Deterministic Point Cloud Diffusion for Denoising

Zheng Liu, Zhenyu Huang, Maodong Pan, and Ying He

***Abstract*—Diffusion-based generative models have achieved remarkable success in image restoration by learning to iteratively refine noisy data toward clean signals. Inspired by this progress, recent efforts have begun exploring their potential in 3D domains. However, applying diffusion models to point cloud denoising introduces several challenges. Unlike images, clean and noisy point clouds are characterized by structured displacements. As a result, it is unsuitable to establish a transform mapping in the forward phase by diffusing Gaussian noise, as this approach disregards the inherent geometric relationship between the point sets. Furthermore, the stochastic nature of Gaussian noise introduces additional complexity, complicating geometric reasoning and hindering surface recovery during the reverse denoising process. In this paper, we introduce a deterministic noise-free diffusion framework that formulates point cloud denoising as a two-phase residual diffusion process. In the forward phase, directional residuals are injected into clean surfaces to construct a degradation trajectory that encodes both local displacements and their global evolution. In the reverse phase, a U-Net-based network iteratively estimates and removes these residuals, effectively retracing the degradation path backward to recover the underlying surface. By decomposing the denoising task into directional residual computation and sequential refinement, our method enables faithful surface recovery while mitigating common artifacts such as over-smoothing and under-smoothing. Extensive experiments on synthetic and real-world datasets demonstrate that our method achieves state-of-the-art performance in both quantitative metrics and visual quality.**

***Index Terms*—Point cloud denoising, diffusion models, residual diffusion, 3D deep learning, point cloud processing.**

I. Introduction

With the increasing availability of 3D scanning technologies, including laser scanners and depth cameras, point clouds have become an essential data representation in various fields like digital twins [1], [2], manufacturing [3], [4], and robotics [5], [6]. Nonetheless, the scanning process inherently introduces noise as a result of measurement errors, which diminishes data quality and complicates subsequent tasks, such as point cloud understanding [7], [8], [9], [10], completion [11], and reconstruction [12].

Over recent decades, considerable progress has been made; however, conventional denoising methods [13], [14], [15], [16], [17], [18] remain constrained by their reliance on local data and their inefficacy in addressing intricate noise patterns. Furthermore, the majority of these traditional approaches necessitate manual parameter adjustment, which requires expert knowledge for optimal configuration. This adjustment process is both time-consuming and labor-intensive, yet essential for obtaining satisfactory denoising outcomes, consequently restricting the applicability of these traditional methods.

Over the past few years, learning-based approaches have emerged as powerful alternatives to traditional denoising techniques. Among them, displacement-based methods have gained prominence for their conceptual simplicity and strong performance. These methods estimate the point-wise displacement of noisy points from the underlying surface and apply the inverse displacement for denoising, either in a single inference pass [19], [20] or through iterative refinement [21]. However, these approaches face a fundamental trade-off. Single-pass methods often leave noise and outliers, resulting in artifacts like protrusions or false edges. Iterative methods, while more effective at noise removal, tend to over-smooth geometry, causing surface shrinkage or loss of fine details. This inherent tension between restoration effectiveness and detail preservation limits the performance of displacement-based denoising methods.

Recent advances have generated considerable interest in leveraging diffusion models for point cloud denoising. These methods generally adopt a forward process that adds Gaussian noise to clean point clouds and a reverse process that estimates the data distribution gradient to reconstruct the surface [22], [23]. While these models excel in preserving complex shapes and fine details, their stochastic nature in the forward process overlooks the deterministic displacements between noisy points and the true surface. Specifically, the injection of Gaussian noise during the forward process increases the stochasticity of the diffusion, as it neglects the structured geometric relationships between noisy and clean point sets. This omission not only diminishes the interpretability of the diffusion process but also compromises the effectiveness of the reverse denoising. The accumulated randomness introduced throughout the forward diffusion complicates surface recovery, leading to artifacts such as uneven point densities, spurious bumps in smooth regions, and false geometric features.

Received 23 June 2025; revised 17 September 2025; accepted 11 October 2025. Date of publication 16 October 2025; date of current version 6 February 2026. This work was supported in part by the National Key R&D Program of China under Grant 2024YFA1016300, in part by the National Natural Science Foundation of China under Grant 12471359, in part by the Aeronautical Science Fund of China under Grant 20240009052001, and in part by the Ministry of Education, Singapore, through Academic Research Fund Grant under Grant RT19/22. Recommended for acceptance by P.-S. Wang. *(Corresponding author: Maodong Pan.)*

Zheng Liu and Zhenyu Huang are with the School of Computer Science, China University of Geosciences (Wuhan), Wuhan 430079, China (e-mail: liu.zheng.jojo@gmail.com; huangzhenyu@cug.edu.cn).

Maodong Pan is with the School of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: maodong@nuaa.edu.cn).

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/huangzygiti/DPCD.

Digital Object Identifier 10.1109/TVCG.2025.3621633

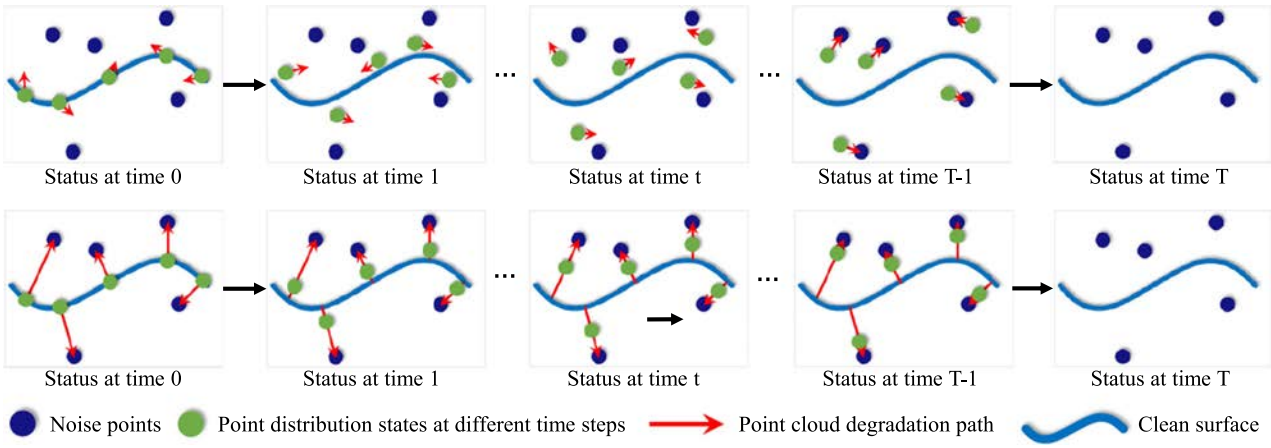

Fig. 1. Key insight and motivation. Top: Existing diffusion-based methods inject Gaussian noise during the forward process, disrupting the geometric structure of point clouds and increasing ambiguity in the reverse phase, which makes accurate restoration difficult. Bottom: In contrast, our method establishes a directional degradation path in the forward process, enabling the reverse phase to progressively restore the underlying surface along a clear, well-defined trajectory. This design leads to superior denoising performance and better preservation of fine geometric details.

To address these challenges, we reformulate point cloud denoising within the Residual Diffusion Model (RDM) [24], which comprises two tightly integrated stages: forward residual addition and reverse residual removal. In 3D domains, residual directions are geometrically meaningful and crucial for surface recovery, so our model incorporates both geometric interpretability and directional guidance. In the forward process, we establish a deterministic degradation path by iteratively adding residual vectors—defined as point-wise displacements from the clean surface—thus linking clean and noisy point clouds. This geometry-aware formulation overcomes the limitations of stochastic forward diffusion in prior methods, as shown in Fig. 1. In the reverse process, we train a U-Net to traverse this path in the opposite direction, progressively predicting and subtracting residuals to recover the underlying surface. By breaking down the denoising task into directional computation (forward) and residual prediction (reverse), our geometry-driven approach produces high-fidelity results while effectively minimizing over- and under-denoising artifacts.

We argue that our method is not a direct extension of image-based diffusion methods [24], [25], [26]. Instead, it is specifically designed to address the unique characteristics of point cloud denoising. For images, Gaussian noise provides a reasonable degradation model; however, in point clouds, the relationship between clean and noisy data is governed by structured geometric displacements. Applying Gaussian noise in the forward process ignores these intrinsic geometric relations, resulting in a mismatch between the assumed degradation model and the actual noise distribution. Furthermore, the stochastic nature of Gaussian noise introduces additional randomness into the reverse denoising phase, which complicates geometric reasoning and hinders accurate surface recovery. To overcome these limitations, we establish a deterministic degradation path guided by residual displacements between clean and noisy point sets. This design eliminates the adverse effects of random noise, leverages underlying geometric structures, and enables a more stable and geometry-consistent denoising process specifically tailored to point cloud data.

Our main contributions are summarized below:

- We reformulate point cloud denoising as a deterministic residual diffusion process, decomposing it into a sequence of directional computation and prediction subproblems. This formulation facilitates high-fidelity restoration with fine-grained details while suppressing over- and under-smoothing artifacts.
- We introduce a deterministic degradation path in the forward process, which establishes an explicit trajectory from the clean to the noisy point cloud. This trajectory provides directional guidance for the reverse denoising process, where a U-Net progressively predicts and removes residuals along the learned path.
- We validate our approach on synthetic and real-world datasets. Extensive experiments demonstrate that our approach outperforms state-of-the-art baselines in both quantitative accuracy and visual quality.

