Guiding Point Cloud Denoising with Learned Structural Priors

Chuchen Guo[1], **Zheng Liu**[1*], **Ying He**[2]

[1]School of Computer Science, China University of Geosciences (Wuhan)
[2]College of Computing and Data Science, Nanyang Technological University
guocconly1@foxmail.com, liu.zheng.jojo@gmail.com, yhe@ntu.edu.sg

Abstract

Recovering precise surface geometry from corrupted point clouds remains a core challenge in 3D vision. Although existing denoising techniques achieve remarkable success, balancing noise removal with preserving intricate geometric details continues to pose difficulties. A critical limitation of current methods is that their adaptive feature aggregation mechanisms rely heavily on intermediate network features that have not been explicitly regularized, resulting in unstable guidance signals. This instability restricts the capability of the network to optimally differentiate true geometric details from noise. To overcome this limitation, we propose a novel deep learning framework that explicitly learns *structured representations* as robust priors to guide feature refinement. Our approach first derives a set of representative local structural primitives from input features by means of a learned codebook. This learned *structured representation* then serves as a robust conditional signal, directing a subsequent feature fusion mechanism to dynamically aggregate information in a structure-aware manner, thereby more effectively discerning noise and meticulously reconstructing geometric details. Extensive experiments on several benchmarks have demonstrated the superiority of our framework over existing advanced techniques in terms of detail preservation and noise suppression.

Code — https://github.com/git-guocc/PGD

Introduction

Point clouds serve as an essential representation for the digital 3D world, underpinning many cutting-edge applications such as autonomous driving, robotic perception, high-fidelity reconstruction, and digital twins. However, raw point cloud data acquired through sensors (e.g., LiDAR, depth cameras) often contain significant noise due to physical limitations and environmental factors. Such noise severely distorts the underlying surface geometry, negatively impacting critical downstream tasks including surface reconstruction, object recognition, and scene understanding. Therefore, effective denoising of point clouds is crucial for fully harnessing their potential.

Recent deep learning-based approaches for point cloud denoising, ranging from single-step regression models (de Silva Edirimuni et al. 2024; Guo et al. 2025; Mao et al. 2022; Zhang et al. 2021) to iterative refinement methods (de Silva Edirimuni et al. 2023; Luo and Hu 2021; Vogel et al. 2024; Zhou et al. 2025), have significantly advanced performance. A common strength of these models lies in their sophisticated adaptive feature aggregation mechanisms, such as attention (Vaswani et al. 2017) or dynamic convolutions (Wang et al. 2019). However, a key observation reveals a fundamental limitation: these adaptive mechanisms typically rely on intermediate network features that lack explicit regularization. Consequently, these intermediate guidance signals remain unstable and ambiguous, limiting the network's ability to consistently distinguish between intricate geometric details and noise. This constraint inherently caps the achievable denoising quality of existing methods.

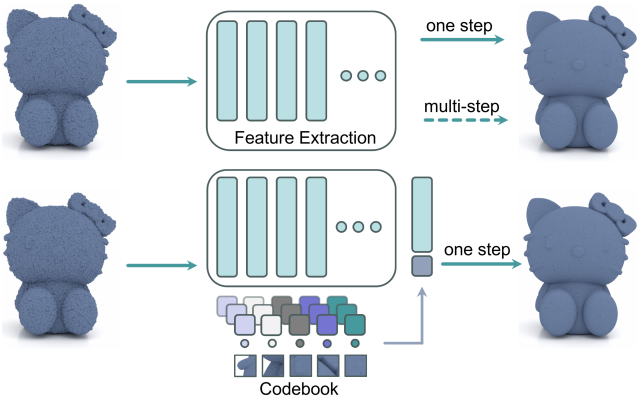

Figure 1: A conceptual illustration of our core motivation. (Top) Existing denoising paradigms, whether single-step or multi-step, often rely on an adaptive network that operates directly on noise-corrupted features. The lack of a clear, high-quality guidance signal can lead to the loss of fine-grained details. (Bottom) Our proposed approach first distills a set of structural primitives via a learned codebook to form a robust *structural prior*. This high-quality prior then explicitly guides the main network, enabling it to recover richer and more accurate details.

Motivated by this limitation, we propose a fundamentally novel denoising paradigm that explicitly *separates* the denoising process into two distinct stages: *Structure Under-*

*Corresponding Author.

standing and *Guided Refinement*. Our key insight is that providing a stable and explicit structural prior significantly improves the network's denoising performance (see Fig. 1). To this end, our framework introduces an explicit step for learning a *structured representation* to serve as a high-quality prior for guidance.

This process begins by distilling a vocabulary of regularized, representative local *structural primitives* from the input features by means of a learned codebook (i.e., vector quantization). The output is a robust *structured representation*, which effectively regularizes the underlying geometric patterns to form a high-quality understanding of the local geometry. Subsequently, we use this stable and clear *structured representation* as an explicit, high-quality conditional signal to guide a powerful conditional attention network. This network can now perform subsequent feature fusion and refinement tasks based on a reliable *structural prior*.

The unique advantage of our method is that the subsequent feature refinement process is guided by clear, high-quality prior knowledge. Since the guidance signal itself (the *structured representation*) already contains judgments about the type of local geometry (e.g., plane, edge, or corner), the conditional attention network can more targetedly aggregate and enhance features, thereby achieving more precise preservation of true details and more thorough removal of noise.

