NeuralIndicator: Implicit Surface Reconstruction from Neural Indicator Priors

Shi-Sheng Huang [1] **Guo Chen** [1] **Li Heng Chen** [1] **Hua Huang** [1]

Abstract

The neural implicit surface reconstruction from unorganized points is still challenging, especially when the point clouds are *incomplete* and/or *noisy* with complex topology structure. Unlike previous approaches performing neural implicit surface learning relying on *local* shape priors, this paper proposes to utilize *global* shape priors to regularize the neural implicit function learning for more reliable surface reconstruction. To this end, we first introduce a differentiable module to generate a smooth indicator function, which *globally* encodes both the indicative prior and local SDFs of the entire input point cloud. Benefit from this, we propose a new framework, called *NeuralIndicator*, to jointly learn both the smooth indicator function and neural implicit function simultaneously, using the global shape prior encoded by smooth indicator function to effectively regularize the neural implicit function learning, towards reliable and high-fidelity surface reconstruction from unorganized points without any normal information. Extensive evaluations on synthetic and real-scan datasets show that our approach consistently outperforms previous approaches, especially when point clouds are incomplete and/or noisy with complex topology structure.

1. Introduction

Surface reconstruction from unordered 3D point clouds continues to be a critical research topic in the computer graphics and vision communities, with many impressive approaches achieving huge progress in the past few decades (Carr et al., 2001; Kazhdan et al., 2006; Kazhdan & Hoppe, 2013; Huang et al., 2019; Lu et al., 2019). The recent works that formulate the latent geometry as a neural implicit function (i.e. signed distance function, SDF) (Park et al., 2019), have demonstrated the benefits to represent shapes in arbitrary topology and at any resolution, and inspired subsequent works for point cloud surface reconstruction in an either supervised (Dai & Nießner, 2019; Erler et al., 2020; Jiang et al., 2020; Duan et al., 2020; Peng et al., 2020; Chen & Zhang, 2021; Chen et al., 2022; Xiao et al., 2022; Huang et al., 2023) or unsupervised (Williams et al., 2019; Atzmon & Lipman, 2020; Gropp et al., 2020; Hanocka et al., 2020; Atzmon & Lipman, 2021; Wang et al., 2021; Peng et al., 2021; Zhao et al., 2021; Ma et al., 2022a; Chen et al., 2023a; Wang et al., 2023) manner.

Compared with the supervised approaches, the unsupervised learning doesn't need any extra supervised label and gets rid of generalization problem, which has receiving more and more research attention these years. SAL(Atzmon & Lipman, 2020) and its variants (Gropp et al., 2020; Atzmon & Lipman, 2021; Lipman, 2021) are probably one of first few attempts to perform such unsupervised learning. Recently, some works introduce geometry priors to regularize neural implicit function learning, such as deep geometric priors (Williams et al., 2019), self-similarity pattern (Hanocka et al., 2020; Zhao et al., 2021), Iso-Points (Wang et al., 2021), predictive context prior (PCP (Ma et al., 2022a)), On-Surface prior (Ma et al., 2022b), Thin Plate function (Chen et al., 2023a) or MLS function (Wang et al., 2023) etc, thus achieving much better surface reconstruction quality. However, such approaches still couldn't achieve reliable surface reconstruction when input point cloud is incomplete and/or noisy with complex topology.

In this paper, we introduce a new framework to learn the neural implicit function, which aims at more reliable and accurate surface reconstruction even for incomplete and/or noisy points with complex topology structure. Different from previous approaches that use local shape priors, our motivation is to explore more effective *global* shape priors from the entire shape itself, to regularize the neural implicit function learning. Our key observation is that indicator function encodes the *global* geometry and topology priors of the entire shape, which can be used to guide the neural implicit function learning. However, a naive indicator function (Peng et al., 2021) is discontinues without exact gradient definition, which would easily introduce discontinues surface reconstruction for the neural implicit function learning.

---

[*] Equal contribution [1] School of Artificial Intelligence, Beijing Normal University, Beijing, China. Correspondence to: Hua Huang <huahuang@bnu.edu.cn>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

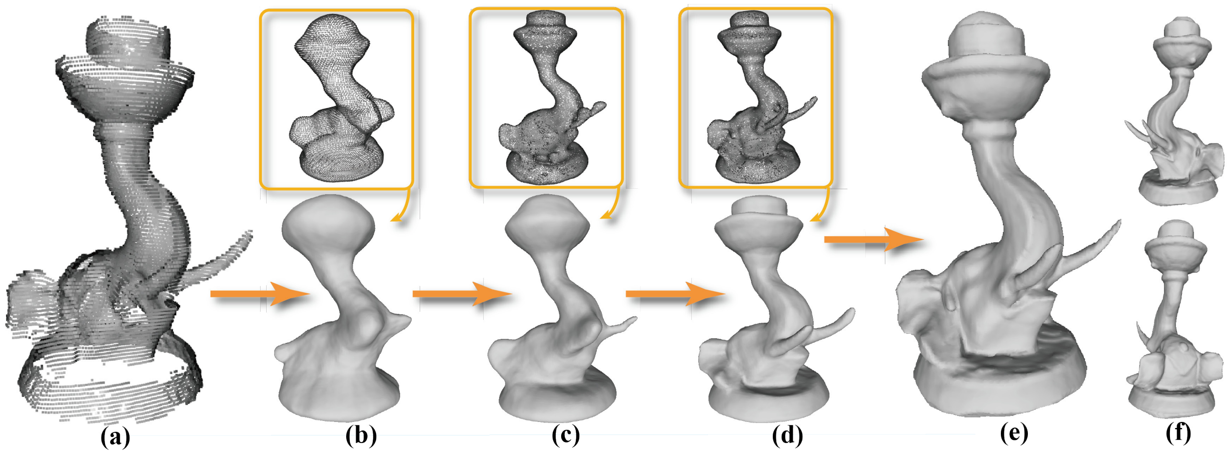

Figure 1. We present a new neural implicit function learning strategy, called *NeuralIndicator*, for surface reconstruction directly from unordered point cloud in an unsupervised manner. Starting from a given incomplete point cloud (a), our *NeuralIndicator* learns the *global* shape priors from the entire point cloud via a smooth indicator function, and simultaneously use such effective priors (top colored) to regularize the neural implicit function learning in an iterative way (b)-(d) for the final *complete* surface reconstruction (e), with fine geometric details preserved as show in two different views (f). The joint learning of both smooth indicator function and neural implicit function enables reliable surface reconstruction, even for incomplete and/or noisy scanned point clouds with complex topology structure.

To overcome such drawback, we introduce a more reliable *smooth* indicator function, which is obtained by applying smooth constraints on the gradient domain. The smooth indicator function approximates the local SDFs, and inherits the global indicative priors at the same time, which serves a suitable basis to encode the global shape priors for the entire point cloud. What's more, we also propose a differentiable generation module, which enables differentiable smooth indicator function generation (SIFG) from a set of oriented points (Fig. 1), with the oriented points served as learning parameters. Based on the SIFG, we further introduce a new joint learning framework, called *NeuralIndicator*, which leverages effective neural indicator priors to learn both the smooth indicator function and neural implicit function. With such regularization, our *NeuralIndicator* can reliably *fit* the input point cloud for high-fidelity surface reconstruction, even for incomplete and/or noisy input with complex topology structure. Note that both the smooth indicator function and neural implicit function are learnt directly from the input point cloud (Fig. 2) , in an unsupervised fashion without any point normals input or supervision.

To show the effectiveness, we have extensively evaluated our approach on both the public synthetic (ABC (Koch et al., 2019), FAMOUS (Erler et al., 2020), Reconbench (Berger et al., 2013), Thingi10K (Zhou & Jacobson, 2016)) and challenging real-scan (DPoint (Wu et al., 2015)) point cloud datasets, by comparing with previous neural implicit surface reconstruction approaches such as Point2Mesh (Hanocka et al., 2020), Iso-Points (Wang et al., 2021), SAP (Peng et al., 2021), PCP (Ma et al., 2022a), NeuralTPS (Chen et al., 2023a) and Neural-IMLS (Wang et al., 2023), and also the traditional surface reconstruction approach such as PSR (Kazhdan et al., 2006) and iPSR (Hou et al., 2022) respectively. Our approach consistently outperforms those previous approaches for surface reconstruction from point cloud in both quantitatively and qualitatively, even when input point clouds are incomplete and/or noisy with complex topology structure.

2. Related Work

Implicit Surface Reconstruction. The traditional implicit surface representation can generate smooth and water-tight shape surfaces, and has been the mainstream method since the pinoor work of PSR (Kazhdan et al., 2006; Kazhdan & Hoppe, 2013). Subsequent works RBF (Dinh et al., 2002; Carr et al., 2001), MLS (Merry et al., 2014; Lu et al., 2019), SDF (Curless & Levoy, 1996; Huang et al., 2021) and variational implicit surface (Zagorchev & Goshtasby, 2011; Evrard et al., 2018) etc., have also achieved impressive surface reconstruction. Very recently, iPSR (Hou et al., 2022) provides an iterative learning framework for accurate point normal estimation. Different from those traditional implicit surface reconstruction approaches, our work adopts neural implicit function (Wang et al., 2021; Hanocka et al., 2020) to represent shape geometry, and aims at more accurate surface reconstruction using neural implicit surface learning than traditional implicit surface reconstruction approaches.