II. Related Work

A. Point Cloud Denoising

Given the extensive body of literature on point cloud denoising, a review is beyond the scope of this work. We refer interested readers to [27] for a comprehensive review. Instead, we focus on three representative categories of methods that are most relevant to our approach.

Estimating displacements for denoising: PointCleanNet [19] removes outliers and estimates displacements for the remaining points using a distance-minimizing loss function. PointFilter [28] adopts a similar architecture but introduces a bilateral loss that incorporates both point and normal information, effectively preserving sharp edges. FCNet [29] reduces feature noise through a teacher–student training strategy with feature domain constraints to guide the student network in learning clean features. PointFilterNet [20] learns three filtering coefficients to generate denoised point coordinates based on the local neighborhood of each noisy point. RePCD [30] employs a recurrent architecture to learn multiscale geometric features for effective

displacement estimation. IterativePFN [21] models progressive smoothing using a step-wise loss that gradually refines point positions toward the clean surface. Wang et al. [31] proposed a random screening-based feature aggregation approach that merges dense and sparse point features to improve the quality of denoising. Recently, PathNet [32] introduces a reinforcement learning framework to select optimal relocation paths, guiding noisy points more effectively toward the underlying surface. More recently, ASDN [33] introduces an adaptive strategy that ceases denoising already cleaned points to prevent over-denoising, while continuing to refine points that remain noisy.

Two-phase point cloud denoising: Lu et al. [34] divided the point cloud into two subsets and estimated multiple normals only on a selected feature set to better preserve geometric edges. Wei et al. [35] designed a dual-network architecture that filters noisy normals by leveraging geometric priors. To further improve normal quality, Zhou et al. [36] and Zhang et al. [37] proposed a similar pipeline in which normal filtering and refinement are treated as separate sub-tasks. More recently, joint denoising and normal estimation have gained attention. Edirimuni et al. [38] employed a contrastive learning framework where the decoder simultaneously outputs denoised points and refined normals. Alternatively, PCDNF [39] and PN-Internet [40] adopt the multi-task learning framework with dual interactive branches to jointly restore points and filter normals. A mesh denoising method, ResGEM, which alternates between normal and vertex position optimization, offers insights for dual-branch approaches in point cloud denoising. ResGEM employs a two-branch pipeline that sequentially recovers local geometric details and corrects vertex distortions. This approach integrates multi-scale edge-conditioned convolutions with complex decoding layers, effectively balancing denoising performance and computational efficiency. Most recently, Li et al. [41] proposed CustomBF, a hybrid approach that overcomes the limitations of the classic bilateral filter by employing a per-point customization strategy. This strategy uses multi-graph encoders to adapt the filter components.

Inferring underlying clean surfaces for denoising: Total Denoising [42] projects noisy points onto clean surfaces in an unsupervised manner using spatial priors. DMRDenoise [43] identifies an underlying downsampled surface, which serves as the basis for reconstructing the clean surface. ScoreDenoise [44] estimates the score of the noise-convolved distribution from the noisy input and applies gradient ascent to recover the clean surface. Zhao et al. [45] introduced a method that perturbs latent-space features and exploits shared structures to reconstruct the clean surface. PD-Flow [46] learns a mapping from Euclidean to latent space via normalizing flows to facilitate noise removal. DeepResampling [47] employs a learned gradient field to iteratively project noisy points toward the underlying surface. More recently, Mao et al. [48] proposed an invertible network, which can disentangle noise in the latent space for noise removal. Wang et al. [49] introduced an implicit filtering method that learns intrinsic signed distance fields from noisy point clouds for noise removal. Most recently, an improved Octree U-Net [50] has been proposed to jointly perform point cloud upsampling and cleaning, offering simplified architecture and markedly higher efficiency compared to patch-wise denoising methods.

B. Diffusion Models From Images to Point Clouds

Diffusion models, originally designed for image generation [51], [52], [53], have since been extended to tasks such as image restoration and super-resolution. Luo et al. [26] introduced a method that transforms a high-quality image into a degraded version with fixed Gaussian noise as the mean state by constructing a mean-reverting stochastic differential equation (SDE). The corresponding reverse-time SDE is then simulated to restore the original image from the low-quality one, without relying on any task-specific prior knowledge. ResShift [25] achieves image super-resolution by transferring the residual between high- and low-resolution images, reducing the number of diffusion steps and eliminating the need for acceleration techniques during inference.

Recent studies have extended their applicability to point cloud processing, demonstrating significant potential across a range of 3D tasks. Luo et al. [54] introduced an autoencoder framework with a diffusion model as the decoder for point cloud generation. Chu et al. [55] proposed DiffComplete, a conditional diffusion model that balances realism, multi-modality, and geometric fidelity for point cloud completion. IPoD [56] integrates point diffusion with implicit field learning, improving 3D reconstruction through self-conditioning and iterative refinement. Qu et al. [57] developed PUDM, a conditional diffusion model for point cloud upsampling that learns the transformation between dense point sets and noise in an iterative manner.

Despite these advances, applying diffusion models to point cloud denoising remains relatively underexplored. P2P-Bridge [22] introduces a Diffusion Schrödinger Bridge framework tailored to point clouds, modeling an optimal transport trajectory between noisy and clean point sets for effective denoising. More recently, Wang et al. [23] proposed an adaptive score-based diffusion method that dynamically adjusts denoising step sizes based on estimated noise levels, and progressively restores the clean surface using a trained score network. Despite promising performance, existing diffusion-based methods share a fundamental limitation: their forward diffusion stage injects random noise, which lacks geometric interpretability for point cloud denoising.

III. Background

Building on the groundbreaking work by Ho et al. [51], diffusion probabilistic models have demonstrated significant success in the field of image processing. These models define a forward procedure that incrementally adds Gaussian noise to an input image I_0 across T steps, along with a trained reverse process designed to recover the original image. The forward process is defined as:

$$q(I_{1:T}|I_0) := \prod_{t=1}^{T} q(I_t|I_{t-1}). \tag{1}$$

The reverse process can also be described as a Markov chain, formulated as:

$$p_\theta(I_{0:T}) := p_\theta(I_T) \prod_{t=1}^{T} p_\theta(I_{t-1}|I_t), \tag{2}$$

The loss function in [51], derived from maximizing the likelihood of the data distribution $p_\theta(I_0)$, is defined as

$$L(\theta) := \mathbb{E}_{I_0\sim q(I_0),\epsilon\sim N(\mathbf{0},\mathbf{I})}[||\,\epsilon - \epsilon_\theta(I_t,t)\,||^2], \tag{3}$$

where ϵ_θ is the noise estimated by the network. This estimated noise can be used for iterative sampling in the reverse process. We refer readers to the original work [51] for further details and discussions.

More recently, Liu et al. [24] proposed residual diffusion models, which decouple the traditional diffusion process into two components: residual diffusion and noise diffusion. Specifically, the residual diffusion models separate the forward process into a deterministic residual component, which models the transition from the clean image I_0 to its degraded counterpart I_{in}, and a stochastic noise component. The residual is defined as $I_{res} = I_{in} - I_0$. The joint probability distributions of the forward process is given by:

$$q(I_{1:T}|I_0, I_{res}) := \prod_{t=1}^{T} q(I_t|I_{t-1}, I_{res}), \tag{4}$$

where each step follows:

$$q(I_t|I_{t-1}, I_{res}) := \mathcal{N}(I_t; I_{t-1} + \alpha_t I_{res}, \beta_t^2\mathbf{I}), \tag{5}$$

with α_t and β_t controlling the contribution of the residual and the injected noise, respectively. In this forward process, residuals I_{res} and Gaussian noise ϵ are gradually added to I_0, generating the noisy sequence $\{I_t\}$.

The reverse process aims to restore the denoised image from the noisy input. Assuming the network predicts both the residual I^θ_{res} and noise ϵ_θ, we can obtain the estimated denoised image $I^\theta_0 = I_t - \bar\alpha_t I^\theta_{res} - \bar\beta_t\epsilon_\theta$ with $\bar\alpha_t = \sum_{i=1}^t \alpha_t$, $\bar\beta_t = \sqrt{\sum_{i=1}^t \beta_i^2}$. Given I^θ_0 and I^θ_{res}, the reverse step is formulated as:

$$p_\theta(I_{t-1}|I_t) := q_\sigma(I_{t-1}|I_t, I^\theta_0, I^\theta_{res}), \tag{6}$$

where the transfer distribution for sampling I_{t-1} from I_t is

$$q_\sigma(I_{t-1}|I_t, I_0, I_{res}) = \mathcal{N}(I_{t-1}; I_0 + \bar\alpha_{t-1} I_{res} + \sqrt{\bar\beta_{t-1}^2 - \sigma_t^2}\frac{I_t - (I_0 + \bar\alpha_t I_{res})}{\bar\beta_t}, \sigma_t^2 I), \tag{7}$$

where $\sigma_t^2 = \eta\cdot\beta_t^2\bar\beta_{t-1}^2/\bar\beta_t^2$, and $\eta\in[0,1]$ controls the stochasticity of the reverse process (e.g., $\eta = 1$ for stochastic sampling, $\eta = 0$ for deterministic inference). The training objective includes two loss terms:

$$L_{res}(\theta) := \mathbb{E}[\lambda_{res}\,||\,I_{res} - I^\theta_{res}(I_t,t,I_{in})\,||^2], \tag{8}$$

$$L_\epsilon(\theta) := \mathbb{E}[\lambda_\epsilon\,||\,\epsilon - \epsilon_\theta(I_t,t,I_{in})\,||^2], \tag{9}$$

where hyperparameters λ_{res}, $\lambda_\theta\in\{0,1\}$. For a complete theoretical derivation and empirical analysis, we refer readers to the original work [24].