Fundamentally, our work shifts the research focus in point cloud denoising from the process of denoising to the information that guides it. While prior work has concentrated on optimizing the algorithms for noise removal, our approach emphasizes the critical importance of the guidance signal itself. We demonstrate that by first learning and then leveraging a high-quality, structured representation of the geometry, a more robust and precise restoration can be achieved. This shift in perspective offers a conceptually novel and promising direction for tackling the long-standing challenge of point cloud denoising, distinguishing our paradigm from existing mainstream methods.

Our main contributions can be summarized as follows:

- We propose a novel point cloud denoising paradigm, the core of which is to explicitly guide the adaptive feature fusion process with a learned *structured representation*, addressing the limitation of guidance signals that lack explicit structural regularization in existing methods.
- We design a specific network architecture that effectively combines vector quantization (codebook learning) and conditional attention mechanisms, successfully verifying the effectiveness of our proposed *decouple-and-guide* paradigm.
- Extensive experiments on several benchmark datasets show that our method achieves state-of-the-art levels in both quantitative metrics and visual quality, especially demonstrating excellent capabilities in preserving complex geometric details, thus confirming the great potential and value of our method.

Related Work

Deep Learning for Point Cloud Denoising

Deep learning has emerged as the dominant paradigm for point cloud denoising, largely supplanting traditional methods (Fleishman, Drori, and Cohen-Or 2003; Alexa et al. 2001) that struggle with complex noise and fine details. Modern deep learning methods (Wang et al. 2025) can be broadly categorized into single-step and multi-step approaches. Although architecturally diverse, we analyze these methods from the perspective of their internal guidance mechanisms, which we identify as a common area with potential for improvement.

Single-step Denoising Methods Single-step methods aim to learn a direct mapping from a noisy point cloud to its clean counterpart. Early works like PointCleanNet (Rakotosaona et al. 2020) and PCDNF (Liu et al. 2023) predicted displacement vectors for noisy points. Subsequent methods introduced more sophisticated techniques. For instance, PointFilter (Zhang et al. 2021) incorporated a bilateral loss to regularize the learning process. Others have explored learning in latent space; PDFlow (Mao et al. 2022) uses normalizing flows to disentangle noise from features, while PD-LTS (Mao et al. 2024) leverages an invertible neural network to uncover clean latent codes. ASDN (Guo et al. 2025) introduced an adaptive stopping strategy to prevent over-smoothing.

Despite their different strategies, these methods share a common characteristic: their internal adaptive mechanisms—whether a simple MLP, a graph convolution, or a complex flow-based transformation—operate on features derived directly from the noisy input. Without an intermediate step to explicitly regularize these features into stable structural priors, the guidance signals generated can be volatile. This makes it challenging for the network to consistently distinguish fine-grained geometric details from noise, often leading to a trade-off between noise removal and detail preservation.

Multi-step Denoising Methods To overcome the limitations of single-step regression, such as the *regression-to-the-mean* effect, multi-step iterative methods have been proposed. These methods decompose the complex denoising task into a sequence of more manageable refinements (Liu et al. 2025). Inspired by generative modeling, score-based methods like ScoreDenoise (Luo and Hu 2021) train a network to estimate the score function of the noisy data distribution, and then iteratively update point positions along this score field. Similarly, diffusion-based approaches, such as P2P-Bridge (Vogel et al. 2024), model denoising as a learned stochastic process. Other works adopt an architectural approach; for instance, IterativePFN (de Silva Edirimuni et al. 2023; Zhou et al. 2025) employs a cascade of denoising modules, where each module refines the output of the previous one.

While these iterative approaches effectively mitigate challenges like the RTM effect, they still share a fundamental trait: the guidance for each step is reactively generated based on the current, partially-denoised state. For example,