Supervised Neural Implicit Surface Reconstruction. DeepSDF (Park et al., 2019) and its subsequent works (Jiang et al., 2020) have shown impressive effectiveness to reconstruct surface using neural implicit representation (Chen

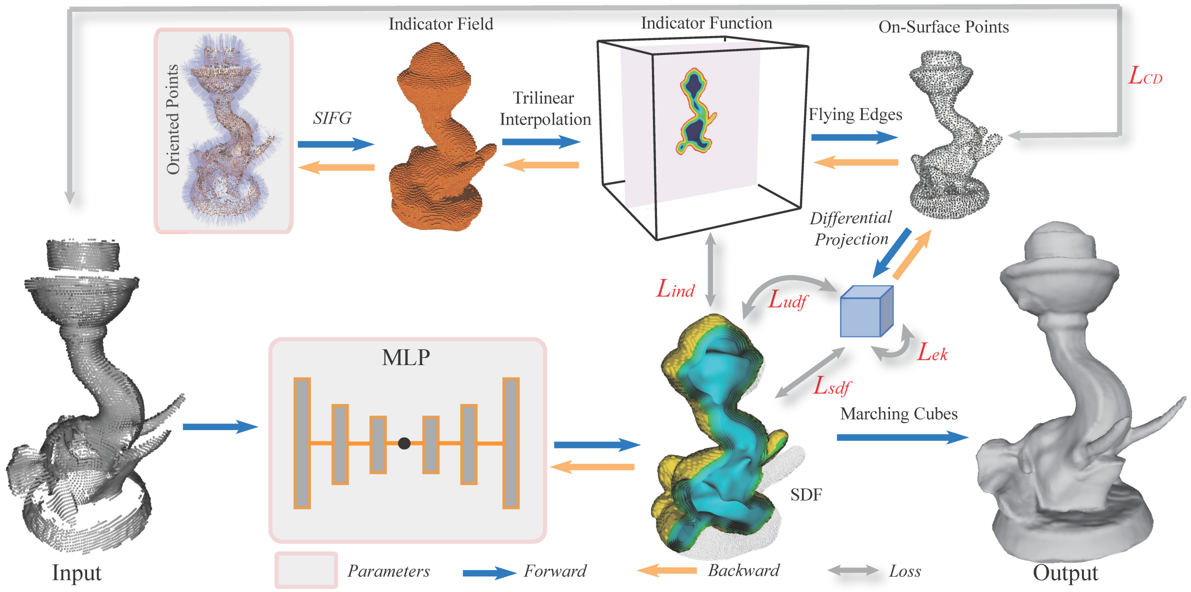

Figure 2. The system overview of our approach. For an input point cloud, our *NeuralIndicator* efficiently learns the effective neural indicator priors, which are used to regularize the neural implicit function optimization. Note that our system is an end-to-end unsupervised learning framework, with oriented points and MLP parameters θ of the neural implicit function $f(x, \theta)$ serving as trainable network parameters.

et al., 2021; Landgraf et al., 2022; Chen et al., 2023b). This inspired many works such as Convolutional Occupancy Networks (Peng et al., 2020), SA-ConvONet (Tang et al., 2021), Points2Surf (Erler et al., 2020), NMC (Chen & Zhang, 2021), CurriculumDeepSDF (Duan et al., 2020) and NDC (Chen et al., 2022) to perform neural implicit surface reconstruction in a supervised manner. Recently, NKSR (Huang et al., 2023) adopts an adaptive and hierachical neural kernel field for surface reconstruction from point cloud. Our work is inspired by the success of those data-driven approaches but focuses on surface reconstruction in an unsupervised learning manner.

Unsupervised Neural Implicit Surface Reconstruction. SAL (Atzmon & Lipman, 2020) and its variants (Gropp et al., 2020; Atzmon & Lipman, 2021) provide a sign agnostic learning to learn the signed distance function in a unsupervised way, achieving impressive surface reconstruction results. The subsequent works adopt to use more extra priors, such as Wasserstein Distance function (Williams et al., 2019), smoothness (Poursaeed et al., 2020), self-similarity shape prior (Hanocka et al., 2020; Zhao et al., 2021), neural tangent kernel (NTK) (Chu et al., 2021), iso-points (Wang et al., 2021), predictive context priors (Ma et al., 2022a), for better surface reconstruction. OnSurf (Ma et al., 2022b) introduced an on-surface decision function (ODF) as shape prior for surface reconstruction from point cloud, while the ODF module still need ground truth signed distance supervision to train, which is limited to cover the geometric variations when generalizing to unseen point clouds. Recently, NeuralTPS (Chen et al., 2023a) adopted Thin Plate Surface (TPS) function and Neural-IMLS (Wang et al., 2023) used MLS function as priors for surface reconstruction. But they often lead to oversmooth surface reconstruction results without too-much fine-grained details. Besides, Peng et al. (Peng et al., 2021) introduce Shape-As-Point neural solver for surface reconstruction, but only on naive indicator function, which would easily introduce unsatisfied surface reconstruction results. Different from those previous approaches, our method proposes to leverage a smooth indicator function as *global* shape priors to regularize neural implicit function learning, which can achieve more reliable and accurate surface reconstruction results.

3. NeuralIndicator

Given unordered point cloud set $\mathbf{X} = \{x_i\}$, our goal is to learn a neural implicit function $f(x, \theta)$ for surface reconstruction from $\mathbf{X}$. Our key solution is to introduce a smooth indicator function χ_P, and leverage the *global* shape priors from χ_P to regularize $f(x, \theta)$ learning. Fig. 2 demonstrates the overview of our full system.

Smooth Indicator Function. Inspired by previous works (Kazhdan, 2005; Kazhdan et al., 2006; Kazhdan & Hoppe, 2013), for an oriented point set $\mathbf{P} = \{(p_i, n_i)\}$, we seek to generate a *smooth* version of indicator function χ_P, where its gradient field $\bigtriangledown \chi_P$ approximates $\mathbf{P}$' normal

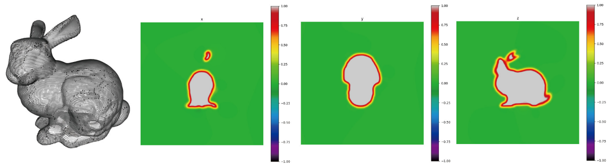

Figure 3. A tiny example of the smooth indicator field generated from oriented points, which are sampled from the bunny model (top left). We show one slice view in x-,y-,z-plane of the indicator field. Note that the indicator values are *smooth* especially for the shape boundary areas.

vector filed $\mathbf{V}$, i.e., $\mathbf{V}(p) = \sum_i \delta(p - p_i)n_i$, $\delta(\cdot)$ is Kronecker Delta function. We adopt to add smooth constraint to its gradient domain, by minimizing the volume integration $E(M) = \frac{1}{|M|}\int_M |H\chi_P(x)|^2 dx$ to be zero, where $H\chi_P$ is the Hessian matrix as:

$$H\chi_P = [\frac{\partial \bigtriangledown \chi_P}{\partial x_1} \quad \frac{\partial \bigtriangledown \chi_P}{\partial x_2} \quad \frac{\partial \bigtriangledown \chi_P}{\partial x_3}].$$

So specifically, we generate an smooth indicator function χ_P (see the property of SIF in Appendix A) from oriented points set $\mathbf{P}$ by optimizing the energy function as:

$$E = \frac{1}{|M|}\int_M |\bigtriangledown \chi_P(x) - \mathbf{V}(x)|^2 dx + \frac{\lambda}{|M|}\int_M |H\chi_P(x)|^2 dx. \quad (1)$$

About Point Normals Computation. During the learning of $f(x, \theta)$, we don't explicitly compute point normals for the input point cloud, but learn to optimize the oriented point set $\mathbf{P} = \{(p_i, n_i)\}$ to generate smooth indicator function, by regularizing the input point cloud locating on-surface of the smooth indicator function. So basically, we optimize normal information for the oriented point set $\mathbf{P}$, and let the smooth indicator function generated by the oriented point set $\mathbf{P}$ to fit the input point cloud, and we don't need any point normal input for the point cloud.

3.1. Smooth Indicator Function Generation

Since directly optimizing Eq. (1) to obtain χ_P is challenging, we proposed to estimate it in a differentiable way with two steps: (1) we first solve a Partial Differential Equation from E to obtain a smooth indicator filed, and then (2) recover the smooth indicator function by trilinearly interpolating the indicator filed.