IV. Method

In this section, we introduce the notation and define the residual diffusion model, including the forward and reverse processes on point clouds. Let $\mathcal{P}$ and $\mathcal{P}_{GT}$ represent the noisy and clean point clouds, respectively. To generate the inputs for our method, we divide $\mathcal{P}$ into overlapping patches $\{P_{in}\}$, where the corresponding clean patch is denoted as P_{GT}. The residual displacement is defined as the difference between the noisy patch and its clean counterpart:

$$P_{res} = P_{in} - P_{GT}, \tag{10}$$

which captures the geometric discrepancy that we aim to estimate and remove during denoising. In the forward process, the clean point cloud patch $P_0 = P_{GT}$ is gradually degraded into the noisy patch $P_T = P_{in}$, guided by the residual displacement P_{res}, as shown in Fig. 2. In the reverse process, starting from the noisy input P_T, a U-Net is employed to progressively predict the residual displacement P^t_{res} at each timestep t, which is then used to compute the previous state P_{t-1}. This reverse step is repeated until the final denoised patch P_0 is obtained, as shown in Fig. 2. Since the denoised patches $\{P_0\}$ overlap, we employ the stitching strategy outlined in [21] to reconstruct the complete denoised point cloud.

A. Forward Residual Diffusion

The forward process is defined as a deterministic sequence that degrades the clean point cloud P_0 into the noisy input P_T. At each timestep t, the residual displacement P_{res} is incrementally added to the current state P_{t-1}, scaled by a diffusion coefficient α_t, to simulate the progressive degradation as follows:

$$P_t = P_{t-1} + \alpha_t P_{res}, \tag{11}$$

where α_t controls the diffusion strength, determining the extent to which the residual is incorporated at each step. To facilitate a smooth and stable transition, we adopt a linearly increasing schedule for α_t. This allows the residual to be introduced gradually in the early stages and more strongly in later steps, effectively guiding the point cloud toward the noisy target.

Unfolding the forward process step by step yields:

$$\begin{aligned} P_T &= P_{T-1} + \alpha_T P_{res} \\ &= P_{T-2} + (\alpha_{T-1} + \alpha_T) P_{res} \\ &= \dots \\ &= P_0 + \bar\alpha_t P_{res}, \end{aligned} \tag{12}$$

where $\bar\alpha_t = \sum_{i=1}^t \alpha_i$ denotes the cumulative residual contribution up to timestep t. When $t = T$, we have $\bar\alpha_T = 1$ and $P_T = P_{in}$, completing the degradation process. The forward formulation (12) ensures that the residual displacement is introduced gradually and controllably, allowing for an interpretable and stable reverse denoising process.

B. Reverse Denoising and Training Objective

The reverse process aims to restore the clean point cloud P_0 from the noisy input P_T by progressively estimating and removing the residual introduced during the forward diffusion. At each timestep t, a neural network predicts the residual displacement P^t_{res}, which is used to compute the previous state P_{t-1} from P_t:

$$P_{t-1} = P_t - \alpha_t P^t_{res}, \tag{13}$$

where α_t controls the step-wise correction strength, and P^t_{res} denotes the predicted residual at timestep t. By recursively applying the update step (13), the complete reverse process from

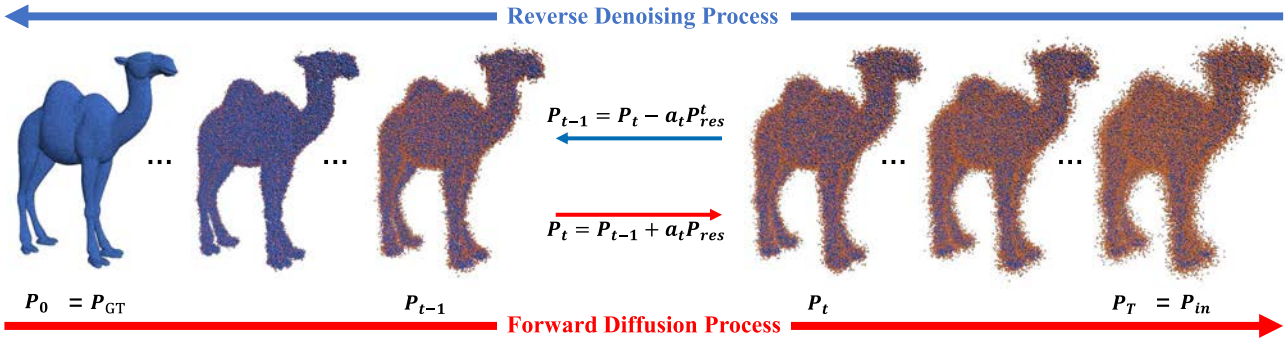

Fig. 2. Illustration of the proposed residual diffusion model for point cloud denoising. The forward process gradually degrades the clean point cloud $P_0 = P_{GT}$ into the noisy input $P_T = P_{in}$, guided by the residual displacement $P_{res} = P_{in} - P_{GT}$. The reverse denoising process restores P_T back to P_0 by progressively removing the predicted residual displacement P^t_{res} at each timestep.

P_T to P_0 can be written as:

$$\begin{aligned} P_0 &= P_1 - \alpha_1 P^1_{res} \\ &= P_2 - (\alpha_1 P^1_{res} + \alpha_2 P^2_{res}) \\ &= \dots \\ &= P_T - \sum_{t=1}^{T} \alpha_t P^t_{res}. \end{aligned} \tag{14}$$

Training objective: The network is trained to predict the residual displacement P^t_{res} from the current point cloud state P_t and timestep t. The objective function is defined as:

$$L_{res}(\theta) = || P_{res} - P^t_{res} ||^2, \tag{15}$$

where P_{res} is the ground-truth residual displacement, and P^t_{res} is the predicted residual at timestep t.

Inference: During inference, the process starts with the state $P_T = P_{in}$. The network then iteratively predicts the residual sequence $\{P^t_{res}\}^T_{t=1}$ over T timesteps. These predictions are sequentially applied via the reverse process (14) to progressively restore the clean point cloud P_0.

C. Network Architecture

Main backbone: As shown in Fig. 3, the network takes the point cloud state P_t and timestep t as input, and outputs the predicted residual P^t_{res}. The input P_t is first encoded into a high-dimensional feature via a position embedding block, while t is processed by a time embedding block to generate scale and shift factors. These factors modulate the spatial feature, which are passed through a U-Net backbone for residual prediction. The U-Net is composed of $L = 4$ encoder and decoder layers arranged symmetrically.

Position and time embedding: The input coordinates P_t are encoded using sinusoidal position embedding, followed by an MLP to extract the feature f_{emb} as

$$f_{\text{emb}} = \text{MLP}(\text{Emb}(P_t)), \tag{16}$$

where $\text{Emb}(\cdot)$ denotes the sinusoidal position encoding introduced in [58]. Compared to directly applying an MLP to raw coordinates, this high-dimensional encoding improves the representation of spatial relationships and relative geometric structures.

The timestep t is encoded via a time embedding block and mapped to modulation parameters $\{\text{scale}, \text{shift}\}$, which are used to transform the spatial feature as:

$$f_0 = \text{scale} \cdot f_{\text{emb}} + \text{shift}. \tag{17}$$

This formulation enables the network to incorporate temporal information by conditioning spatial features on the current diffusion timestep, yielding a timestep-aware representation f_0 that facilitates residual prediction.

Encoder: As illustrated in Fig. 3, each encoder layer E_l receives the output f_{l-1} from the previous encoder layer and produces the encoded feature f_l as follows:

$$f_l = \text{E}_l(f_{l-1}), l = 1, 2, \dots, L. \tag{18}$$

Each encoder layer consists of several feature aggregation blocks. For each point $p_i \in P^{l-1}_t$, where P^{l-1}_t denotes the downsampled point set at layer $l-1$, we collect its K-nearest neighbors in Euclidean space, denoted as p^K_i. To encode local geometric structure, we first construct a neighborhood-based representation and apply an MLP to extract local features:

$$f^k_{p_i} = \text{MLP}\left(\bar{p}_i \parallel p^K_i \parallel \bar{p}_i - p^K_i \parallel \text{norm}(\bar{p}_i - p^K_i)\right), \tag{19}$$

where $\bar{p}_i$ denotes the replicated coordinates of p_i, $\parallel$ denotes feature concatenation, and $\text{norm}(\cdot)$ computes the Euclidean distance. We then concatenate the local feature $f^k_{p_i}$ with the point-wise feature f^k_i to form the feature $\tilde{f}^k_i$. To refine $\tilde{f}^k_i$, an MLP followed by a softmax function is employed to adaptively identify crucial elements, defined as:

$$\tilde{f}_i = \text{SumPooling}(\text{Softmax}(\text{MLP}((\tilde{f}^k_i) \odot \tilde{f}^k_i)), \tag{20}$$

where $\odot$ denotes the Hadamard product, $\text{SumPooling}(\cdot)$ squeezes the input to a single feature by summation.