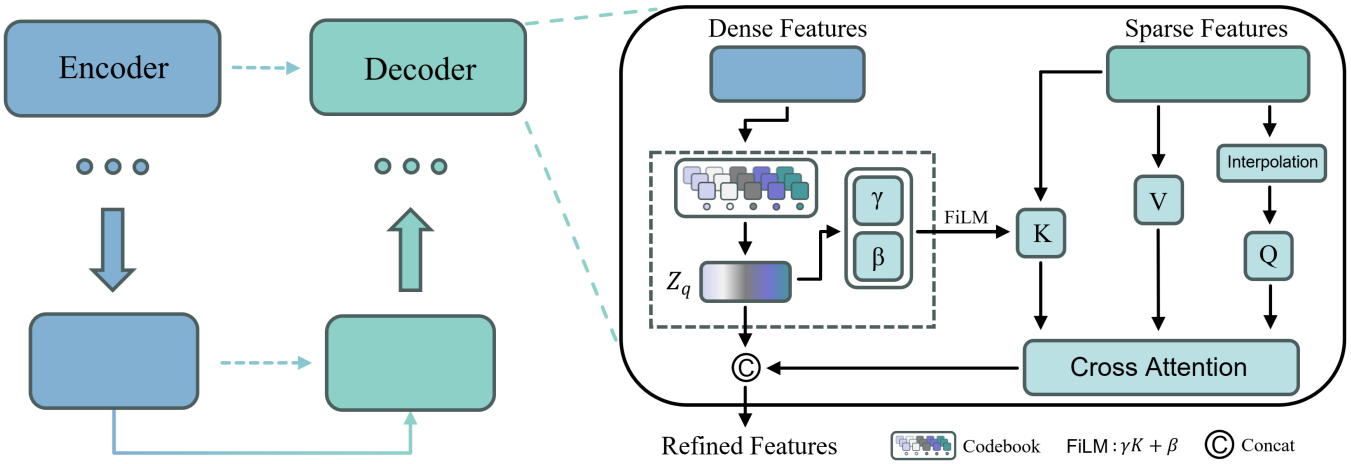

Figure 2: The overall architecture of our proposed denoising framework, which follows the *decouple–and–guide* paradigm. On the left, we show the global U-Net backbone with an encoder, a decoder, and skip connections. On the right, the magnified view details our Guided Upsampling Block. Dense features from the skip connection are first quantized by a Codebook (VQ) module, yielding a soft-assigned *structural prior* ($\mathbf{Z}_q$).. This prior serves as a conditional signal: a FiLM layer predicts (γ, β) from $\mathbf{Z}_q$ to modulate the Key vectors $\mathbf{K}$ (i.e., FiLM($\mathbf{Z}_q$): $\gamma\mathbf{K} + \beta$). The attention output is then fused with the *structural prior* via concatenation to form the Refined Features for the current decoding stage. By guiding every upsampling stage with a stable, learned understanding of local geometry, the network achieves robust noise removal while preserving fine-grained structures.

a score network estimates the gradient based on the current state X_t, and each module in a cascaded architecture operates on the output from the previous stage. Consequently, the guidance signal at each iteration, while progressively improving, lacks a stable, explicit structural reference. This can cause the propagation of errors from earlier stages and may limit the final recovery quality. Our work departs from this paradigm by first learning a stable structural prior, and then using this consistent, high-quality reference to guide the restoration process at each stage.

Vector Quantization for Structural Representation Learning

Vector Quantization (VQ), a classic technique revived by VQ-VAE (Van Den Oord, Vinyals et al. 2017), maps continuous features to a finite, discrete set of learned embeddings known as a codebook. Each *codeword* in this codebook acts as a learned prototype. In the context of point clouds, these codewords can be interpreted as canonical representations of local geometric patterns, or the *structural primitives* we conceptualized in our introduction. This mapping inherently performs regularization and implicit denoising by forcing noisy features to conform to a learned, standardized geometric form. While the success of VQ has inspired its application in 3D generative modeling, such as for autoregressive mesh generation in MeshGPT (Siddiqui et al. 2023), our work employs it from a fundamentally different perspective. Instead of for generation, we leverage VQ within a discriminative task to extract a high-quality conditional prior, directly addressing the challenge of generating stable guidance from dynamic, unregularized features.

Conditional Feature Modulation

To operationalize the guidance part of our framework, we employ a conditional feature modulation mechanism, realized through Feature-wise Linear Modulation (FiLM) (Perez et al. 2018). This technique learns to dynamically adjust network activations via an affine transformation derived from a conditioning signal. While FiLM is widely used to inject high-level semantic guidance (e.g., class labels) (Perez et al. 2018), our novelty lies not in the modulation mechanism itself, but in the nature of our conditioning signal. We are the first, to our knowledge, to use a robust, structured representation of local geometry, distilled via VQ, as the conditional input to modulate a denoising network, thereby directly leveraging a high-quality *structural prior* for feature refinement.

Method

Overview

As outlined in our introduction, the core philosophy of our proposed method is to decouple the complex denoising task into two distinct, sequential stages: (1) *Structure Understanding*, where the network learns to extract a robust *structural prior* from the noisy input, and (2) *Guided Refinement*, where this learned prior is used to explicitly guide the detailed restoration of the point cloud. This paradigm shifts the focus from direct, end-to-end regression to a more deliberate, knowledge-guided process. An overview of our framework is presented in Figure 2.

Formally, let $\mathbf{X}_{\text{noisy}} \in \mathbb{R}^{n\times 3}$ be a local patch extracted from a noisy point cloud, and $\mathbf{X}_{\text{clean}} \in \mathbb{R}^{n\times 3}$ be its corresponding ground truth. Our denoising framework is an end-to-end model, denoted by F, which maps the noisy input to its re-

Algorithm 1: Codebook Module: Forward Pass and EMA Update

Input: A batch of feature vectors $\mathbf{Z}_e = \{\mathbf{z}_e^{(i)}\}_{i=1}^n$.
Parameters: Learnable codebook $\mathbf{E} = \{\mathbf{e}_k\}_{k=1}^K$, fixed temperature τ=0.1 , EMA decay μ.
Output: Quantized features $\mathbf{Z}_q$ (with STE for backward pass).

for $i = 1$ **to** n **do**
    Compute cosine similarity: $s_{i,k} \leftarrow \frac{(\mathbf{z}_e^{(i)})^\top \mathbf{e}_k}{\|\mathbf{z}_e^{(i)}\| \|\mathbf{e}_k\|}$.
    Compute weights: $w_{i,k} \leftarrow \frac{\exp(s_{i,k}/\tau)}{\sum_{j=1}^K \exp(s_{i,j}/\tau)+\varepsilon}$.
    Compute quantized vector: $\mathbf{z}_q^{(i)} \leftarrow \sum_{k=1}^K w_{i,k}\, \mathbf{e}_k$.
end for
Assemble the set of true quantized vectors: $\mathbf{Z}_q \leftarrow \{\mathbf{z}_q^{(i)}\}_{i=1}^n$.

