

Unsupervised Point Cloud Reconstruction via Recurrent Multi-step Moving Strategy

Zheng Liu, Jianjun Zhang, Ming Zhang[†], Runze Ke, Chengcheng Yu, Ligang Liu

Abstract—Point cloud reconstruction is an ingredient in geometry modeling, computer graphics, and 3D vision. In this paper, we propose a novel unsupervised learning method called the Recurrent Multi-Step Moving Strategy, which progressively moves query points toward the underlying surface to accurately learn unsigned distance fields (UDFs) for point cloud reconstruction. Specifically, we design a recurrent network for UDF estimation that integrates a multi-step strategy for query movement. This model treats query movement as a trajectory prediction process, establishing dependencies between the current query move decision and the previous path, thus utilizing temporal information to improve UDF estimation accuracy. Further, we design distance and gradient regularization losses to ensure the precision, consistency, and continuity of the estimated UDFs. Extensive evaluations, comparisons, and ablation studies are conducted to show the superiority of our method over the competing approaches in terms of reconstruction accuracy and generality. Our unsupervised reconstruction method outperforms many supervised techniques and demonstrates efficacy across diverse scenarios, including single-object, indoor, and outdoor benchmarks. Our source code is available at <https://github.com/xiluocug/MSMS>.

Index Terms—Point cloud reconstruction, unsigned distance field, 3D deep learning, point clouds.

I. INTRODUCTION

THANKS to the advancements in scanning equipment, point clouds have become a common form of data acquisition in real-world scenarios [1], [2]. These acquired point clouds often require reconstruction to generate meshes, known as surface reconstruction. This is a fundamental problem in computer graphics and 3D vision. The reconstructed meshes, derived from the raw point clouds, have many applications. This innovation directly impacts key multimedia applications such as AR/VR, gaming, animation, and 3D simulation [3], where high-quality 3D reconstructions are crucial.

Surface reconstruction techniques can be broadly classified into two categories: explicit [4] and implicit [5]–[7] approaches, with implicit-based techniques being the dominant.

Zheng Liu, Runze Ke, and Chengcheng Yu are with School of Computer Science, China University of Geosciences (Wuhan), Wuhan 430079, China (e-mail: liu.zheng.jojo@gmail.com; 111826@cug.edu.cn; 20201003732@cug.edu.cn).

Jianjun Zhang is with National Engineering Research Center of Geographic Information System, China University of Geosciences (Wuhan), Wuhan 430079, China (e-mail: jianjunzhang@cug.edu.cn).

Ming Zhang is with Guangzhou Urban Planning & Design Survey Research Institute Co., Ltd. and Guangdong Enterprise Key Laboratory for Urban Sensing, Monitoring and Early Warning, Guangzhou 510060, China (e-mail: ming.zhang@gzpi.com.cn).

Ligang Liu is with School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230052, China (e-mail: lgl Liu@ustc.edu.cn).

[†]Corresponding author.

Implicit reconstruction methods, a key advancement in surface reconstruction, typically involve the computation of implicit function representations from raw point clouds. These implicit representations, which are mathematical functions that define the surface as the zero level set, are then used in conjunction with the marching cube algorithm and its variants to extract the reconstructed meshes. Implicit reconstruction methods have advanced over the long term in recent years due to the proliferation of deep learning techniques. In deep surface reconstruction, the three forms of neural implicit representations most frequently utilized are signed distance functions (SDFs) [8]–[13], occupancies (Occs) [14]–[18], and unsigned distance functions (UDFs). Implicit reconstruction techniques based on SDFs and Occs typically perform better when reconstructing watertight surfaces, which are surfaces without any holes or gaps, ensuring a continuous and complete surface. In contrast, because of its unique ability to not distinguish between inside and outside, UDF is a more ideal representation for depicting open surfaces, complex structures, and scenarios.

The existing UDF-based reconstruction techniques [19]–[21] rely heavily on the ground-truth UDFs or even large-scale meshes during the training phase, which limits their generalization, resulting in unsatisfactory results in complex structures, scenarios, and unseen objects. Recently, the self-supervised approaches [22]–[24] have received great interest and attention. These self-supervised approaches can learn UDFs from raw point clouds directly, which are particularly valuable in complex structures and scenarios where the ground-truth UDFs are scarce or expensive to derive. Although these self-supervised approaches can achieve impressive results without ground-truth UDFs, most of them employ a single-step moving strategy to pull the query points onto the underlying surface directly; the one-step moving distance is the estimated unsigned distance. However, this single-step strategy, while effective in some aspects, may lead to problems such as jitter and discontinuity in the estimated UDFs. This can result in bumps in the flat areas of the final reconstruction and flaws in the recovery of geometric features, highlighting the need for further research and improvement.

To address the aforementioned issues, we propose a novel multi-step moving strategy to accurately estimate UDFs from raw point clouds without relying on supervised information. Our multi-step strategy incrementally moves query points toward the underlying surface while simultaneously learning and optimizing the UDF and gradient field throughout the sequential moving process. We design a recurrent neural network architecture combined with a multi-step strategy to model sequential query movements. Our approach creates a

dependency between current and previous movement information, effectively leveraging both structural and temporal information to achieve more accurate UDF estimation. Moreover, we introduce regularization losses to constrain the distance and gradient fields, enforcing the smoothness and continuity of the estimated UDF's zero-level set while ensuring better consistency in the non-zero-level sets.

Our main contributions are summarized as follows:

- We propose a recurrent multi-step moving strategy, integrating a recurrent neural network with multi-step query movement. This approach progressively learns more accurate unsigned distance fields (UDFs) from raw point clouds without supervision.
- We propose Spatial Transformer (ST) and Recurrent Steps Aggregation (RSA) modules to model multi-step query movement sequentially, establishing dependencies between the current move decision and the previous path, thus utilizing temporal information to improve UDF estimation accuracy.
- We introduce two regularization losses to ensure the accuracy, continuity, and consistency of estimated UDFs.
- We conduct extensive experiments to show that our method achieves state-of-the-art performance on several benchmarks qualitatively and quantitatively.

II. RELATED WORK

In the last decades, numerous of traditional surface reconstruction techniques [5], [6], [25]–[29], have been proposed. However, most existing traditional methods typically rely on manual parameter tuning, which requires specialist knowledge for parameter selection. In addition, the time-consuming and laborious tuning procedure is crucial for producing promising results, resulting in limited generalizability. Therefore, we focus on reviewing learning-based surface reconstruction methods that closely relate to our approach, despite the availability of many traditional methods. For a comprehensive review of traditional methods, readers can refer to [30].

A. SDF-based Implicit Surface Reconstruction

In implicit field learning for surface reconstruction, signed distance fields (SDFs), which are able to discriminate between the interior and exterior regions of the underlying surface, are frequently utilized. Some SDF-based learning methods require ground-truth signed distance values as supervision data to learn implicit fields. DeepSDF [8] introduces a probabilistic auto-decoder to refine SDFs through latent vector optimization of 3D shapes. Based on DeepSDF, DeepLS [10] utilizes a grid to manage independent latent codes, each corresponding to a specific local area. Point2Surf [31] improves reconstruction accuracy by prior learning that combines fine-grained local and coarse-grained global data. POCO [11] utilizes point cloud convolution and potential vector computation to improve reconstruction accuracy and generalization capacity. Several approaches [32], [33] ensure a zero Hessian determinant for surface-near points, aligning gradients to yield a coarse yet faithful shape swiftly. These approaches can produce reconstruction results by progressively diminishing the singular Hessian weight. In addition, neural networks have been leveraged

to improve traditional methods. For example, the integration of implicit moving least-squares (IMLS) into neural networks by DOG [34], DeepIMLS [12], and NeuralIMLS [13] facilitates the estimation of SDFs directly from raw point clouds.

B. UDF-based Implicit Surface Reconstruction

Owing to its unsigned nature, UDF is more widely applicable, particularly while handling open surfaces and intricate situations. NDF [19] provides a differentiable function to represent UDFs and leverages neural networks to learn feature representations for approximating UDFs, thereby overcoming the non-differentiability problem of UDF near the zero-level set. Venkatesh et al. [20] proposed a closest-surface-point (CSP) representation. Their method utilizes the CSP function to identify the closest surface point to the query point and then the straightforward prediction of the unsigned distance value through simple Euclidean distance computation. Guillard et al. [35] proposed a fast and differentiable method, dubbed MeshUDF, to mesh UDFs and parameterize meshes to model non-manifold surfaces. Their approach enables direct surface reconstruction using UDFs without additional post-processing steps, thereby achieving efficient surface reconstruction. GeoUDF [36] introduces a point cloud upsampling method to capture local geometric information and employs this information as guidance for learning UDFs and the corresponding gradient fields. HSDF [37] begins by extracting multi-scale grid features to capture shape information, subsequently predicting the UDF and sign separately. This approach decomposes SDF into UDF and sign components, leveraging surface proximity to determine sign, effectively integrating the characteristics of UDF and SDF to deal with open surfaces and complex topologies. NeUDF [38] is capable of directly learning UDFs from 2D images and refining them through probability density and pixel color information. Additionally, NeUDF reduces surface gradient instability by substituting UDF surface normals with vectors obtained from interpolated UDF gradients and the normals of nearby points. UODF (Unsigned Orthogonal Distance Field) [39] is a novel implicit representation, distinct from traditional SDF and UDF, defined by the shortest distances from each point to the nearest surface in three orthogonal directions. This representation offers a six-sided perspective that delineates external and internal structural characteristics.

