Sparse Point Cloud Completion via Decoupled Geometry Completion and Detail Refinement

Sheng Liu, Meng Wang, Ruihui Li*, Fan Wu, Nan Hu, Zhuo Tang

College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China

Abstract

In the canonical 3D task of point cloud completion, sparser partial inputs introduce greater ambiguity in geometric and semantic cues, often resulting in floating artifacts or geometric collapse during reconstruction. Existing methods aim to generate a single dense point set that both aligns with the surface and exhibits uniform distribution, yet entangling these objectives in a single network hinders clear optimization. Viewing this as a multi-objective problem, we propose a decoupled two-stage architecture: a coarse generator followed by a dense refiner. The coarse generator extracts multi-scale features and predicts a surface-aligned coarse shape, while the dense refiner enhances local fidelity via a novel upsampling module that captures both local patterns and global dependencies. Extensive experiments on benchmark datasets demonstrate that our method significantly outperforms state-of-the-art approaches, especially when dealing with highly sparse inputs.

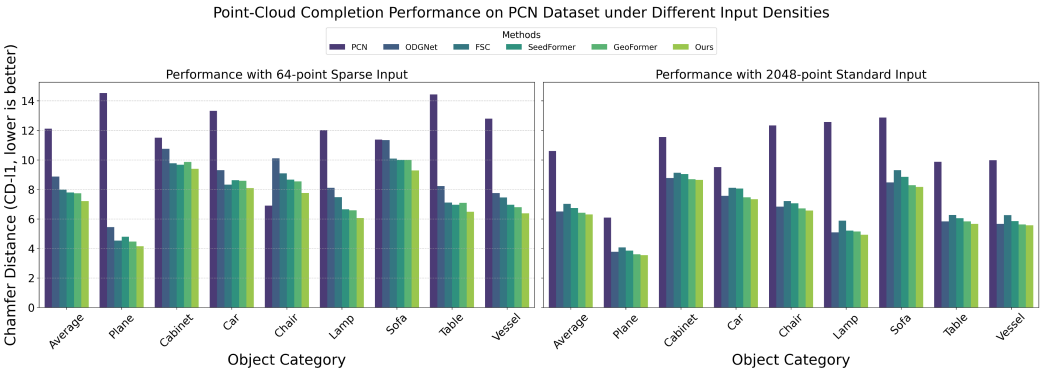
Keywords:
3D visions, Point cloud completion, Geometric Modeling

1. Introduction

Point clouds serve as a fundamental representation for human perception and interpretation of three-dimensional (3D) environments. They are typically captured through various sensing modalities, including time-of-flight (ToF) cameras,

*Corresponding author.

Email addresses: liusheng23@hnu.edu.cn (Sheng Liu), willem@hnu.edu.cn (Meng Wang), liruihui@hnu.edu.cn (Ruihui Li), wufan@hnu.edu.cn (Fan Wu), hunan5@hnu.edu.cn (Nan Hu), ztang@hnu.edu.cn (Zhuo Tang)



stereo vision systems, and LiDAR scanners. However, due to the inherent limitations of these sensors—such as restricted viewpoints and occlusions—only partial observations of an object's surface are usually obtained. Consequently, reconstructing complete 3D shapes from incomplete point clouds has become a crucial task in 3D computer vision. In recent years, point cloud completion has garnered increasing attention and remains a vibrant area of ongoing research.

This paper focuses on a challenging subtask of point cloud completion, which is to recover complete geometric structures from highly sparse point cloud structures. The fewer the input defective points, the more ambiguous the geometric and semantic information they contain, making it particularly difficult to restore complete and accurate geometric structures.