Smooth Indicator Field. According to the Euler-Lagrange formulation, optimizing the energy function E can be obtained by solving a Partial Differential Equation as:

$$\Delta\chi_P + \lambda \bigtriangledown \cdot \sum_i \frac{\partial \bigtriangledown \chi_P}{\partial x_i} = \bigtriangledown \cdot V$$

$$\Rightarrow \Delta\chi_P + \lambda \bigtriangledown \cdot (\Delta\chi_P, \Delta\chi_P, \Delta\chi_P) = \bigtriangledown \cdot V \quad (2)$$

For an efficient solution, we adopt the spectrum method (Canuto et al., 2007) i.e. Fast Fourier Transform (FFT) to solve the above PDE over a uniform voxel grid, which have already been optimized to support GPUs, TPU and mainstream deep learning framework. For brevity, we denote $\hat{\chi}_P = FFT(\chi_P)$ as the Fast Fourier Transform (FFT) of the indicator function χ_P, and obtain χ_P following:

$$\chi_P = IFFT(\hat{\chi}_P)$$

$$\hat{\chi}_P(\omega) = G(\omega) \odot \frac{i\omega \cdot \hat{\mathbf{V}}}{-2\pi|\omega|^2(1 + 2\lambda\pi i \sum_j \omega_j)}$$

$$G(\omega) = exp(-2\frac{\sigma^2|\omega|^2}{r^2}) \quad (3)$$

with $\omega = (\omega_1, \omega_2, \omega_3) \in R^3$ is the spectral frequencies, $IFFT(\cdot)$ represent the inverse Fast Fourier Transform (IFFT) operation, $\hat{\mathbf{V}} = FFT(\mathbf{V})$, $\odot$ denote element-wise product and $G(\cdot)$ represents a Gaussian smoothing kernel of bandwidth σ at grid resolution r. Please refer to Appendix B for more details.

Trilinear Interpolation. After estimating the indicator filed, we propose to recover the *continues* indicator function by linearly interpolating the indicator filed. Specifically, for any position $x = (x_1, x_2, x_3) \in R^3$ we search the voxel v that x locates with the eight corner points coordinates are $\{\lfloor x_1 \rfloor \pm 1, \lfloor x_2 \rfloor \pm 1, \lfloor x_3 \rfloor \pm 1\}$. Then we compute the indicator value of position x as $\chi_p(x) = \frac{1}{|v|}\sum_i w(v_i - x)\chi_P(v_i)$, where $w(\cdot)$ is a trilinear interpolation weight function. In this way, we obtain the final continues indicator function χ_P that is generated from an oriented point set P. An tiny example for the smooth indicator function generated from an oriented point set sampled from a bunny model, is illustrated in Fig. 3.

3.2. Neural Indicator Prior

For an pair of implicit functions, i.e. the indicator function χ_P and signed distance function $f(x, \theta)$, we can formulate $f(x, \theta) = (-1)^{\chi_P} * f^u$ with f^u represents the absolute part of $f(x, \theta)$. Thus we propose to regularize both the signed part $f^s = (-1)^{\chi_P}$ and absolute part f^u with effective loss measurements from the indicator function χ_P respectively.

Indicator Loss L_{ind}. Since the signed part f^s of $f(x, \theta)$ is directly related with the indicator function χ_P as $f^s = (-1)^{\chi_P}$, we introduce the binary cross entropy $\mathcal{H}$ as indicator loss to regularize the signed part f^s as:

$$L_{ind} = \frac{1}{|\mathcal{G}|}\sum_i^{\mathcal{G}} \mathcal{H}(\sigma(\chi_P(x_i), [f^s(x_i) > 0]),$$

with $\mathcal{G}$ is point sampling set, $\sigma(\cdot)$ is the logistic function to convert the sign logits to probabilities, $[f^s(x_i) > 0]$ is 1 if x_i is outside the surface and 0 otherwise.

Absolute Distance Loss L_{udf}. For the regularization of the absolute part $f^u(x) = min_{y \in \partial f(x,\theta)} |y - x|^2$, we first introduce a *differential projection* $\Gamma : \mathcal{P}_s \rightarrow \mathcal{P}_s^{'}$, and then define an absolute distance loss L_{udf} as:

$$L_{udf} = \frac{1}{|\Omega|} \sum_{i}^{\Omega} |dist(p_i, \mathcal{P}_s^{'}) - dist(p_i, \mathbf{X})|^2,$$

with $dist(p_i, \mathbf{X}) = min_{x \in \mathbf{X}} |p_i - x|^2$ is distance function from p_i to point cloud set $\mathbf{X}$ and Ω is point sampling set. We adopt the back-propagation mechanism for the on-surface points P_s following (Remelli et al., 2020), by approximating the gradient between on-surface points P_s to any point p using the inverse surface normal (Remelli et al., 2020) $\frac{\partial P_s}{\partial p} = -n_p$, where n_p is the normal of the on-surface point P_s. In this way, we enable the differential back-propagation of the absolute distance loss L_{udf}. Please see more details in Appendix C.

SDF Loss L_{sdf} and Eikonal Loss L_{ek}. Since the projected on-surface point set $\mathcal{P}_s^{'}$ locate the on-surface of the signed distance function, we introduce a regularization for $\mathcal{P}_s^{'}$ as SDF loss $L_{sdf} = \frac{1}{|\mathcal{P}_s^{'}|} \sum_{p_i \in \mathcal{P}_s^{'}} |f(p_i, \theta)|^2$. Beside, we also regularize $\mathcal{P}_s^{'}$ according to the Eikonal equation (Atzmon & Lipman, 2020) as $L_{ek} = \frac{1}{|\mathcal{P}_s^{'}|} \sum_{p_i \in \mathcal{P}_s^{'}} |1 - |J_M(p_i)||^2$ with $J_M(\cdot)$ is the Jacobin vector for each point.

CD Loss L_{CD}. For a reliable indicator function regularization, we use a extra data term loss, i.e., a Chamfer Distance loss L_{CD} between the on-surface point set $\mathcal{P}_s$ and input point cloud $\mathbf{X}$, to encourage input point cloud locating to the latent surface of the indicator function. Specifically, we define it as $L_{CD} = \frac{1}{|\mathcal{P}_s|} \sum_{i}^{\mathcal{P}_s} |dist(p_i, \mathbf{X})|^2$. Note that this L_{CD} is mainly used to regularize the smooth indicator function χ_P learning, but not to regularize the neural implicit function $f(x, \theta)$. As we shown in the ablation study in Sec. 4.4, L_{CD} takes effects for more reliable smooth indicator function χ_P learning, which subsequently helps for more accurate neural implicit function $f(x, \theta)$ learning.

3.3. Training

We formulate a unified framework to learn the neural implicit function $f(x, \theta)$ together with the smooth indicator function χ_P in a unsupervised fashion.

Losses. We use regularization losses mainly from our NeuralIndicator during the optimization. Specifically, we formulate the loss used in the optimization as :

$$L = L_{ind} + \lambda_1 L_{udf} + \lambda_2 L_{sdf} + \lambda_3 L_{CD} + \lambda_4 L_{ek} \quad (4)$$

including the Indicator loss L_{ind}, absolute distance loss L_{udf}, SDF loss L_{sdf}, Eikonal loss L_{ek} and CD loss L_{CD} defined above. λ_i $(i = 1, .., 4)$ are weight parameters to balance different loss items. Please see more details in the supplementary materials.

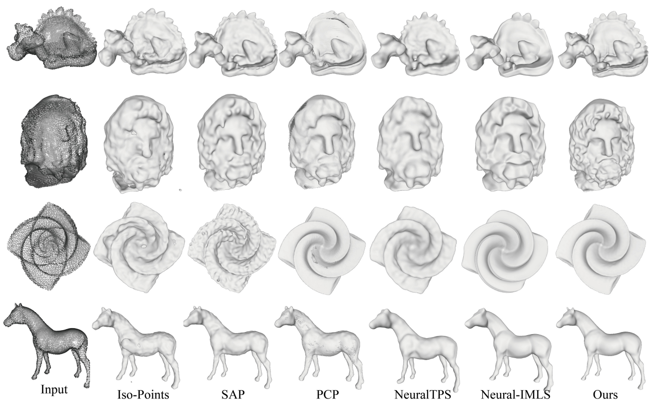

Figure 4. Visual comparisons of surface reconstruction results evaluated on the *eg-dragon*, *serapis*, *flower* and *horse* models from the FAMOUS dataset using the five compared approaches.

4. Evaluation and Analysis

Dataset. We use five public point cloud datasets, including both synthetic datasets (ABC (Koch et al., 2019), FAMOUS (Erler et al., 2020), Reconbench (Berger et al., 2013), Thingi10K (Zhou & Jacobson, 2016)) and real-scan dataset (DPoint (Wu et al., 2015) dataset), to evaluate our approach. As like Point2Surf (Erler et al., 2020)) and SAP (Peng et al., 2021), we use the test subset (with 100 meshes) in ABC, 22 diverse meshes in FAMOUS and test subset (with 100 meshes) in Thingi10K to perform the evaluation. Since ABD, FAMOUS, Reconbench and Thingi10K have the ground truth data, we mainly use these four datasets to perform the quantitatively evaluation. For the DPoint dataset (Wu et al., 2015) consisting 20 real-scan challenging incomplete and noisy point cloud data, we choose to perform the qualitative evaluation.