We adopt stacked aggregation blocks to extract local features, yielding intermediate features $\{f^1_{l-1}, f^2_{l-1}\}$, where

$$f^j_{l-1} = \left\{\tilde{f}_i\right\}^{n_l}_{i=1}, \quad j \in \{1, 2\}, \tag{21}$$

and n_l denotes the number of points in the downsampled set P^{l-1}_t. We then integrate the aggregated features f^1_{l-1}, f^2_{l-1}, and the original input feature f_{l-1} to compute the fused representation:

$$\tilde{f}_l = \text{MLP}(f^1_{l-1} \parallel f^2_{l-1}) + \text{MLP}(f_{l-1}), \tag{22}$$

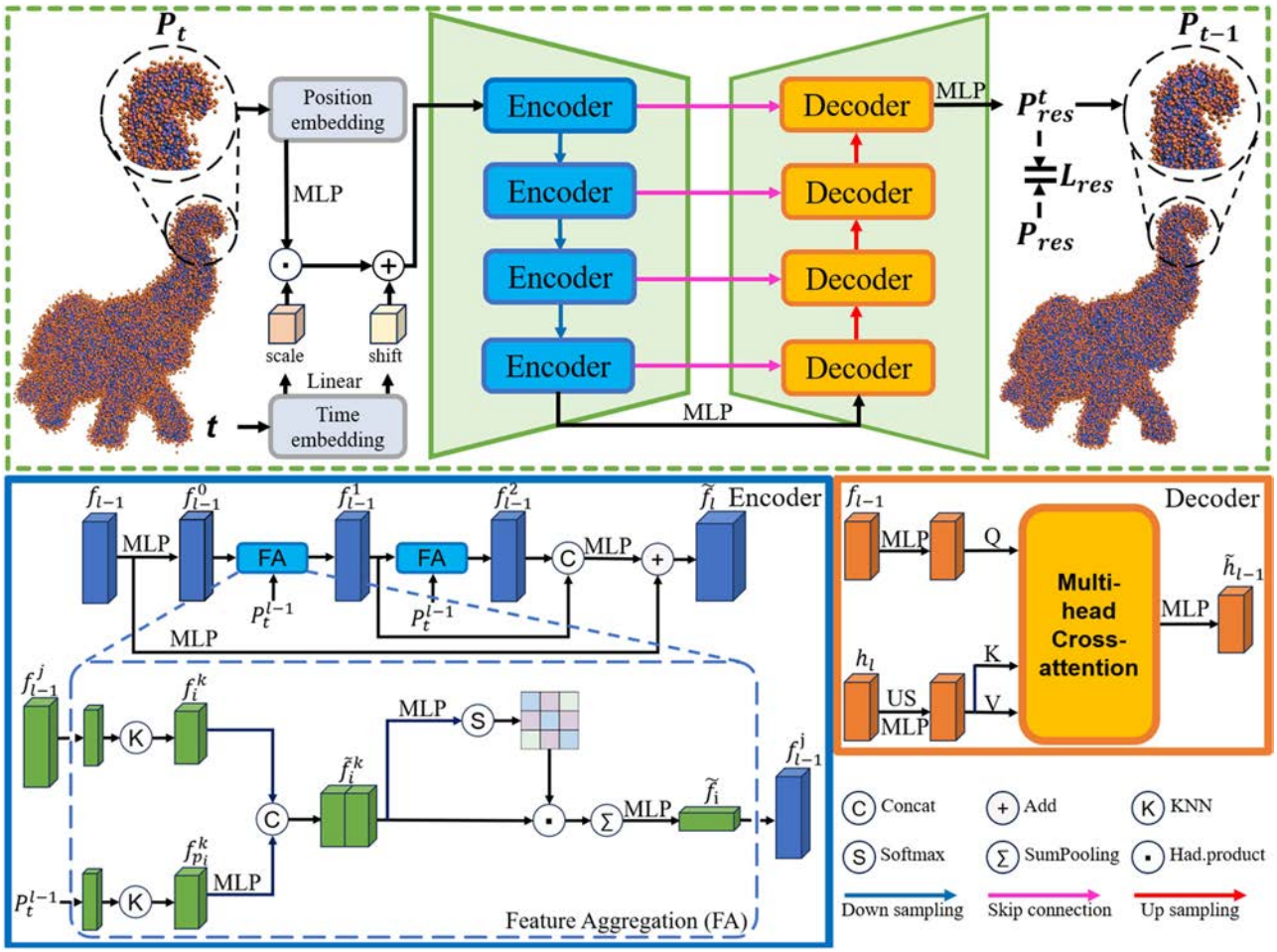

Fig. 3. Illustration of our network architecture. Our network comprises position and time embedding blocks, followed by a U-Net backbone with $L = 4$ symmetrically arranged encoder and decoder layers. Given the point cloud state P_t and the diffusion timestep t, the embedding blocks generate the initial feature representation f_0. This feature is then fed into the U-Net to predict the residual displacement P_{res}^t at the current timestep t.

Finally, a downsampling operation is applied to $\tilde{f}_l$ to obtain the encoder feature as $f_l = \mathrm{DS}(\tilde{f}_l)$.

Decoder: As illustrated in Fig. 3, each decoder layer D_l takes as input the decoder output from the previous layer h_l and the corresponding encoder feature f_{l-1}, and produces the decoded feature h_l as follows:

$$h_{l-1} = \mathrm{D}_l(h_l, f_{l-1}), l = 1, 2, \ldots, L. \tag{23}$$

Specifically, given the decoder feature h_l and the corresponding encoder feature f_{l-1}, we treat f_{l-1} as the query and the upsampled decoder feature $\mathrm{US}(h_l)$ as the key-value pair. These inputs are projected into a shared feature space via a linear transformation $\varphi(\cdot)$:

$$\{Q, K, V\} = \{\varphi(f_{l-1}), \varphi(\mathrm{US}(h_l)), \varphi(\mathrm{US}(h_l))\}. \tag{24}$$

We then apply multi-head cross-attention (MHCA) to compute an intermediate feature:

$$\widetilde{h}_{l-1} = \mathrm{MLP}(\mathrm{MHCA}(Q, K, V)). \tag{25}$$

Finally, the output decoder feature h_{l-1} is obtained via a skip connection with the encoder feature:

$$h_{l-1} = \mathrm{MLP}(\widetilde{h}_{l-1} \parallel f_{l-1}). \tag{26}$$

V. Experiments

A. Datasets and Settings

Training dataset: We train our framework using the PU-Net dataset [59], which consists of 40 meshes. Point clouds are generated from these meshes at resolutions of 10 K, 30 K, and 50 K, yielding a total of 120 training point clouds. Noisy data is introduced by applying Gaussian noise with a standard deviation ranging from 0.05 to 0.2 times the radius of the bounding sphere. Due to its patch-based nature, our method divides input point clouds into smaller, overlapping patches for training, allowing the model to efficiently learn local features and geometric structures. This enables effective training of the residual diffusion model on a dataset of 120 samples, with sufficient generalization despite the limited number of training samples.

Testing datasets: We conduct comparison experiments on the PU-Net dataset [59] at 10 K and 50 K resolutions, resulting in 40 point clouds. To assess the generalization capability of our method, we also evaluate it on real-scanned datasets, including the Kinect dataset [60] and the indoor SceneNN dataset [61]. Additionally, to assess the performance of our method on real-world data, we use the Paris-rue-Madame dataset [62], which

TABLE I
QUANTITATIVE EVALUATION ON PU-NET WITH CD AND P2M METRICS ($\times 10^{-4}$). THE BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY.

Method	10K points						50K points					
	1% noise		2% noise		2.5% noise		1% noise		2% noise		2.5% noise	
	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M
PCN	3.686	1.599	7.926	4.759	10.486	6.987	1.103	0.646	1.978	1.370	3.203	2.486
GPDNet	2.310	0.714	4.284	1.855	5.837	3.066	1.049	0.635	3.288	2.503	5.085	4.134
DMR	4.712	2.196	5.085	2.523	5.277	2.669	1.205	0.762	1.443	0.970	1.696	1.190
Pointfilter	2.461	0.730	3.534	1.155	4.099	1.505	0.758	0.432	0.907	0.507	1.099	0.629
Score	2.522	0.754	3.683	1.380	4.232	1.904	0.716	0.400	1.288	0.833	1.445	0.958
PDFlow	2.126	0.674	3.246	1.324	3.627	1.702	0.651	0.416	1.270	0.921	1.874	1.426
IterativePFN	2.055	0.501	3.043	0.843	3.353	1.046	0.605	0.302	0.803	0.436	1.015	0.588
Invertible	1.825	0.496	2.567	0.824	2.849	1.034	0.496	0.306	0.706	0.447	**0.918**	**0.592**
StraightPCF	1.904	0.545	2.672	0.929	2.925	1.120	0.620	0.389	0.811	0.551	0.987	0.668
P2P-Bridge	2.284	0.686	3.202	1.114	3.531	1.386	0.586	0.330	0.902	0.580	1.165	0.803
Ours	**1.781**	**0.457**	**2.473**	**0.751**	**2.756**	**0.979**	**0.464**	**0.287**	**0.657**	**0.433**	1.120	0.776

features real Paris street scenes captured with a 3D mobile laser scanner.