if training is active **then**
    for $k = 1$ **to** K **do**
        Compute weighted average of inputs: $\bar{\mathbf{z}}_k \leftarrow \frac{\sum_i w_{i,k} \mathbf{z}_e^{(i)}}{\sum_i w_{i,k}+\varepsilon}$.
        Update codeword via EMA: $\mathbf{e}_k \leftarrow \mu\, \mathbf{e}_k + (1-\mu)\bar{\mathbf{z}}_k$.
    end for
end if

▷ The returned value is numerically equal to $\mathbf{Z}_q$ in the forward pass.
return $\mathbf{Z}_e + \mathrm{sg}(\mathbf{Z}_q - \mathbf{Z}_e)$ ▷ sg is the stop-gradient operator.

stored counterpart:

$$\widehat{\mathbf{X}}_{\text{clean}} = F\left(\mathbf{X}_{\text{noisy}}\right) \tag{1}$$

Our model F is realized using a U-Net backbone. The key innovation lies in how this U-Net architecture internally performs the two conceptual stages of our paradigm. Specifically, the *Structure Understanding* stage occurs within each upsampling block of the decoder: dense features passed from the corresponding encoder stage via a skip connection are first processed by a vector quantization module to produce a soft-assigned structural prior, $\mathbf{Z}_q \in \mathbb{R}^{n\times d}$. This prior represents a regularized and stable understanding of the local geometry. Immediately following, in the *Guided Refinement* stage, this high-quality prior $\mathbf{Z}_q$ is utilized as a dynamic conditional signal to actively modulate the subsequent feature fusion and refinement process within the very same upsampling block. This design ensures that the restoration process is explicitly guided by a stable structural representation. The entire framework is trained end-to-end, with the details of the internal guidance mechanism elaborated in the following section.

Notations Let n denote the number of points in a local patch. Since each point typically yields one feature vector, we use $N = n$ for the number of vectors in a set. The superscript l will be used to index the levels of the U-Net hierarchy where necessary, but is omitted for brevity when discussing a generic layer.

Structural Prior Extraction with Vector Quantization

Local Geometric Feature Encoding The feature encoding process begins by projecting the input patch $\mathbf{X}_{\text{noisy}}$ into a higher-dimensional feature space via an MLP. These initial features are then processed by the U-Net encoder, which consists of a stack of four hierarchical encoder blocks. Each block aggregates local neighborhood information to learn richer geometric features and then downsamples the point set via Farthest Point Sampling (FPS). This process yields a series of multi-scale feature sets, $\{\mathbf{Z}_e^l\}$, which are passed to the corresponding decoder stages via skip connections.

Vector-Quantization Codebook The core of our *Structure Understanding* stage is the Codebook module, placed within each of the four *Upsampling* blocks in the decoder. At each stage, it processes the dense feature map arriving from the corresponding skip connection, which we denote as a set of n feature vectors $\mathbf{Z}_e = \{\mathbf{z}_e^{(i)}\}_{i=1}^n$. The Codebook module contains a learnable codebook, defined as a set of K embedding vectors (or codewords), $\mathbf{E} = \{\mathbf{e}_k \in \mathbb{R}^d\}_{k=1}^K$, where K is the codebook size and d is the feature dimensionality.

Our module employs a soft quantization scheme. For each input vector $\mathbf{z}_e^{(i)}$, we compute its cosine similarity $s_{i,k}$ with every codeword $\mathbf{e}_k$:

$$s_{i,k} = \frac{(\mathbf{z}_e^{(i)})^\top \mathbf{e}_k}{\|\mathbf{z}_e^{(i)}\| \|\mathbf{e}_k\|}. \tag{2}$$

These scores are converted into weights via a temperature-controlled softmax, where τ is a fixed temperature hyper-parameter:

$$w_{i,k} = \frac{\exp(s_{i,k}/\tau)}{\sum_{j=1}^K \exp(s_{i,j}/\tau)+\varepsilon}, \tag{3}$$

where a small ε is added for numerical stability. The final quantized feature $\mathbf{z}_q^{(i)}$ is the weighted average of all codewords, $\mathbf{z}_q^{(i)} = \sum_{k=1}^K w_{i,k}\mathbf{e}_k$. The set of all such vectors forms the *structural prior* $\mathbf{Z}_q$. This soft assignment leads to smoother codebook updates and helps alleviate the issue of codeword collapse.

