Trajectory Flow: Geometry-Constrained Surface Reconstruction via Unsigned Distance Fields

Chengcheng Yu, Zixu Dun, Zheng Liu, and Ying He, *Member, IEEE*

***Abstract*—Neural implicit methods reconstruct surfaces from point clouds by learning a continuous function, usually signed or unsigned distances. For unsigned distance field (UDF) reconstruction, many self-supervised methods typically estimate distance through local, gradient-guided displacement regression from query points to the surface. Because these corrections are local, they can become unstable near the zero level set and accumulate errors under sparse or non-uniform sampling. We address this limitation by reformulating UDF learning as a geometrically constrained trajectory-evolution problem, shifting from static local regression to global path-consistent modeling. Specifically, we construct a bidirectional linear flow that promotes shortest-path trajectories between surface samples and query points. Under this formulation, distance prediction is interpreted as continuous state evolution along deterministic trajectories, inherently preserving geometric consistency and mitigating local error accumulation. This perspective connects UDF estimation to neural ODEs, where geometry is recovered through continuous dynamics, with the learned transport following structured trajectories induced by an implicit geometric field. To address sparse and non-uniform sampling, we further introduce a flow-guided iterative densification strategy that progressively upgrades shortest-path modeling from discrete point-to-point approximations to accurate point-to-surface projections, enabling implicit geometric completion. Extensive experiments on synthetic benchmarks and real-world scans demonstrate that our method achieves state-of-the-art performance in both reconstruction accuracy and robustness.**

***Index Terms*—Surface reconstruction, point clouds, neural unsigned distance fields, straight flow, neural ODE.**

I. Introduction

With the rapid advancement of 3D acquisition technologies, such as LiDAR scanners and depth cameras, point clouds have become a fundamental representation in computer graphics, 3D vision, photogrammetry, autonomous driving, and SLAM. A central problem in geometry processing is to reconstruct continuous and structurally consistent surfaces from raw point samples. This problem underlies numerous downstream applications, including virtual and augmented reality [1], manufacturing [2], digital twins [3], [4], and robotics [5], [6]. Despite substantial progress, achieving high-fidelity and robust reconstruction remains challenging, especially for 3D models with complex geometry and topology.

Recent progress in deep learning has established neural implicit representations (NIRs) as a dominant paradigm for surface reconstruction [7], [8], [9], [10]. A large class of NIR-based reconstruction methods represents the target surface as the zero level set of either a signed distance function (SDF) or an unsigned distance field (UDF). SDFs encode both distance and an inside–outside sign, making them well suited for closed, oriented surfaces. In contrast, UDFs dispense with sign information and are therefore more appropriate for representing open surfaces, non-manifold structures, and internal geometries, where sign ambiguities are unavoidable. By modeling geometry as a continuous distance field, these formulations provide adaptive resolution and implicit topological regularization within a unified framework.

Existing NIR-based reconstruction methods can be broadly categorized into supervised [11], [12], [13], [14] and self-supervised approaches [15], [16], [17], [18]. Supervised methods achieve strong performance but require ground-truth annotations, which are expensive and often difficult to obtain. Their dependence on curated datasets may also limit generalization to unseen categories and complex real-world scenarios.

To alleviate these limitations, self-supervised methods with geometric regularization, such as IGR [15], NSH [19], and DiGS [20], directly fit neural implicit fields to point clouds. Benefiting from the spectral bias of deep networks, they tend to recover low-frequency structures first, promoting smooth surfaces. However, these fitting-based approaches remain sensitive to input quality and often rely on accurate normal estimation. In the presence of noise or unreliable normals, optimization can become unstable, leading to artifacts, fragmented geometry, and topological inconsistencies.

This challenge is particularly pronounced for UDF reconstruction, which aims at recovering accurate surface distances without relying on sign information. A representative line of work estimates distances by regressing point-wise displacements from query points toward the surface [16], [17], [21]. These approaches assume that following the predicted gradient direction of an implicit field approximates a shortest path to the underlying surface, allowing the resulting displacement to serve as a proxy for distance. Although effective, this

Received 1 April 2026; revised 17 June 2026; accepted 3 August 2026. Date of publication 6 August 2026; date of current version 4 September 2026. The work of Zheng Liu was supported in part by the Natural Science Foundation of Hubei Province under Grant 2026AFB495. The work of Ying He was supported in part by the Ministry of Education, Singapore, under its Academic Research Fund Grant RG17/26. Recommended for acceptance by Y.-S. Liu.*(Corresponding author: Zheng Liu.)*

Chengcheng Yu, Zixu Dun, and Zheng Liu are with the School of Computer Science, China University of Geosciences (Wuhan), Wuhan 430079, China (e-mail: yccszbd@cug.edu.cn; dzx203725@gmail.com; liu.zheng.jojo@gmail.com).

Ying He is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: yhe@ntu.edu.sg).

Our source code is available at https://github.com/yccszbd/Trajectory-Flow.

Digital Object Identifier 10.1109/TVCG.2026.3721433

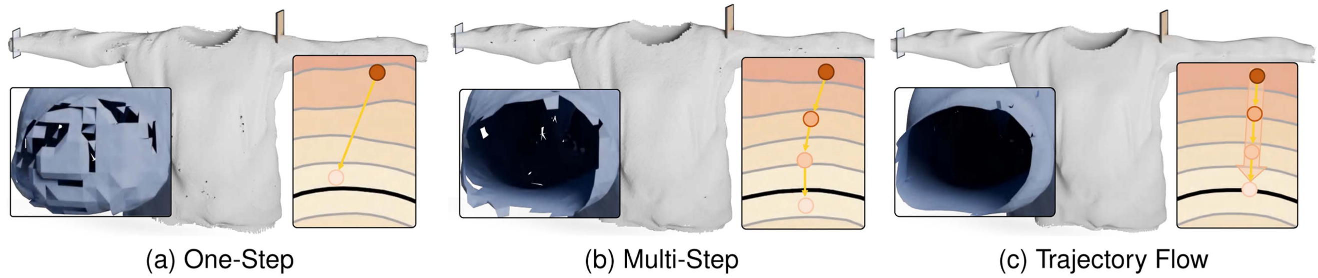

Fig. 1. Motivation of the proposed geometry-constrained trajectory flow for unsigned distance field (UDF) reconstruction. (a) One-step and (b) multi-step displacement regression methods rely on locally estimated gradients, which can be unstable near the surface and lead to inaccurate UDF estimation and reconstruction artifacts. (c) In contrast, the proposed trajectory flow models distance as continuous evolution along deterministic shortest-path trajectories, promoting trajectory consistency and enabling more accurate and robust surface reconstruction.

formulation is inherently local. Methods such as Neural-Pull [16] and CAP-UDF [17] rely on locally estimated gradients, which are sensitive to noise and can oscillate near the zero level set, leading to unstable displacement predictions. LevelsetUDF [21] improves local inconsistency through gradient alignment, and Multipull [22] extends correction through multi-step updates. Nevertheless, these methods still construct trajectories through incremental local corrections without globally consistent constraints, which can result in inconsistent paths and accumulated errors in distance estimation.

These observations suggest that implicit distance should not be modeled solely via local corrections, but instead through a globally transport process. Motivated by this insight, we reformulate UDF learning as a trajectory evolution problem under geometric constraints. Rather than predicting instantaneous displacements, we model the evolution of query points along structured paths toward the surface. Fig. 1 shows that local displacement regression suffers from gradient instability near the surface, whereas the proposed trajectory flow mitigates this issue by enforcing consistent shortest-path transport.

Specifically, we construct linear reference trajectories between surface samples and query points and learn a bidirectional, time-conditioned flow to model the corresponding transport. During inference, the learned flow transports query points backward toward near-surface states, from which the UDF is estimated. This trajectory-based formulation replaces local gradient correction with global transport modeling, improving optimization stability, reducing error accumulation, and yielding more robust implicit distance estimation.

However, reliable trajectory construction becomes difficult under sparse and non-uniform sampling, where nearest-neighbor correspondences provide only coarse approximations of the underlying surface. To address this issue, we further introduce a flow-guided iterative densification strategy that jointly refines the flow field and the sampling distribution. By progressively enriching surface samples and improving the correspondence between query points and the latent geometry, the proposed strategy upgrades coarse point-to-point supervision into more accurate point-to-surface approximations, improving robustness under degenerate sampling conditions. We validate the proposed method on synthetic benchmarks and real-world scans, achieving state-of-the-art performance in reconstruction accuracy and robustness.

Our main contributions are summarized as follows:

- We reformulate self-supervised UDF learning as a geometry-constrained trajectory evolution problem, replacing local displacement regression with globally structured transport.
- We introduce a bidirectional, time-conditioned flow formulation that models linear reference trajectories between surface samples and query points, enabling self-supervised learning of reverse transport toward the underlying surface.
- We develop a flow-guided iterative densification strategy that progressively refines coarse point-wise supervision into accurate point-to-surface approximations, enhancing robustness to sparse and non-uniform sampling.

II. Related Work

Over the past decades, a large body of traditional surface reconstruction methods has been developed [23], [24], [25], [26], [27], [28], [29], [30]. Despite their effectiveness, these approaches often require careful parameter tuning and may lack robustness across diverse datasets. With the success of deep neural networks, neural implicit representations have emerged as a powerful alternative for surface reconstruction. In light of this paradigm shift, we focus our review on neural implicit methods and related flow-based approaches that are most relevant to our work, rather than exhaustively covering traditional techniques.

A. Implicit Learning From 3D Supervision

Surface reconstruction from point clouds using neural implicit representations has been extensively studied in recent years. Early learning-based approaches rely on supervised training with ground-truth signed distance fields or occupancy labels. Representative methods [11], [12], [31], [32], [33], [34], [35], [36], [37] encode shapes into global latent codes and decode them into continuous implicit fields. While effective within restricted shape families, such global representations are sensitive to domain shift and often generalize poorly to geometries outside the training distribution. To improve scalability and cross-shape generalization, subsequent works [14], [38], [39], [40], [41] adopt local representations by decomposing point clouds into grids or patches. By reducing global reconstruction into localized fitting problems, these methods better capture fine-scale geometry and exhibit improved robustness to structural variations. More recently, several methods [13], [42], [43] explore vector field regression from point clouds, representing shapes as explicit vector fields by predicting one-step displacements from query points to the surface. Despite their empirical success, these supervised approaches rely on high-quality ground-truth annotations, which are costly to obtain and limit applicability in real-world scenarios. Furthermore, their dependence on curated datasets and complex training pipelines often restricts generalization beyond the training domain.

B. Implicit Learning From Raw Point Clouds

Regularization-based Implicit Reconstruction: A prevailing paradigm directly fits neural implicit fields from raw point clouds under geometric regularization [18], [44], [45]. Typical regularizers include gradient orientation alignment [46], [47], unit-norm constraints [15], [48], and smoothness or structural priors [19], [20], [49], [50]. Specifically, IGR [15] constrains gradient norms with normal supervision, while SIREN [46] improves high-frequency representation via periodic activations and tailored initialization. DiGS [20] introduces Laplacian regularization to stabilize training without normals, and Neural-Hessian [19] further suppresses spurious variations through second-order gradient constraints. Most of the aforementioned methods model implicit fields as signed distance functions, which limits their ability to represent open surfaces or complex geometries. DUDF [51] extends the framework to unsigned distance fields and alleviates gradient instability near the surface via hyperbolic scaling, while SSDF [52] introduces Monge–Ampère regularization to enhance geometric fidelity. Overall, this line of work relies on neural fitting capacity under various regularization, and remains sensitive to input quality, sampling irregularity, and network initialization.