Comparing Approaches. We choose five representative approaches, including PSR (version 13.8), iPSR, Point2Mesh, Iso-Points, SAP, OnSurf, PCP, NeuralTPS and Neural-IMLS. Although Point2Mesh (Hanocka et al., 2020) didn't perform neural implicit function learning, we also choose Point2Mesh for comparison since it's a state-of-the-art unsupervised learning surface reconstruction approach.

4.1. Quantitative Evaluation

We first made a qualitative evaluation by measuring the surface reconstruction quality on the four synthetic dataset.

Accuracy Metrics. To evaluate the surface reconstruction

Table 1. The quantitative surface reconstruction accuracy in terms of CD1, CD2($\times 100$) and F-Score, evaluated on ABC, FAMOUS, Thingi10K and Reconbench dataset, using six different surface reconstruction approaches, with P(%), R(%) , F(%) represents Presion, Recall and F-score for short respectively.

Method	ABC					FAMOUS					THINGI10K					RECONBENCH				
	CD1$^{\downarrow}$	CD2$^{\downarrow}$	P$^{\uparrow}$	R$^{\uparrow}$	F$^{\uparrow}$	CD1$^{\downarrow}$	CD2$^{\downarrow}$	P$^{\uparrow}$	R$^{\uparrow}$	F$^{\uparrow}$	CD1$^{\downarrow}$	CD2$^{\downarrow}$	P$^{\uparrow}$	R$^{\uparrow}$	F$^{\uparrow}$	CD1$^{\downarrow}$	CD2$^{\downarrow}$	P$^{\uparrow}$	R$^{\uparrow}$	F$^{\uparrow}$
PSR (Kazhdan & Hoppe, 2013)	2.6	0.39	71	67	69	2.3	0.44	68	70	69	1.8	1.95	55	64	59	1.05	1.09	74	73	73
iPSR (Hou et al., 2022)	1.7	0.19	77	78	77	1.1	**0.03**	78	76	77	1.0	0.16	81	**82**	82	0.71	**0.48**	87	84	85
ISO (Wang et al., 2021)	3.3	0.95	62	64	63	3.1	0.79	70	74	72	1.3	0.88	69	75	72	1.08	1.12	69	74	71
P2M (Hanocka et al., 2020)	3.1	0.84	64	66	65	3.5	0.91	67	69	68	1.4	1.06	66	64	65	1.20	1.79	63	67	65
SAP (Peng et al., 2021)	2.5	0.55	65	69	67	2.4	0.68	66	67	66	1.9	0.25	80	78	79	0.77	0.52	82	83	83
PCP (Ma et al., 2022a)	1.9	0.23	74	72	73	1.2	0.04	74	78	76	1.1	0.29	76	80	78	0.85	0.65	80	79	79
OnSurf (Ma et al., 2022b)	2.0	0.31	71	72	72	1.4	0.07	73	75	74	1.2	0.33	74	72	73	1.01	1.23	70	78	74
NTPS (Chen et al., 2023a)	1.8	0.25	75	70	73	1.1	0.05	75	**79**	77	1.1	0.27	74	78	76	0.94	1.18	72	80	76
N-ILMS (Wang et al., 2023)	1.6	0.18	77	79	78	1.2	0.04	78	75	76	1.2	0.36	75	81	79	0.79	0.54	83	**86**	84
Ours	**1.5**	**0.11**	**78**	**80**	**79**	**1.0**	**0.03**	**79**	78	**78**	**0.9**	**0.15**	**84**	**82**	**83**	**0.65**	0.49	**88**	84	**86**

accuracy, we adopt three public accuracy metrics, including L1-Chamfer Distance (CD1), L2-Chamfer Distance (CD2) (Aanæs et al., 2016) and F-Score (F, with both Precision and Recall), to evaluate the surface reconstruction quality between the reconstructed surface and ground-truth surface. For the F-score accuracy, we set the distance threshold as $\mu = 1.5\%$ of the maximum length of objects' bounding box to compute the Precision and Recall numerical values. We uniformly sample 1×10^5 points from both the reconstructed surface and ground truth to calculate all of those three accuracy metrics as like Point2Surf (Erler et al., 2020) and PCP (Ma et al., 2022a).

Comparison Results. Table 1 summarizes the average Chamfer Distance (CD1 and CD2) and F-Score for the different surface reconstruction approaches evaluated on the four synthetic datasets respectively. Specifically, our approach achieves the lowest CD1, CD2 error and highest F-score in all of the four synthetic datasets evaluation, which is better than other approaches including Point2Mesh, Iso-Points, OnSurf, PCP, SAP, NeuralTPS and Neural-IMLS, and even slightly better than PSR (and iPSR) given ground truth point normal. Note that our approaches doesn't suffer from the point normal problem as PSR.

4.2. Qualitative Evaluation

We also made qualitative evaluations on the surface reconstruction results to see how their results are visually different. Fig. 4 shows several visual comparing results from FAMOUS dataset (with ground truth point normal). Since the point cloud in FAMOUS often contain a high level of noise, the surface reconstruction could be difficult. We can see that PCP's results contain many tiny holes (*eg-dragon*, *flower* and *horse* models), Iso-Points and SAP's results are a bit noisy. The results of NeuralTPS and Neural-IMLS are often too smooth without fine-grained details. In contrast, ours is not influenced by the noise sampling points too much and consistently leads to *complete*, *fine-grained*, and *regularized* surface reconstruction results.

Fig. 5 shows several visual comparing results from DPoint dataset. This dataset consists of real-scan point clouds, and most models in this dataset are noisy, incomplete and with complex topology structure, which is challenging for most surface reconstruction approaches. Our approach can reliably recover accurate topology structures in all of the four models with fine geometric details, which are much better than the other four approaches.

From the above qualitative evaluation, we can see that our approach can achieve consistently better visual surface reconstruction results than the other four approaches, for incomplete or noisy point clouds in complex topology structure. Please refer to our supplementary materials to see more visual comparison results using different comparing approaches, including synthetic dataset (with both ground truth and estimated point normal) and real-scan dataset (with estimated point normal) respectively.

Table 2. The CD2$^{\downarrow}$($\times 10^5$) accuracy on ABC, FAMOUS, Thingi10k and Reconbench datasets using different approaches.

Method	P2S	LIG	NDC	NKSR	Ours
ABC	1803.3	45.7	12.8	**9.2**	121.7
FAMOUS	1321.5	51.2	**20.1**	38.9	30.2
Thingi10k	1285.1	62.6	**15.4**	94.5	151.6
Reconbench	120.5	5.1	**1.3**	5.2	4.9
mean	1132.6	41.1	**12.4**	36.9	77.0

4.3. Comparison with Supervised Approaches

We also compare our approach with some latest supervised learning based approaches, including Point2Surf (P2S) (Erler et al., 2020), LIG (Jiang et al., 2020), NDC (Chen et al., 2022), and NKSR (Huang et al., 2023). Table 2 shows the average CD2 accuracy of surface reconstruction results using the four comparing approaches. The state-of-the-art supervised approaches like LIG, NDC and NKSR can achieve better surface reconstruction accuracy than our approach. This is reasonable since supervised learning approaches would obtain data-driven information from dataset to en-