The training of our framework is fully end-to-end. Gradients with respect to the feature encoder are passed through the quantization operation by means of a Straight-Through Estimator (STE) (Bengio, Léonard, and Courville 2013). In backpropagation, the STE treats the quantization as an identity map, allowing the gradient from the main reconstruction loss to pass through it unmodified. It is worth noting that our approach differs from the original VQ-VAE framework in its training objective. We found that for our discriminative denoising task, introducing an additional commitment loss was not necessary and did not consistently benefit performance. Therefore, our model is trained purely with the final

Points Sparsity	10K points						50K points					
Noise Level	1% noise		2% noise		2.5% noise		1% noise		2% noise		2.5% noise	
Method	*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
GDPNet	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	1.289	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
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
PD-LTS	<u>1.825</u>	0.496	2.567	0.824	2.849	1.034	<u>0.496</u>	0.306	0.706	0.447	0.936	0.605
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
3DMambaIPF	1.989	<u>0.477</u>	2.995	0.803	3.262	0.992	0.589	<u>0.291</u>	0.755	<u>0.405</u>	0.928	**0.531**
ASDN	1.836	0.490	<u>2.508</u>	<u>0.776</u>	<u>2.697</u>	<u>0.961</u>	0.498	0.308	<u>0.686</u>	0.448	**0.850**	<u>0.575</u>
Ours	**1.797**	**0.469**	**2.430**	**0.729**	**2.626**	**0.904**	**0.479**	**0.286**	**0.615**	**0.404**	<u>0.881</u>	0.604

Table 1: Quantitative evaluation on PUNet (Yu et al. 2018) with CD and P2M metrics ($\times 10^{-4}$). The best and second-best results are highlighted in bold and underlined, respectively.

reconstruction loss, where the codebook embeddings $\mathbf{e}_k$ are updated exclusively using an Exponential Moving Average (EMA) of the incoming features. This streamlined process ensures the gradient path to the encoder remains uninterrupted while the codebook is kept stable, effectively realizing our *decouple-and-guide* philosophy. The entire procedure is detailed in Algorithm 1.

Guided Denoising with Conditional Attention

Once the *structural prior* $\mathbf{Z}_q$ is extracted from the dense skip-connection features, the next stage is to leverage this high-quality information to guide the feature refinement process. This is the core of our *Guided Refinement* phase and is realized within each Guided Upsampling Block in the decoder. This block takes two main streams of information as input: the dense features $\mathbf{Z}_e$ from the skip connection (from which $\mathbf{Z}_q$ is derived) and the sparse features, which we denote as $\mathbf{Z}_s$, from the preceding, deeper decoder layer. The central mechanism for this guided process is a conditional cross-attention layer.

Conditional Cross-Attention. Our conditional attention mechanism is designed to dynamically fuse information from the sparse feature stream under the explicit guidance of the *structural prior*. First, we generate the Query ($\mathbf{Q}$), Key ($\mathbf{K}$), and Value ($\mathbf{V}$) vectors. The Key and Value vectors are derived directly from the sparse features $\mathbf{Z}_s$ via linear projections. The Query vectors are derived from an interpolated version of these sparse features $\mathbf{Z}_s$, passed through a separate linear projection.

Next, we perform the crucial guidance injection step using a Feature-wise Linear Modulation (FiLM) layer (Perez et al. 2018). The *structural prior* $\mathbf{Z}_q$ serves as the conditioning signal. It is passed through the FiLM layer—a small MLP—to predict a scaling vector γ and a bias vector β, whose dimensions are broadcastable to the Key vectors.

$$(\gamma, \beta) = \text{FiLM}(\mathbf{Z}_q). \tag{4}$$

These parameters perform an element-wise affine transformation on the Key vectors, yielding the modulated keys $\mathbf{K}'$:

$$\mathbf{K}' = \gamma \odot \mathbf{K} + \beta. \tag{5}$$

This modulation dynamically re-weights the information retrieval space. We deliberately modulate only the Key vectors to guide the attention's selection criteria, ensuring that the aggregated Value vectors remain pristine representations of the source information. Finally, a standard scaled dot-product attention operation (Vaswani et al. 2017) is performed:

$$\mathbf{Z}_{\text{enhance}} = \text{Attention}(\mathbf{Q}, \mathbf{K}', \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}(\mathbf{K}')^{\mathsf{T}}}{\sqrt{d_k}}\right)\mathbf{V}. \tag{6}$$

The output of this attention mechanism is an enhanced feature representation, where d_k is the dimension of each attention head $\mathbf{Z}_{\text{enhance}}$.

Dual-Stream Feature Fusion. The output of the conditional attention, $\mathbf{Z}_{\text{enhance}}$, represents the dynamically refined contextual information. To form the final output of the Guided Upsampling Block, we fuse this with the stable structural information. Specifically, we concatenate the enhanced features with the *structural prior* that guided them:

$$\mathbf{Z}_{\text{fused}} = \text{Concat}\left(\mathbf{Z}_{\text{enhance}}, \mathbf{Z}_q\right). \tag{7}$$

This fused representation, containing both the robust learned pattern and the contextually refined details, is then processed by a final MLP to produce the output features for the current decoder stage.

Training Objectives

Our framework is trained end-to-end by minimizing a reconstruction loss between the denoised output and the ground-truth point cloud. For this purpose, we employ the recently proposed InfoCD loss (Lin et al. 2023), an information-theoretic variant of Chamfer Distance that quantifies the distributional alignment between two point sets.