Displacement-based Implicit Reconstruction: Another line of self-supervised research estimates implicit distance fields by learning point-wise displacements from query points toward clean surface samples. Neural-Pull [16] and its variants [53], [54], [55] learn signed distance fields (SDFs) by shifting query points toward the underlying surface along predicted gradient directions. To improve stability and reconstruction quality, subsequent works introduce additional geometric constraints. Zhou et al. [17] proposed a progressive strategy for unsigned distance field (UDF) learning with consistency regularization, achieving strong performance on open surfaces and complex geometries. LevelsetUDF [21] alleviates gradient oscillation near surfaces via gradient alignment between surface and query points. Building upon this idea, MSMS [56] and Multipull [22] extend alignment constraints along multi-step displacement trajectories, while I-Filtering [57] introduces level-set filtering to enhance local correction consistency within neighboring regions.

Despite improved stability via displacement supervision and gradient alignment, existing methods fundamentally rely on incremental local updates. Their trajectories are constructed through successive gradient corrections, without enforcing global geometric consistency or a unified evolution process. In contrast, we reformulate implicit distance estimation as a continuous evolution of point distributions under linear geometric constraints. By embedding shortest-path structure into a deterministic forward process, our method produces globally consistent trajectories that naturally satisfy the geometric properties of unsigned distance fields.

C. Flow-Based Methods for Point Cloud Processing

Flow-based models learn continuous-time transformations that transport a simple prior distribution (e.g., Gaussian noise) to a target data distribution. In contrast to diffusion models, which rely on stochastic forward and reverse processes [58], [59], [60], flow-based approaches define deterministic transport dynamics governed by ordinary differential equations, enabling exact likelihood evaluation and efficient sampling [61]. Flow matching methods, including Rectified Flow [62] and Flow Matching [63], parameterize velocity fields along probability paths between source and target distributions. By supervising the transport dynamics, these methods improve training stability and sampling efficiency. More recently, Zhu et al. [61] incorporated trajectory optimization strategies into differential equation–based frameworks, emphasizing the critical role of transport path design in improving model performance.

Flow-based formulations have recently been extended to 3D point cloud processing. Wu et al. [64] rectified transport trajectories and distilled them into a single-step generator for efficient point cloud synthesis. Edirimuni et al. [65] modeled point cloud denoising as linear flow transport by learning constant velocity fields that map noisy inputs directly to clean surfaces. Sun et al. [66] unified point cloud registration and multi-part assembly under a conditional generative framework by learning continuous point-wise velocity fields to recover target configurations.

Unlike generation- or restoration-oriented approaches, our method leverages the continuity and reversibility of flow dynamics to model neural implicit representations from raw point clouds. Instead of using flows for generative transport, we employ deterministic trajectories to encode shortest-path structures between point distributions, enabling stable and accurate implicit surface reconstruction.

III. Method

A. Straight-Line Transport for UDF Estimation

We formulate UDF estimation as a velocity field learning problem under a straight-line transport paradigm. In the continuous setting, given a query point $q \in \mathbb{R}^3$ and a surface $\mathcal{S} \subset \mathbb{R}^3$, the UDF is defined as

$$f(q) = \min_{p \in \mathcal{S}} \|q - p\|.$$

We reformulate this problem as a continuous transport process. Specifically, the evolution of a query point over time $t \in [0, 1]$ is modeled as a trajectory $x_t \in \mathbb{R}^3$ governed by a velocity field v:

$$\frac{dx_t}{dt} = v(x_t, t).$$

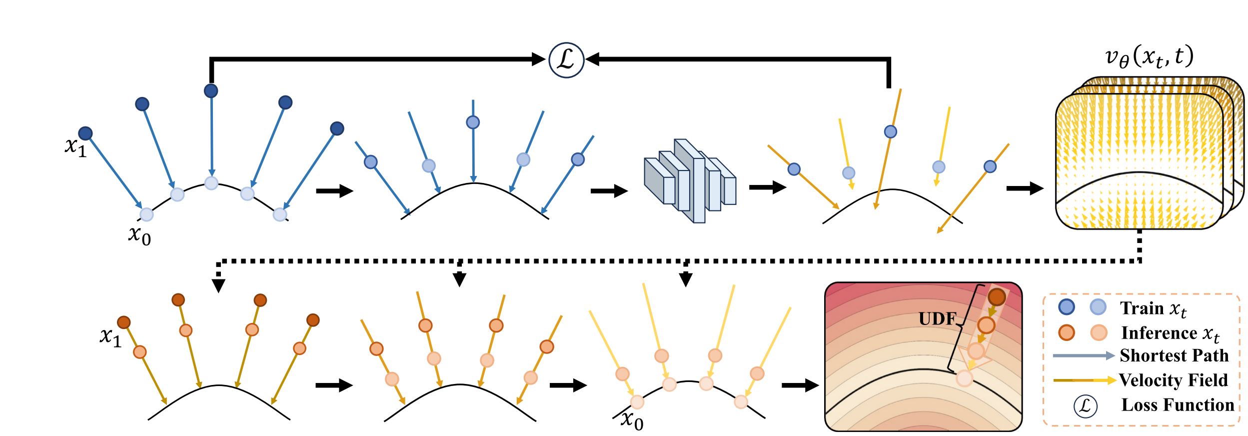

Fig. 2. Pipeline of the proposed framework. During training, shortest trajectories are constructed from query points to their nearest surface samples, with intermediate states x_t obtained via linear interpolation. The network learns a time-conditioned velocity field $v_\theta(x_t, t)$ along these trajectories. During inference, query points are transported toward the underlying surface via reverse integration of the learned velocity field, and the unsigned distance is computed as the displacement between the initial and converged positions.

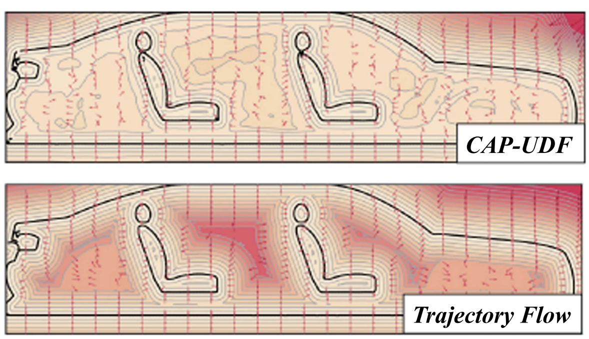

Fig. 3. UDFs and gradient fields on a 2D slice of a model from the ShapeNet Cars category, with CAP-UDF shown on the top and Trajectory Flow on the bottom. Benefiting from trajectory consistency induced by the ODE-based flow, our method produces more consistent gradient directions and improves the stability of surface reconstruction.

The goal is to learn a velocity field that transports the query point $x_1 := q$ to its closest surface point $x_0 := \mathrm{NN}(q; \mathcal{S})$, where $\mathrm{NN}(\cdot)$ is the nearest neighbor operator. To this end, we consider the straight-line trajectory connecting x_0 and x_1:

$$x_t = tx_1 + (1-t)x_0, \quad t \in [0, 1]. \tag{1}$$

This construction is geometrically optimal, as the straight-line segment corresponds to the shortest path in Euclidean space. Consequently, the UDF is directly given by the displacement magnitude:

$$f(q) = \|x_1 - x_0\|. \tag{2}$$

Furthermore, the induced velocity along this trajectory is constant:

$$v(x_t, t) = \frac{dx_t}{dt} = x_1 - x_0.$$

Beyond a transport-based formulation for UDF estimation, the ODE perspective introduces an implicit trajectory consistency constraint. For a sufficiently regular (e.g., locally Lipschitz) velocity field, the induced dynamics admit a unique solution, implying a non-crossing property of trajectories that encourages coherent evolution of query points. In geometrically complex regions where locally constructed point-to-point trajectories may become ambiguous or inconsistent, the learned flow reorganizes them into a more coherent transport structure while preserving the overall point-to-surface mapping. This continuous formulation imposes a mild smoothness prior on the transport process, reducing sensitivity to high-frequency local perturbations and alleviating reliance on unstable UDF gradients. As illustrated in Fig. 3, this leads to more consistent gradient behavior and improves the stability of surface reconstruction.

B. Flow Learning and Reverse Integration

Based on the continuous transport formulation, we present a discrete implementation on point clouds with corresponding learning and inference procedures.

In practice, the input point cloud $P \subset \mathbb{R}^3$ is treated as a discrete sampling of the underlying surface $\mathcal{S}$. For a query point q, we define the terminal state as $x_1 := q$, and approximate the source state using its nearest neighbor in P as $x_0 := \mathrm{NN}(q; P)$.

To enable numerical computation, we discretize the continuous transport process using an explicit Euler scheme. Let $\mathcal{T} = \{t_k\}_{k=0}^{K}$ denote a sequence of time steps with $t_0 = 0$ and $t_K = 1$. To allocate higher resolution near the surface, we adopt a non-uniform time schedule:

$$\Delta t_k = \frac{2k}{K(K+1)}, \quad t_k = \frac{k(k+1)}{K(K+1)}, \quad k = 1, \ldots, K.$$

The resulting discrete trajectory is given by the linear transport path defined in (1):

$$x_{t_k} = t_k x_1 + (1 - t_k)x_0, \quad k = 1, \ldots, K.$$

Flow Learning: As established in Section III-A, the ideal velocity remains constant along the trajectory. We therefore supervise the parameterized velocity field v_θ using the reference displacement $(x_1 - x_0)$, leading to the following objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{t_k \sim \mathcal{T}, q \sim Q}\left[\|v_\theta(x_{t_k}, t_k) - (x_1 - x_0)\|_2^2\right]. \tag{3}$$

The training procedure is illustrated in the top panel of Fig. 2.