C. Unsupervised Implicit Field Learning

Although supervised implicit field learning techniques have made great progress, unsupervised methods are becoming increasingly popular because they don't require labeled data and are more flexible. Without using supervised data, Neural-Pull [40] introduces a new paradigm for directly learning SDF by shifting query points onto the underlying surface along their gradient. However, Neural-Pull struggles with reconstructing sparse point clouds, open surfaces, and scenarios. To address the issue, Zhou et al. [24] proposed a progressive strategy for learning UDF and applying consistency loss to constrain the learned implicit field. Their approach yields excellent reconstruction performance on open surfaces and scenarios.

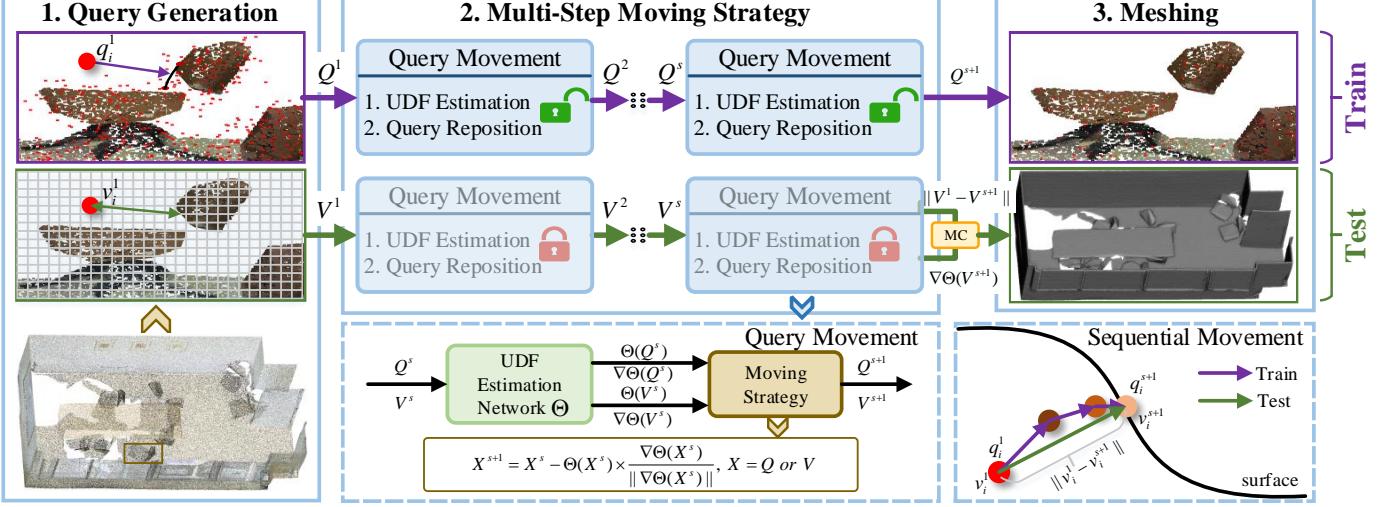


Fig. 1. The proposed method is divided into three phases: Query Generation, Multi-step Moving Strategy (MSMS), and Meshing. MSMS includes several cascaded Query Movement modules, each comprising two operations: UDF Estimation and Query Reposition. The diagram in the right corner demonstrates the sequential movement of the query point under MSMS. During training, queries are moved to the surface in an unsupervised manner to optimize the UDF Estimation Network. In the testing phase, the method predicts the unsigned distance and gradient direction for each grid vertex, followed by mesh extraction using Marching Cube (MC).

LevelSetUDF [41] employs level set projections for regularization, resulting in more continuous UDF field learning. Recently, Tian et al. [23] proposed a two-stage projection strategy to learn UDFs based on the locally optimal projection (LOP), which is robust to sparse point clouds. More recently, NeuralGF [42] introduces a multiple-step moving strategy to facilitate query points approximating the underlying surface for oriented normal field estimation. The aforementioned self-supervised methods typically rely on a single- or multiple-step moving strategy to shift query points onto the underlying surfaces for learning implicit fields. However, none of these methods take into account temporal information during the movement process, so they ignore the correlation and consistency during the movement steps, limiting the accuracy and robustness of the implicit field learning.

III. METHODOLOGY

A. Method Workflow

The workflow of our approach is illustrated in Fig. 1. Given a raw point cloud $P \in \mathbb{R}^{N_P \times 3}$, we apply Gaussian sampling to select a fixed count of points and then introduce perturbations to generate the initial query points $Q^1 = \{q_i^1\} \in \mathbb{R}^{N_Q \times 3}$. Then, we employ a multi-step moving strategy to predict the sequential movement of queries $\mathbf{Q} = \{Q^s\}$, where Q^s denotes queries at step s . At each step s , query movement consists of two operations: UDF estimation and query reposition. For the queries Q^s , our UDF estimation network Θ first predicts their unsigned distance $\Theta(Q^s)$ and gradient $\nabla\Theta(Q^s)$. Then, we move these queries along their estimated gradient by their predicted distance.

During the test phase, we initialize 3D grids to partition the point cloud at a specific resolution, generating the initial query set V^1 . Subsequently, we use the trained network Θ and multi-step movement strategy to predict the sequential query movements $\mathbf{V} = \{V^s\}$. This process allows us to compute the

unsigned distance at each grid vertex, facilitating the extraction of the zero-level set. Finally, the Marching Cube algorithm [43] is employed to produce the triangular mesh.

B. Recurrent UDF Estimation Network

Our recurrent UDF estimation network, paired with the multi-step strategy, is illustrated in Fig. 2. This recurrent neural network is designed to learn displacement vectors that iteratively move queries toward the underlying surface over several steps. At step s , the current queries are fed into a Transformer-based feature extractor, generating the per-point feature set H^s to establish spatial relationships. The cascaded Recurrent Step Aggregation (RSA) modules at level l combine the current step's features with those of previous steps, forming a multi-step temporal information model and yielding the fused structural and sequential information $\{H_l^s\}$. We then map the structural and sequential features to derive unsigned distances $\Theta(Q^s)$ and compute gradients $\nabla\Theta(Q^s)$ via backpropagation. By utilizing these distances and gradients, we relocate the current queries Q^s towards the underlying surface, yielding the updated queries Q^{s+1} as the input for the next step. Multi-step supervision [44], [45] is employed to ensure displacement precision. Thus, our recurrent network excels in UDF learning through the above principles.

1) *Transformer-based Feature Extraction:* We propose a transformer-based feature extractor, termed ST, as shown in Fig. 2. Spatial Position Encoding (SPE) is first introduced to embed the feature with spatial structural relationship as

$$R^s = \text{MLP}(Q^s : P^s : Q^s - P^s : \|Q^s - P^s\|), \quad (1)$$

where R^s denotes spatial encoding features, P^s represents the nearest neighbors of queries in Q^s within the raw point cloud P , and $:$ denotes the concatenation operation.

Next, we map 3D coordinates into deep features using MLP and Linear layers, yielding key F_k^s , value F_v^s , and query F_q^s .