Benefiting from the rapid advancement of deep learning techniques and the emergence of high-quality point cloud completion datasets[1, 2, 3, 4, 5, 6], point cloud completion has witnessed the development of various robust solutions[7, 8, 9, 10, 11, 12, 13, 14, 15], significantly outperforming traditional completion approaches. We observe that most existing methods[1, 7, 8, 9, 10, 16, 17, 3] follow a common paradigm: a single-branch encoder first maps the partial point cloud into a latent feature space, followed by iterative application of a single decoder module to reconstruct the complete point cloud. However, when the number of input points is severely limited, the extracted features tend to be ambiguous. In such cases, the completion module not only needs to infer and generate dense points from sparse input, but also must ensure that the generated points are uniformly distributed over the object's surface. It is challenging for a single network to simultaneously fulfill these requirements. Consequently, these frameworks often struggle under extremely sparse input conditions (e.g, 64 points), leading to incomplete reconstructions or the presence of floating artifacts.

After revisiting the sparse point cloud completion task, we propose decomposing it into two sub-tasks: (1) generating a coarse but surface-aligned completion, and (2) refining the coarse output to better fit local details and suppress floating artifacts. To this end, we design a cascaded network composed of a coarse generator and a dense refiner to sequentially achieve these sub-goals. Specifically, the coarse generator consists of two parallel branches: a Contextual Branch that captures global geometric information, and a Detail-oriented Branch that extracts local fine-grained features. These multi-scale features are then fused and passed through SDG blocks proposed in [18] to generate a coarse point cloud. The dense refiner, following a similar architecture to existing completion frameworks, re-encodes the coarse output and progressively decodes it from latent space into a dense and complete point cloud. The overall loss is computed by comparing the two-stage outputs with ground truth at multiple scales, enforcing a balanced optimization between local shape structure and fine geometric details.

Compared with existing methods, our decoupled cascaded pipeline assigns smaller and more specific tasks to each sub-network, enabling them to focus on their respective objectives. Moreover, the cascade structure allows mutual enhancement between the two stages during training, leading to substantial performance improvements. Extensive experimental results demonstrate that our method outperforms state-of-the-art approaches on several public benchmarks for sparse point cloud completion.

The main contributions of this work are summarized as follows:

- We propose a novel two-stage decoupled architecture for sparse point cloud completion, which disentangles the task into coarse geometry generation and fine detail refinement. This approach effectively mitigates issues like geometric collapse and floating artifacts common in single-stage methods.
- We design a coarse generator with a dual-branch encoder that captures both global contextual information and fine-grained local details, ensuring the generation of a structurally sound geometric scaffold.
- We introduce a dense refiner equipped with a novel upsampling module that progressively enhances point cloud density and refines local surface details while maintaining global shape consistency.
- Extensive experiments on benchmark datasets demonstrate that our method significantly outperforms state-of-the-art approaches, particularly in the challenging scenario of completing point clouds from highly sparse inputs.

2. Related Works

2.1. Point Cloud Learning

Point cloud learning has attracted growing attention in recent years due to its importance in 3D computer vision, robotics, and autonomous driving.

Unlike images, point clouds are unordered, sparse, and irregular, which poses unique challenges for representation learning. To address these, a variety of methods, particularly supervised and self-supervised learning techniques, have been proposed. Supervised learning has traditionally been the dominant paradigm, enabling precise feature extraction through labeled data. Common architectures include pointwise MLPs, hierarchical models, graph-based methods, and transformer-based designs. These models typically rely on feedforward or sequential processing to capture spatial context and structural patterns.

However, obtaining large-scale annotated 3D datasets remains a bottleneck, especially for tasks like point cloud completion that require detailed geometric supervision. To mitigate this, self-supervised learning (SSL) has emerged as a promising alternative. SSL methods design proxy tasks that allow models to learn geometric and contextual features from unlabeled data, often with direct relevance to completion. For example, Huang et al. proposed STRL, which learns 3D representations from temporally correlated frames [19], while Wang et al.'s OcCo masks regions of point clouds and trains the model to reconstruct them from occluded views [20]. Mixing and Disentangling (MD) [21] encourages the network to recover original structures from composite shapes, reinforcing disentangled spatial reasoning.

Recent advances have focused on generative SSL frameworks tailored for 3D reconstruction tasks. PointMAE [22] and Point-M2AE [23] adopt masked autoencoders to learn reconstruction-oriented features, while PointGPT [24] formulates auto-regressive point generation as pretraining. These approaches have shown strong transferability to downstream tasks like shape completion, making them particularly relevant for this work.