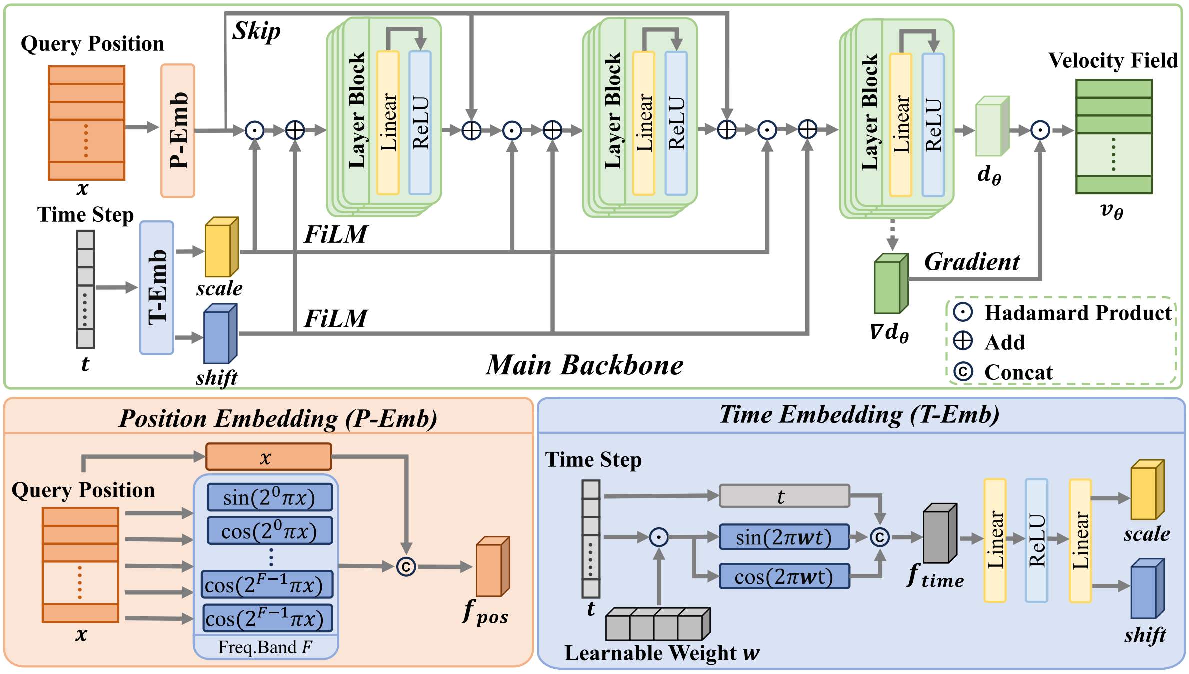

Fig. 4. Architecture of the proposed velocity field network. The query position x is encoded using position embedding, while the timestep t provides time embedding. The backbone integrates spatial and temporal information via FiLM modulation and positional skip connections. The network predicts a scalar field $d_\theta(x,t)$, whose gradient defines the velocity field $v_\theta(x,t)$.

Reverse Integration: At inference time, we recover the surface projection by integrating the learned velocity field backward from the query point:

$$x_{t_{k-1}} = x_{t_k} - \Delta t_k\, v_\theta(x_{t_k}, t_k), \quad k = K, \ldots, 1.$$

This yields the estimated source state:

$$x_0 = x_1 - \sum_{k=1}^{K} \Delta t_k\, v_\theta(x_{t_k}, t_k). \tag{4}$$

According to (2), the unsigned distance is given by the displacement magnitude induced by the learned transport:

$$f(q) = \|x_1 - x_0\|. \tag{5}$$

The reverse integration and UDF estimation process is illustrated in the bottom panel of Fig. 2.

Overall, the proposed framework establishes a unified pipeline that connects trajectory construction, velocity supervision, and reverse integration, ensuring consistency between training and inference while enabling accurate and numerically stable UDF estimation on point clouds.

C. Network Architecture

As shown in Fig. 4, the proposed network takes a query position x and timestep t as input and predicts a velocity field $v_\theta(x,t)$. The backbone is a multilayer perceptron (MLP) that integrates spatial and temporal information via position embedding, time embedding, skip connections, and Feature-wise Linear Modulation (FiLM) [67].

The input position x is mapped to a high-dimensional representation using sinusoidal position embedding with frequency bandwidth F:

$$f_{\text{pos}}(x) = \Big[x, \sin(2^0\pi x), \cos(2^0\pi x), \ldots, \sin(2^{F-1}\pi x), \cos(2^{F-1}\pi x)\Big], \tag{6}$$

which enables modeling of high-frequency spatial variations [68]. The encoded feature in (6) is used to initialize the network input:

$$h_0 = f_{\text{pos}}(x).$$

The backbone MLP consists of L layers, where the feature is updated as:

$$h_{l+1} = \sigma(W_l h_l + b_l), \quad l = 0, \ldots, L-1,$$

with $\sigma(\cdot)$ denoting a non-linear activation function. To preserve spatial information across layers, skip connections are introduced as

$$\mathcal{S}_l(h_l) = \begin{cases} \text{Concat}(h_l, h_0), & l \in L_s, \\ h_l, & \text{otherwise}, \end{cases}$$

where Concat$(\cdot)$ denotes feature concatenation and L_s denotes the set of layers where skip connections are applied.

The timestep t is encoded using a sinusoidal embedding with a learnable frequency vector $\mathbf{w} \in \mathbb{R}^D$:

$$f_{\text{time}}(t) = [t, \sin(2\pi\mathbf{w}t), \cos(2\pi\mathbf{w}t)]\,. \tag{7}$$

The time embedding in (7) is mapped into FiLM parameters:

$$(scale, shift) = \text{MLP}\big(f_{\text{time}}(t)\big),$$

which depend only on t and are shared across layers. Temporal modulation is applied through the FiLM operator:

$$\mathcal{T}_l(z_l) = \begin{cases} z_l \odot scale + shift, & l \in L_t, \\ z_l, & \text{otherwise}, \end{cases}$$

where $\odot$ denotes element-wise multiplication and L_t denotes the set of layers where FiLM is applied.

Combining spatial skip connections and temporal modulation, the feature transformation at layer l is expressed as

$$h_{l+1} = \sigma\big(W_l \mathcal{T}_l\big(\mathcal{S}_l(h_l)\big) + b_l\big)\,, \quad l = 0, \ldots, L-1.$$

The final feature is mapped to a scalar field:

$$d_\theta(x, t) = \psi(h_L),$$

where $\psi(\cdot)$ denotes the output activation function. The velocity field is then defined as:

$$v_\theta(x, t) = d_\theta(x, t) \nabla_x d_\theta(x, t), \tag{8}$$

where $\nabla_x d_\theta(x, t)$ is computed via differentiation. This construction induces a conservative vector field, leading to stable and geometrically consistent trajectory evolution.

Lemma 1: For any fixed t, the velocity field $v_\theta(x, t)$ defined in (8) is curl-free with respect to x.

Proof: Since $v_\theta(x, t) = \nabla_x(\frac{1}{2} d_\theta(x, t)^2)$, it is a gradient field, implying $\nabla_x \times v_\theta(x, t) = 0$. □

Thus, the velocity field can be interpreted as a gradient flow induced by the energy function $E(x, t) = \frac{1}{2} d_\theta(x, t)^2$. The dynamics $\frac{dx}{dt} = v_\theta(x, t)$ correspond to gradient descent on $E(x, t)$ with respect to x, establishing a connection to energy-based modeling where $d_\theta(x, t)$ defines an implicit energy landscape. Consequently, the induced trajectories follow stable descent directions toward the underlying surface, providing a principled explanation for the improved reconstruction quality.

D. Iterative Densification for Trajectories

We propose an iterative densification process that transforms discrete point-based supervision into continuous surface-level constraints via learned transport. The input point cloud provides a discrete sampling of the underlying surface $\mathcal{S}$, where shortest paths constructed from point pairs may deviate from the true surface under sparse and non-uniform conditions. Under the linear ODE formulation, flow trajectories exhibit a non-crossing property [62], while neural networks exhibit a low-frequency bias. These properties encourage smooth transport along the underlying manifold, enabling implicit geometric completion.

Based on this observation, shortest-path supervision is formulated as an iterative process. At each iteration, the velocity field is fitted from the current point cloud and used to reversely transport query points toward the surface, progressively densifying the geometry. This process transitions supervision from discrete point constraints to continuous surface-level constraints, improving shortest-path estimation.

Let the point cloud at iteration n be denoted as P^n. The velocity field is fitted by the framework introduced in Section III-B as

$$v_\theta^n \leftarrow \text{Fitting}(P^n).$$

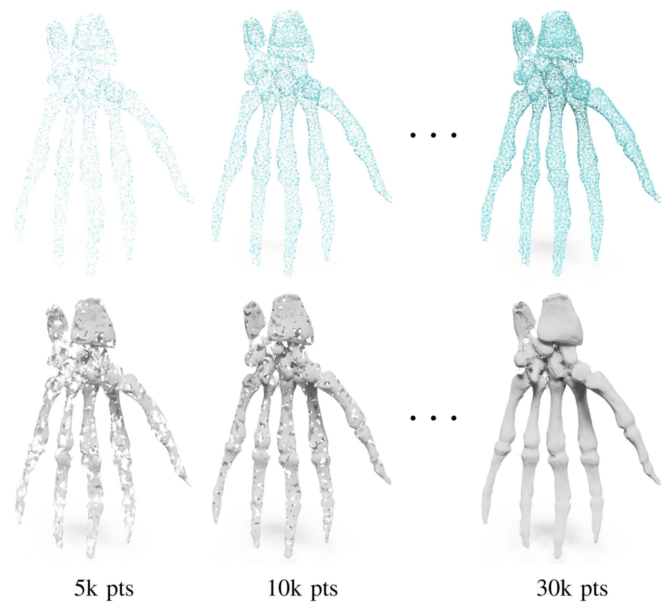

Fig. 5. Iterative densification process. Top: input point cloud P^0 and densified point cloud P^1 and P^N across iterations. Bottom: reconstructed surfaces from UDFs induced by the corresponding velocity fields v_θ^0, v_θ^1, and v_θ^N.

Query points Q^n are sampled in the neighborhood of P^n and reversely transported using v_θ^n via (4):

$$Q^n \xrightarrow{v_\theta^n} \hat{Q}^n,$$

yielding candidate surface points $\hat{Q}^n$ for densifying P^n. To mitigate instability caused by projection bias in early stages, only points satisfying the UDF constraint are retained:

$$P^{n+1} = P^n \cup \left\{ \hat{q} \in \hat{Q}^n \mid f^n(\hat{q}) < \epsilon \right\},$$

where $f^n(\cdot)$ denotes the UDF induced by v_θ^n in (5) and ϵ is a threshold. The densification step is written as

$$P^{n+1} \leftarrow \text{Reflow}(P^n, v_\theta^n).$$

The process alternates between velocity fitting and point cloud densification:

$$P^0 \xrightarrow{\text{Fitting}} v_\theta^0 \xrightarrow{\text{Reflow}} P^1 \xrightarrow{\text{Fitting}} v_\theta^1 \cdots \xrightarrow{\text{Reflow}} P^N \xrightarrow{\text{Fitting}} v_\theta^N, \tag{9}$$

where N denotes the number of iterations. The full process is illustrated in Fig. 5. Due to the sparsity of the initial point cloud, the reconstructed surfaces exhibit severe fragmentation. As iterative densification proceeds, the point cloud becomes progressively enriched, leading to improved surface reconstruction quality.

As a result, the densification process in (9) shifts supervision from discrete samples to surface-level constraints, encouraging meaningful shortest-path trajectories. By alleviating sparsity and sampling irregularity, it improves reconstruction completeness, smoothness, and topological consistency.

TABLE I
QUANTITATIVE COMPARISON ON THE SHAPENET CARS CATEGORY. † INDICATES SUPERVISED METHODS, WHILE THE OTHERS ARE SELF-SUPERVISED. BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY.

Method	L1-CD ↓	$FS^{0.01}$ ↑	$FS^{0.005}$ ↑	NC ↑
NVF†	3.910	94.91%	86.03%	88.92%
GeoUDF†	2.810	99.58%	90.95%	88.79%
DUDF	3.972	96.21%	79.25%	88.31%
DEUDF	<u>2.803</u>	99.52%	90.57%	<u>91.31%</u>
CAP-UDF	3.013	<u>99.73%</u>	84.36%	87.59%
LevelsetUDF	2.817	99.51%	<u>91.14%</u>	91.25%
Ours	**2.711**	**99.89%**	**92.08%**	**92.21%**

TABLE II
QUANTITATIVE COMPARISON ON THE MGN DATASET. † INDICATES SUPERVISED METHODS, WHILE THE OTHERS ARE SELF-SUPERVISED. BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY.