Baselines: We compare our approach to state-of-the-art methods, including PCN [19], GPDNet [63], DMR [43], Pointfilter [28], Score [44], PDFlow [46], IterativePFN [21], Invertible [48], StraightPCF [64], and P2P-Bridge [22]. To ensure fair comparisons, we use the publicly available codes of these methods and retrain them on the same datasets.

Implementation: The method is implemented using PyTorch and trained on an NVIDIA GeForce RTX 4090 GPU. We use the Adam optimizer with a learning rate of 1×10^{-4}. The model consists of 611 K parameters and is trained for a maximum of 300 epochs with a batch size of 48. Prior to training, point clouds are normalized to fit within a unit sphere. To sample patches of size 1 K, we apply the FPS (Farthest Point Sampling) and KNN (K-Nearest Neighbors) algorithms. For a fair comparison of computational costs, all methods are evaluated on the same GPU.

B. Quantitative Results

For quantitative evaluation, we adopt Chamfer Distance (CD) and Point-to-Mesh (P2M) distance as the error metrics to assess denoising accuracy and completeness, where lower values indicate superior denoising performance.

Numerical evaluations: We first evaluate our method on the PUNet dataset [59], with results recorded in Table I. As shown, our approach outperforms the competing methods in most cases, achieving lower CD and P2M values, except under the 2.5% noise level at 50 K resolution. This indicates that, in general, our approach produces more representative results on the synthetic dataset with varying noise levels.

We also evaluate our method on the Kinect dataset [60], with results presented in Table II. Our method achieves state-of-the-art results on both metrics, demonstrating its significant denoising capabilities and robust generalization performance on real-world scanned data.

We conducted comparative experiments on the indoor SceneNN dataset [61] to further evaluate the generalization ability of our method on unseen data. Using 93 indoor scenes as test data with a resolution of 100,000 and a noise level of 0.1%, we demonstrate that our method achieves competitive results in unseen indoor scenarios, effectively removing noise and outperforming competing approaches, as shown in Table III.

TABLE II
QUANTITATIVE EVALUATION ON KINECT DATA WITH CD AND P2M METRICS ($\times 10^{-4}$)

Method	Kinect	
	CD	P2M
PCN	13.73	8.75
GDPNet	14.83	8.69
Pointfilter	13.77	7.91
Score	13.22	8.18
PDFlow	13.34	8.69
IterativePFN	13.20	8.43
Invertible	13.51	8.96
StraightPCF	12.96	8.59
P2P-Bridge	12.92	7.64
Ours	**12.89**	**7.59**

TABLE III
QUANTITATIVE EVALUATION OF INDOOR SCENARIOS (SCENENN DATA) USING CD AND P2M METRICS ($\times 10^{-4}$)

Method	SceneNN	
	CD	P2M
IterativePFN	0.60	0.44
Invertible	0.58	0.42
StraightPCF	0.54	**0.38**
P2P-Bridge	0.63	0.48
Ours	**0.51**	0.40

TABLE IV
NUMERICAL COMPARISON AMONG COMPETING METHODS UNDER VARIOUS NOISE TYPES. THE UNIT OF CD AND P2M IS 10^{-4}.

Noise Type	Method	10K points				50K points			
		1% noise		2% noise		1% noise		2% noise	
		CD	P2M	CD	P2M	CD	P2M	CD	P2M
Anisotropic	IterativePFN	2.603	0.611	4.932	2.315	0.611	0.309	0.825	0.456
	StraightPCF	2.371	0.665	5.214	2.672	0.631	0.396	0.955	0.591
	P2P-Bridge	2.871	0.915	**4.206**	**1.939**	0.608	0.348	0.947	0.621
	Ours	**2.215**	**0.577**	4.857	2.724	**0.470**	**0.292**	**0.726**	**0.490**
Laplace	IterativePFN	2.429	0.618	**3.486**	**1.182**	0.662	**0.343**	**1.051**	**0.618**
	StraightPCF	2.246	0.668	4.011	1.731	0.714	0.446	1.676	1.181
	P2P-Bridge	2.613	0.843	3.883	1.721	0.709	0.422	1.286	0.884
	Ours	**2.095**	**0.578**	3.577	1.742	**0.582**	0.388	1.217	0.776
Uniform	IterativePFN	1.355	0.399	2.644	0.584	0.444	0.254	0.598	0.297
	StraightPCF	1.315	0.438	2.376	0.625	0.500	0.337	0.588	0.369
	P2P-Bridge	1.768	0.588	2.829	0.827	0.465	0.277	0.571	0.315
	Ours	**1.225**	**0.383**	**2.197**	**0.527**	**0.365**	**0.253**	**0.457**	**0.283**
Discrete	IterativePFN	1.219	0.394	**1.911**	**0.546**	0.347	0.254	0.424	**0.278**
	StraightPCF	1.213	0.430	1.971	0.599	0.458	0.329	0.559	0.373
	P2P-Bridge	1.598	0.599	2.227	0.799	0.389	0.281	0.471	0.321
	Ours	**1.131**	**0.380**	1.952	0.571	**0.323**	**0.253**	**0.412**	0.280

Robustness to noise types: Noisy points are often contaminated by various noise types, such as anisotropic Gaussian, Laplace, uniform distribution, and discrete noise, all of which are common in real-world scenarios. To evaluate the robustness of our method, we test it on the PU-Net dataset with points at 10 K and 50 K resolutions, using noise parameters in the work [47]. The results, summarized in Table IV, demonstrate that our method consistently producing the best and second-best results across different noise conditions. This highlights the robustness of our approach, particularly in handling diverse noise types.

C. Qualitative Comparions

In Fig. 4, we present the denoising results on the PU-Net [59] dataset, using 50 K points with a 2% noise level. The color map represents the P2M distances, demonstrating that our method closely approximates the ground truth. Compared to all competing methods, our approach recovers cleaner geometric features and avoids introducing artifacts in smooth areas, marking a significant improvement over existing techniques.

In Fig. 5, we show the results on the Kinect dataset [60]. As shown, competing methods tend to blur fine-grained details and introduce visual artifacts to varying extents. In contrast, our method produces visually excellent results, effectively preserving small-scale features while maintaining smooth regions and avoiding unnatural artifacts.

In Fig. 6, we demonstrate results on the indoor SceneNN dataset [61]. As shown, our method effectively suppresses noise and recovers scene geometry with greater fidelity compared to competing approaches, highlighting the enhanced robustness of our method on unseen indoor data.

In Fig. 7, we present the results on the street dataset [62], which contains stronger noise and complex infrastructure, such as roads, buildings, and traffic signs. As shown, our method effectively removes heavy noise while better preserving structural shapes compared to competing methods. These results highlight our method's superior visual quality and robust performance in complex scenarios.

Overall, across all the tested datasets, our method consistently produces visually cleaner results with fewer unnatural artifacts. This shows the effectiveness of our approach in high-fidelity restoration, even in real-world scenarios.

D. Ablation Studies

We conduct ablation experiments on several variants of our method to assess the impact of different module configurations and parameter settings, as summarized in Table V. All experiments are performed on the PUNet dataset at 10 K resolution. The ablation study is organized into five parts.

Encoder selection: To evaluate the effectiveness of our encoder–decoder architecture, we replace it with several well-known alternatives, including PointNet++[65], Point Transformer (PTv1) [66], and Point Transformer V2 (PTv2) [67]. As shown in Table V, our architecture enables more effective information exchange between geometric and semantic feature spaces, thereby facilitating the learning of more representative features. It is important to note that our primary contribution lies in proposing a generalizable residual diffusion paradigm that can be seamlessly integrated with diverse encoder–decoder architectures, rather than introducing a bespoke architecture. Consequently, we do not benchmark against the latest encoder–decoder variants (e.g., Point Transformer V3 [68]), as the development of more advanced encoding–decoding

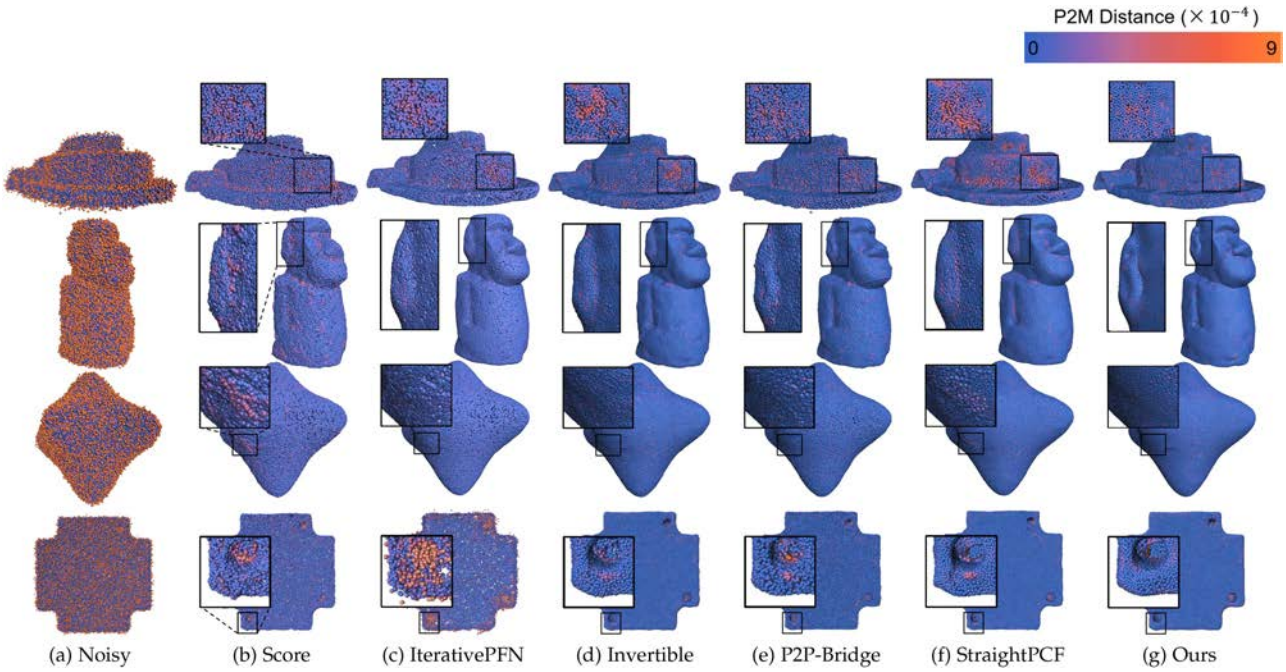

Fig. 4. Comparison of tested approaches based on P2M distance for 50K-resolution shapes with 2% Gaussian noise on PU-Net. The zoomed-in views show that our approach recovers cleaner geometric features and avoids introducing artifacts in smooth areas. Darker points (i.e., more blue) indicate smaller distance from ground truth.