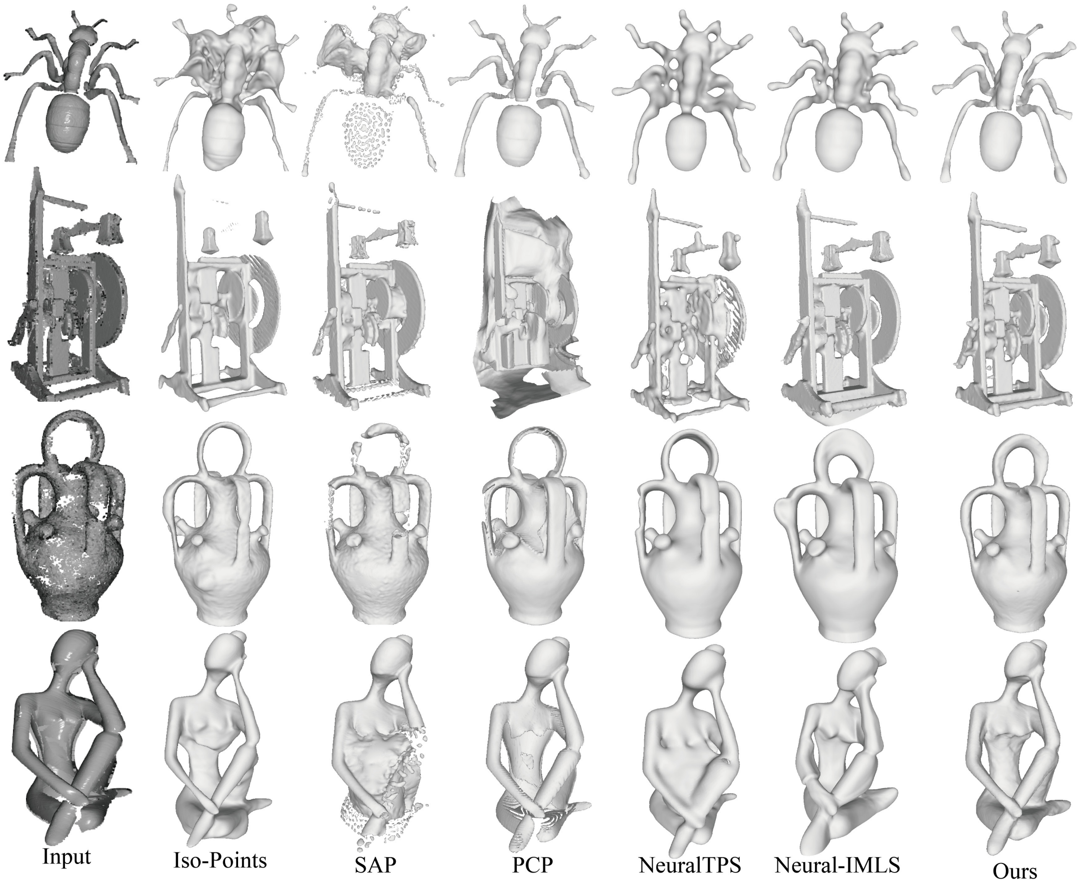

Figure 5. Representative visual comparison results of surface reconstruction evaluated on the *ant*, *clock*, *vase*, and *yoga* models (from top to bottom rows respectively) of the DPoint dataset (Wu et al., 2015).

hance the individual surface reconstruction quality, which is missing for unsupervised learning approaches like ours. But for dataset that didn't pre-trained by those supervised approaches, their surface reconstruction results will significantly decrease. For example, Ours achieves better surface reconstruction accuracy than LIG and NKSR in FAMOUSE and Reconbench dataset, when they are not pre-trained on these two dataset. Fig. 7 also shows some visual surface reconstruction results from DPoint dataset using these four approaches. Our approach can achieve better surface reconstruction with complete and fine-grained results than other three supervised learning approaches. Please refer to the supplementary materials[1] to see more results.

Table 3. The surface reconstruction accuracy (CD1 and CD2 ($\times 10^3$)) for our approach under different module variants.

Method	CD1$\downarrow$	CD2$\downarrow$	Method	CD1$\downarrow$	CD2$\downarrow$
SIFG-32	1.47	1.05	w/o L_{udf}	1.67	2.13
SIFG-64	1.40	0.85	w/o L_{sdf}	1.41	0.90
SIFG-128	1.30	0.56	w/o L_{CD}	1.52	1.21
SIFG-256	1.28	0.51	w/o L_{ek}	1.32	0.67
w/o L_{ind}	1.38	0.72	**FULL**	**1.31**	**0.59**

4.4. Ablation Study and Analysis

SIFG Resolution. We first conduct study on different voxel resolution in the SIFG module. Specifically, we set voxel resolution as 32×, 64×, 128× and 256× voxels in the unit box, and perform the surface reconstruction with such four system variants (called SIFG-32, SIFG-64, SIFG-128 and SIFG-256 respectively) on the Test dataset. As shown in Table 3, we can see that both the CD1 and CD2 numerical values decrease from sparse to dense voxel resolutions, which means that the more dense SIFG resolution will lead to the better surface reconstruction quality for our approach. But the accuracy in both CD1 and CD2 only get slight improvement from 128× to 256×. Considering that SIFG-256 will takes much time cost than SIFG-128 but doesn't achieve significant surface reconstruction quality improvement, we

---

[1] https://shishenghuang.github.io/index/nid-supp.pdf

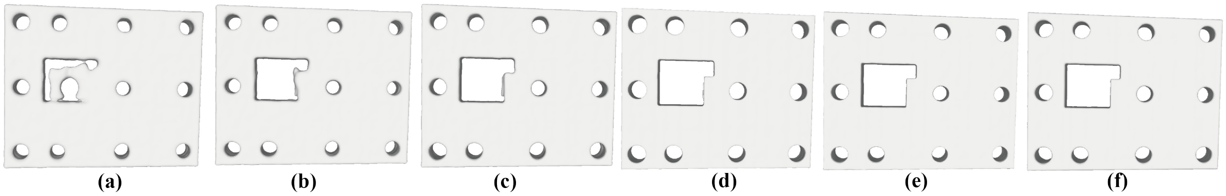

Figure 6. Visual surface reconstruction results from model 11827 in ABC dataset using our approaches getting rid of one specific neural indicator priors including L_{udf} (a), L_{CD} (b), L_{sdf} (c), L_{ek} (d) and L_{ind}(e) respectively, comparing with the result of our full system (f).

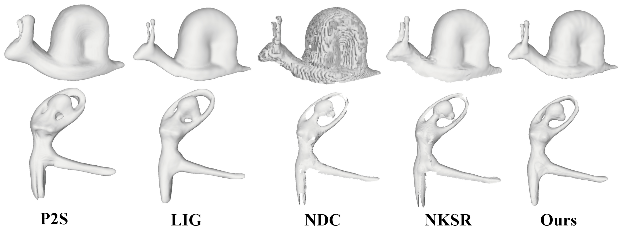

Figure 7. Two visual surface reconstruction results from DPoint dataset using four different surface reconstruction approaches, including P2S, LIG, NDC and Ours (from left to right).

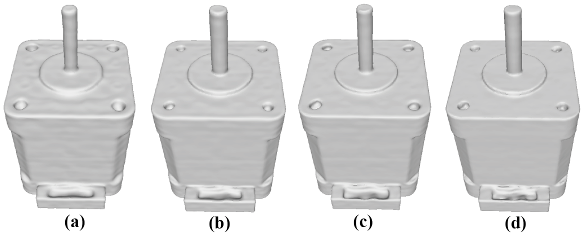

Figure 8. A tiny example of surface reconstruction for model 10218 in ABC dataset using different SIFG resolutions by our approach, including 32× (a), 64× (b), 128× (c) and 256× (d).

adopt to set the SIFG resolution to 128 for a better balance of both surface reconstruction quality and efficiency in our full system. Fig. 8 shows a tiny visual surface reconstruction results of model 10218 from ABC dataset, using our approach with different SIFG resolutions.

Neural Indicator Priors. To evaluate the influence of the neural indicator priors (including L_{ind}, L_{udf}, L_{sdf}, L_{CD} and L_{ek}) on our full system, we conduct experiments on the Test dataset by getting rid of one specific prior loss. Table 3 also shows the average CD1 and CD2 accuracy metrics for system variant without L_{ind}, L_{udf}, L_{sdf}, L_{CD} and L_{ek} respectively. We can see that L_{udf}, L_{sdf}, L_{CD} will make major influence on the final surface reconstruction quality, while L_{ind} and L_{ek} make minor influence. Fig. 6 show a visual surface reconstruction result of model 11827 from ABC dataset, using our approach by getting rid of one specific neural indicator prior.

Noises. We also study how the ambiguous level of point noisy influences the performance of our approach. Here we conduct experiments on the six dataset variants beforementioned using our approach, and include previous approaches (PSR, Iso-Points, SAP and PCP) as baseline for comparison. Table 4 shows the CD2 accuracy metrics of the surface reconstruction results for different approaches. It can be seen that different level of point noisy influence all methods. But generally, our approach achieves consistently lowest CD2 accuracy metrics, demonstrating that our approach performs consistently better for reliable surface reconstruction than previous approaches. Fig. 9 shows a tiny visual results from Thingi10k dataset under different noisy variant using our approach.

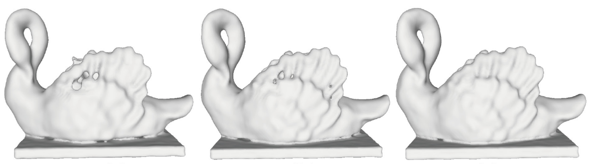

Figure 9. A visual surface reconstruction results using our approach from Thingi10k dataset under different noisy variants, including strong noise version (column a), varying noise version (column b) and original models (column c).

Table 4. The quantitative surface reconstruction accuracy CD2($\times 100$) on dataset variants using different surface reconstruction approaches.