Method	L1-CD ↓	$FS^{0.01}$ ↑	$FS^{0.005}$ ↑	NC ↑
NVF†	2.003	99.95%	97.07%	98.81%
GeoUDF†	2.010	99.96%	96.42%	98.48%
DUDF	3.084	99.90%	87.51%	98.73%
DEUDF	1.811	99.94%	<u>98.44%</u>	98.87%
CAP-UDF	1.824	99.88%	98.34%	<u>98.98%</u>
LevelsetUDF	<u>1.778</u>	<u>99.97%</u>	98.43%	98.85%
Ours	**1.772**	**99.98%**	**98.46%**	**99.40%**

IV. EXPERIMENTS

A. Experimental Setup

Implementation Details: All experiments are conducted on a workstation equipped with an NVIDIA RTX 4090 GPU (24 GB) and an Intel i7-12700 K CPU. The network is implemented as a multilayer perceptron (MLP) with depth $L = 12$ and width $H = 256$, and operates over $K = 5$ discrete time steps. Position embedding uses a frequency bandwidth of $F = 5$, with skip connections applied at layers $L_s = [4, 8]$. The time variable is embedded using a learnable weight $D = 16$, and injected via FiLM modulation at layers $L_t = [0, 4, 8]$. ReLU is adopted as the activation function for both the hidden layers $\sigma(\cdot)$ and the output layer $\psi(\cdot)$. The model is trained for 60 k iterations using the Adam optimizer, with an initial learning rate of 1×10^{-3}, followed by cosine decay with a warm-up phase of 1 k iterations.

In the standard experiments (Sections IV-B and IV-C), reconstruction is performed directly from the input point clouds without iterative densification. A total of 30 k input points are used, uniformly sampled from the ground-truth surface for synthetic datasets. For scanned datasets, point clouds are directly sampled to 30 k points. Densification is enabled only for sparse-input experiments, including the densification study (Section IV-F) and robustness evaluation (Section IV-G). The target point count is set to 30 k, and $2k$ points are inserted per iteration. Unless otherwise specified, the UDF threshold is fixed at $\epsilon = 0.005$.

Surface Extraction: For UDF-based methods, surfaces are extracted using the CAP-UDF zero-level set extraction [17] at a resolution of 256^3. The same configuration is used for all UDF baselines for fair comparison. For SDF-based methods, Marching Cubes is applied at the same resolution.

Evaluation Metrics: Following prior work [17], [52], we evaluate reconstruction quality using L1-Chamfer Distance (L1-CD), Normal Consistency (NC), and F-score (FS). For each tested mesh, 100 k points are sampled for evaluation. L1-CD measures the mean bidirectional L1 distance between predicted and ground-truth point clouds (scaled by 10^3), while NC computes the mean cosine similarity between predicted and ground-truth normals (reported in %). FS is defined as the harmonic mean of precision and recall, and reported at thresholds $FS^{0.01}$ and $FS^{0.005}$. For real-world scans where only point cloud ground truth is available, NC is not reported.

B. Non-Watertight Surface Reconstruction

Our comparisons on non-watertight surface reconstruction are conducted using both synthetic and real-world datasets. We compare our method against two categories of state-of-the-art approaches: supervised and self-supervised UDF reconstruction methods. The supervised baselines include NVF [13] and GeoUDF [69], which rely on ground-truth supervision to train generalizable models. The self-supervised baselines consist of DUDF [51], DEUDF [47], CAP-UDF [17], and LevelsetUDF [21], where neural implicit fields are optimized independently for each test shape.

Evaluation on ShapeNet: ShapeNet [70] is a synthetic dataset containing diverse CAD models. Following the evaluation protocol of NDF [71] and CAP-UDF [17], we evaluate on Cars category, which contains a high proportion of shapes with complex internal and multi-layered structures. We randomly select 10 shapes from this category and sample 30 k surface points from each as input. Quantitative results are reported in Table I, where our method achieves the best performance across all evaluated metrics. Qualitative comparisons are shown in Fig. 6. GeoUDF produces surfaces with noticeable holes, while DUDF generates overly rough reconstructions (Fig. 6(b) and (c)). CAP-UDF and LevelsetUDF introduce artifacts on the outer or internal layers (Fig. 6(e) and (f)), degrading the visual quality of the reconstructed surfaces. In contrast, our method reconstructs nested multi-layer surfaces more faithfully, preserves distinct internal structures, and avoids such artifacts, particularly in piecewise-smooth regions.

Evaluation on MGN: MGN [72] is a synthetic dataset composed of open garment surfaces. We randomly select 20 shapes and sample 30 k points from each shape as input. Quantitative results are reported in Table II, where our method achieves the best performance across all metrics. Qualitative comparisons are presented in Fig. 7. NVF and GeoUDF fail to recover fine geometric details, while DUDF and CAP-UDF introduce topological artifacts, particularly near open boundaries such as garment cuffs (see the zoom-in views in Figs. 7(a), (b), (c), and (e)). In contrast, our method preserves the topological correctness of open surfaces while maintaining clean boundary contours, highlighting its effectiveness in reconstructing thin surfaces with complex boundaries, such as clothing.

Evaluation on 3D Scene: To further evaluate performance on real-world data, we conduct experiments on the 3D Scene dataset [73], which contains RGB-D videos and reconstructed

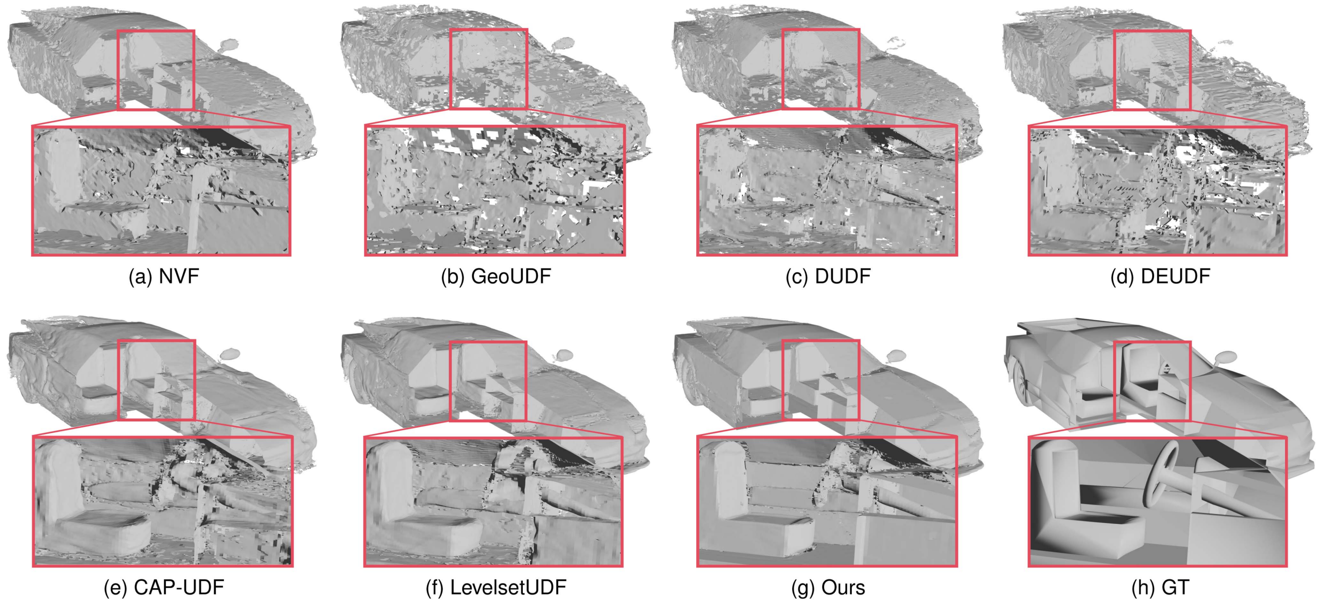

Fig. 6. Qualitative comparison on the ShapeNet Cars category with multi-layered structures. Zoomed-in views show that our method accurately reconstructs nested surfaces and preserves distinct internal layers without visual artifacts.

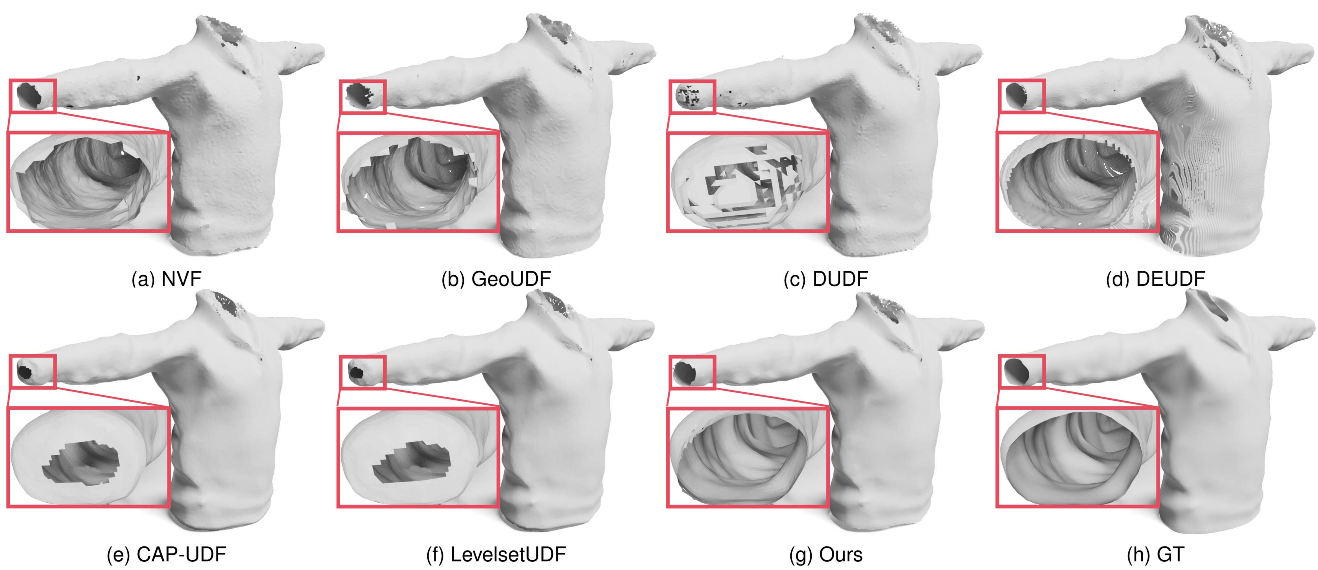

Fig. 7. Qualitative comparison on the MGN dataset for thin surfaces with complex boundaries. The zoomed-in views show that our method preserves the topological correctness of open surfaces while maintaining clean boundary contours.

surfaces from diverse environments (e.g., Burgher, Lounge, and Copyroom). Following the standard protocol, all methods are evaluated at sampling densities of 500 and 1,000 points/m^2. Quantitative results are reported in Table III. Our method ranks first at 500 points/m^2, achieving the best performance in L1-CD, FS$^{0.005}$, and NC, and obtains the second-best result in FS$^{0.01}$ at 1,000 points/m^2. Qualitative comparisons in Fig. 8 show that most competing methods introduce noticeable artifacts in real-world scenes, whereas our method better preserves geometric structures and smooth regions, demonstrating improved robustness.