In summary, the evolution from supervised to generative self-supervised learning has laid a robust foundation for point cloud completion by enabling high-quality feature learning without extensive annotations.

Although these approaches have laid a solid foundation for feature representation, how to effectively leverage such powerful representations within a generative framework specifically tailored for extreme sparsity remains an open question—one that constitutes the core contribution of this work.

2.2. Point Cloud Completion

The goal of point cloud completion is to recover a fine-grained and globally consistent complete point cloud from a scanned partial observation. The subtask addressed in this paper—recovering the full geometry from an extremely sparse input—represents a more challenging variant of the canonical completion problem. In this subsection, we briefly review the recent advances in point cloud completion.

Point cloud completion algorithms based on deep learning can be categorized into three types. The first type is voxel-based methods[25, 26, 7], which can effectively and directly handle large-scale point cloud data. However, for the completion task, the incomplete point cloud information requires high-granularity semantic and geometric features for completion. Moreover, voxel-based networks are prone to the issue of a large number of zero voxel values occupying memory, leading to redundant voxel space, significant memory consumption, and computational burden.

The second type is view-based methods[27, 28, 29], which can obtain richer and more comprehensive perspective information, and mitigate the impact of noise and outliers. The success of 2D convolutional neural networks has inspired many researchers to adopt projection-based approaches for point cloud processing, where multi-view representations of point clouds are generated and subsequently fused to enhance the performance of point cloud completion networks. Such methods can obtain different information from multi-angle images. Their performance depends on the angles and the number of views. Moreover, the geometric information within the point cloud is prone to folding during the projection stage, which affects the accuracy of the point cloud completion network.

The third type is point-based methods. Specifically, methods based on MLP have a simple structure and can clearly predict the approximate shape of the complete point cloud, but they tend to ignore the topological structure of objects. Graph convolution-based methods can effectively deal with the topological structure and geometric features of point cloud data, but they are sensitive to noise and other outliers, leading to incorrect completion results. Moreover, the network size is relatively large and the number of parameters is high. GAN-based methods[30] do not require additional prior knowledge and can automatically learn and generate realistic geometric shapes and details. However, the training process is relatively complex and requires higher computational power. Transformer-based methods[16, 31, 32, 33] can capture global feature correlations in point clouds and handle point clouds of different densities and sizes. However, they are not

good at representing the position information of point clouds, which can lead to an excessive number of parameters or heavy computational load.

While these methods have achieved promising results on standard benchmarks, most assume moderately dense input (e.g., 2048 points) and struggle when faced with extreme sparsity (e.g., 64 points). Under such conditions, existing architectures tend to generate ambiguous or floating structures due to entangled optimization objectives. In contrast to these approaches, our work provides a novel perspective on completing shapes under extreme sparsity by explicitly disentangling coarse-shape generation from detail refinement.

3. Methods

3.1. Overview

In this section, we describe the structure and implementation details of our framework. Our two-stage pipeline consists of a coarse generator and a dense refiner, as shown in Fig. 1. We will next introduce the implementation details and key ideas behind this approach.