Method	PSR	ISO	SAP	PCP	Ours
ABC-var-noise	3.31	0.62	0.51	0.48	**0.40**
ABC-max-noise	4.01	1.05	0.63	0.59	**0.52**
F-var-noise	1.78	0.19	0.67	0.08	**0.06**
F-max-noise	3.52	0.59	0.29	0.31	**0.25**
T10k-var-noise	2.75	0.51	0.30	0.39	**0.28**
T10k-max-noise	3.54	0.83	0.48	0.52	**0.46**
mean	3.15	0.58	0.38	0.39	**0.33**

4.5. Large Scale Scenes

To evaluate the ability to reconstruct large scale scenes, we adopt the grid-based learning strategy like LIG (Jiang et al., 2020) to perform geometry learning grid-by-grid us-

Table 5. The CD1 and CD2 ($\times 10^4$) accuracy of surface reconstruction results using PSR (Kazhdan & Hoppe, 2013), PCP (Ma et al., 2022a) and our approach, on the 3D Scene dataset.

Method	Burghers		Lounge		Copyroom		Stonewall		Totempole	
	CD1	CD2	CD1	CD2	CD1	CD2	CD1	CD2	CD1	CD2
PSR	2.31	6.21	0.87	7.64	0.29	9.15	0.25	10.9	0.25	13.76
PCP	0.91	2.73	**0.06**	**0.61**	**0.07**	0.78	0.07	0.61	0.08	1.12
Ours	**0.85**	**2.05**	0.07	0.71	**0.07**	**0.67**	**0.06**	**0.55**	**0.07**	**1.01**

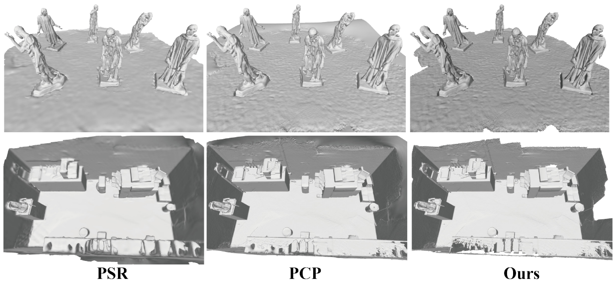

Figure 10. Some visual comparison results for large scale point cloud reconstruction from 3D Scene dataset.

ing our approach, and choose PSR (Kazhdan & Hoppe, 2013) and PCP (Ma et al., 2022a) for comparison on the 3D Scene (Zhou & Koltun, 2013) dataset. Table 5 shows the surface reconstruction accuracy values (CD1 and CD2) for PSR, PCP and our approach on the 3D Scene dataset. On average, with the aid of grid-based learning of LIG (Jiang et al., 2020), our approach achieves the similar level of surface reconstruction accuracy as PCP (Ma et al., 2022a), but is much better than PSR (Kazhdan & Hoppe, 2013). One possible reason that our approach doesn't significantly outperform PCP, would due to the independent geometry learning for each grid voxel as LIG (Jiang et al., 2020) did, which often leads to in-consistent geometry reconstruction across grid neighbors thus decreasing the total surface reconstruction accuracy. Fig. 10 shows some visual comparing results for point cloud scenes from 3D Scene using PSR, PCP and our approach respectively.

4.6. Time Analysis

On average, our approach consumes about $6G$ GPU memory to train the neural implicit function for a point cloud with 10,000 points. It takes about one hour for our system to finish $10,000$ training epochs on a desktop computer with an NVIDIA GeForce RTX 3060 12G GPU. This is slower than SAP (average 30min), slightly slower than Iso-Points (average less then one hour), but faster than Point2Mesh (average 1.5 hours) and PCP (average 2.5 hours). Note that we did not use GPU-based parallel acceleration techniques in our current implementation, so our system can be further accelerated by making the on-surface points extraction (such as marching cubes) be paralleled for GPU computation if seriously required. Besides, it is also straightforward to perform the point projection in parallel, which would further accelerate our whole system. Please refer to the supplementary materials for a further study on the computational cost of our system.

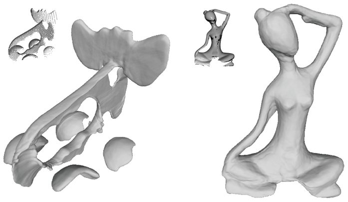

Figure 11. Two failure cases of surface reconstruction for point clouds with large holes (right) or incomplete scans (left) by our approach.

4.7. Limitation

One main limitation of our approach is that our current solution only incorporate geometry priors. If a point cloud is highly incomplete due to occlusions, our approach could not successfully reconstruct a complete surface (Fig. 11 left). Besides, large holes located near the boundary areas (Fig. 11 right) would not be filled by our approach, since we did not employ any priors indicating whether this kind of hole belongs to the empty space or the underlying surface. So it is an interesting direction to incorporate more structure priors, such as symmetry, structure pattern priors, etc., to further improve the surface reconstruction quality.

5. Conclusion

In this paper, we have presented a new neural implicit surface optimization approach, by introducing *global* shape priors regularization from a neural indicator function generation (*NeuralIndicator*) in a unsupervised fashion. With effective neural indicator priors, we show that our approach is reliable to reconstruct *complete*, *fine-detailed* shape surfaces, even from incomplete and/or noisy point clouds with complex topology structure, which is better than previous surface reconstruction approaches.

We hope that our approach could inspire the subsequent works to extract 3D geometry priors in a unsupervised manner, which could benefit applications such as 3D shape generation, manipulation or even online 3D reconstruction.

Impact Statement

This paper proposes a new unsupervised learning approach for point cloud reconstruction, which utilizes global shape prior encoded by a smooth indicator function to regularize the neural implicit surface learning, and achieves *complete*, *fine-detailed* point cloud reconstruction results even from incomplete and/or noisy point clouds with complex topology structure. With extensive evaluation, the proposed approach can achieve better surface reconstruction results than previous approaches, which becomes a new state-of-the-art point cloud reconstruction approach in a unsupervised manner. The key contribution of this paper is to introduce an way to encode global shape priors, and demonstrate the effectiveness for robust and high-fidelity point cloud reconstruction. This can inspire subsequent works to extract effective 3D geometry priors in an unsupervised manner, and would benefit many applications such as shape generation, manipulation or even online 3D reconstruction.

Acknowledgments

We thank all the anonymous reviewers for their constructive comments. This work was supported by Natural Science Foundation of China (Project Number 62202057).

References

Aanæs, H., Jensen, R. R., Vogiatzis, G., Tola, E., and Dahl, A. B. Large-scale data for multiple-view stereopsis. *IJCV*, 120(2):153–168, 2016.

Atzmon, M. and Lipman, Y. SAL: sign agnostic learning of shapes from raw data. In *IEEE CVPR*, pp. 2562–2571, 2020.

Atzmon, M. and Lipman, Y. SALD: sign agnostic learning with derivatives. In *ICLR*, 2021.

Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., and Silva, C. T. A benchmark for surface reconstruction. *ACM TOG*, 32(2):20:1–20:17, 2013.

Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. *Spectral methods: fundamentals in single domains*. Springer Science & Business Media, 2007.

Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. Reconstruction and representation of 3d objects with radial basis functions. In *ACM SIGGRAPH*, pp. 67–76, 2001.

Chen, C., Liu, Y.-S., and Han, Z. Unsupervised inference of signed distance functions from single sparse point clouds without learning priors. In *IEEE CVPR*, pp. 17712–17723, 2023a.

Chen, H., Wu, R., Grinspun, E., Zheng, C., and Chen, P. Y. Implicit neural spatial representations for time-dependent pdes. In *ICML*, pp. 5162–5177. PMLR, 2023b.

Chen, Y., Fernando, B., Bilen, H., Mensink, T., and Gavves, E. Neural feature matching in implicit 3d representations. In *ICML*, pp. 1582–1593. PMLR, 2021.

Chen, Z. and Zhang, H. Neural marching cubes. *ACM TOG*, 40(6):1–15, 2021.

Chen, Z., Tagliasacchi, A., Funkhouser, T., and Zhang, H. Neural dual contouring. *ACM TOG*, 2022.

Chu, L., Pan, H., and Wang, W. Unsupervised shape completion via deep prior in the neural tangent kernel perspective. *ACM TOG*, 40(3):1–17, 2021.

Curless, B. and Levoy, M. A volumetric method for building complex models from range images. In *ACM SIGGRAPH*, pp. 303–312, 1996.

Dai, A. and Nießner, M. Scan2mesh: From unstructured range scans to 3d meshes. In *IEEE CVPR*, pp. 5574–5583, 2019.

Dinh, H. Q., Turk, G., and Slabaugh, G. G. Reconstructing surfaces by volumetric regularization using radial basis functions. *IEEE TPAMI*, 24(10):1358–1371, 2002.

Duan, Y., Zhu, H., Wang, H., Yi, L., Nevatia, R., and Guibas, L. J. Curriculum deepsdf. In *ECCV*, pp. 51–67, 2020.

Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N. J., and Wimmer, M. Points2surf learning implicit surfaces from point clouds. In *ECCV*, volume 12350, pp. 108–124, 2020.

Evrard, F., Denner, F., and Van Wachem, B. Surface reconstruction from discrete indicator functions. *IEEE TVCG*, 25(3):1629–1635, 2018.

Gropp, A., Yariv, L., Haim, N., Atzmon, M., and Lipman, Y. Implicit geometric regularization for learning shapes. In *ICML*, volume 119, pp. 3789–3799, 2020.