C. Watertight Surface Reconstruction

To evaluate our approach on watertight surfaces, we compare against representative SDF- and UDF-based reconstruction methods on four widely used datasets. SDF-based baselines include IGR [15], SIREN [46], NSH [19], StEik [74], and PG-SDF [49], while UDF-based methods include CAP-UDF [17], DUDF [51], and LevelsetUDF [21].

Evaluation on Stanford 3D Scanning Repository and ABC: We further evaluate our method on two watertight benchmarks: the Stanford 3D Scanning Repository [32] and the ABC

TABLE III
QUANTITATIVE COMPARISON ON THE 3D SCENE DATASET AT TWO SAMPLING DENSITIES. † INDICATES SUPERVISED METHODS, WHILE THE OTHERS ARE SELF-SUPERVISED. BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY.

	500/m^2				1000/m^2			
Method	L1-CD ↓	$\text{FS}^{0.01}$ ↑	$\text{FS}^{0.005}$ ↑	NC ↑	L1-CD ↓	$\text{FS}^{0.01}$ ↑	$\text{FS}^{0.005}$ ↑	NC ↑
NVF†	3.214	98.10%	85.14%	91.55%	3.357	97.98%	80.12%	93.11%
GeoUDF†	2.503	98.36%	91.61%	92.33%	3.178	95.44%	81.04%	93.03%
DUDF	3.535	99.45%	84.74%	93.12%	3.571	99.19%	82.59%	93.06%
DEUDF	2.372	99.57%	93.46%	93.65%	2.511	99.44%	93.70%	93.51%
CAP-UDF	2.068	99.36%	96.13%	93.27%	2.193	99.03%	95.10%	93.01%
LevelsetUDF	1.944	99.96%	97.26%	93.80%	2.054	**99.95%**	96.46%	93.65%
Ours	**1.909**	**99.97%**	**97.28%**	**94.31%**	**2.012**	99.93%	**96.48%**	**94.29%**

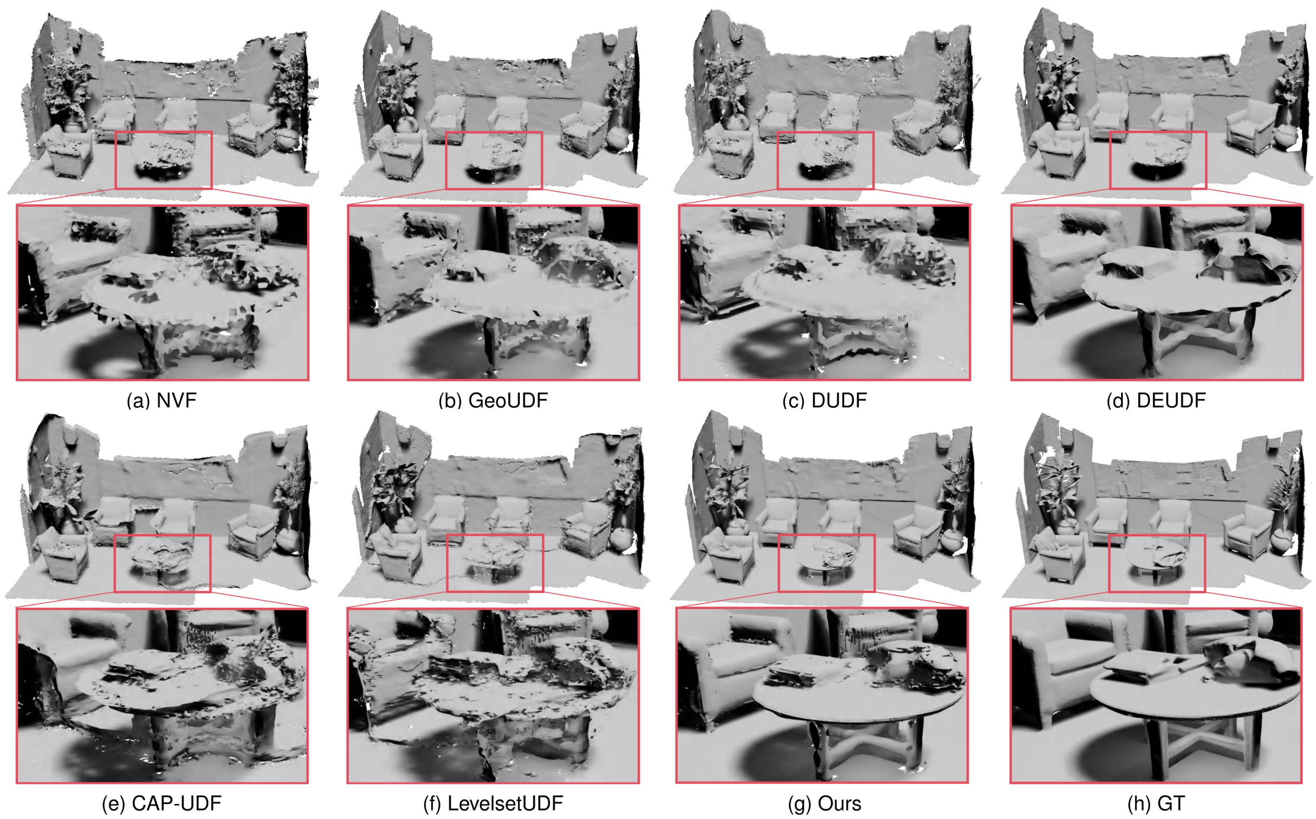

Fig. 8. Qualitative comparison on the 3D Scene dataset under complex real-world scenarios. The zoomed-in views show that our method suppresses visible artifacts, preserves geometric structures, and reconstructs smooth surfaces.

dataset [75]. The Stanford dataset contains 22 curated watertight models, while the ABC dataset comprises a large collection of CAD shapes, from which we select 100 representative models. For both datasets, 30 k points are uniformly sampled from each shape as input. Quantitative results are reported in Table IV. On the Stanford dataset, our method achieves the best performance in L1-CD, $\text{FS}^{0.01}$, and $\text{FS}^{0.005}$, and ranks second in NC. On the ABC dataset, it achieves the best results across all metrics. Qualitative comparisons (Figs. 9 and 10) show that competing methods tend to oversmooth surfaces or fail to preserve fine structures. On the Stanford dataset, our method better retains high-frequency details such as embossed text and brick textures (Fig. 9(g)). On the ABC dataset, prior methods exhibit structural breakage or fail to reconstruct thin structures and small holes (Fig. 10(d)–(f)), whereas our method preserves correct topology and more accurately recovers sharp features and smooth regions.

Evaluation on SRB and Real Scans: The Surface Reconstruction Benchmark (SRB) [76] evaluates reconstruction methods under realistic scanning conditions, including missing data and measurement noise. To further assess robustness to real-world interference, we also include the Real Scans dataset [77], which contains scan misalignments and sensor noise from real acquisition. Quantitative results are reported in Table V. Our method achieves the best performance across all metrics on both datasets. Qualitative comparisons are shown in Fig. 11. LevelsetUDF struggles to recover correct topology, while CAP-UDF is sensitive to noise and introduces noticeable surface artifacts.

TABLE IV
QUANTITATIVE COMPARISON ON STANFORD 3D SCANNING REPOSITORY AND THE ABC DATASET. BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY.

	Stanford 3D Scanning Repository				ABC			
Method	L1-CD ↓	$\text{FS}^{0.01}$ ↑	$\text{FS}^{0.005}$ ↑	NC ↑	L1-CD ↓	$\text{FS}^{0.01}$ ↑	$\text{FS}^{0.005}$ ↑	NC ↑
IGR	6.422	88.22%	77.82%	96.82%	10.475	70.24%	51.04%	90.41%
SIREN	4.601	93.46%	84.85%	84.85%	7.102	80.25%	68.98%	92.39%
StEik	4.326	95.40%	90.87%	96.92%	5.974	89.47%	70.74%	93.11%
NSH	3.866	97.97%	94.24%	**97.92%**	2.711	98.81%	<u>93.06%</u>	<u>98.12%</u>
DUDF	2.779	99.82%	93.27%	96.46%	4.898	90.96%	72.73%	92.91%
PG-SDF	2.459	99.75%	95.51%	96.53%	3.314	97.04%	88.15%	96.53%
DEUDF	2.319	99.78%	96.43%	97.53%	2.925	97.58%	90.75%	96.87%
CAP-UDF	2.320	99.95%	96.33%	97.28%	2.742	<u>99.73%</u>	91.04%	97.68%
LevelsetUDF	<u>2.263</u>	<u>99.97%</u>	<u>96.50%</u>	97.03%	<u>2.706</u>	<u>99.73%</u>	91.40%	97.75%
Ours	**2.241**	**99.98%**	**96.54%**	<u>97.78%</u>	**2.542**	**99.92%**	**93.64%**	**98.37%**

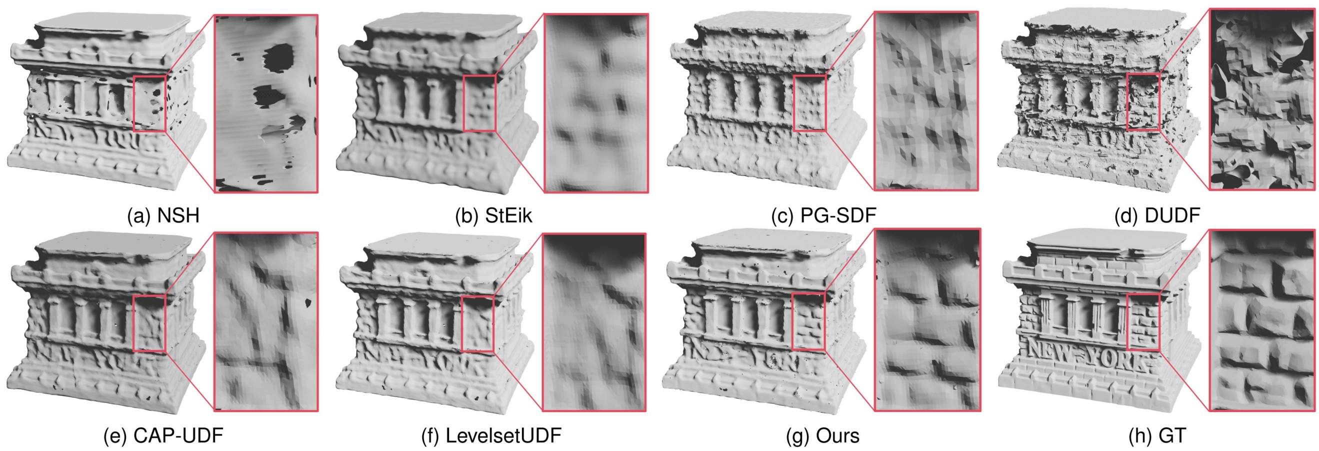

Fig. 9. Qualitative comparison on Stanford 3D Scanning Repository. The zoomed-in views show our method preserves fine details and high-frequency geometric features.