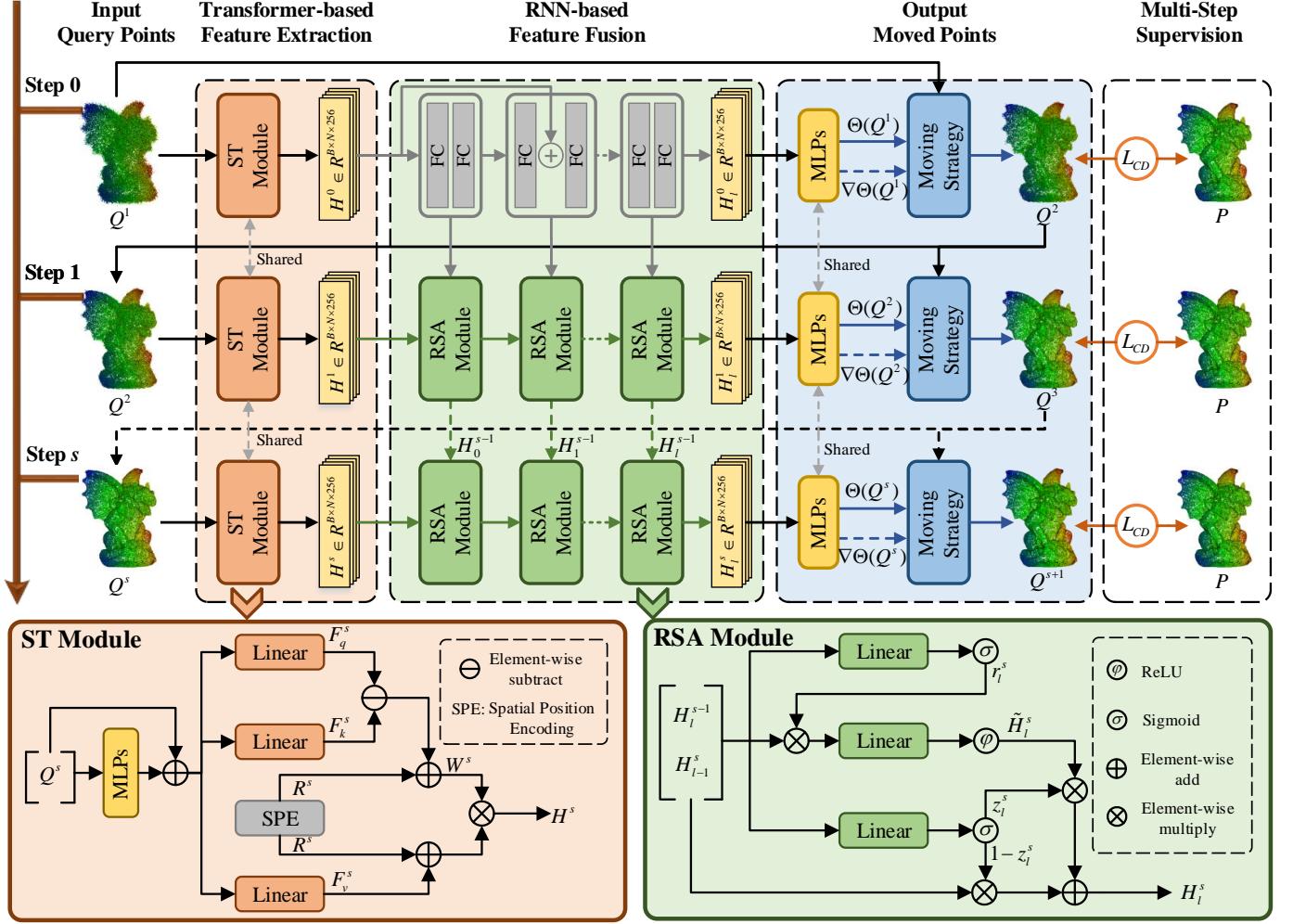


Fig. 2. Illustration of our recurrent UDF estimation network, incorporating the multi-step moving strategy. In each recursion, the Transformer-based Feature Extraction (ST) and RNN-based Feature Fusion (RSA) modules are used to learn feature embedding for UDF estimation. Then, Query Reposition pulls the queries to the underlying surface, with supervision on the relocated queries. By cascading multiple recursions, information from multiple movements is integrated, facilitating accurate UDF learning.

Then, we employ attention [46] to adaptively identify crucial elements within \$R^s\$ to produce contextual feature \$H^s\$ as

$$H^s = W^s \cdot (F_v^s + R^s), \quad (2)$$

where \$W^s = (F_q^s - F_k^s) + R^s\$. As expected, ST guarantees that the extracted features \$H^s\$ are both contextual and discriminative by capturing spatial dependencies.

2) *RNN-based Feature Fusion*: Single-step query movements relying solely on the current feature often lead to errors in estimating displacement, especially with large movements. In contrast, our multi-step strategy incorporates information from current and prior steps. Inspired by the Recurrent Neural Network (RNN) [47], we treat multi-step movement data sequentially. Thus, we implement RNN-based feature fusion through a series of Recurrent Step Aggregation (RSA) units. The detailed structure of the RSA unit is shown in Fig. 2. Given the input features from the current step \$H_{l-1}^{s-1}\$ and the previous step \$H_l^{s-1}\$, we employ GRUs to memorize and forget information based on the input features selectively and yield the refined feature embedding \$H_l^s\$ for the next layer \$l\$.

Specifically, GRU calculates an update gate \$z_s\$ and a reset gate \$r_s\$ to encode and forget information as follows:

$$z_s^s = \sigma(W_z(H_{l-1}^s : H_{l-1}^{s-1}) + b_z), \quad (3)$$

$$r_s^s = \sigma(W_r(H_{l-1}^s : H_l^{s-1}) + b_r), \quad (4)$$

where \$W_z\$ and \$W_r\$ are weight matrices, \$b_z\$ and \$b_r\$ are biases, and \$\sigma\$ is the sigmoid activation function. We then compute the candidate hidden state \$\tilde{H}_l^s\$ using reset gate \$r_s^s\$ as

$$\tilde{H}_l^s = \phi(W_h(r_s^s \cdot H_{l-1}^{s-1} : H_{l-1}^s) + b_h), \quad (5)$$

where \$W_h\$ is a weight matrix, \$b_h\$ is a bias, and \$\phi\$ is activation function ReLU. Finally, the hidden state \$H_l^s\$ for the current step is derived from the integration of the candidate hidden state \$\tilde{H}_l^s\$ and the input feature \$H_l^s\$ via the update gate \$z_s^s\$ as

$$H_l^s = z_s^s \cdot \tilde{H}_l^s + (1 - z_s^s) \cdot H_{l-1}^s. \quad (6)$$

C. Multi-step Moving Strategy

After obtaining the structural and sequential feature \$H_l^s\$, the unsigned distance and gradient vectors \$(\Theta(Q^s)\$ and \$\nabla\Theta(Q^s))

of the current-step queries can be derived by MLPs and back-propagation. Inspired by Neural-Pull [40], we can reposition the queries for the next step as follows:

$$Q^{s+1} = Q^s - \Theta(Q^s) \times \frac{\nabla \Theta(Q^s)}{\|\nabla \Theta(Q^s)\|}, \quad (7)$$

where Q^{s+1} represents the moved queries.

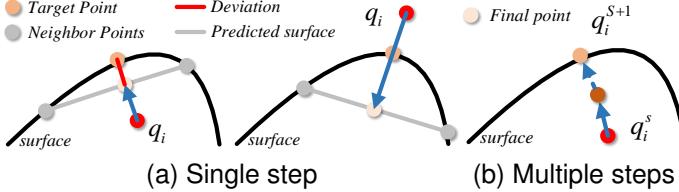


Fig. 3. Comparison of single-step and multi-step moving strategies. (a) Two primary challenges with the single-step scheme. (b) Our multi-step strategy can progressively pull query points to the underlying surface more accurately.

To show the superiority of our multi-step movement strategy, we compare it to the single-step scheme. As depicted in Fig. 3, the single-step scheme often struggles to precisely align query points with the target surface, particularly when the queries are distant from the surface. These artifacts may cause queries to either drift away from or penetrate the surface, leading to inaccuracies in learned UDFs and defects in reconstructed meshes. Therefore, the adoption of the multi-step strategy is not just a choice, but a necessity. Our strategy treats the query movement as a trajectory prediction process, integrating multi-step information to determine the direction and distance of the next step. The key to our strategy is the process of progressively pulling the queries toward the underlying surface in a coarse-to-fine manner, which significantly improves the UDF learning, surpassing the limitations of the single-step scheme.

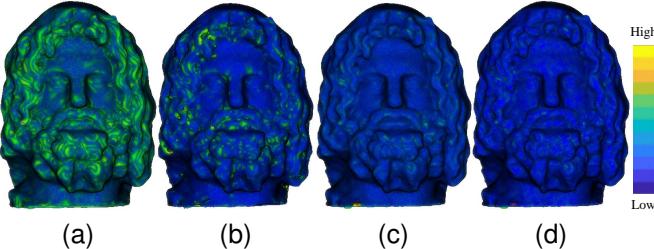


Fig. 4. Visual comparison of error maps for different moving strategies: (a) single-step, (b) multi-step, (c) multi-step with loss constraints, and (d) multi-step with loss constraints and sequential supervision.

Error accumulation mitigation. As we know, the multi-step moving strategy may introduce deviations at each step, which accumulate over iterations and ultimately degrade prediction accuracy. To address this issue, we employ two key mechanisms. First, we formulate distance field (10) and gradient field (11) loss constraints to maintain directional consistency in query point movement while preventing surface penetration; see Section III-D for details. Second, we introduce a sequential supervision strategy (9) to ensure the query points aligned with the underlying surface in each moving step.

These two mechanisms collaboratively control the cumulative errors generated in each movement step, thereby making the predictions progressively more reliable. Fig. 4 visualizes the error maps across different moving strategies, quantifying the deviation between moved query points and the underlying surface. As we can see, our multi-step moving strategy with loss constraints and sequential supervision can achieve more accurate convergence to the underlying surface, which mitigates error accumulation effectively.

D. Training Losses

For training our proposed network, we construct a loss function that integrates a data fidelity term with distance and gradient regularizers. The total loss is expressed as:

$$\mathcal{L} = \mathcal{L}_{CD} + \lambda_1 \mathcal{L}_{Distance} + \lambda_2 \mathcal{L}_{Gradient}, \quad (8)$$

where λ_1, λ_2 are hyperparameters that control the importance of term. These regularizers ensure the accuracy, consistency, and continuity of the estimated UDF, as shown in Fig. 5.

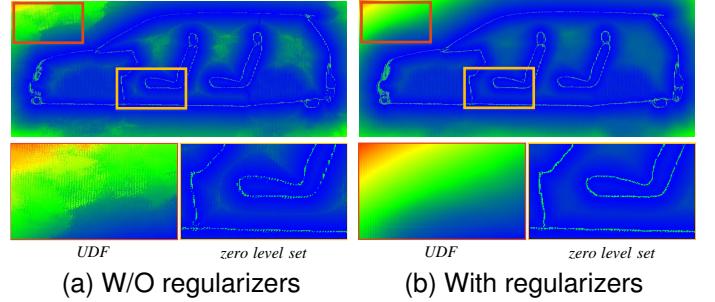


Fig. 5. Effect of $\mathcal{L}_{Distance}$ and $\mathcal{L}_{Gradient}$ regularizers. We visualize the UDF field and its zero-level set, estimated (a) without and (b) with the two regularizers. Points colored darker (i.e., more blue) indicate the smaller distance from the underlying surface.