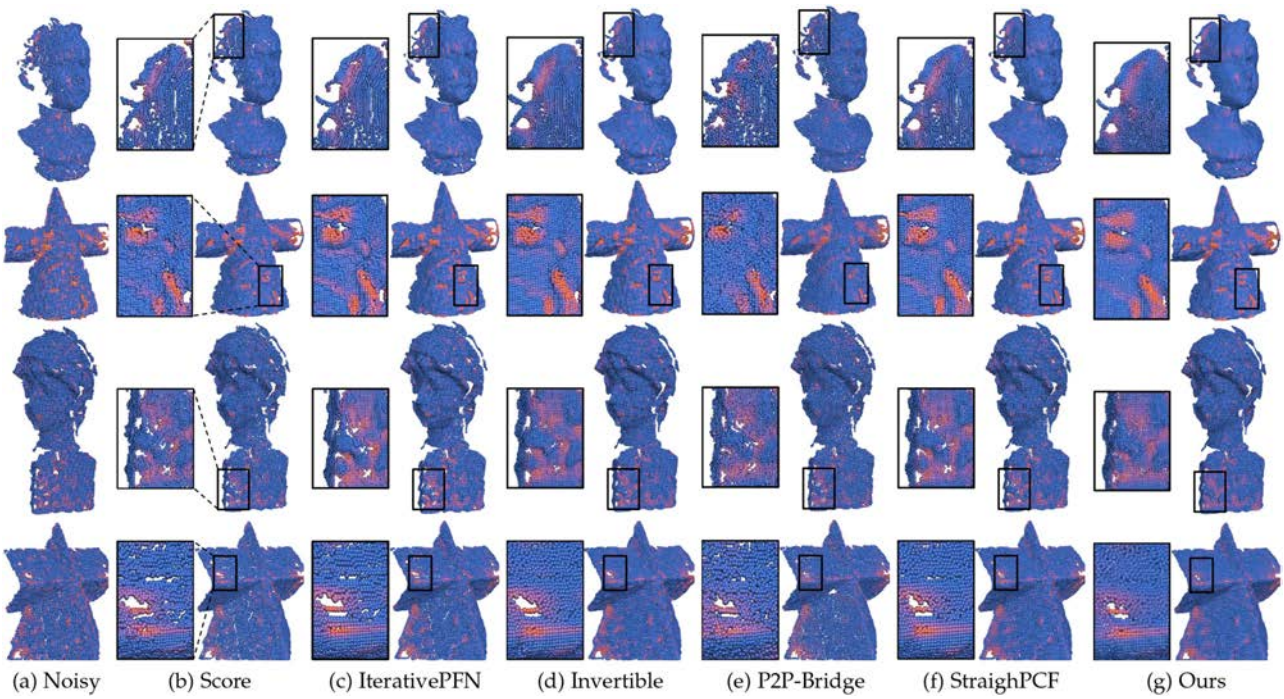

Fig. 5. Comparison of tested approaches on Kinect data. The zoomed-in views show that our approach better preserves fine-grained details while producing cleaner results.

architectures represents a separate and parallel line of research, which we leave for future exploration.

Selection of sampling steps: We investigate the impact of different sampling steps in the reverse process on denoising performance. The results indicate that performance with 10 and 50 steps is worse than with 30 steps. With too few sampling steps, information propagation is inadequate, leading to poor noise removal, as shown in Fig. 8(a). On the other hand, with too many steps, excessive recovery can introduce outliers and potentially damage surface shapes, as shown in Fig. 8(c). Therefore, we

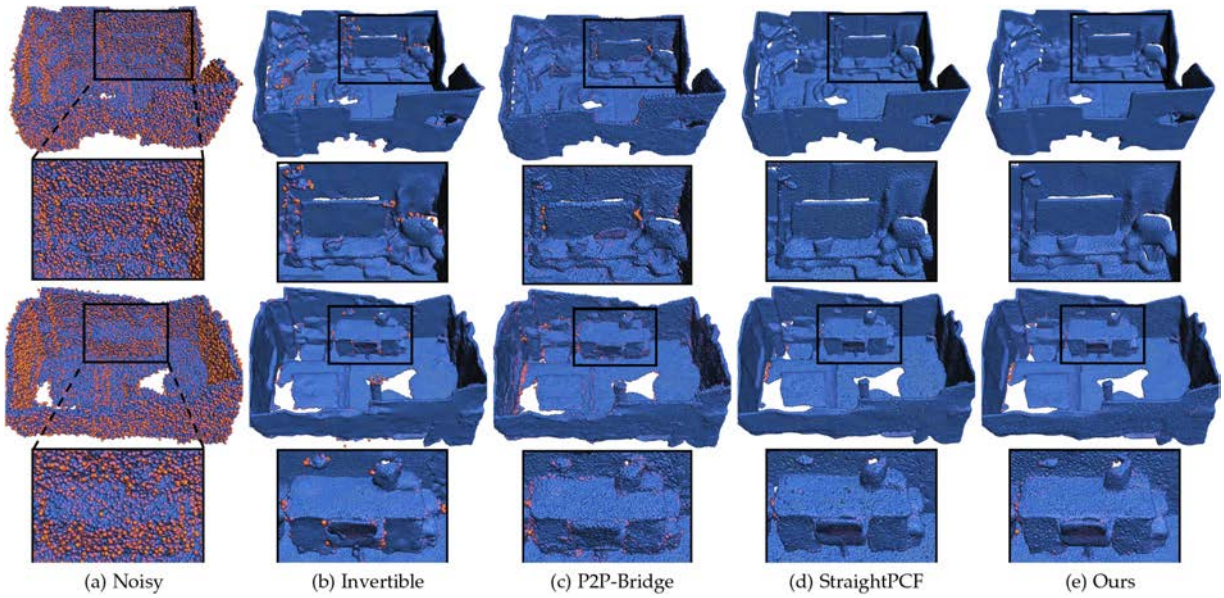

Fig. 6. Comparison of tested approaches on indoor scenarios. The zoomed-in views show that our method suppresses noise more effectively and recovers scene geometry more faithfully than competing approaches.

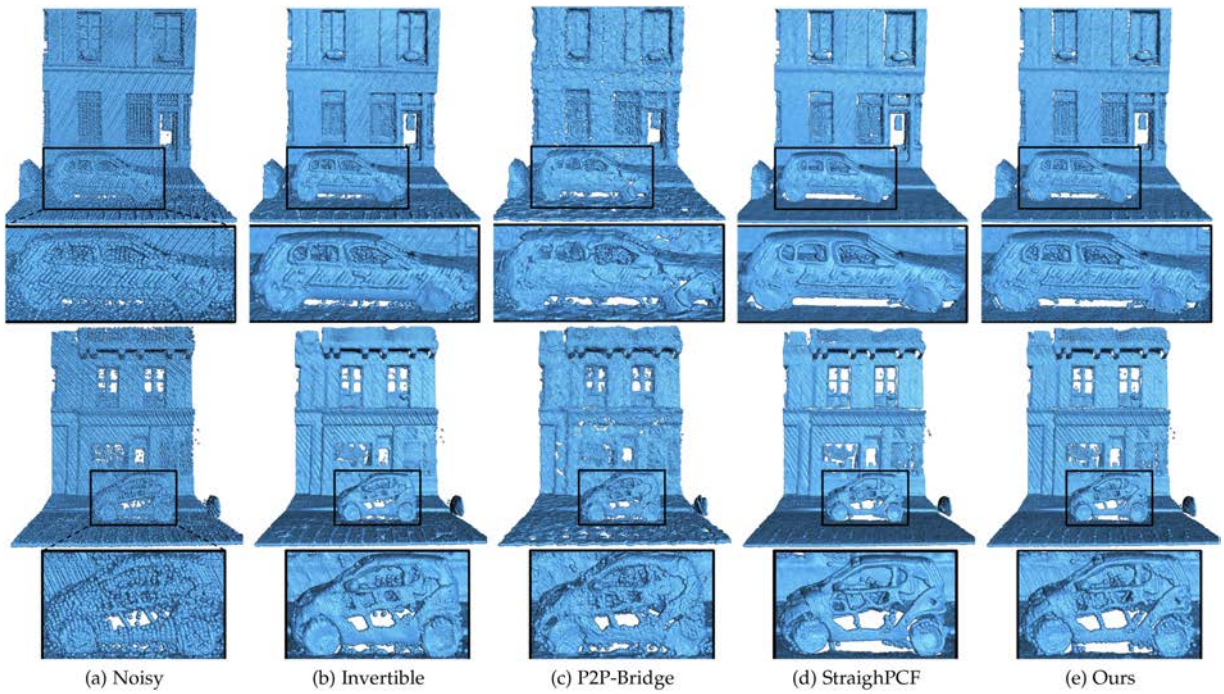

Fig. 7. Comparison of tested approaches on real-world scenarios. The zoomed-in views show that our approach removes noise more effectively and preserves structural shapes better than competing methods.

select 30 steps as the optimal choice for our default sampling strategy.