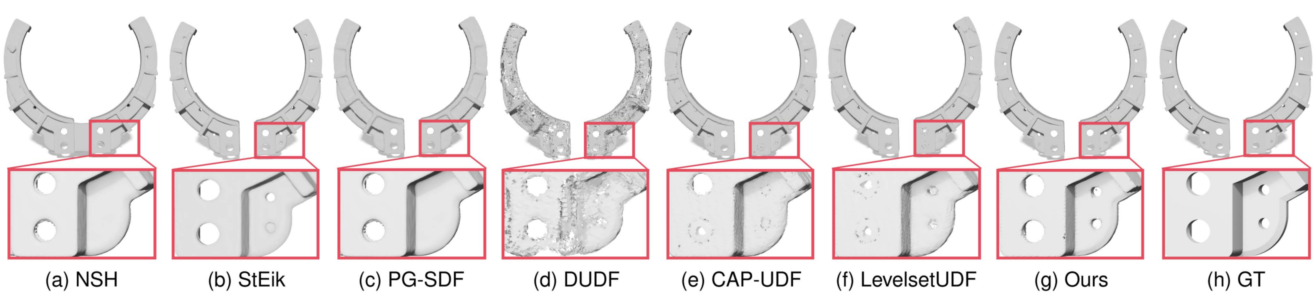

Fig. 10. Qualitative comparison on the ABC dataset for CAD surfaces. The zoomed-in views show that our method accurately reconstructs sharp edges, corners, and piecewise smooth surfaces.

Our method suppresses noise while preserving the underlying surface structure.

D. Ablation Studies

We conduct ablation studies to analyze the impact of timestep selection, network modules, network architecture and the iterative densification strategy. The results are summarized in Table VI, with the default implementation following the settings in Section IV-A. All variants are evaluated on ten representative models selected from the Stanford 3D Scanning Repository to ensure a consistent comparison.

Effect of Timestep K: We evaluate different timestep settings to determine the appropriate transport discretization. As shown in Table VI, a small timestep number fails to accurately model the continuous transport process, resulting in large errors. Increasing K improves reconstruction quality, but the performance gain gradually saturates. Empirically, $K = 5$ provides the best balance between reconstruction accuracy and computational efficiency.

TABLE V
QUANTITATIVE COMPARISON ON SRB AND REAL SCANS DATASETS. BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY.

Method	SRB L1-CD ↓	SRB $FS^{0.01}$ ↑	SRB $FS^{0.005}$ ↑	Real Scans L1-CD ↓	Real Scans $FS^{0.01}$ ↑	Real Scans $FS^{0.005}$ ↑
IGR	11.214	60.11%	40.24%	10.975	60.14%	42.21%
SIREN	7.887	80.24%	65.11%	7.424	79.11%	60.24%
StEik	4.547	96.14%	88.24%	4.147	95.14%	80.11%
NSH	3.325	97.56%	91.32%	3.331	96.96%	92.04%
DUDF	3.131	98.60%	91.40%	3.452	96.66%	90.41%
PG-SDF	4.394	87.12%	79.47%	3.120	98.66%	91.59%
DEUDF	2.843	99.11%	93.12%	2.524	99.88%	92.33%
CAP-UDF	2.888	99.24%	92.91%	2.550	99.87%	92.55%
LevelsetUDF	2.807	99.31%	92.91%	2.546	99.86%	92.34%
Ours	**2.712**	**99.46%**	**94.08%**	**2.510**	**99.89%**	**92.58%**

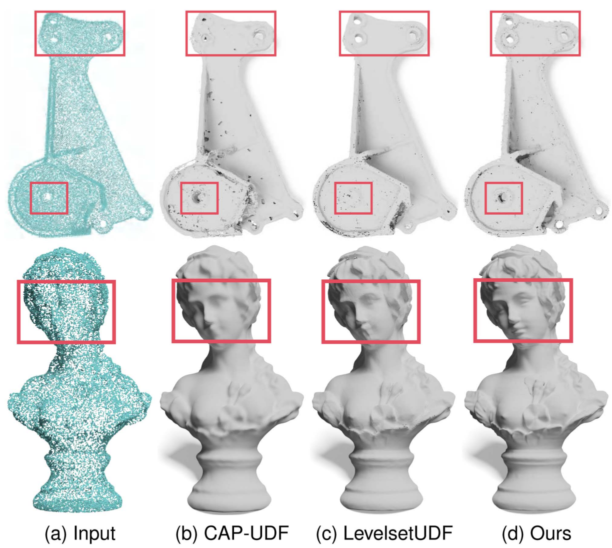

Fig. 11. Qualitative comparison under non-ideal inputs on SRB (top) and Real Scans (bottom). Our method preserves geometric features while suppressing noise and reconstruction artifacts.

Effect of Network Modules: We study the contributions of Position Embedding(P-Emb) and Time Embedding(T-Emb) by comparing different model variants. As shown in Table VI, removing either component degrades reconstruction performance, while removing both causes the most severe degradation. Specifically, Position Embedding is essential for preserving high-frequency details, whereas Time Embedding enables the model to capture temporal dynamics during the evolution process. The combination of both modules yields the best reconstruction quality, demonstrating their complementary roles.

Effect of Network Architecture: We further analyze the influence of network capacity by varying the network depth $L \in \{8, 12\}$ and hidden dimension $H \in \{128, 256, 512\}$. As reported in Table VI, smaller models (e.g., $L = 8, H = 128$) lack sufficient representational capacity to model fine structures. Increasing network capacity generally improves reconstruction quality. However, expanding the hidden dimension from 256 to 512 provides only marginal improvement while significantly increasing computational cost.Therefore, we adopt $L = 12$ and $H = 256$ as the default architecture.

TABLE VI
ABLATION STUDY ON TIMESTEP, NETWORK MODULES, NETWORK ARCHITECTURE FOR RECONSTRUCTION QUALITY ON STANFORD SCANNING REPOSITORY

Ablation	L1-CD ↓	$FS^{0.01}$ ↑	$FS^{0.005}$ ↑	NC ↑
$K = 1$	2.121	98.87%	97.22%	95.12%
$K = 3$	1.967	99.85%	97.64%	95.87%
$K = 10$	**1.912**	99.98%	98.54%	97.59%
w/o P-Emb	1.957	99.85%	98.34%	94.54%
w/o T-Emb	1.946	99.91%	98.48%	96.08%
w/o P-Emb&T-Emb	2.019	99.73%	97.62%	94.12%
$L = 8, H = 128$	2.001	99.75%	98.46%	93.22%
$L = 8, H = 256$	1.982	99.83%	98.43%	95.51%
$L = 8, H = 512$	1.971	99.85%	98.46%	96.14%
$L = 12, H = 128$	1.930	99.93%	98.48%	95.11%
$L = 12, H = 512$	1.914	**99.99%**	**98.64%**	97.08%
w Densification	3.154	99.42%	87.79%	95.69%
w/o Densification	3.635	96.22%	81.38%	94.66%
Default	1.913	**99.99%**	98.57%	**97.79%**

TABLE VII
MODEL COMPLEXITY AND EFFICIENCY ANALYSIS ON STANFORD SCANNING REPOSITORY

Method	Complexity: Params. (M)	Complexity: FLOPs (G)	Efficiency: Train. (min)	Efficiency: Infer. (s)	Quality: L1-CD ↓	Quality: $FS^{0.01}$ ↑
StEik	1.4	82.8	15.9	16.8	4.326	95.40%
DUDF	**0.5**	**26.6**	**6.7**	28.1	2.779	99.82%
CAP-UDF	0.5	27.5	9.4	**15.6**	2.320	99.95%
MutilPull	2.6	151.99	64.3	34.2	2.308	99.96%
Ours	1.0	59.3	14.5	53.9	**2.241**	**99.98%**

Effect of Iterative Densification: We conduct ablation on sparse inputs of 5 k points by comparing models trained with densification (up to 30 k points) and without it. As shown in Table VI, removing densification significantly degrades performance. Without densification, sparse inputs fail to provide sufficient surface constraints, leading to fragmented reconstructions. In contrast, iterative densification progressively fills geometric gaps and improves supervision, resulting in more complete and stable surfaces.

E. Complexity and Efficiency Analysis

Table VII compares the model complexity, efficiency, and reconstruction quality of different overfitting-based methods.

TABLE VIII
PERFORMANCE COMPARISON UNDER IDENTICAL NETWORK SCALES, EVALUATING THE EFFECT OF MODEL CAPACITY

Network Scale	Method	L1-CD ↓	$\mathrm{FS}^{0.01}$ ↑	$\mathrm{FS}^{0.005}$ ↑	NC ↑
$L = 8, H = 256$	CAP-UDF	2.003	99.23%	97.36%	95.93%
	LevelsetUDF	1.991	99.34%	98.12%	**96.21%**
	Ours	**1.982**	**99.83%**	**98.43%**	95.51%
$L = 12, H = 256$	CAP-UDF	1.965	99.96%	98.48%	97.32%
	LevelsetUDF	1.921	99.97%	98.55%	97.52%
	Ours	**1.913**	**99.99%**	**98.57%**	**97.79%**

Only the field query time on the evaluation grid is reported as inference time, while Marching Cubes surface extraction is excluded for all methods. DUDF and CAP-UDF achieve relatively high efficiency but yield lower reconstruction accuracy, while StEik requires more parameters and still performs worse than our method. MultiPull, another multi-step projection-based approach, employs a larger model and substantially higher computational cost than ours. Despite using fewer parameters (1.0 M vs. 2.6 M) and lower FLOPs (59.3 G vs. 151.9 G), our method achieves superior reconstruction performance. Although the proposed trajectory-flow framework incurs additional inference cost due to its multi-step evolution process, it achieves the best L1-CD and $\mathrm{FS}^{0.01}$ among all compared methods, demonstrating a favorable trade-off between reconstruction quality and computational efficiency.

Impact of Network Capacity: We further examine whether the performance gains are due to increased network capacity. CAP-UDF, LevelsetUDF, and our method are evaluated under both (8×256) and (12×256) architectures. As shown in Table VIII, while all methods benefit from larger networks, our method consistently outperforms the competing approaches under both settings. This suggests that the improvement mainly stems from the proposed trajectory-flow modeling rather than increased model capacity.

F. Densification Strategy Analysis

Our method targets object-level and local-scene neural implicit surface reconstruction rather than large-scale scanning or downstream tasks such as object recognition and 3D detection. Following the standard protocol, the input density of $30k$ points is commonly used to evaluate neural implicit reconstruction methods. Denser point clouds (e.g., $100k$ points), as employed by classical methods such as Screened Poisson Reconstruction [23], would substantially increase training and querying costs while exceeding the reconstruction regime considered here. To accommodate sparser inputs, we apply the proposed densification strategy to upsample the point cloud to 30 k points, ensuring fair comparison, sufficient geometric constraints, and computational efficiency.

Effect of UDF Threshold: The threshold ϵ controls the strictness of the filtering process by determining whether candidate points are sufficiently close to the underlying surface for insertion. As shown in Fig. 12, disabling this mechanism introduces numerous erroneous candidates and causes noticeable reconstruction artifacts. A small threshold ($\epsilon = 0.001$) over-rejects valid points and impairs surface completion, while a large threshold ($\epsilon = 0.01$) admits off-surface samples and degrades reconstruction quality. In practice, $\epsilon = 0.005$ achieves a good trade-off between stability and completeness and is used in all experiments.