1) *CD Loss*: For each moving step, we enforce the CD loss to quantify the discrepancy between the current relocated queries and the raw point cloud, which is formulated as

$$\mathcal{L}_{CD} = \frac{1}{S} \sum_{s=1}^S \left(\frac{1}{N_Q} \sum_{q_i^{s+1} \in Q^{s+1}} \min_{p_j \in P} \|q_i^{s+1} - p_j\| + \frac{1}{N_P} \sum_{p_j \in P} \min_{q_i^{s+1} \in Q^{s+1}} \|p_j - q_i^{s+1}\| \right), \quad (9)$$

where p_j denotes the nearest neighbor to the relocated query within the raw point cloud P .

2) *Distance Loss*: Based on the observation that the unsigned distance of a query point indicates its shortest distance to the surface, we introduce the following distance loss

$$\mathcal{L}_{Distance} = \frac{1}{S} \sum_{s=1}^S \left(\frac{1}{N_Q} \sum_{q_i^s \in Q^s} \max\{(\Theta(q_i^s) - \min \|q_i^s - p_j\|), 0\} \right). \quad (10)$$

This distance loss can address the inaccurate UDF estimation caused by incorrect query reposition. As we can see, the

L_{CD} value in the case of Fig. 6b is equal to that in Fig. 6c, leading to incorrect query reposition.

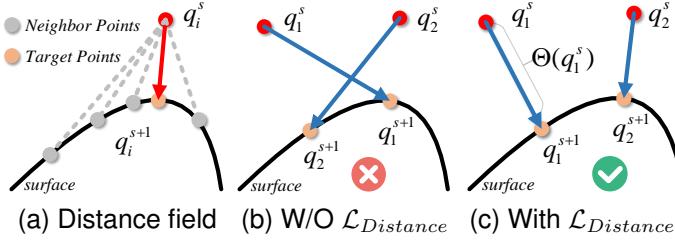


Fig. 6. (a) illustrates how the unsigned distance from the query point is less than that to neighbor points. (b) and (c) depict the repositioning of query points without and with the distance regularizer.

3) *Gradient Loss*: The joint gradient loss function, which includes three regularizers to optimize the gradient field of UDF, is formulated as

$$\mathcal{L}_{\text{Gradient}} = L_{\text{mag}} + L_{\text{dir}} + L_{\text{lap}}. \quad (11)$$

The *gradient normalization loss* L_{mag} aims to normalize the estimated UDF gradients across the domain, ensuring consistent gradient magnitudes. This regularizer is defined as

$$L_{\text{mag}} = \frac{1}{S} \sum_{s=1}^S \left(\frac{1}{N_Q} \sum_{q_i^s \in Q^s} (\|\nabla \Theta(q_i^s)\| - 1)^2 \right). \quad (12)$$

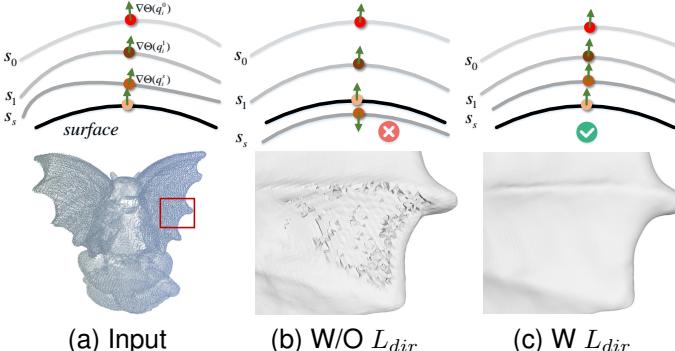


Fig. 7. (a) The original UDF level sets are displayed at the top, accompanied by the raw point cloud beneath. (b) Optimized UDF results, including level sets at the top and reconstructions at the bottom, without incorporating the gradient direction consistency loss. (c) Results where this loss is applied.

The *gradient direction consistency loss* L_{dir} encourages consistent gradient direction in predicted sequential queries, thus enhancing the alignment and continuity of the estimated UDF's level sets; see Fig. 7 for illustration. This gradient regularizer is defined as follows:

$$L_{\text{dir}} = \frac{1}{S-1} \sum_{s=1}^{S-1} \left(\frac{1}{N_Q} \sum_{i=1}^{N_Q} \left(1 - \frac{\nabla \Theta(q_i^s) \cdot \nabla \Theta(q_i^{s+1})}{\|\nabla \Theta(q_i^s)\| \cdot \|\nabla \Theta(q_i^{s+1})\|} \right) \right). \quad (13)$$

The *Laplacian smoothness loss* aims to minimize high-frequency disturbances and enforce smooth surfaces in the estimated UDF, formulated as

$$L_{\text{lap}} = \frac{1}{S} \sum_{s=1}^S \left(\frac{1}{N_Q} \sum_{q_i^s \in Q^s} \|\Delta \Theta(q_i^s)\|^2 \right). \quad (14)$$

IV. EXPERIMENTS AND DISCUSSIONS

Datasets and metrics. We evaluate the effectiveness of our point cloud reconstruction method across various benchmarks, including synthetic (Sec. IV-A), real scan (Sec. IV-B), indoor scene (Sec. IV-C), and outdoor scene (Sec. IV-D) datasets. We also conduct ablation studies to assess the contributions of our designed modules and loss functions (Sec. IV-F). For fairness comparison, the query point sampling strategy from CAP-UDF [24] is employed during training, and the same 3D grid resolution is applied during testing. Reconstructed mesh quality is evaluated quantitatively using L1 Chamfer Distance (L1CD), L2 Chamfer Distance (L2CD), Normal Consistency (NC), and F-Score with 0.005 and 0.01 thresholds.

Implementation details. Experiments are conducted on an Ubuntu system with an Intel i7-12700K processor and dual NVIDIA 4090 GPUs. Our method employs PyTorch and the ADAM optimizer, with a learning rate of 1×10^{-3} , a batch size of 5000, and 1×10^5 epochs. The multi-step moving strategy uses $S = 3$ steps.

A. Evaluation on Synthetic Shape Reconstruction

ShapeNet [48] provides a comprehensive 3D object dataset, featuring over 3×10^6 models from roughly 3,135 categories, including chairs, tables, and cars. The rich annotations make ShapeNet a famous benchmark for point cloud reconstruction. Analogous to CAP-UDF, we choose the car category from ShapeNet for assessing our reconstruction results, due to its extensive range of multi-layered and open shapes.

TABLE I
QUANTITATIVE EVALUATION ON SHAPENET CAR DATASET (L2CD $\times 10^4$).

Method	L2CD \downarrow		F-Score \uparrow		NC \uparrow
	Mean	Median	F1 0.005	F1 0.01	
Input	0.363	0.355	48.50	88.34	-
WatertightGT	2.628	2.293	68.82	81.60	-
GT	0.076	0.074	95.70	99.99	-
NDF [19]	0.202	0.193	77.40	97.97	79.1
GIFS [49]	0.128	0.123	88.05	99.31	-
CAP-UDF [24]	0.119	0.114	88.55	99.82	82.5
LevelSetUDF [41]	0.098	0.097	92.18	99.90	85.0
Ours	0.079	0.079	96.40	99.90	87.6

We compare our method with four state-of-the-art approaches: NDF [19], GIFS [49], LevelSetUDF [41], and CAP-UDF [24]. The quantitative results are listed in Table. II. Our method outperforms on metrics such as L2CD, F-Score, and NC. A visual comparison in Fig. 8 illustrates the reconstruction quality. Notably, our method reconstructs more complete car tires and smoother interior seats than others.

B. Evaluation on Real Scan Reconstruction

SRB [50] is a real scanning dataset that primarily includes Gargoyle, Dancing Children, Quasimoto, Anchor, and Daratech point clouds. Compared to synthetic data [48], real-world data typically exhibit more significant noise and uneven point distributions, which can further verify our method's efficacy.

