	0.827	2.327	0.030	0.857	2.234	0.024	0.913	2.278	0.034	0.878
PCP [3]	1.339	0.031	0.929	0.432	0.014	**0.934**	0.405	0.014	**0.914**	0.266	0.014	**0.957**	1.089	0.029	**0.954**
NP [4]	0.897	0.025	0.883	0.855	0.022	0.887	0.479	0.018	0.862	0.434	0.018	0.929	1.604	0.032	0.923
Ours	**0.133**	**0.011**	**0.934**	**0.120**	**0.008**	0.926	**0.111**	**0.009**	0.913	**0.082**	**0.009**	**0.957**	**0.203**	**0.013**	0.944

(a) Input Points   (b) L_{pull} (c) L_{CD} (c) $L_{CD} + L_{zero}$ (d) $L_{CD} + L_{zero} + L_{field}$
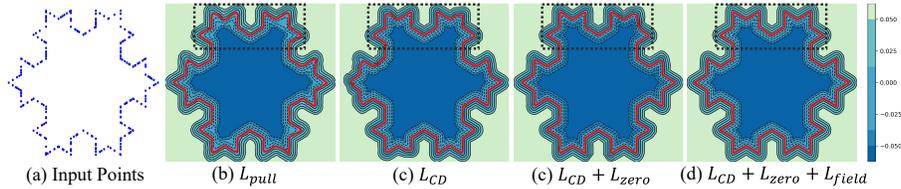
Fig. 12: The 2D level sets show the distance field learned by different losses. The red lines represent the learned zero level set.

4.4 Ablation Studies

We conduct ablation studies on the FAMOUS dataset to demonstrate the effectiveness of our proposed implicit filter and explore the effect of some important hyperparameters. We report the performance in terms of L1 and L2 Chamfer distance ($CD_{L1}, CD_{L2} \times 10^3$), normal consistency (NC), and F-Score (F-S.).

Effect of Eikonal loss. We select the L_{CD} to prevent the degeneration of the gradient since it both constrains the value and the gradient of the SDF. It also guides how to pull

Table 6: Effect of the Eikonal term.

Loss	CD_{L1}	CD_{L2}	F-S.	NC
w/ Eikonal, w/o CD	0.009	0.021	0.738	0.899
w/ Eikonal, w/ CD	0.008	0.009	0.774	0.910
w/o Eikonal, w/ CD	**0.007**	**0.008**	**0.778**	**0.911**

the query point onto the surface. Therefore we omit the Eikonal term used in previous methods like the IGR [14], SIREN [38], and DIGS [5] which have no other direct supervision for the gradient. To verify this selection, we conduct the following experiments by trade-off these two functions. With the experimental results in Tab. 6, we find that only applying the Eikonal term is not as effective as CD alone. At the same time combining the Eikonal term with CD does not further enhance the experiment results, but the difference is small.

Effect of level set filtering. To justify the effectiveness of each term in our loss function. We report the results trained by different combinations in Tab. 7. The L_{CD} is more applicable for training SDF from raw point clouds. The zero-level filter can help remove the noise and keep the geometric features. Filtering across non-zero level sets can improve the overall consistency of the entire signed distance field. Since we assume all input points lie on the surface, the function L_{dist} is also necessary. Fig. 12 shows a 2D comparison of these losses, showing

that our filter loss functions can reconstruct a field that is aligned at all level sets and maintains geometric characteristics.

Effect of the bidirectional projection. To validate our bidirectional projection distance, we report the results in Tab. 8. The numerical comparisons show that projecting the distance to both normals can improve the reconstruction quality. Note that only using $d(\bar{\boldsymbol{p}})$ can also improve the results.

Table 7: Effect of different losses.

Loss	CD_{L1}	CD_{L2}	F-S.	NC
L_{pull}	0.012	0.083	0.742	0.884
L_{CD}	0.010	0.031	0.757	0.891
$L_{CD} + L_{zero}$	0.008	0.018	0.772	0.905
$L_{CD} + L_{zero} + L_{field}$	0.008	0.011	0.769	0.908
Ours	**0.007**	**0.008**	**0.778**	**0.911**

Table 8: Effect of bidirectional projection.

	$d(\bar{\boldsymbol{p}})$	$d_{bi}(\bar{\boldsymbol{p}})$
CD_{L1}	0.010	**0.007**
CD_{L2}	0.024	**0.008**
F-S.	0.726	**0.778**
NC	0.890	**0.911**

Weight of level set projection loss. We explore the effect of the L_{CD} loss function by adjusting the weight α_3 in Eq. (11). We report our results with different candidates $\{0, 1, 10\}$ in Tab. 9, where 0 means we do not use the L_{CD} to constrain the gradient. The comparisons in Tab. 9 show that although our implicit filter can directly learn SDFs, it is better to adopt the L_{CD} for a more stable field. However, if the weight is too large, the filtering effect will decrease. It is recommended to select weights ranging from 1 to 10, which is usually adequate. For the weights α_1 and α_2, setting them to 1 is always necessary.