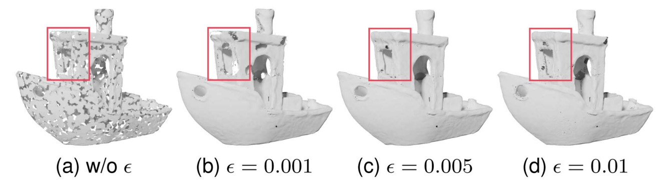

Fig. 12. Effect of the threshold ϵ on iterative densification. A moderate value ($\epsilon = 0.005$) achieves the best trade-off between reconstruction stability and surface completeness.

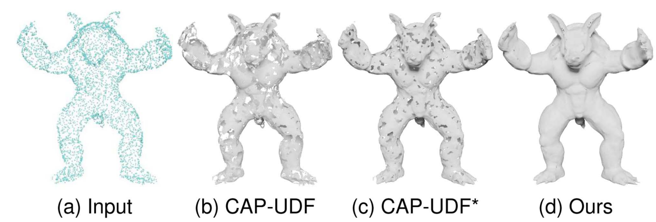

Fig. 13. Generalization of the densification strategy. (a) Input point cloud ($5k$ points). (b) CAP-UDF. (c) CAP-UDF*, i.e., CAP-UDF equipped with the proposed densification, showing improved surface completeness. (d) Our method, achieving the best reconstruction quality through the combination of densification and flow-guided projections.

Generalization and Plug-and-Play Capability: The proposed densification strategy depends only on UDF values and gradients, making it readily applicable to other neural implicit reconstruction methods. As shown in Fig. 13, integrating it into CAP-UDF improves surface completeness and reconstruction quality, demonstrating its plug-and-play effectiveness. While the strategy generalizes well across UDF-based methods, the best results are achieved within our full framework, where flow-guided projections provide more accurate candidate points for densification.

G. Robustness Evaluation

Robustness to Challenging Conditions: To further evaluate robustness, we conduct experiments under six challenging input conditions, including outliers, disconnected components, non-uniform sampling, noisy real scans, high-genus topology, and structural missing data. These settings cover a broad spectrum of non-ideal geometric and sampling scenarios encountered in practice. Fig. 14 presents qualitative comparisons of DUDF, CAP-UDF, and the proposed method under these conditions. Noisy real scans are particularly challenging, as measurement noise corrupts surface observations and degrades geometric fidelity. Structural missing regions introduce another difficulty: although our densification strategy improves geometric coverage, it cannot fully recover severely under-observed structures, such as corners and edges. In contrast, under the remaining degradation types, the proposed trajectory flow demonstrates strong adaptability and consistently maintains stable reconstruction quality across diverse inputs.

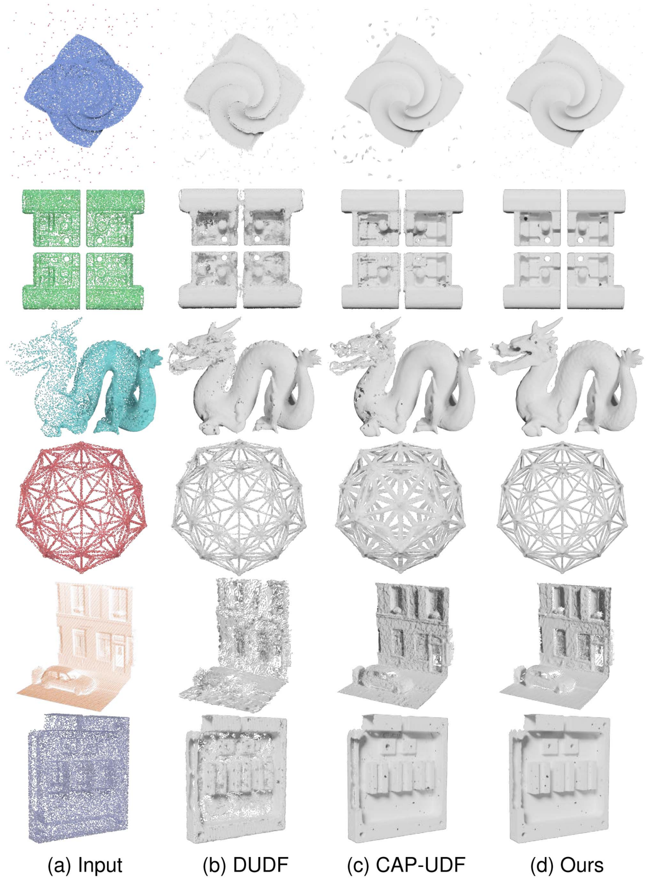

Fig. 14. Robustness comparison under challenging input conditions. From top to bottom: point clouds with outliers, disconnected components, non-uniform sampling, high-genus topology, severe noise, and structural missing data.

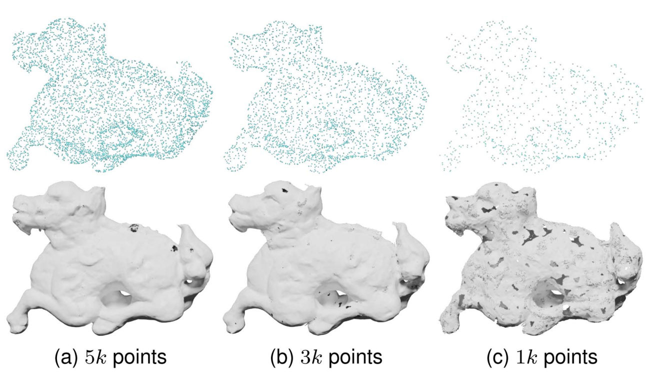

Fig. 15. Reconstruction under varying input sparsity. The first row shows input point clouds with $5k$, $3k$, and $1k$ points from left to right. The second row shows the corresponding reconstruction results, demonstrating that the proposed densification strategy preserves overall geometry under sparse inputs, while quality degrades under extremely sparse observations.

Robustness to Input Sparsity: Fig. 15 presents reconstruction results under varying input sparsity levels. When sufficient geometric cues are available (Figs. 15(a) and (b)), the proposed trajectory-flow framework with the densification strategy successfully recovers the overall shape and major geometric structures. Under extremely sparse input (Fig. 15(c)), however, the lack of observations limits the recovery of fine details and missing structures.

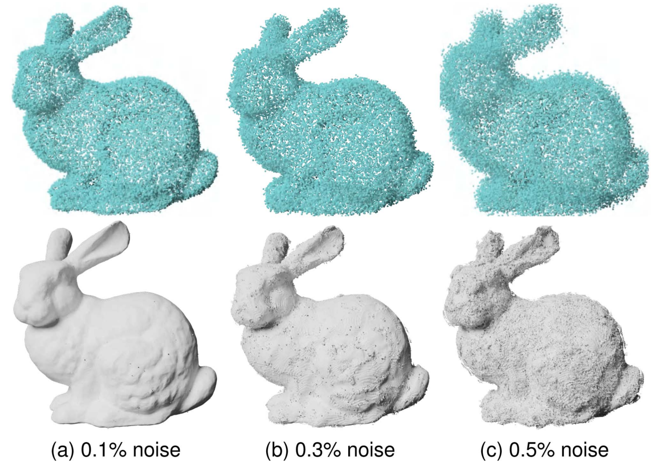

Fig. 16. Reconstruction under varying noise levels. The first row shows input point clouds with 0.1%, 0.3%, and 0.5% Gaussian noise. The second row shows the corresponding reconstructions, illustrating the robustness of the proposed method to moderate noise while preserving geometric details.

TABLE IX
QUANTITATIVE COMPARISON WITH RECENT STATE-OF-THE-ART SUPERVISED METHODS† ON THE STANFORD 3D SCANNING REPOSITORY. BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	L1-CD ↓	$FS^{0.01}$ ↑	$FS^{0.005}$ ↑	NC ↑
HSDF†	3.587	96.11%	88.74%	93.99%
DITTO†	3.062	97.74%	92.16%	95.22%
LoSF†	2.661	99.87%	94.30%	96.07%
Ours	**2.241**	**99.98%**	**96.54%**	**97.78%**

Robustness to Noise: Fig. 16 presents reconstruction results under varying noise levels. As shown in Figs. 16(a) and (b), the proposed method suppresses noise while preserving geometric details. When the noise level exceeds the scale of local geometric features, reconstruction quality degrades and some details become blurred (Fig. 16(c)).

H. Trajectory-Flow Versus Supervised Methods

To further validate the effectiveness of the proposed trajectory flow, we compare it with recent supervised reconstruction methods, including HSDF [44], DITTO [78], and LoSF [40]. As reported in Table IX, the proposed method consistently achieves the best performance across all evaluation metrics. These results demonstrate that the trajectory-flow formulation can achieve superior reconstruction accuracy without requiring external supervision or learned priors.

V. LIMITATIONS AND FUTURE WORK

The primary limitation of our method is inference efficiency. Unlike single-step approaches, the proposed trajectory-flow framework relies on a multi-step evolution process, resulting in repeated network evaluations and increased inference time. Future work will investigate adaptive sampling and geometry-aware step-size strategies to improve efficiency while preserving reconstruction quality.

Another limitation arises under severely noisy and under-observed inputs, as shown in Fig. 14. These cases are fundamentally constrained by the lack of reliable observations, which limits the recoverable geometry and fine details. Future work will explore geometry-driven restoration densification to better handle noisy and incomplete inputs.

A further limitation of the proposed trajectory flow is that projection accuracy gradually decreases with increasing distance from the target surface, as longer transport trajectories accumulate larger integration errors and exhibit greater geometric ambiguity, as shown in Fig. 3. Since reconstruction primarily relies on near-surface projections, this effect has limited influence on the final reconstruction quality. In addition, a fixed integration budget may not be optimal for query points with different distances or noise levels. Future work will explore adaptive integration schemes that allocate computation according to query-point location and local geometric complexity.

VI. Conclusion

We present a new perspective on self-supervised implicit surface reconstruction by reformulating distance estimation as a globally constrained trajectory evolution process. Instead of local displacement regression, we model unsigned distance prediction as deterministic transport along geometrically consistent trajectories between surface samples and query points. This formulation embeds shortest-path structure into the evolution dynamics, providing an intrinsic constraint that stabilizes optimization and mitigates local errors.

Our bidirectional linear flow enforces global consistency, while the iterative densification strategy progressively refines geometric structure from sparse samples to continuous surfaces. Together, these components transform shortest-path modeling from local approximation to global transport, improving robustness under challenging sampling conditions. Extensive experiments demonstrate state-of-the-art performance across multiple benchmarks.

Beyond empirical gains, our formulation suggests a broader perspective, implicit distance fields can be understood as the outcome of geometric evolution rather than scalar regression. This viewpoint highlights the potential of integrating geometric constraints and continuous dynamics into neural implicit representations, opening new directions for stable and interpretable surface reconstruction.

References

[1] J. Liu et al., "EverywhereAR: A visual authoring system for creating adaptive ar game scenes," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 10, pp. 7297–7310, Oct. 2025.

[2] H. Cao, X. Xia, G. Wu, R. Hu, and L. Liu, "ScanBot: Autonomous reconstruction via deep reinforcement learning," *ACM Trans. Graph. (SIGGRAPH)*, vol. 42, no. 4, pp. 157–1, 2023.