We provide a numerical comparison with the competing methods, including IGR [51], Point2Mesh [52], SPSR [53],

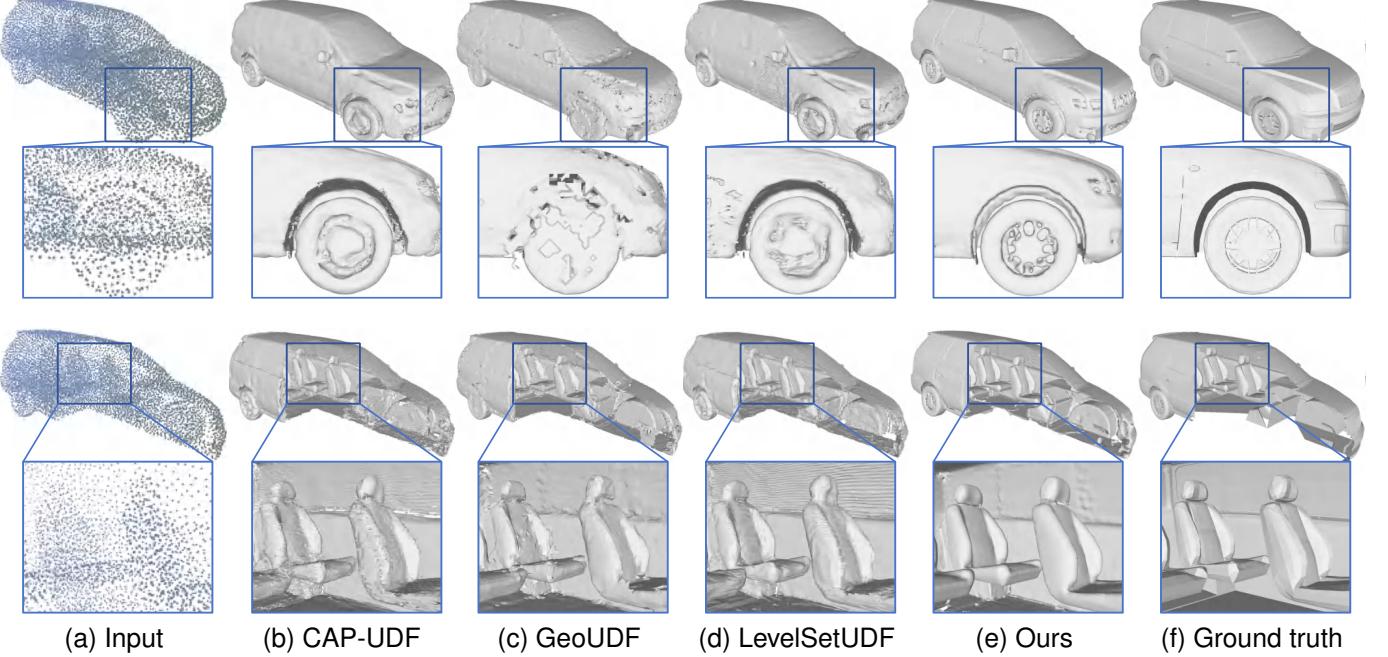


Fig. 8. Visual comparison of Car in ShapeNet dataset. From left to right: raw point cloud, results of CAP-UDF, GeoUDF, LevelSetUDF, our method, and ground truth. The zoomed views highlight that our method exhibits improved piecewise smoothness while effectively preserving geometric details compared to the competing approaches.

TABLE II
QUANTITATIVE EVALUATION ON SRB DATASET.

Method	L1CD \downarrow	F-Score \uparrow	NC \uparrow
IGR [51]	0.178	75.5	86.9
Point2Mesh [52]	0.116	64.8	-
SPSR [53]	0.232	73.5	-
SAP [54]	0.076	83.0	88.6
Neural-Pull [40]	0.106	79.7	87.2
NDF [19]	0.238	68.6	80.4
CAP-UDF [24]	0.073	84.5	88.6
LevelSetUDF [41]	0.071	85.1	91.0
Ours	0.069	89.4	90.6

and SAP [54], as shown in Table II. Our method achieves the best results in terms of L1CD and F-Score and is nearly the second-best result in NC. Fig. 9 shows the visual comparison. As we can see, our method prevents visible artifacts, showing superior performance in preserving sharp edges and corners and recovering smooth regions.

C. Evaluation on Indoor Scene Reconstruction

The indoor scenes [55] are crucial to validate our method’s generalization capability, which includes RGB-D videos and 3D surfaces from diverse settings such as Burgher, Lounge, Copyroom, Cactusgarden, Stonewall, Totempole, Readin-groom, and Fountain. We test our method and the competing ones at two densities: 500 points/m² and 1000 points/m².

As demonstrated in Table III, our method achieves superior performance in the Copyroom, Stonewall, and Totempole scenes according to both L2CD and NC metrics. In addition, our method consistently ranks in the top two for the Burgher and Lounge scenes, underscoring its robustness and superiority

in scene reconstruction. The qualitative comparison illustrates that our method has the capability to reconstruct more continuous surfaces with clearer boundaries, as shown in Fig. 10.

D. Evaluation on Outdoor Scene Reconstruction

UrbanScene3D [57] offers a comprehensive urban scene perception dataset comprising over 128,000 high-resolution images across 16 diverse urban areas, spanning 136 square kilometers. Equipped with accurate LiDAR scans and diverse imagery, it allows us to visually compare outdoor scenarios.

In contrast to single objects and indoor scenes, outdoor scenes, characterized by their open and irregular geometries, demand methods with robust topological reconstruction capabilities. Fig. 11 illustrates that our method surpasses the traditional technique [53] by reconstructing complex scenes without generating superfluous surfaces and accurately preserving geometric topology, thereby highlighting our superior performance in outdoor reconstruction.

E. Evaluation on Challenging Scenarios

Low-density, complex surfaces, and high occlusion scenarios present challenges in the reconstruction task. Fig. 12 presents the results of our robustness test under these conditions. As shown, our method outperforms competing approaches, demonstrating superior reconstruction quality, while other methods exhibit one or more limitations. The robustness test shows that our approach is robust to these challenges, consistently producing visually appealing results.

F. Ablation Studies

We conduct ablated experiments to validate the framework design, module structure, the effectiveness of loss function,

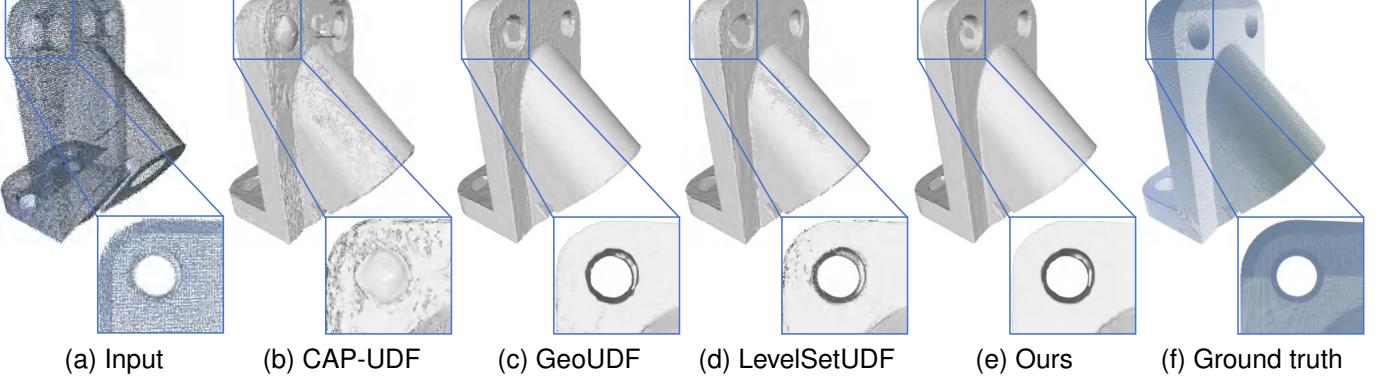


Fig. 9. Visual comparison of Anchor in SRB dataset. From left to right: raw point cloud, results of CAP-UDF, GeoUDF, LevelSetUDF, our method, and ground truth. The zoomed views highlight that our method not only preserves structural features more effectively but also prevents artifacts, particularly in piecewise smooth regions.

TABLE III
QUANTITATIVE EVALUATION ON INDOOR SCENES ($L2CD \times 10^3$).

	Method	Burgher $L2CD \downarrow$	$NC \uparrow$	Lounge $L2CD$	NC	Copyroom $L2CD$	NC	Stonewall $L2CD$	NC	Totempole $L2CD$	NC
500/m ²	ConvONet [17]	26.97	0.905	9.044	0.894	10.08	0.885	17.70	0.909	2.165	0.937
	LIG [51]	3.080	0.840	6.729	0.831	4.058	0.810	4.919	0.878	9.38	0.887
	DeepLS [10]	0.714	0.923	10.88	0.814	0.552	0.907	0.673	0.951	21.15	0.927
	NDF [19]	0.546	0.917	0.314	0.921	0.242	0.907	0.226	0.949	1.049	0.939
	OnSurf [56]	0.609	0.930	0.529	0.926	0.483	0.908	0.666	0.955	2.025	0.954
	CAP-UDF [24]	0.192	0.911	0.099	0.911	0.120	0.902	0.069	0.958	0.131	0.954
	LevelSetUDF [41]	0.161	0.916	0.087	0.933	0.102	0.920	0.061	0.964	0.114	0.960
	MSMS	0.128	0.923	0.065	0.930	0.074	0.921	0.047	0.970	0.092	0.967
1000/m ²	ConvONet [17]	27.46	0.907	9.54	0.894	10.97	0.892	20.46	0.905	2.054	0.943
	LIG [51]	3.055	0.835	6.972	0.833	3.61	0.810	5.032	0.879	9.58	0.887
	DeepLS [10]	0.401	0.920	6.103	0.848	0.609	0.901	0.320	0.954	0.601	0.950
	NDF [19]	1.168	0.901	0.393	0.910	0.269	0.908	0.509	0.936	2.020	0.922
	OnSurf [56]	1.339	0.929	0.432	0.934	0.405	0.914	0.266	0.957	1.089	0.954
	CAP-UDF [24]	0.191	0.910	0.092	0.927	0.113	0.911	0.066	0.962	0.139	0.955
	LevelSetUDF [41]	0.146	0.921	0.068	0.941	0.093	0.925	0.050	0.970	0.107	0.960
	Ours	0.125	0.923	0.059	0.946	0.069	0.936	0.048	0.972	0.094	0.961