The InfoCD loss is realized through a symmetric formulation. It first defines a one-sided term, $\mathscr{L}_{\text{single}}(\mathbf{X} \to \mathbf{Y})$, which measures how well the point set $\mathbf{Y}$ represents the distribution of point set $\mathbf{X}$:

$$\mathscr{L}_{\text{single}}(\mathbf{X} \to \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} \left(-\log \frac{\exp(-\alpha \cdot d(\mathbf{x}_i, \mathbf{Y}))}{\sum_{j=1}^{n} \exp(-\alpha \cdot d(\mathbf{x}_j, \mathbf{Y}))} \right), \tag{8}$$

where $d(\mathbf{p}, \mathbf{P}) = \min_{\mathbf{q} \in \mathbf{P}} \|\mathbf{p} - \mathbf{q}\|_2^2$ is the squared Euclidean distance from a point to its nearest neighbor in a point set, and α is a scaling factor.

Our final reconstruction loss, $\mathscr{L}_{\text{recon}}$, is the symmetric average of this term. For brevity in the following equation, let $\widehat{\mathbf{X}} = \widehat{\mathbf{X}}_{\text{clean}}$ and $\mathbf{X} = \mathbf{X}_{\text{clean}}$. The loss is then defined as:

$$\mathscr{L}_{\text{recon}} = \frac{1}{2} \left(\mathscr{L}_{\text{single}}(\widehat{\mathbf{X}} \to \mathbf{X}) + \mathscr{L}_{\text{single}}(\mathbf{X} \to \widehat{\mathbf{X}}) \right). \tag{9}$$

This formulation encourages each predicted point to be a plausible member of the entire ground-truth distribution. The Codebook module is trained implicitly: gradients from this reconstruction loss are passed to the feature encoder via the STE, while the codebook itself is updated via EMA.

Points Sparsity	**10K points**			
Noise Level	**1% noise**		**2% noise**	
Method	***CD*↓**	***P2M*↓**	***CD*↓**	***P2M*↓**
StraightPCF	<u>2.791</u>	0.867	4.078	1.285
PD-LTS	2.843	<u>0.823</u>	4.151	1.265
P2P-Bridge	2.882	0.967	4.476	1.361
ASDN	2.926	0.854	<u>4.023</u>	<u>1.256</u>
Ours	**2.701**	**0.756**	**3.962**	**1.159**

Table 2: Quantitative evaluation on PCNet (Rakotosaona et al. 2020) with CD and P2M metrics ($\times 10^{-4}$) for 10K points. The best and second-best results are highlighted in bold and underlined, respectively.

Experimental Results

Performance on Synthetic Data We first evaluate our method on two synthetic benchmarks to assess its core performance and its ability to generalize to unseen geometries.

Our primary evaluation is conducted on the PUNet dataset (Yu et al. 2018) under various levels of isotropic Gaussian noise. As shown in Table 1, our framework establishes a new state-of-the-art, achieving superior or highly competitive performance across all evaluated scenarios. The performance gap is particularly evident under high noise conditions (e.g., 2.5%), where many competing methods suffer from significant performance degradation, while our method maintains its robustness. The qualitative results in Figure 3 visually corroborate these quantitative findings, illustrating our method's superior capability in preserving intricate geometric details.

To further validate the generalization capability of our learned model, we test it directly on the PCNet dataset (Rakotosaona et al. 2020) without any fine-tuning. The quantitative results, presented in Table 2, again demonstrate a strong and consistent performance, with our method outperforming the majority of competing approaches.

Points Sparsity		**10K points**			
Noise Level		**1% noise**		**2% noise**	
Noise Type	**Method**	***CD*↓**	***P2M*↓**	***CD*↓**	***P2M*↓**
Ani	**StraightPCF**	1.923	0.529	2.753	0.999
	PD-LTS	<u>1.833</u>	0.503	2.630	0.878
	P2P-Bridge	2.312	0.708	3.230	1.151
	ASDN	1.847	<u>0.497</u>	<u>2.592</u>	<u>0.853</u>
	Ours	**1.806**	**0.477**	**2.497**	**0.786**
Lap	**StraightPCF**	2.249	0.665	**3.051**	<u>1.241</u>
	PD-LTS	2.158	0.624	3.090	**1.232**
	P2P-Bridge	2.637	0.855	3.908	1.727
	ASDN	<u>2.145</u>	<u>0.598</u>	<u>3.080</u>	1.284
	Ours	**2.098**	**0.573**	3.226	1.419
Uni	**StraightPCF**	0.673	0.389	1.877	0.567
	PD-LTS	**0.635**	**0.368**	<u>1.787</u>	0.477
	P2P-Bridge	1.084	0.560	2.293	0.662
	ASDN	0.700	0.385	1.791	<u>0.471</u>
	Ours	<u>0.672</u>	<u>0.376</u>	**1.760**	**0.456**
Dis	**StraightPCF**	<u>0.678</u>	0.391	1.711	0.556
	PD-LTS	**0.643**	**0.371**	<u>1.596</u>	0.475
	P2P-Bridge	1.087	0.580	1.982	0.676
	ASDN	0.709	0.386	1.635	<u>0.472</u>
	Ours	0.680	<u>0.375</u>	**1.595**	**0.462**

Table 3: Numerical evaluation of different noise patterns: Anisotropic Gaussian (Ani), Laplace (Lap), Uniform (Uni), and Discrete (Dis) for 10K points. The unit of CD and P2M is 10^{-4}.

Robustness to Different Noise Types To further assess the robustness of our framework, we extend our evaluation on the PUNet dataset to a wider variety of challenging noise distributions beyond the standard isotropic Gaussian noise Table 1. We conduct comprehensive comparisons under four other challenging noise types: anisotropic Gaussian, Laplace, uniform, and discrete noise.