Impact of random perturbations: To evaluate the impact of stochastic perturbations, we conduct an ablation study by introducing diffuse Gaussian noise into the forward process. The results are summarized in Table V, with visual comparisons shown in Fig. 9. We observe a decline in both quantitative metrics and visual quality upon the introduction of noise. This degradation occurs because, while stochasticity enhances diversity in generative modeling, it negatively affects denoising. Specifically, introducing randomness during the forward degradation phase disrupts the directional geometric cues, increasing ambiguity in the reverse process. These findings underscore the importance of employing a geometry-aware, deterministic degradation path within our diffusion-based denoising framework.

Depth choice: To evaluate the impact of network depth on denoising performance, we conduct an ablation study focusing

TABLE V
NUMERICAL RESULTS OF THE ABLATION STUDIES ON PU-NET AT 10 K RESOLUTION

Ablation	10K points					
	1% noise		2% noise		2.5% noise	
	CD	P2M	CD	P2M	CD	P2M
Encoder of PointNet++	1.794	0.470	2.509	0.762	2.810	1.011
Encoder of PTv1	1.790	0.465	2.497	0.764	2.806	0.997
Encoder of PTv2	1.788	0.459	**2.450**	0.758	**2.750**	0.981
10 sample	1.799	0.472	2.506	0.791	2.850	1.064
50 sample	1.789	0.466	2.499	0.773	2.834	1.038
3 layers	1.801	0.462	2.500	0.759	2.787	0.989
5 layers	**1.781**	**0.449**	2.485	0.755	2.753	0.988
CD Loss	1.811	0.491	2.556	0.814	2.853	1.044
Add 0.1% Gaussian noise	1.796	0.463	2.525	0.771	2.807	1.005
Default	**1.781**	0.457	2.473	**0.751**	2.756	**0.979**

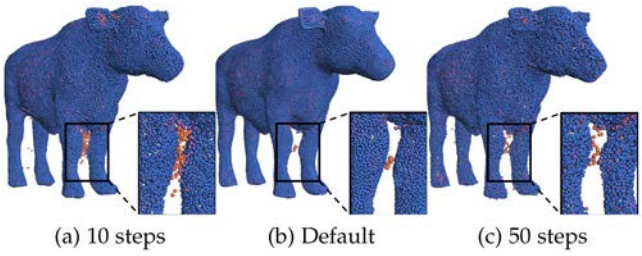

Fig. 8. Impact of sampling step selection on denoising performance.

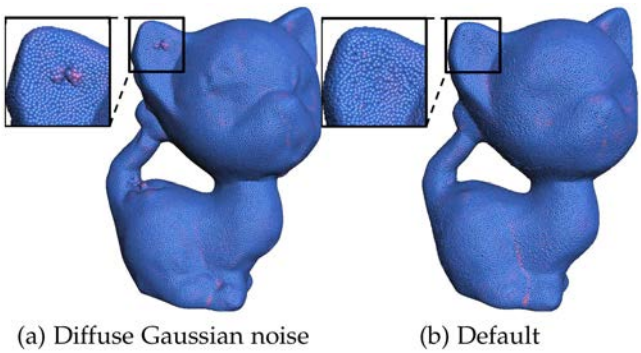

Fig. 9. Impact of diffusive Gaussian noise in the forward process on denoising performance.

on the depth selection of our encoder-decoder architecture. Our analysis reveals that optimal performance is achieved with a network depth of four layers. When the network depth is insufficient, the network cannot fully learn the geometric information and structural features of the point clouds. On the other hand, when the network is too deep, excessive downsampling leads to the blurring of important geometric details of the point clouds and increases unnecessary computational cost.

Selection of loss functions: We emphasize that our method does not rely on a strict one-to-one correspondence between noisy and clean points. Our method is designed to predict residual displacements directly, which become more accurate as the reverse process proceeds. Since Chamfer Distance (CD) loss supervises the reconstructed clean point cloud rather than the residual displacement itself, it does not align with our formulation. Therefore, MSE loss is more consistent with our theoretical framework and better suited for our method, as shown in Table V.

TABLE VI
COMPARISON OF PARAMETER COUNT AND INFERENCE TIME BETWEEN OUR METHOD AND COMPETING APPROACHES

Method	Param. (M)	Time (s)
PCN	27.91	10.12
DMR	0.23	**0.28**
Pointfilter	1.39	5.73
Score	**0.18**	0.29
PDFlow	0.47	1.18
IterativePFN	3.22	1.85
Invertible	0.68	0.74
StraightPCF	0.53	2.35
P2P-Bridge	26.44	0.73
Ours	0.61	5.09

E. Complexity and Efficiency Analysis

We evaluate model complexity (total number of parameters) and efficiency (inference time) on synthetic 10K-point clouds corrupted with 1% Gaussian noise. The results are summarized in Table VI. As shown, Score uses the fewest parameters and DMR achieves the fastest inference, but both yield lower denoising accuracy. Our method employs slightly more parameters than Score and, owing to the repeated passes required by the reverse sampling process, incurs a longer inference time than competing approaches. Despite this overhead, it offers a favorable accuracy–efficiency trade-off. Future work will focus on accelerating inference to further improve overall performance.

F. Limitations

Although our residual diffusion denoising method achieves state-of-the-art performance, there are several limitations to our approach. The first limitation is the relatively high inference time. The reverse denoising phase involves multiple iterations of network processing for sampling, resulting in slower inference speeds compared to direct inference methods, which is a common drawback of diffusion-based approaches. Another limitation is the suboptimal denoising performance under high-density (50 K points) and high-noise (2.5%) conditions. High-density point clouds usually contain richer geometric details; however, when severe noise is present, the noise and geometric details become highly entangled, making them difficult to separate. Consequently, our method occasionally misinterprets noise as geometric information, which degrades denoising performance in such challenging conditions.

VI. CONCLUSION

We have introduced a geometry-driven diffusion framework for point cloud denoising, which significantly improves the

quality of denoised point clouds. By establishing a deterministic degradation path in the forward process and utilizing a U-Net architecture to progressively restore the clean surface during the reverse process, our approach achieves high-fidelity denoising, effectively preserving fine details and suppressing artifacts. Extensive experiments on both synthetic and real-world datasets demonstrate that our method outperforms state-of-the-art denoising techniques, showcasing its robustness and effectiveness.

Our geometry-driven diffusion framework not only provides a novel solution to point cloud denoising, but also lays the foundation for future advancements in the broader field of point cloud processing. As a versatile framework, it can be extended to tasks such as point cloud completion, upsampling, and surface reconstruction. Furthermore, integrating our framework with self-supervised or unsupervised learning paradigms could enhance its generalizability, particularly in real-world scenarios where clean ground truth data are scarce or unavailable. We believe that our approach will have a lasting impact on point cloud processing, especially in applications requiring high-quality 3D data, such as 3D vision, robotics, and digital modeling.

ACKNOWLEDGMENT

The author thank the anonymous reviewers for their constructive feedback.

REFERENCES

[1] 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," in *Proc. ACM SIGGRAPH Conf.*, 2023, pp. 1–11.

[2] 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.

[3] Z. Yu, Z. Qin, L. Zheng, and K. Xu, "Learning instance-aware correspondences for robust multi-instance point cloud registration in cluttered scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 19605–19614.

[4] H. Cao, X. Xia, G. Wu, R. Hu, and L. Liu, "ScanBot: Autonomous reconstruction via deep reinforcement learning," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–16, 2023.

[5] 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.*, vol. 42, no. 6, pp. 1–16, 2023.

[6] 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.

[7] P.-S. Wang, "OctFormer: Octree-based transformers for 3D point clouds," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–11, 2023.

[8] B. Pang, Z. Zheng, Y. Li, G. Wang, and P.-S. Wang, "Neural Laplacian operator for 3D point clouds," *ACM Trans. Graph. Asia*, vol. 43, no. 6, pp. 1–14, 2024.

[9] D. Shao, X. Lu, W. Wang, X. Liu, and A. S. Mian, "TriCI: Triple cross-intra branch contrastive learning for point cloud analysis," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 09, pp. 5321–5333, Sep. 2025.

[10] J. Jiang et al., "PCoTTA: Continual test-time adaptation for multi-task point cloud understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 96229–96253.

[11] Y. Liu et al., "PointCG: Self-supervised point cloud learning via joint completion and generation," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 10, pp. 6648–6660, Oct. 2025.

[12] 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.