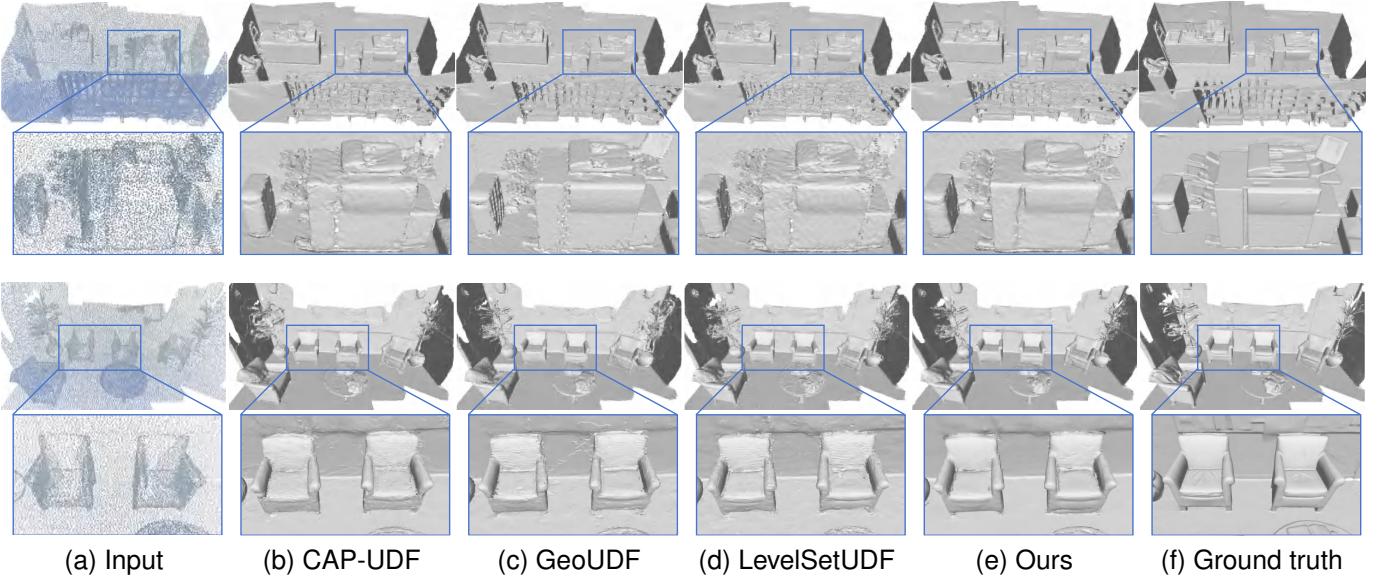


Fig. 10. Visual comparison of indoor scenes. From left to right: raw point cloud, results of CAP-UDF, GeoUDF, LevelSetUDF and our method, ground truth. The zoomed views highlight that our method outperforms the others in terms of both noise removal and geometry feature preservation.

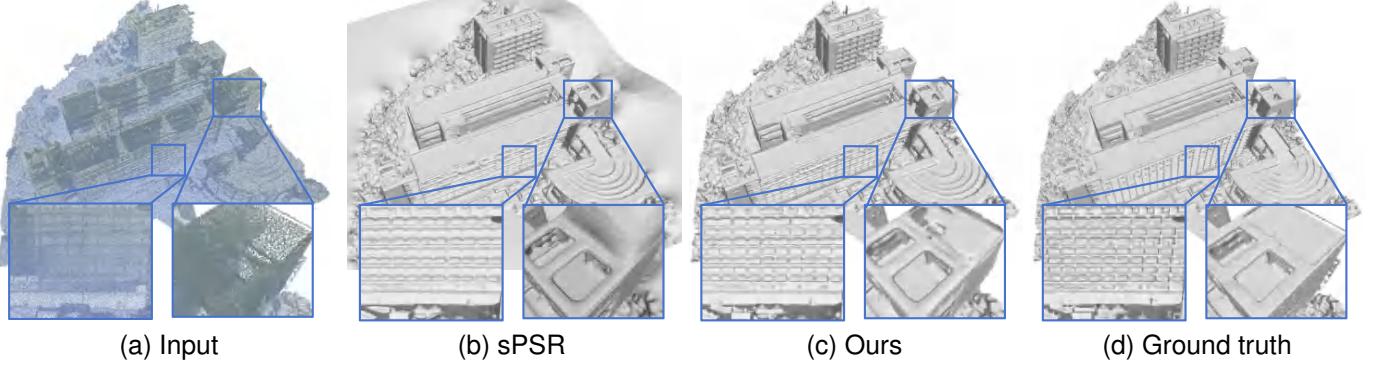


Fig. 11. Visual comparison on UrbanScene3D dataset. From left to right: raw point cloud, results of screened Poisson Surface Reconstruction (sPSR) and our method, ground truth. The zoomed views highlight our method's ability to handle complex topology, preserve details, and achieve artifact-free reconstruction.

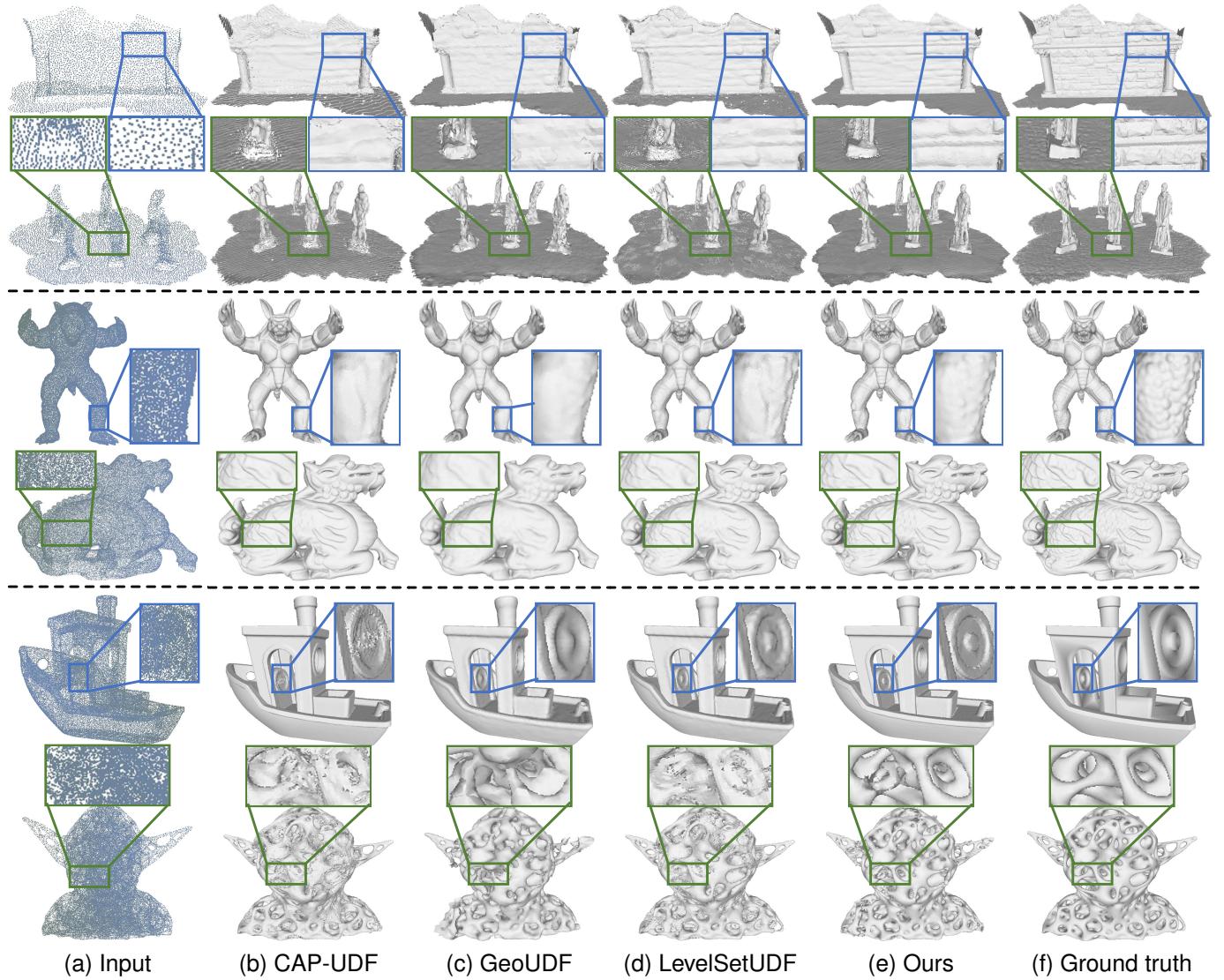


Fig. 12. Robustness test on challenging scenarios. From top to bottom: low density, complex surfaces, and high occlusion. The zoomed-in views highlight that our method achieves superior reconstruction quality, whereas other methods exhibit one or more limitations.

key parameter choice. These experiments aim to validate the individual contribution of each component of our method. All ablation studies are carried out on the ShapeNet Car dataset.

The effect of MSMS. Our method relies heavily on the Multi-Step Moving Strategy (MSMS). As depicted in Table IV, four key comparisons—baseline vs. A1, A2 vs. A4, A3 vs.

A5, and A6 vs. Full Model—demonstrate MSMS’s superiority over the single-step scheme. Specifically, A1 surpasses the baseline in L2CD by 0.027 and in NC by 1.8 points, achieving this without additional constraints. These results highlight the effectiveness of MSMS.