The detailed quantitative results are presented in Table 3. As these results indicate, our method consistently achieves superior or highly competitive performance across all tested noise distributions and levels. This demonstrates that our *decouple-and-guide* paradigm is not over-fitted to a specific

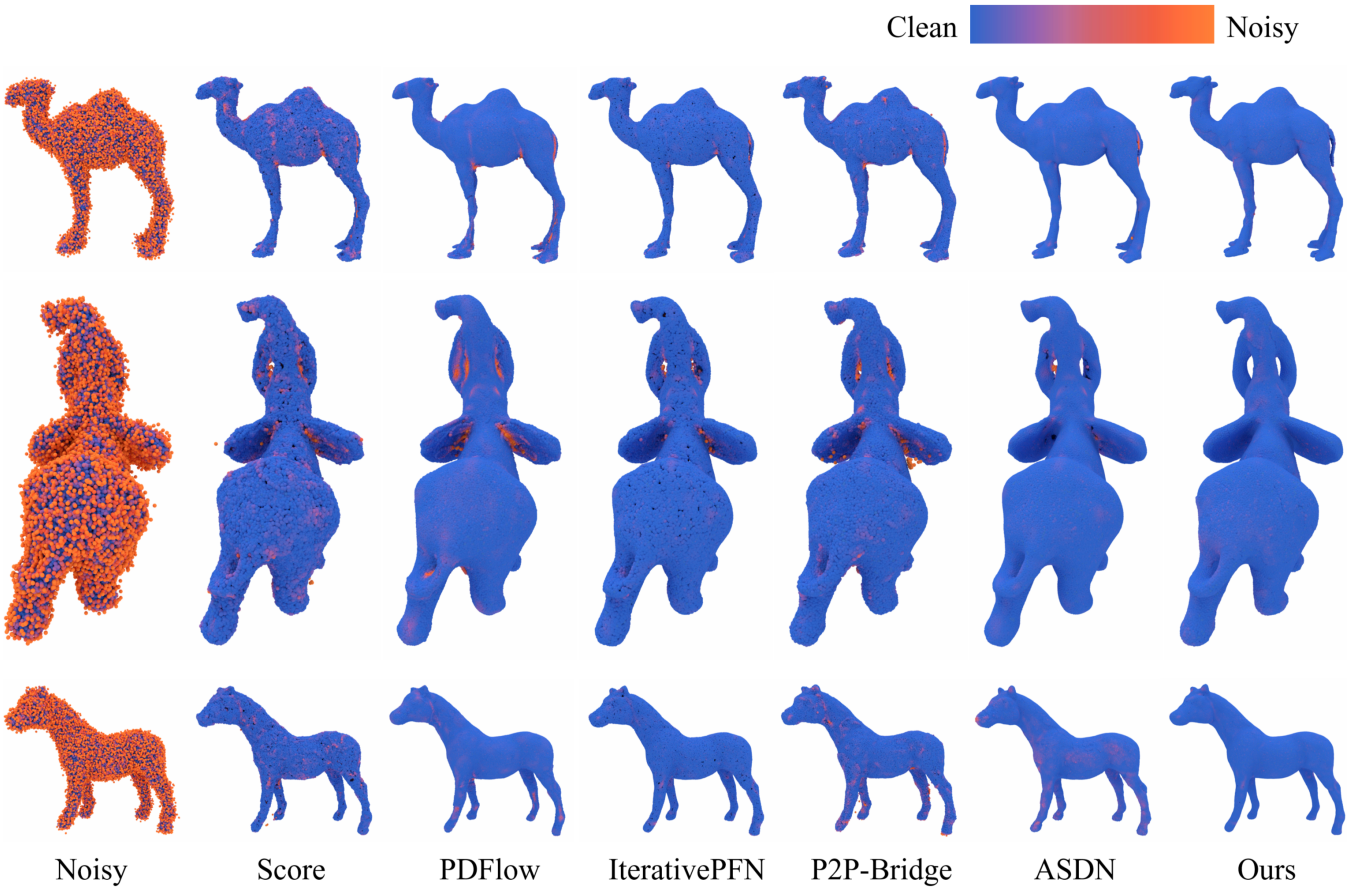

Figure 3: Visual results of point-wise P2M distance for shapes at 50K resolution with 2% Gaussian noise.

noise model. By first learning to recognize and represent underlying geometric structures via the codebook, our model is less sensitive to the statistical properties of the noise itself, enabling it to maintain strong denoising capabilities in a variety of challenging scenarios.

Ablation Studies

To validate our core design choices, we conduct a series of ablation studies on the PUNet dataset, with the results summarized in Table 4. The step-wise analysis provides a clear chain of evidence for our *decouple-and-guide* paradigm. Starting from a baseline U-Net, we first observe that introducing the VQ-distilled prior yields a significant performance gain. The subsequent replacement of simple feature concatenation with our FiLM-based conditional guidance brings further improvement. Finally, our full model, which incorporates a final dual-stream fusion of both the structural prior and the enhanced features, achieves the best results.