Effect of filter parameters. We compare the effect of different parameters σ_n, σ_p in Tab. 10. The diagonal weight for σ_p means the length of the diagonal of the bounding box for the local patch mentioned in [48]. The results indicate that the method is relatively robust to parameter variation in a certain range.

Table 9: Effect of weight α_3.

α_3	CD_{L1}	CD_{L2}	F-S.	NC
0	0.008	0.013	0.758	0.903
1	**0.007**	0.011	0.772	0.910
10	**0.007**	**0.008**	**0.778**	**0.911**
100	0.008	0.009	0.774	0.909

Table 10: Effect of filter parameters σ_n and σ_p.

		CD_{L1}	CD_{L2}	F-S.	NC
σ_n	15°	**0.007**	**0.008**	**0.778**	**0.911**
	30°	**0.007**	0.011	0.771	0.907
	45°	0.008	0.012	0.764	0.903
	60°	0.008	0.010	0.767	0.901
σ_p	max	**0.007**	**0.008**	**0.778**	**0.911**
	diagonal	0.008	0.011	0.763	0.904

5   Conclusion

We introduce implicit filtering on SDFs to reduce the noise of the signed distance field while preserving geometry features. We filter the distance field by minimizing the weighted bidirectional projection distance, where we can generate sampling points on the zero level set and neighbor points on non-zero level sets by the pulling procedure. By leveraging the Chamfer distance, we address the issue of gradient degeneration problem. The visual and numerical comparisons demonstrate our effectiveness and superiority over state-of-the-art methods.

Acknowledgements

References

1. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
2. Atzmon, M., Lipman, Y.: SALD: sign agnostic learning with derivatives. In: 9th International Conference on Learning Representations, ICLR 2021 (2021)
3. Baorui, M., Yu-Shen, L., Matthias, Z., Zhizhong, H.: Surface reconstruction from point clouds by learning predictive context priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
4. Baorui, M., Zhizhong, H., Yu-Shen, L., Matthias, Z.: Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In: International Conference on Machine Learning (ICML) (2021)
5. Ben-Shabat, Y., Hewa Koneputugodage, C., Gould, S.: Digs: Divergence guided shape implicit neural representation for unoriented point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19323–19332 (2022)
6. Boulch, A., Marlet, R.: Poco: Point convolution for surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6302–6314 (June 2022)
7. Chabra, R., Lenssen, J.E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., Newcombe, R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16. pp. 608–625. Springer (2020)
8. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
9. Chao, C., Yu-shen, L., Zhizhong, H.: Latent partition implicit with surface codes for 3d representation. In: European Conference on Computer Vision (ECCV) (2022)
10. Chen, C., Liu, Y.S., Han, Z.: Gridpull: Towards scalability in learning implicit representations from 3d point clouds. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2023)
11. Chen, Z., Tagliasacchi, A., Zhang, H.: Bsp-net: Generating compact meshes via binary space partitioning. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
12. Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N.J., Wimmer, M.: Points2Surf: Learning implicit surfaces from point clouds. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 108–124. Springer International Publishing, Cham (2020)
13. Fleishman, S., Cohen-Or, D., Silva, C.T.: Robust moving least-squares fitting with sharp features. ACM transactions on graphics (TOG) **24**(3), 544–552 (2005)

14. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. In: Proceedings of Machine Learning and Systems 2020, pp. 3569–3579 (2020)
15. Huang, H., Li, D., Zhang, H., Ascher, U., Cohen-Or, D.: Consolidation of unorganized point clouds for surface reconstruction. ACM transactions on graphics (TOG) **28**(5), 1–7 (2009)
16. Huang, H., Wu, S., Gong, M., Cohen-Or, D., Ascher, U., Zhang, H.: Edge-aware point set resampling. ACM transactions on graphics (TOG) **32**(1), 1–12 (2013)
17. Huang, J., Gojcic, Z., Atzmon, M., Litany, O., Fidler, S., Williams, F.: Neural kernel surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4369–4379 (2023)
18. Jiang, C.M., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T.: Local implicit grid representations for 3d scenes. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020)
19. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG) **32**(3), 1–13 (2013)
20. Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., Panozzo, D.: Abc: A big cad model dataset for geometric deep learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9601–9611 (2019)
21. Koneputugodage, C.H., Ben-Shabat, Y., Campbell, D., Gould, S.: Small steps and level sets: Fitting neural surface models with point guidance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21456–21465 (2024)
22. Li, S., Gao, G., Liu, Y., Liu, Y.S., Gu, M.: Gridformer: Point-grid transformer for surface reconstruction. In: Proceedings of the AAAI Conference on Artificial Intelligence (2024)
23. Liao, B., Xiao, C., Jin, L., Fu, H.: Efficient feature-preserving local projection operator for geometry reconstruction. Computer-Aided Design **45**(5), 861–874 (2013)
24. Lindell, D.B., Van Veen, D., Park, J.J., Wetzstein, G.: Bacon: Band-limited coordinate networks for multiscale scene representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16252–16262 (2022)
25. Lipman, Y., Cohen-Or, D., Levin, D., Tal-Ezer, H.: Parameterization-free projection for geometry reconstruction. ACM Transactions on Graphics (TOG) **26**(3), 22–es (2007)
26. Liu, S.L., Guo, H.X., Pan, H., Wang, P.S., Tong, X., Liu, Y.: Deep implicit moving least-squares functions for 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1788–1797 (2021)
27. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353 (1998)
28. Ma, B., Liu, Y.S., Han, Z.: Reconstructing surfaces for sparse point clouds with on-surface priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6315–6325 (2022)
29. Ma, B., Zhou, J., Liu, Y.S., Han, Z.: Towards better gradient consistency for neural signed distance functions via level set alignment. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
30. Marschner, Z., Sellán, S., Liu, H.T.D., Jacobson, A.: Constructive solid geometry on neural signed distance fields. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–12 (2023)

31. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)

32. Mi, Z., Luo, Y., Tao, W.: Ssrnet: Scalable 3d surface reconstruction network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 970–979 (2020)

33. Öztireli, A.C., Guennebaud, G., Gross, M.: Feature preserving point set surfaces based on non-linear kernel regression. In: Computer graphics forum. vol. 28, pp. 493–501. Wiley Online Library (2009)

34. Peng, S., Jiang, C., Liao, Y., Niemeyer, M., Pollefeys, M., Geiger, A.: Shape as points: A differentiable poisson solver. Advances in Neural Information Processing Systems **34**, 13032–13044 (2021)

35. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: European Conference on Computer Vision (ECCV) (2020)

36. Preiner, R., Mattausch, O., Arikan, M., Pajarola, R., Wimmer, M.: Continuous projection for fast l1 reconstruction. ACM Trans. Graph. **33**(4), 47–1 (2014)

37. Pumarola, A., Sanakoyeu, A., Yariv, L., Thabet, A., Lipman, Y.: Visco grids: Surface reconstruction with viscosity and coarea grids. Advances in Neural Information Processing Systems **35**, 18060–18071 (2022)

38. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Neural Information Processing Systems,Neural Information Processing Systems (Jun 2020)

39. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems **33**, 7537–7547 (2020)

40. Tang, J., Lei, J., Xu, D., Ma, F., Jia, K., Zhang, L.: Sa-convonet: Sign-agnostic optimization of convolutional occupancy networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)

41. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Stoll, C., Theobalt, C.: Patchnets: Patch-based generalizable deep implicit 3d shape representations. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16. pp. 293–309. Springer (2020)

42. Wang, X., Cheng, Y., Wang, L., Lu, J., Xu, K., Xiao, G.: Edge preserving implicit surface representation of point clouds. arXiv preprint arXiv:2301.04860 (2023)

43. Wang, Y., Rahmann, L., Sorkine-Hornung, O.: Geometry-consistent neural shape representation with implicit displacement fields. In: The Tenth International Conference on Learning Representations. OpenReview (2022)

44. Wang, Z., Zhou, S., Park, J.J., Paschalidou, D., You, S., Wetzstein, G., Guibas, L., Kadambi, A.: Alto: Alternating latent topologies for implicit 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 259–270 (2023)

45. Williams, F., Gojcic, Z., Khamis, S., Zorin, D., Bruna, J., Fidler, S., Litany, O.: Neural fields as learnable kernels for 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18500–18510 (2022)

46. Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., Panozzo, D.: Deep geometric prior for surface reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10130–10139 (2019)

47. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: Ec-net: an edge-aware point set consolidation network. In: Proceedings of the European conference on computer vision (ECCV). pp. 386–402 (2018)
48. Zhang, D., Lu, X., Qin, H., He, Y.: Pointfilter: Point cloud filtering via encoder-decoder modeling. IEEE Transactions on Visualization and Computer Graphics **27**(3), 2015–2027 (2020)
49. Zhou, J., Ma, B., Liu, Y.S., Fang, Y., Han, Z.: Learning consistency-aware unsigned distance functions progressively from raw point clouds. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)