TABLE IV
ABLATION EXPERIMENTS ON MSMS AND LOSS FUNCTIONS.

Variant	MSMS	$\mathcal{L}_{Distance}$	$\mathcal{L}_{Gradient}$	ShapeNet	
				L2CD	NC
Baseline				0.119	85.0
A1	✓			0.092	86.8
A2		✓		0.106	85.8
A3			✓	0.098	86.8
A4	✓	✓		0.091	87.0
A5	✓		✓	0.086	87.2
A6		✓	✓	0.092	86.9
Full Model	✓	✓	✓	0.079	87.6

The effect of loss functions. Fig. 5 illustrates that the $\mathcal{L}_{Distance}$ and $\mathcal{L}_{Gradient}$ constraints aid our method in achieving a more accurate, continuous, and coherent UDF. Table IV shows that the $\mathcal{L}_{Gradient}$ constraint notably improves the NC metric, as observed in comparisons between (Baseline, A1, A2) and (A1, A4, A5). Comparing the Full Model with the Baseline highlights the improved performance of our MSMS, reinforced by the two regularization terms.

TABLE V
ABLATION EXPERIMENTS ON ST AND RSA MODULES.

Variant	ST	RSA	ShapeNet	
			L2CD	NC
Baseline			0.119	85.0
A1	✓		0.092	86.8
A2		✓	0.083	87.0
Full Model	✓	✓	0.079	87.6

The effect of core modules. Transformer Feature Extraction (ST) and RNN-based Feature Fusion (RSA) modules are two core modules in our UDF Estimation Network. In contrast, the Baseline employs several linear layers for UDF estimation. As indicated in Table V, ST and RSA both result in notable performance enhancements, with improvements of (0.027, 1.8) and (0.036, 2.0) in L2CD and NC, respectively. RSA plays a more crucial role in the performance boost, given its role in integrating multi-step movement data from query points.

TABLE VI
ABLATION EXPERIMENTS ON STEP NUMBER CHOICE.

S	1	2	3	4
L2CD	0.092	0.082	0.079	0.080
NC	86.9	87.3	87.6	87.1

Step number choice. In evaluating the effect of movement step count S in MSMS, we vary S across 1, 2, 3, and 4, as summarized in Table VI. Upon analysis, our model achieves optimal performance at $S = 3$. Optimal model performance was observed at $S = 3$. A lower step count restricts the aggregation of multi-step temporal information, potentially degrading performance, whereas a higher count may result in model instability.

G. Complexity and Efficiency Analysis

To comprehensively assess the trade-offs introduced by our multi-step moving strategy, we conduct experiments to evaluate model complexity (including the total number of parameters and memory consumption) and efficiency (including training and inference time) of the tested methods, with the results summarized in Table VII.

Training time. Intuitively, the training time of our multi-step moving method should be much longer than that of single-step methods. However, due to the combination of our multi-step strategy and regularization constraints, our method can converge quickly after only a few steps. As a result, our training time (18.48 minutes) remains highly competitive—comparable to single-step methods such as CAP-UDF and LevelSetUDF, and significantly shorter than GeoUDF (238.62 minutes), which requires learning global priors across the entire dataset.

Inference time. Due to the repeated inference passes in the multi-step pipeline, our method has a longer inference time compared to competing approaches. To reduce computational overhead, after the initial prediction pass, we use the gradients of the eight corners of each cube to determine if a surface intersects the cube, allowing us to filter out empty cubes. As a result, subsequent inference passes only require distance predictions for the corner points of cubes near the surface, significantly reducing inference time. Combined with our acceleration technique, the inference time of our multi-step method remains acceptable.

Memory consumption. The integration of Transformer and RNN structures in our method leads to a higher parameter count compared to CAP-UDF and LevelSetUDF, which rely solely on fully connected layers. In addition, our multi-step strategy requires intermediate computations and storage, resulting in a memory usage of 2.66G, slightly higher than CAP-UDF (1.99G) and LevelSetUDF (2.34G), but significantly lower than GeoUDF (4.48G).

Accuracy. Our method, while slightly more computationally expensive, achieves superior reconstruction accuracy. It yields the lowest L2CD of 0.079, outperforming CAP-UDF (0.114), GeoUDF (0.082), and LevelSetUDF (0.097). In addition, it achieves the highest F-score (96.40) and NC (87.6), surpassing all competing methods. These results highlight the effectiveness of our method in producing precise and smooth reconstructed surfaces.

To summarize, although our multi-step strategy introduces extra computational costs, it effectively balances precision and effectiveness. The improvements in L2CD, F-score, and Normal Consistency (NC) validate the substantial upgrade in reconstruction quality, showing our method highly applicable for practical scenarios. Future research aims to further optimize memory usage and inference speed of our method.

V. CONCLUSION

In this study, we introduce an unsupervised method to learn neural implicit fields from raw point clouds. Specifically, we propose a multi-step moving strategy to learn the unsigned distance field (UDF) by gradually moving 3D queries toward

TABLE VII
MODEL COMPLEXITY AND EFFICIENCY ANALYSIS ON THE SHAPENETCAR DATASET.

Method	Params. (K)	Memory (G)	Training time (min)	Inference time (s)	Reconstruction Quality		
					L2CD ↓	F-Score ↑	NC ↑
CAP-UDF [24]	581.24	1.99	15.30	133.86	0.114	88.55	82.5
GeoUDF [36]	1019.66	4.48	238.62	79.26	0.082	94.25	86.3
LevelSetUDF [41]	581.24	2.34	17.35	130.75	0.097	92.18	85.0
Ours	710.94	2.66	18.48	200.35	0.079	96.40	87.6

the underlying surface. This strategy conceptualizes query point movement as a temporal path prediction task, utilizing a recurrent neural network to capture movement dependencies and integrate temporal information over sequential steps. In conjunction with the proposed recurrent network, we design several loss functions to ensure accuracy, continuity, and consistency in UDF estimation. Our method achieves a new state-of-the-art for point cloud reconstruction on several benchmarks, with substantial improvements in visual quality and numerical evaluation.

Building upon the wealth of opportunities for future exploration, the proposed recurrent multi-step moving strategy can be applied to numerous computer graphics and 3D vision problems—such as oriented normal estimation, point cloud denoising, and completion—thus opening up a variety of exciting research directions. Furthermore, the accurate estimation of UDFs from heavily corrupted point clouds remains a challenge area for research.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China (2022YFB3904100), Guangdong Enterprise Key Laboratory for Urban Sensing, Monitoring and Early Warning (2020B121202019).

REFERENCES

- [1] Z. Liu, Y. Zhao, S. Zhan, Y. Liu, R. Chen, and Y. He, “PCDNF: Revisiting learning-based point cloud denoising via joint normal filtering,” *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 8, pp. 5419–5436, 2024.
- [2] Z. Wei, H. Chen, L. Nan, J. Wang, J. Qin, and M. Wei, “PathNet: Path-selective point cloud denoising,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 6, pp. 4426–4442, 2024.
- [3] C. Chen, A. Jin, Z. Wang, Y. Zheng, B. Yang, J. Zhou, Y. Xu, and Z. Tu, “SGSR-Net: Structure semantics guided lidar super-resolution network for indoor lidar slam,” *IEEE Trans. Multimedia*, vol. 26, pp. 1842–1854, 2024.
- [4] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, “The ball-pivoting algorithm for surface reconstruction,” *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 4, pp. 349–359, 1999.
- [5] M. Kazhdan and H. Hoppe, “Screened Poisson surface reconstruction,” *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, 2013.
- [6] F. Hou, C. Wang, W. Wang, H. Qin, C. Qian, and Y. He, “Iterative Poisson surface reconstruction (iPSR) for unoriented points,” *ACM Trans. Graph.*, vol. 41, no. 4, pp. 128:1–13, 2022.
- [7] H. Tian, Z. Qin, R. Yi, C. Zhu, and K. Xu, “Tensorformer: Normalized matrix attention transformer for high-quality point cloud reconstruction,” *IEEE Trans. Multimedia*, vol. 27, pp. 718–730, 2025.
- [8] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “DeepSDF: Learning continuous signed distance functions for shape representation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 165–174.
- [9] M. Atzmon and Y. Lipman, “SAL: Sign agnostic learning of shapes from raw data,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 2562–2571.
- [10] R. Chabra, E. Lenssen, Jan E. and Ilg, T. Schmidt, J. Straub, S. Lovegrove, and R. Newcombe, “Deep local shapes: Learning local sdf priors for detailed 3d reconstruction,” in *Proc. Eur. Conf. Comput. Vis.*, 2020.
- [11] A. Boulch and R. Marlet, “POCO: Point convolution for surface reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.
- [12] S.-L. Liu, H.-X. Guo, H. Pan, P.-S. Wang, X. Tong, and Y. Liu, “Deep implicit moving least-squares functions for 3d reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
- [13] Z. Wang, P. Wang, P.-S. Wang, Q. Dong, J. Gao, S. Chen, S. Xin, C. Tu, and W. Wang, “Neural-IMLS: Self-supervised implicit moving least-squares network for surface reconstruction,” *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 8, pp. 5018–5033, 2024.
- [14] S. Li, G. Gao, Y. Liu, Y.-S. Liu, and M. Gu, “GridFormer: Point-grid transformer for surface reconstruction,” in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 3163–3171.
- [15] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3d reconstruction in function space,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 4460–4470.
- [16] S. Lionar, D. Emtsev, D. Svilarkovic, and S. Peng, “Dynamic plane convolutional occupancy networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 1829–1838.
- [17] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutional occupancy networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 523–540.
- [18] J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, “Sa-convolnet: Sign-agnostic optimization of convolutional occupancy networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 6504–6513.
- [19] J. Chibane, A. Mir, and G. Pons-Moll, “Neural unsigned distance fields for implicit function learning,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21638–21652.
- [20] R. Venkatesh, T. Karmali, S. Sharma, A. Ghosh, R. V. Babu, L. A. Jeni, and M. Singh, “Deep implicit surface point prediction networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 12653–12662.
- [21] F. Zhao, W. Wang, S. Liao, and L. Shao, “Learning anchored unsigned distance functions with gradient direction alignment for single-view garment reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 12674–12683.
- [22] J. Zhou, B. Ma, Y.-S. Liu, Y. Fang, and Z. Han, “Learning consistency-aware unsigned distance functions progressively from raw point clouds,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 16481–16494.
- [23] H. Tian, C. Zhu, Y. Shi, and K. Xu, “SuperUDF: Self-supervised UDF estimation for surface reconstruction,” *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 9, pp. 5965–5975, 2024.
- [24] J. Zhou, B. Ma, S. Li, Y.-S. Liu, Y. Fang, and Z. Han, “CAP-UDF: Learning unsigned distance functions progressively from raw point clouds with consistency-aware field optimization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 7475–7492, 2024.
- [25] A. C. Öztireli, G. Guennebaud, and M. Gross, “Feature preserving point set surfaces based on non-linear kernel regression,” *Comput. Graph. Forum*, vol. 28, no. 2, pp. 493–501, 2009.
- [26] N. Salman, M. Yvinec, and Q. Merigot, “Feature preserving mesh generation from 3d point clouds,” *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1623–1632, 2010.
- [27] R. Xu, Z. Wang, Z. Dou, C. Zong, S. Xin, M. Jiang, T. Ju, and C. Tu, “RFEPS: Reconstructing feature-line equipped polygonal surface,” *ACM Trans. Graph.*, vol. 41, no. 6, pp. 1–15, 2022.
- [28] C. Lv, W. Lin, and B. Zhao, “Voxel structure-based mesh reconstruction from a 3d point cloud,” *IEEE Trans. Multimedia*, vol. 24, pp. 1815–1829, 2022.