Model Variant	***CD*↓**	***P2M*↓**
(a) Baseline	1.99 8	0.495
(b) + VQ (w/ Concat Fusion)	1.885	0.483
(c) + VQ & FiLM (w/o Final Fusion)	1.815	0.476
(d) Ours (Full Model)	**1.797**	**0.469**

Table 4: Ablation studies on the effectiveness of our core components. We evaluate on the PUNet dataset (10k points, 1% Gaussian noise). Lower CD and P2M values ($\times 10^{-4}$) are better. (a) is a baseline U-Net where our Guided Upsampling Block is replaced by a standard self-attention block. (b) adds our VQ module but uses simple concatenation for fusion instead of FiLM. (c) further incorporates FiLM-based guidance but omits the final fusion with the prior. (d) is our full proposed model.

Conclusion

We have introduced a point cloud denoising framework built on a *decouple–and–guide* paradigm. First, a vector-quantized codebook distills a soft-assigned *structural prior* from noisy features; this prior then conditionally steers a U-Net decoder via FiLM-modulated cross-attention, allowing the network to suppress noise while preserving fine-grained geometry. Comprehensive experiments confirm the effectiveness of this design. Future work could involve a deeper analysis of the learned structural priors to better understand their geometric representations, further unlocking the potential of such prior-guided schemes.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. We gratefully acknowledge the support from the National Key R&D Program of China (Grant No. 2024YFA1016300), the National Natural Science Foundation of China (Grant No. 12471359), and the Ministry of Education, Singapore, under its Academic Research Fund Grant (RT19/22).

References

Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; and Silva, C. T. 2001. Point set surfaces. In *Proceedings Visualization, 2001. VIS'01.*, 21–29. IEEE.

Bengio, Y.; Léonard, N.; and Courville, A. C. 2013. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *CoRR*, abs/1308.3432.

de Silva Edirimuni, D.; Lu, X.; Li, G.; Wei, L.; Robles-Kelly, A.; and Li, H. 2024. StraightPCF: Straight Point Cloud Filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20721–20730.

de Silva Edirimuni, D.; Lu, X.; Shao, Z.; Li, G.; Robles-Kelly, A.; and He, Y. 2023. IterativePFN: True Iterative Point Cloud Filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13530–13539.

Fleishman, S.; Drori, I.; and Cohen-Or, D. 2003. Bilateral mesh denoising. In *ACM SIGGRAPH 2003 Papers*, 950–953.

Guo, C.; Zhou, W.; Liu, Z.; and He, Y. 2025. You Should Learn to Stop Denoising on Point Clouds in Advance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39: 3212–3219.

Lin, F.; Yue, Y.; Zhang, Z.; Hou, S.; Yamada, K.; Kolachalama, V. B.; and Saligrama, V. 2023. InfoCD: A Contrastive Chamfer Distance Loss for Point Cloud Completion. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Liu, Z.; Huang, Z.; Pan, M.; and He, Y. 2025. Deterministic Point Cloud Diffusion for Denoising. *IEEE Trans. Vis. Comput. Graph.*, 1–14.

Liu, Z.; Zhao, Y.; Zhan, S.; Liu, Y.; Chen, R.; and He, Y. 2023. PCDNF: Revisiting Learning-based Point Cloud Denoising via Joint Normal Filtering. *IEEE Trans. Vis. Comput. Graph.*, 30(8): 5419–5436.

Luo, S.; and Hu, W. 2021. Score-based point cloud denoising. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 4583–4592.

Mao, A.; Du, Z.; Wen, Y.-H.; Xuan, J.; and Liu, Y.-J. 2022. PD-Flow: A Point Cloud Denoising Framework with Normalizing Flows. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 13663.

Mao, A.; Yan, B.; Ma, Z.; and He, Y. 2024. Denoising Point Clouds in Latent Space via Graph Convolution and Invertible Neural Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5768–5777.

Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Rakotosaona, M.-J.; La Barbera, V.; Guerrero, P.; Mitra, N. J.; and Ovsjanikov, M. 2020. PointCleanNet: Learning to denoise and remove outliers from dense point clouds. *Comput. Graph. Forum*, 39(1): 185–203.

Siddiqui, Y.; Alliegro, A.; Artemov, A.; Tommasi, T.; Sirigatti, D.; Rosov, V.; Dai, A.; and Nießner, M. 2023. MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers. *arXiv preprint arXiv:2311.15475*.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*, volume 30, 6000–6010.

Vogel, M.; Tateno, K.; Pollefeys, M.; Tombari, F.; Rakotosaona, M.-J.; and Engelmann, F. 2024. P2P-Bridge: Diffusion Bridges for 3D Point Cloud Denoising. In *European Conference on Computer Vision (ECCV)*.

Wang, J.; Fei, B.; de Silva Edirimuni, D.; Liu, Z.; He, Y.; and Lu, X. 2025. A Survey of Deep Learning-based Point Cloud Denoising. *CoRR*.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.*, 38(5): 1–12.

Yu, L.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2018. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2790–2799.

Zhang, D.; Lu, X.; Qin, H.; and He, Y. 2021. Pointfilter: Point cloud filtering via encoder-decoder modeling. *IEEE Trans. Vis. Comput. Graph.*, 27(3): 2015–2027.

Zhou, Q.; Yang, W.; Fei, B.; Xu, J.; Zhang, R.; Liu, K.; Luo, Y.; and He, Y. 2025. 3DMambaIPF: A State Space Model for Iterative Point Cloud Filtering via Differentiable Rendering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 10843–10851.