[13] 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.

[14] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 176:1–7, 2009.

[15] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. R. Zhang, "Edge-aware point set resampling," *ACM Trans. Graph.*, vol. 32, no. 1, pp. 9:1–9:12, 2013.

[16] Z. Liu et al., "A feature-preserving framework for point cloud denoising," *Comput-Aided Des.*, vol. 127, 2020, Art. no. 102857.

[17] H. Chen, M. Wei, Y. Sun, X. Xie, and J. Wang, "Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 11, pp. 3255–3270, Nov. 2020.

[18] X. Lu, S. Schaefer, J. Luo, L. Ma, and Y. He, "Low rank matrix approximation for 3D geometry filtering," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 4, pp. 1835–1847, Apr. 2022.

[19] M.-J. Rakotosaona, V. La Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, "PointCleanNet: Learning to denoise and remove outliers from dense point clouds," *Comput. Graph. Forum*, vol. 39, no. 1, pp. 185–203, 2020.

[20] X. Wang, X. Fan, and D. Zhao, "PointFilterNet: A filtering network for point cloud denoising," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 3, pp. 1276–1290, Mar. 2023.

[21] D. de Silva Edirimuni, 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 Recognit.*, 2023, pp. 13530–13539.

[22] M. Vogel, K. Tateno, M. Pollefeys, F. Tombari, M.-J. Rakotosaona, and F. Engelmann, "P2P-Bridge: Diffusion bridges for 3d point cloud denoising," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 184–201.

[23] Z. Wang, M. Li, S. Xin, and C. Tu, "Adaptive and iterative point cloud denoising with score-based diffusion model," in *Proc. Conf. Comput. Graph. Forum*, 2025, Art. no. e70149.

[24] J. Liu, Q. Wang, H. Fan, Y. Wang, Y. Tang, and L. Qu, "Residual denoising diffusion models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 2773–2783.

[25] Z. Yue, J. Wang, and C. C. Loy, "Resshift: Efficient diffusion model for image super-resolution by residual shifting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 13294–13307.

[26] Z. Luo, F. K. Gustafsson, Z. Zhao, J. Sjölund, and T. B. Schön, "Image restoration with mean-reverting stochastic differential equations," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 23045–23066.

[27] J. Wang, B. Fei, D. de Silva Edirimuni, Z. Liu, Y. He, and X. Lu, "A survey of deep learning-based point cloud denoising," 2025, *arXiv:2508.17011*.

[28] D. Zhang, X. Lu, H. Qin, and Y. He, "Pointfilter: Point cloud filtering via encoder-decoder modeling," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 3, pp. 2015–2027, Mar. 2021.

[29] X. Wang, W. Cui, R. Xiong, X. Fan, and D. Zhao, "FCNet: Learning noise-free features for point cloud denoising," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 11, pp. 6288–6301, Nov. 2023.

[30] H. Chen, Z. Wei, X. Li, Y. Xu, M. Wei, and J. Wang, "RePCD-Net: Feature-aware recurrent point cloud denoising network," *Int. J. Comput. Vis.*, vol. 130, no. 3, pp. 615–629, 2022.

[31] W. Wang, W. Pan, X. Liu, K. Su, B. Rolfe, and X. Lu, "Random screening-based feature aggregation for point cloud denoising," *Comput. Graph.*, vol. 116, pp. 64–72, 2023.

[32] 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, Jun. 2024.

[33] C. Guo, W. Zhou, Z. Liu, and Y. He, "You should learn to stop denoising on point clouds in advance," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 3212–3219.

[34] D. Lu, X. Lu, Y. Sun, and J. Wang, "Deep feature-preserving normal estimation for point cloud filtering," *Comput-Aided Des.*, vol. 125, 2020, Art. no. 102860.

[35] M. Wei, H. Chen, Y. Zhang, H. Xie, Y. Guo, and J. Wang, "GeoDualCNN: Geometry-supporting dual convolutional neural network for noisy point clouds," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 2, pp. 1357–1370, Feb. 2023.

[36] H. Zhou et al., "Refine-net: Normal refinement neural network for noisy point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 946–963, Jan. 2023.

[37] Y. Zhang et al., "Norest-Net: Normal estimation neural network for 3-d noisy point clouds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 2, pp. 2246–2258, Feb. 2025.

[38] D. d. S. Edirimuni, X. Lu, G. Li, and A. Robles-Kelly, "Contrastive learning for joint normal estimation and point cloud filtering," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 8, pp. 4527–4541, Aug. 2024.
[39] 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, Aug. 2024.
[40] C. Yi, Z. Wei, J. Qiu, H. Chen, J. Wang, and M. Wei, "PN-Internet: Point-and-normal interactive network for noisy point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5704411.
[41] P. Li, Z. Wei, H. Chen, X. Yan, and M. Wei, "Revisiting tradition and beyond: A customized bilateral filtering framework for point cloud denoising," *ACM Trans. Graph.*, vol. 44, no. 4, pp. 1–13, 2025.
[42] P. Hermosilla, T. Ritschel, and T. Ropinski, "Total denoising: Unsupervised learning of 3d point cloud cleaning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 52–60.
[43] S. Luo and W. Hu, "Differentiable manifold reconstruction for point cloud denoising," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 1330–1338.
[44] S. Luo and W. Hu, "Score-based point cloud denoising," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 4583–4592.
[45] T. Zhao, P. Gao, T. Tian, J. Ma, and J. Tian, "From noise addition to denoising: A self-variation capture network for point cloud optimization," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 7, pp. 3413–3426, Jul. 2024.
[46] A. Mao, Z. Du, Y.-H. Wen, J. Xuan, and Y.-J. Liu, "PD-Flow: A point cloud denoising framework with normalizing flows," in *Proc. Eur. Conf. Comput. Vis.*, 2022, Art. no. 13663.
[47] H. Chen, B. Du, S. Luo, and W. Hu, "Deep point set resampling via gradient fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 2913–2930, Mar. 2023.
[48] A. Mao, B. Yan, Z. Ma, and Y. He, "Denoising point clouds in latent space via graph convolution and invertible neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 5768–5777.
[49] J. Wang, X. Lu, M. Wang, F. Hou, and Y. He, "Learning implicit fields for point cloud filtering," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 9, pp. 5408–5420, Sep. 2025.
[50] J. Li, B. Pang, and P.-S. Wang, "Joint point cloud upsampling and cleaning with octree-based cnns," *Comput. Vis. Media*, 2025.
[51] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.
[52] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2020, *arXiv:2010.02502*.
[53] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet, "Diffusion schrödinger bridge with applications to score-based generative modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 17695–17709.
[54] S. Luo and W. Hu, "Diffusion probabilistic models for 3D point cloud generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2837–2845.
[55] R. Chu et al., "Diffcomplete: Diffusion-based generative 3d shape completion," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 75951–75966.
[56] Y. Wu et al., "IPoD: Implicit field learning with point diffusion for generalizable 3D object reconstruction from single RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 20432–20444.
[57] W. Qu, Y. Shao, L. Meng, X. Huang, and L. Xiao, "A conditional denoising diffusion probabilistic model for point cloud upsampling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 20786–20795.
[58] Y. Zou, H. Yu, Z. Yang, Z. Li, and N. Akhtar, "Improved mlp point cloud processing with high-dimensional positional encoding," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 7891–7899.
[59] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-Net: Point cloud upsampling network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2790–2799.
[60] P.-S. Wang, Y. Liu, and X. Tong, "Mesh denoising via cascaded normal regression," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 232–1, 2016.
[61] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, "SceneNN: A scene meshes dataset with annotations," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 92–101.
[62] A. Serna, B. Marcotegui, F. Goulette, and J.-E. Deschaud, "Paris-rue-madame database: A 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods," in *Proc. Int. Conf. Pattern Recognit. Appl. Methods*, 2014, pp. 819–24.
[63] F. Pistilli, G. Fracastoro, D. Valsesia, and E. Magli, "Learning graph-convolutional representations for point cloud denoising," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 103–118.
[64] 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.
[65] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.
[66] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16239–16248.
[67] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer V2: Grouped vector attention and partition-based pooling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, p. 13.
[68] X. Wu et al., "Point transformer V3: Simpler, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 4840–4951.

Zheng Liu received the PhD degree from Central China Normal University. He was a postdoctoral with the 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.

Zhenyu Huang received the BS degree in 2024 from the China University of Geosciences, Wuhan, where he is currently working toward the MS degree. His research interests include geometry processing and deep learning.

Maodong Pan received the PhD degree from the University of Science and Technology of China. He was a postdoctoral fellow with Johannes Kepler University Linz, Austria and Johann Radon Institute for Computational and Applied Mathematics, Austria. He is currently an associate researcher with the Nanjing University of Aeronautics and Astronautics, China. His research interests include computational geometry, computer graphics and isogeometric analysis.

Ying He is currently an associate professor with the College of Computing and Data Science, Nanyang Technological University (NTU), Singapore, where he is also the director with the Centre for Augmented and Virtual Reality (CAVR). His research focuses on geometric computing and analysis, with applications in computer graphics, and 3D vision and computer-aided design. He 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 of several conferences, including Shape Modeling International in 2022, Symposium on Solid and Physical Modeling, in 2022 and 2023, respectively, Geometric Modeling and Processing in 2014 and 2021, respectively, and Conference on Computational Visual Media in 2020.