- [29] R. Fu, K. Hormann, and P. Alliez, “Lfs-aware surface reconstruction from unoriented 3d point clouds,” *IEEE Trans. Multimedia*, vol. 26, pp. 11 415–11 427, 2024.
- [30] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, “A survey of surface reconstruction from point clouds,” *Comput. Graph. Forum*, vol. 36, no. 1, pp. 301–329, 2017.
- [31] P. Erler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Wimmer, “Points2Surf: Learning implicit surfaces from point clouds,” in *Proc. Eur. Conf. Comput. Vis.*, 2020.
- [32] R. Wang, Z. Wang, Y. Zhang, S. Chen, S. Xin, C. Tu, and W. Wang, “Aligning gradient and hessian for neural signed distance function,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2023, pp. 63 515–63 528.
- [33] Z. Wang, Y. Zhang, R. Xu, F. Zhang, P.-S. Wang, S. Chen, S. Xin, W. Wang, and C. Tu, “Neural-Singular-Hessian: Implicit neural representation of unoriented point clouds by enforcing singular hessian,” *ACM Trans. Graph.*, vol. 42, no. 6, 2023.
- [34] P.-S. Wang, Y. Liu, and X. Tong, “Dual octree graph networks for learning adaptive volumetric shape representations,” *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–15, 2022.
- [35] B. Guillard, F. Stella, and P. Fua, “Meshudf: Fast and differentiable meshing of unsigned distance field networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 576–592.
- [36] S. Ren, J. Hou, X. Chen, Y. He, and W. Wang, “GeoUDF: Surface reconstruction from 3d point clouds via geometry-guided distance representation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 14 214–14 224.
- [37] L. Wang, W. Chen, X. Meng, B. Yang, J. Li, L. Gao *et al.*, “HSDF: Hybrid sign and distance field for modeling surfaces with arbitrary topologies,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2022, pp. 32 172–32 185.
- [38] Y.-T. Liu, L. Wang, J. Yang, W. Chen, X. Meng, B. Yang, and L. Gao, “NeUDF: Leaning neural unsigned distance fields with volume rendering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 237–247.
- [39] Y. Lu, L. Wan, N. Ding, Y. Wang, S. Shen, S. Cai, and L. Gao, “Unsigned orthogonal distance fields: An accurate neural implicit representation for diverse 3d shapes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2024, pp. 20 551–20 560.
- [40] M. Baorui, H. Zhizhong, L. Yu-Shen, and Z. Matthias, “Neural-Pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 7246–7257.
- [41] J. Zhou, B. Ma, S. Li, Y.-S. Liu, and Z. Han, “Learning a more continuous zero level set in unsigned distance fields through level set projection,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 3158–3169.
- [42] Q. Li, H. Feng, K. Shi, Y. Gao, Y. Fang, Y.-S. Liu, and Z. Han, “NeuralGF: Unsupervised point normal estimation by learning neural gradient function,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2023, pp. 66 006–66 019.
- [43] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, p. 163–169, 1987.
- [44] D. de Silva Edirumini, X. Lu, Z. Shao, G. Li, A. Robles-Kelly, and Y. He, “IterativePFN: True iterative point cloud filtering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 13 530–13 539.
- [45] P. Xiang, X. Wen, Y.-S. Liu, H. Zhang, Y. Fang, and Z. Han, “Retro-FPN: Retrospective feature pyramid network for point cloud semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 17 826–17 838.
- [46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
- [47] X. Liu, Z. Han, Y.-S. Liu, and M. Zwicker, “Point2Sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, p. 8.
- [48] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q.-X. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “Shapenet: An information-rich 3d model repository,” *ArXiv*, vol. abs/1512.03012, 2015.
- [49] J. Ye, Y. Chen, N. Wang, and X. Wang, “GIFS: Neural implicit function for general shape representation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 12 829–12 839.
- [50] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo, “Deep geometric prior for surface reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 10 122–10 131.
- [51] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, “Implicit geometric regularization for learning shapes,” in *Proc. Int. Conf. Mach. Learn.*, 2020.
- [52] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or, “Point2Mesh: a self-prior for deformable meshes,” *ACM Trans. Graph.*, vol. 39, no. 4, p. 12, 2020.
- [53] M. Kazhdan and H. Hoppe, “Screened poisson surface reconstruction,” *ACM Trans. Graph.*, vol. 32, no. 3, 2013.
- [54] S. Peng, C. Jiang, Y. Liao, M. Niemeyer, M. Pollefeys, and A. Geiger, “Shape as points: A differentiable poisson solver,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 13 032–13 044.
- [55] Q.-Y. Zhou and V. Koltun, “Dense scene reconstruction with points of interest,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–8, 2013.
- [56] B. Ma, Y.-S. Liu, and Z. Han, “Reconstructing surfaces for sparse point clouds with on-surface priors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.
- [57] L. Lin, Y. Liu, Y. Hu, X. Yan, K. Xie, and H. Huang, “Capturing, reconstructing, and simulating: The urbanscene3d dataset,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, p. 93–109.

Zheng Liu is currently an associate professor in China University of Geosciences (Wuhan). He received PhD from Central China Normal University in 2012. From 2013 to 2014, he held a post-doctoral position with School of Mathematical Sciences, University of Science and Technology of China. His research interests include geometry processing, computer graphics, 3D vision and deep learning.



Jianjun Zhang currently a M.S. candidate at China University of Geosciences (Wuhan). He received the B.S. degree from the Beijing Institute of Petrochemical Technology, in 2022. His research interests include geometry processing, 3D vision and deep learning.



Ming Zhang received his PhD degree in nature resource from Beijing Normal University in 2015. From 2015 to 2020, he worked as a senior research scientist with Aramco Beijing Research Center. In 2020, he joined the Innovation Center, Guangzhou Urban Planning and Design Survey Research Institute, mainly focused on urban computing and digital twin city.



Runze Ke received the BS degree from the Hubei University of Technology, Wuhan, in 2024. He is currently working toward an MS degree at the China University of Geosciences (Wuhan). His research interests include 3D vision and point cloud analysis.





Chengcheng Yu received the BS degree from the China University of Geosciences, Wuhan, in 2024. He is currently working toward an MS degree at the China University of Geosciences (Wuhan). His research interests include 3D vision and deep learning.



Ligang Liu is a professor at the University of Science and Technology of China. He received his PhD from Zhejiang University in 2001. He once worked at Microsoft Research Asia, Zhejiang University, and visited Harvard University. His research interests include computer graphics and geometry processing. He serves as the associated editor for journals including IEEE TVCG, IEEE CG&A, CAGD, C&G, The Visual Computer, etc. He served as the conference co-chair of GMP 2017 and the program co-chairs of conferences including Chinagraph 2020, SIAM GD 2019, GMP 2018, CAD/Graphics 2017, CVM 2016, SGP 2015, and SPM 2014. He serves as the steering committee member of GMP and the secretary of the Asiographics Association.