Hanocka, R., Metzer, G., Giryes, R., and Cohen-Or, D. Point2mesh: a self-prior for deformable meshes. *ACM TOG*, 39(4):126:1–126:12, 2020.

Hou, F., Wang, C., Wang, W., Qin, H., Qian, C., and He, Y. Iterative poisson surface reconstruction (ipsr) for unoriented points. *ACM TOG*, 2022.

Huang, J., Huang, S., Song, H., and Hu, S. Di-fusion: Online implicit 3d reconstruction with deep priors. In *IEEE CVPR*, 2021.

Huang, J., Gojcic, Z., Atzmon, M., Litany, O., Fidler, S., and Williams, F. Neural kernel surface reconstruction. In *IEEE CVPR*, pp. 4369–4379, 2023.

Huang, Z., Carr, N., and Ju, T. Variational implicit point set surfaces. *ACM TOG*, 38(4):124:1–124:13, 2019.

Jiang, C. M., Sud, A., Makadia, A., Huang, J., Nießner, M., and Funkhouser, T. A. Local implicit grid representations for 3d scenes. In *IEEE CVPR*, pp. 6000–6009, 2020.

Kazhdan, M. M. Reconstruction of solid models from oriented point sets. In Desbrun, M. and Pottmann, H. (eds.), *SGP*, volume 255, pp. 73–82, 2005.

Kazhdan, M. M. and Hoppe, H. Screened poisson surface reconstruction. *ACM TOG*, 32(3):29:1–29:13, 2013.

Kazhdan, M. M., Bolitho, M., and Hoppe, H. Poisson surface reconstruction. In *SGP*, pp. 61–70, 2006.

Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., and Panozzo, D. ABC: A big CAD model dataset for geometric deep learning. In *IEEE CVPR*, pp. 9601–9611, 2019.

Landgraf, Z., Hornung, A. S., and Cabral, R. S. Pins: Progressive implicit networks for multi-scale neural representations. In *ICML*, pp. 11969–11984. PMLR, 2022.

Lipman, Y. Phase transitions, distance functions, and implicit neural representations. In Meila, M. and Zhang, T. (eds.), *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6702–6712, 2021.

Lu, W., Shi, Z., Sun, J., and Wang, B. Surface reconstruction based on the modified gauss formula. *ACM TOG*, 38(1): 2:1–2:18, 2019.

Ma, B., Liu, Y., Zwicker, M., and Han, Z. Surface reconstruction from point clouds by learning predictive context priors. In *IEEE CVPR*, 2022a.

Ma, B., Liu, Y.-S., and Han, Z. Reconstructing surfaces for sparse point clouds with on-surface priors. In *IEEE CVPR*, pp. 6315–6325, 2022b.

Merry, B., Gain, J., and Marais, P. Moving least-squares reconstruction of large models with gpus. *IEEE TVCG*, 20(2):249–261, 2014.

Park, J. J., Florence, P., Straub, J., Newcombe, R. A., and Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE CVPR*, pp. 165–174, 2019.

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. Convolutional occupancy networks. In *ECCV*, pp. 523–540, 2020.

Peng, S., Jiang, C., Liao, Y., Niemeyer, M., Pollefeys, M., and Geiger, A. Shape as points: A differentiable poisson solver. In *NeurIPS*, pp. 13032–13044, 2021.

Poursaeed, O., Fisher, M., Aigerman, N., and Kim, V. G. Coupling explicit and implicit surface representations for generative 3d modeling. In *ECCV*, volume 12355, pp. 667–683, 2020.

Remelli, E., Lukoianov, A., Richter, S. R., Guillard, B., Bagautdinov, T. M., Baqué, P., and Fua, P. Meshsdf: Differentiable iso-surface extraction. In *NeurIPS*, 2020.

Schroeder, W. J., Maynard, R., and Geveci, B. Flying edges: A high-performance scalable isocontouring algorithm. In *IEEE Symposium on Large Data Analysis and Visualization, LDAV*, pp. 33–40, 2015.

Tang, J., Lei, J., Xu, D., Ma, F., Jia, K., and Zhang, L. Saconvonet: Sign-agnostic optimization of convolutional occupancy networks. In *IEEE CVPR*, pp. 6504–6513, 2021.

Wang, Y., Wu, S., Öztireli, A. C., and Sorkine-Hornung, O. Iso-points: Optimizing neural implicit surfaces with hybrid representations. In *IEEE CVPR*, 2021.

Wang, Z., Wang, P., Wang, P., Dong, Q., Gao, J., Chen, S., Xin, S., Tu, C., and Wang, W. Neural-imls: Self-supervised implicit moving least-squares network for surface reconstruction. *IEEE TVCG*, 2023.

Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., and Panozzo, D. Deep geometric prior for surface reconstruction. In *IEEE CVPR*, pp. 10130–10139, 2019.

Wu, S., Huang, H., Gong, M., Zwicker, M., and Cohen-Or, D. Deep points consolidation. *ACM TOG*, 34(6): 176:1–176:13, 2015.

Xiao, D., Lin, S., Shi, Z., and Wang, B. Learning modified indicator functions for surface reconstruction. *Computers Graphics*, 102:309–319, 2022.

Zagorchev, L. G. and Goshtasby, A. A. A curvature-adaptive implicit surface reconstruction for irregularly spaced points. *IEEE TVCG*, 18(9):1460–1473, 2011.

Zhao, W., Lei, J., Wen, Y., Zhang, J., and Jia, K. Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds. In *IEEE CVPR*, pp. 10256–10265, 2021.

Zhou, Q. and Jacobson, A. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

Zhou, Q.-Y. and Koltun, V. Dense scene reconstruction with points of interest. *ACM TOG*, 32(4):1–8, 2013.

A. Property of Smooth Indicator Function

Our smooth indicator function χ_P has two major properties: (1) it inherits the *global* indicative information to discriminate the inside/outside region of the entire shape, and (2) it approximates the *local* SDFs everywhere near the zero level set geometry (see Lemma I below for details). In this way, our smooth indicator function χ_P encodes both the global indicative prior and local SDFs simultaneously, which can be served as global shape priors of entire point cloud to regularize the neural implicit function learning. Note the oriented point set $\mathbf{P}$ is different from the input point cloud $\mathbf{X}$, where oriented point set $\mathbf{P}$ is parameter set to generate smooth indicator function χ_P, and in the final optimization the zero level set of χ_P also fits $\mathbf{X}$. We have added a new paragraph "Property of Smooth Indicator Function" in Sec. 3 of the main paper to clarify this, please see them in the hightlight version.

Lemma I. The smooth indicator function χ_P approximates the signed distance function $f_M(\theta)$ near the on-surface point every where.

Proof. First of all, according to the definition of Kronecker Delta function $\delta(\cdot)$, we can approximate the Kronecker Delta function $\delta(\cdot)$ using a Gaussian function with a small parameter σ ($|\sigma| < \epsilon$):

$$\delta(x) = \lim_{\sigma \to \infty} \frac{1}{\sigma\sqrt{\pi}} \exp(-\frac{x^2}{\sigma^2}) \approx \frac{1}{\sigma\sqrt{\pi}} \exp(-\frac{x^2}{\sigma^2}).$$

Then for a given oriented point set $P = \{(p_i, n_i)\}$, the normal vector filed $V(x) = \sum_i \delta(x - p_i) n_i$ can also be approximated as :

$$V(x) \approx \begin{cases} 0, & \forall x \notin S_\sigma(x), \\ \frac{1}{\sigma\sqrt{\pi}} \sum_i \exp(-\frac{(x-p_i)^2}{\sigma^2}) n_i, & \forall x \in S_\sigma(x), \end{cases} \tag{5}$$

where $S_\sigma(x) = \{x | \exists i, |x - p_i| \leq 3\sigma\}$ is a point set where every point locates within the 3σ distance from oriented point set $P = \{(p_i, n_i)\}$. Since the learnt signed distance function $f_M(\theta)$ fits the oriented point set $P = \{(p_i, n_i)\}$, we can see $S_\sigma(x)$ as a point set locating near the on-surface of $f_M(\theta)$.