[3] G. Yang, F. Xue, Q. Zhang, K. Xie, C.-W. Fu, and H. Huang, "UrbanBIS: A large-scale benchmark for fine-grained urban building instance segmentation," *ACM Trans. Graph. (SIGGRAPH)*, vol. 42, no. 4, pp. 1–11, 2023.

[4] J. Guo, H. Qin, Y. Zhou, X. Chen, L. Nan, and H. Huang, "Fast building instance proxy reconstruction for large urban scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 11, pp. 7267–7282, Nov. 2024.

[5] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "ROSEFusion: Random optimization for online dense reconstruction under fast camera motion," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–17, 2021.

[6] Y. Tang, J. Zhang, Z. Yu, H. Wang, and K. Xu, "MIPS-Fusion: Multi-implicit-submaps for scalable and robust online neural RGB-D reconstruction," *ACM Trans. Graph. (SIGGRAPH Asia)*, vol. 42, no. 6, pp. 1–16, 2023.

[7] 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.

[8] J. Huang, H.-X. Chen, and S.-M. Hu, "A neural Galerkin solver for accurate surface reconstruction," *ACM Trans. Graph.*, vol. 41, no. 6, pp. 1–16, 2022.

[9] C. Li et al., "MASH: Masked anchored spherical distances for 3D shape representation and generation," in *Proc. Special Int. Group Comput. Graph. Interactive Techn. Conf. Conf. Papers*, 2025, pp. 1–11.

[10] G. Lin et al., "On optimal sampling for learning SDF using mlps equipped with positional encoding," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 9, pp. 6563–6574, Sep. 2025.

[11] 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 Recognit.*, 2019, pp. 165–174.

[12] G. Yuan, Q. Fu, Z. Mi, Y. Luo, and W. Tao, "SSRNet: Scalable 3D surface reconstruction network," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 12, pp. 4906–4919, Dec. 2023.

[13] X. Yang, G. Lin, Z. Chen, and L. Zhou, "Neural vector fields: Generalizing distance vector fields by codebooks and zero-curl regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 7, pp. 5818–5831, Jul. 2025.

[14] G. Lin et al., "Patch-grid: An efficient and feature-preserving neural implicit surface representation," *ACM Trans. Graph.*, vol. 44, no. 2, pp. 1–21, 2025.

[15] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3789–3799.

[16] B. Ma, Z. Han, Y.-S. Liu, and M. Zwicker, "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.

[17] 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, Dec. 2024.

[18] Z. Wang et al., "Neural-IMLS: Self-supervised implicit moving least-squares network for surface reconstruction," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 8, pp. 5018–5033, Aug. 2024.

[19] Z. Wang et al., "Neural-singular-Hessian: Implicit neural representation of unoriented point clouds by enforcing singular Hessian," *ACM Trans. Graph.*, vol. 42, no. 6, pp. 1–14, 2023.

[20] Y. Ben-Shabat, C. H. Koneputugodage, and S. Gould, "Digs : Divergence guided shape implicit neural representation for unoriented point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19301–19310.

[21] 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.

[22] T. Noda, C. Chen, W. Zhang, X. Liu, Y.-S. Liu, and Z. Han, "Multi-Pull: Detailing signed distance functions by pulling multi-level queries at multi-step," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2024, pp. 13404–13429.

[23] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, 2013.

[24] 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. 1–13, 2022.

[25] S. Lin, Z. Shi, and Y. Liu, "Fast and globally consistent normal orientation based on the winding number normal consistency," *ACM Trans. Graph.*, vol. 43, no. 6, pp. 1–19, 2024.

[26] R. Xu et al., "Globally consistent normal orientation for point clouds by regularizing the winding-number field," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–19, 2023.

[27] W. Liu et al., "Consistent point orientation for manifold surfaces via boundary integration," in *Proc. ACM SIGGRAPH Conf. Papers*, 2024, pp. 1–11.

[28] W. Liu et al., "Diffusing winding gradients (DWG): A parallel and scalable method for 3D reconstruction from unoriented point clouds," *ACM Trans. Graph.*, vol. 44, no. 2, pp. 1–18, 2025.
[29] S. Lin, D. Xiao, Z. Shi, and B. Wang, "Surface reconstruction from point clouds without normals by parametrizing the Gauss formula," *ACM Trans. Graph.*, vol. 42, no. 2, pp. 1–19, 2025.
[30] Y. Ma, D. Xiao, Z. Shi, and B. Wang, "Convection augmented Gauss reconstruction for unoriented point clouds," *ACM Trans. Graph.*, vol. 44, no. 5, pp. 1–18, 2025.
[31] A. Boulch and R. Marlet, "POCO: Point convolution for surface reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6292–6304.
[32] 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, pp. 108–124.
[33] 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 Recognit.*, 2019, pp. 4455–4465.
[34] 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 Recognit.*, 2021, pp. 1788–1797.
[35] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Proc. Eur. Conf. Comput. Vis*., 2020, pp. 523–540.
[36] T. Takikawa et al., "Neural geometric level of detail: Real-time rendering with implicit 3D shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11358–11367.
[37] T. MÜLLER, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–15, 2022.
[38] D. Petrov, M. Gadelha, R. Měch, and E. Kalogerakis, "ANISE: Assembly-based neural implicit surface reconstruction," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 8, pp. 4514–4526, Aug. 2024.
[39] J. Huang, Z. Gojcic, M. Atzmon, O. Litany, S. Fidler, and F. Williams, "Neural kernel surface reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4369–4379.
[40] J. Hu et al., "A lightweight UDF learning framework for 3D reconstruction based on local shape functions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2025, pp. 1297–1307.
[41] S. Ren and J. Hou, "Shape as line segments: Accurate and flexible implicit surface representation," in *Proc. Int. Conf. Learn. Representations*, 2025, pp. 1–13.
[42] E. Mello Rella, A. Chhatkuli, E. Konukoglu, and L. Van Gool, "Neural vector fields for implicit surface representation and inference," *Int. J. Comput. Vis.*, vol. 133, no. 4, pp. 1855–1878, 2025.
[43] S. Jin, M. Tuo, Y. Gu, Y. Zhang, and S. Zhang, "VFIR: Vector fields implicit representation for surface reconstruction from point clouds," *IEEE Trans. Vis. Comput. Graph*., vol. 32, no. 7, pp. 5786–5802, Jul. 2026.
[44] L. Wang et al., "HSDF: Hybrid sign and distance field for neural representation of surfaces with arbitrary topologies," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 9, pp. 5215–5228, Sep. 2025.
[45] Y. Lu et al., "Unsigned orthogonal distance fields: An accurate neural implicit representation for diverse 3D shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 20551–20560.
[46] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 7462–7473.
[47] C. Xu, F. Hou, W. Wang, H. Qin, Z. Zhang, and Y. He, "Details enhancement in unsigned distance field learning for high-fidelity 3D surface reconstruction," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 8806–8814.
[48] M. Atzmon and Y. Lipman, "Sal: Sign agnostic learning of shapes from raw data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2562–2571.
[49] C. H. Koneputugodage, Y. Ben-Shabat, D. Campbell, and S. Gould, "Small steps and level sets: Fitting neural surface models with point guidance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21456–21465.
[50] Q. Dong et al., "NeurCADRecon: Neural representation for reconstructing cad surfaces by enforcing zero Gaussian curvature," *ACM Trans. Graph.*, vol. 43, no. 4, pp. 1–17, 2024.
[51] M. Fainstein, V. Siless, and E. Iarussi, "DUDF: Differentiable unsigned distance fields with hyperbolic scaling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 4484–4493.
[52] C. Yang et al., "Monge-ampere regularization for learning arbitrary shapes from point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 8, pp. 6809–6822, Aug. 2025.
[53] 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, Sep. 2024.
[54] 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 Recognit.*, 2022, pp. 6305–6315.
[55] B. Ma, Y.-S. Liu, M. Zwicker, and Z. Han, "Surface reconstruction from point clouds by learning predictive context priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6316–6327.
[56] Z. Liu, J. Zhang, M. Zhang, R. Ke, C. Yu, and L. Liu, "Unsupervised point cloud reconstruction via recurrent multi-step moving strategy," *IEEE Trans. Multimedia*, vol. 28, pp. 972–984, 2026.
[57] S. Li, Y. Liu, G. Gao, M. Gu, and Y.-S. Liu, "I-filtering: Implicit filtering for learning neural distance functions from 3D point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 48, no. 1, pp. 109–126, Jan. 2026.
[58] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 1–12.
[59] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–12.
[60] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet, "Diffusion schrödinger bridge with applications to score-based generative modeling," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 17695–17709.
[61] Z. Zhu, J. Hou, H. Liu, H. Zeng, and J. Hou, "Learning efficient and effective trajectories for differential equation-based image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 10, pp. 9150–9168, Oct. 2025.
[62] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–15.
[63] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–13.
[64] L. Wu et al., "Fast point cloud generation with straight flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 9445–9454.
[65] D. de Silva Edirimuni, X. Lu, G. Li, L. Wei, A. Robles-Kelly, and H. Li, "StraightPCF: Straight point cloud filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 20721–20730.
[66] T. Sun, L. Zhu, S. Huang, S. Song, and I. Armeni, "Rectified point flow: Generic point cloud pose estimation," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2025, pp. 168960–168989.
[67] E. Perez, F. Strub, H.de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3942–3951.
[68] B. Mildenhall, P.P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.
[69] 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. 14168–14178.
[70] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1412.6980*.
[71] 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.
[72] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll, "Multi-Garment net: Learning to dress 3D people from images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5420–5430.
[73] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–8, 2013.
[74] H. Yang, Y. Sun, G. Sundaramoorthi, and A. Yezzi, "StEik: Stabilizing the optimization of neural signed distance functions and finer shape representation," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2023, pp. 13993–14004.
[75] S. Koch et al., "ABC: A big cad model dataset for geometric deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9593–9603.
[76] 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 Recognit.*, 2019, pp. 10122–10131.
[77] Z. Huang, Y. Wen, Z. Wang, J. Ren, and K. Jia, "Surface reconstruction from point clouds: A survey and a benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 9727–9748, Dec. 2024.
[78] J. Shim and K. Joo, "DITTO: Dual and integrated latent topologies for implicit 3D reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 5396–5405.

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

Zixu Dun received the BS degree from the Yangtze University, in 2025. He is currently working toward the MS degree with the China University of Geosciences, Wuhan. His research interests include 3D vision and deep learning.

Zheng Liu received PhD degree from Central China Normal University. He held a postdoctoral position with School of Mathematical Sciences, University of Science and Technology of China. He is currently an associate professor with the School of Computer Science, China University of Geosciences, Wuhan, China. His research interests include geometry processing, 3D deep learning, computer graphics, and 3D vision.

Ying He (Member, IEEE) is an associate professor with the College of Computing and Data Science, Nanyang Technological University , Singapore, where he is also the Director of the Centre for Augmented and Virtual Reality. His research interests include geometric computing and analysis, with applications in computer graphics, 3D vision and computer-aided design. He is/was on the editorial boards of *IEEE Transactions on Visualization and Computer Graphics*, *Computer Graphics Forum*, and *Computational Visual Media*. He has also held leadership roles as General or Program Co-Chair for several conferences, including the Shape Modeling International in 2022, the Symposium on Solid and Physical Modeling in 2022 and 2023, Geometric Modeling and Processing in 2014 and 2021, and the Conference on Computational Visual Media in 2020.