According to the definition of smooth indicator function χ_P, we have:

$$\chi_P = \arg\min_{\chi_P} \frac{1}{M} \int_M |\nabla \chi_P(x) - V(x)|^2 dx + \frac{\lambda}{|M|} \int_M |H\chi_P(x)|^2 dx.$$

Let's re-formulate the above objective energy function in a discrete form using a 3D volumetric grid G with gird length as $\frac{1}{2}\sigma$, we have:

$$\begin{aligned} \chi_P = \arg\min_{\chi_P} \frac{1}{M} \sum_{x_i \in G} |\nabla \chi_P(x_i) - V(x_i)|^2 \\ + \frac{\lambda}{|M|} \sum_{x_i \in G} |H\chi_P(x_i)|^2 \\ \approx \arg\min_{\chi_P} \frac{1}{M} \sum_{x_i \in G} |\nabla \chi_P(x_i) - V(x_i)|^2 + \\ \frac{\lambda}{|M|} \sum_{x_j \in \mathcal{N}_i} |\nabla \chi_P(x_i) - \nabla \chi_P(x_j)|^2, \end{aligned} \tag{6}$$

with $\mathcal{N}_i$ is the neighborhood grid point of x_i. When solving the objective energy function (6) using least square optimization, according to the approximation of normal vector field $V(x)$ in Eq. (5), we can obtain the approximated solution of $\nabla \chi_P$ as:

$$\nabla \chi_P(x_i) \approx \begin{cases} 0, & \forall x_i \notin S_\sigma(x), \\ \frac{1}{(1+\lambda)|\bar{\mathcal{S}}_\sigma(x_i)|} \sum_{x_j \in \bar{\mathcal{S}}_\sigma(x_i)} V(x_j), & \forall x_i \in S_\sigma(x). \end{cases}$$

with $\bar{\mathcal{S}}_\sigma(x_i) = \{x | \forall x \in G, |x - x_i| \leq 3\sigma\}$. Let's consider $\forall x_i \in S_\sigma(x)$ where $S_\sigma(x)$ represents the point set near the on-surface of $f(\theta)$, since $|V(x_j)| \approx 1, \forall x_j \in \bar{\mathcal{S}}_\sigma(x_i)$, we have:

$$|\nabla \chi_P(x_i)| \approx 1, \quad \forall x_i \in S(x), \tag{7}$$

which means the smooth indicator function χ_P satisfies the Eikonal equation[2], and can be seen as a signed distance function. In this way, we can approximate the learnt signed distance function $f_M(\theta)$ using the smooth indicator function χ_P near the on-surface point set $S(x)$ as:

$$f_M(\theta, x) \approx \chi_P(x), \quad \forall x \in S_\sigma(x). \quad \square$$

B. Derivation of PDE Solver

We adopt to solve the PDE using the spectral method (Canuto et al., 2007; Peng et al., 2021). For a function $f(x)$ defined in the three dimensions R^3, i.e., $x := (x_1, x_2, x_3) \in R^3$, the multidimensional Fourier Transform (FFT) and Inverse Fourier Transform (IFFT) for $f(x)$ are defined as:

$$\hat{f}(\omega) := FFT(f(x)) = \int f(x) e^{-2\pi i x \cdot \omega} dx,$$

$$f(x) := IFFT(\hat{f}(\omega)) = \int \hat{f}(\omega) e^{2\pi i x \cdot \omega} d\omega,$$

---

[2] https://en.wikipedia.org/wiki/Signed_distance_function

where $\omega := (\omega_1, \omega_2, \omega_3)$ represents the spectral frequencies corresponding to x. The partial derivatives of $f(x)$ to $x_j, j = 1, 2, 3$ can be calculated as:

$$\frac{\partial}{\partial x_j} f(x) = \int 2\pi i \omega_j \hat{f}(\omega) e^{2\pi i x \cdot \omega} d\omega = IFFT(2\pi i \omega_j \hat{f}(\omega)).$$

The second order partial derivative of $\frac{\partial^2}{\partial^2 x_j} f(x)$ can be calculated as:

$$\begin{aligned} \frac{\partial^2}{\partial^2 x_j} f(x) &= \int -4\pi^2 \omega_j^2 \hat{f}(\omega) e^{2\pi i x \cdot \omega} d\omega \\ &= IFFT(-4\pi^2 \omega_j^2 \hat{f}(\omega)). \end{aligned}$$

And similarly, the third order partial derivative of $\frac{\partial^3}{\partial^2 x_j \partial x_k} f(x)$ can be calculated as:

$$\begin{aligned} \frac{\partial^3}{\partial^2 x_j \partial x_k} f(x) &= \int -8\pi^3 i \omega_j^2 \omega_k \hat{f}(\omega) e^{2\pi i x \cdot \omega} d\omega \\ &= IFFT(-8\pi^3 i \omega_j^2 \omega_k \hat{f}(\omega)). \end{aligned}$$

Based on the above derivatives, we calculate $\Delta \chi_P$ as:

$$\Delta \chi_P = \sum_j^3 \frac{\partial^2}{\partial^2 x_j} \chi_P = IFFT(-4\pi^2 |\omega|^2 \hat{\chi}_P),$$

and calculate $\bigtriangledown \cdot (\Delta \chi_P, \Delta \chi_P, \Delta \chi_P)$ as:

$$\begin{aligned} \bigtriangledown \cdot (\Delta \chi_P, \Delta \chi_P, \Delta \chi_P) &= \sum_j^3 \sum_k^3 \frac{\partial^3}{\partial^2 x_j \partial x_k} f(x) \\ &= IFFT(-8\pi^3 i |\omega|^2 (\omega_1 + \omega_2 + \omega_3) \hat{\chi}_P). \end{aligned}$$

Hence the PDE can be formulated as:

$$\begin{gathered} \Delta \chi_P + \lambda \bigtriangledown \cdot (\Delta \chi_P, \Delta \chi_P, \Delta \chi_P) = \bigtriangledown \cdot V \\ \Rightarrow IFFT(-4\pi^2 |\omega|^2 \hat{\chi}_P) + \\ \lambda IFFT(-8\pi^3 i |\omega|^2 (\omega_1 + \omega_2 + \omega_3) \hat{\chi}_P) \\ = \bigtriangledown \cdot V \end{gathered}$$

By applying the Fourier Transform (FFT) on both sides of the above PDE, we have:

$$\begin{gathered} -4\pi^2 |\omega|^2 (1 + 2\lambda \pi i (\omega_1 + \omega_2 + \omega_3)) \hat{\chi}_P \\ = FFT(\bigtriangledown \cdot V) \end{gathered}$$

Considering that $FFT(\bigtriangledown \cdot V) = 2\pi i (\omega \cdot V)$, the spectral frequency function $\hat{\chi}_P$ can be calculated as:

$$\hat{\chi}_P = G(\omega) \odot \frac{i \omega \cdot \hat{V}}{-2\pi |\omega|^2 (1 + 2\lambda \pi i \sum_j \omega_j)} \tag{8}$$

$$\chi_P = IFFT(\hat{\chi}_P) \tag{9}$$

with $G(\omega) = exp(-2\frac{\sigma^2 |\omega|^2}{r^2})$ a Gaussian smoothing kernel of bandwidth σ for grid resolution of r in the spectral domain to mitigate the ringing effects as a result of the Gibbs phenomenon from rasterizing the point normals V.

C. Absolute Distance Loss L_{udf}

For the regularization of the absolute part $f^u(x) = min_{y \in \partial f(x, \theta)} |y - x|^2$, the key challenge is how to determine the on-surface boundary $\partial f(x, \theta)$ of neural implicit function $f(x, \theta)$, which is often unknown during the neural implicit function learning. Instead of sampling the iso-points (Wang et al., 2021) using local analysis, we proposed to extract the on-surface points from the indicator function χ_P and then project them onto the zero level set of the signed distance function with a *differential projection*.

Specifically, for an efficient on-surface point extraction, we leverage Flying Edges method (Schroeder et al., 2015) to generate the on-surface points $\mathcal{P}_s$ from the indicator function χ_P, which is a state-of-the-art zero level set extraction approach with about 10× time efficiency than Marching Cubes (Curless & Levoy, 1996). But one issue for leveraging the on-surface points $\mathcal{P}_s$ is that $\mathcal{P}_s$ may not exactly locate on zero level set of the signed distance function. We thus adopt to project every point $p_i \in \mathcal{P}_s$ to the zero level set of $f(x, \theta)$ using (Atzmon & Lipman, 2020):

$$p_i^{'} = p_i - \frac{\partial f(x, \theta)}{\partial p} f(p_i, \theta).$$

Since the gradient $\frac{\partial f(x, \theta)}{\partial p}$ can be achieved by the network backward propagation during the neural implicit function learning, such projection $\Gamma : \mathcal{P}_s \to \mathcal{P}_s^{'}$ is a *differential projection* that can be further used in the end-to-end neural implicit function learning. Based on the projected on-surface point set $\mathcal{P}_s^{'}$, we introduce the absolute distance loss L_{udf} as:

$$L_{udf} = \frac{1}{|\Omega|} \sum_i^{\Omega} |dist(p_i, \mathcal{P}_s^{'}) - dist(p_i, \mathbf{X})|^2,$$

with $dist(p_i, \mathbf{X}) = min_{x \in \mathbf{X}} |p_i - x|^2$ is distance function from p_i to point cloud set $\mathbf{X}$ and Ω is point sampling set. We adopt the back-propagation mechanism for the on-surface points P_s following (Remelli et al., 2020), by approximating the gradient between on-surface points P_s to any point p using the inverse surface normal (Remelli et al., 2020) $\frac{\partial P_s}{\partial p} = -n_p$, where n_p is the normal of the on-surface point P_s. In this way, we enable the differential back-propagation of the absolute distance loss L_{udf}.