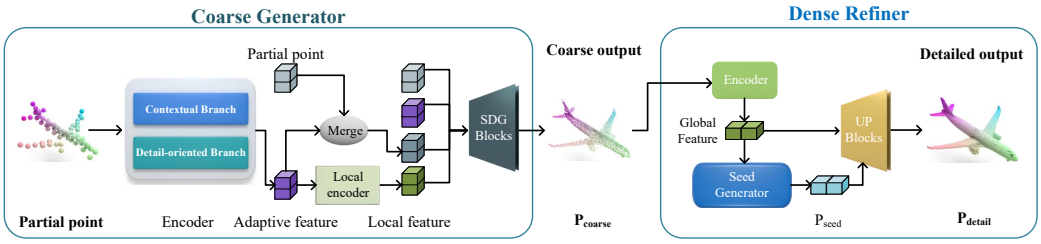


Figure 1: Overview of our two-stage point cloud completion framework. **(Left) Coarse Generator:** The input partial point cloud is encoded by a **Dual-Branch Encoder** that separately captures global contextual geometry and fine-grained local details. These representations are adaptively fused via cross-attention to form an expressive feature that drives the Self-structure Dual-Generator (SDG) Blocks, generating a structurally consistent coarse prediction (P_{coarse}). **(Right) Dense Refiner:** Guided by the global feature, a Seed Generator first produces an initial skeleton (P_{seed}). Then, a cascade of **Upsample Blocks**—each equipped with Multi-Scale Cross-Attention (MSCA) to enable cross-resolution feature interaction across different point densities—progressively reconstructs the dense and detailed output (P_{detail}).

3.2. Coarse Generator

3.2.1. Dual-branch Point Cloud Encoder

In sparse point cloud completion, the first step is to recover geometric information as accurately as possible from highly sparse inputs, striking a balance

between capturing the coarse structural outline and preserving fine-grained local details. To this end, we design a distinctive dual-branch encoder.

The two branches are named the Contextual Branch and the Detail-oriented Branch, specifically designed to extract global geometric information and local fine-grained features from the point cloud. Our goal is to develop a encoder capable of efficiently capturing both global and local features from the point cloud to produce a coarse point cloud feature f_{coarse}:

$$f_{\text{coarse}} = E_{\text{encoder}}(P \in \mathbb{R}^{N\times 3}),$$

where $P \in \mathbb{R}^{N\times 3}$ represents the input point cloud.

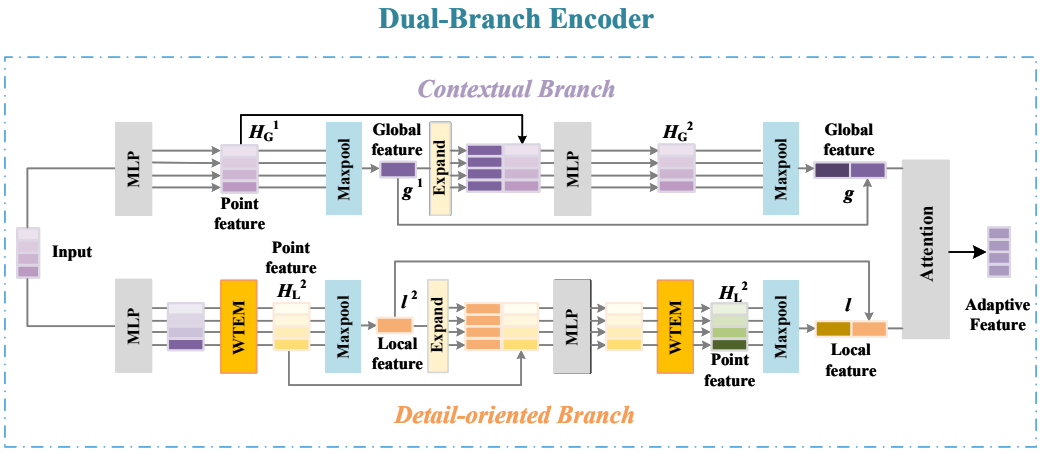


Figure 2: Overview of our Dual-branch encoder

The Contextual Branch focuses on extracting global geometric features, denoted by G and g. This branch captures the overall structure of the point cloud by applying global feature aggregation techniques, such as max pooling and transformer-based global correlation modeling, to produce robust global features.

The Contextual Branch adopts a two-layer stacked architecture inspired by the PCN[1] feature extraction network. In the first layer, a MLP processes the $n \times 3$ dimensional sparse point cloud P to generate the feature matrix $\mathbf{H}_G^1$. The global feature vector g_1 is then obtained via max pooling. In the second layer, g_1 is concatenated with each row of the feature matrix $\mathbf{H}_G^1$, producing an expanded feature matrix $\tilde{\mathbf{H}}_G^1$. Sequential application of two MLPs and a max pooling operation yields the feature vector g_2. Finally, g_1 and g_2 are fused to form the global feature vector G for the Contextual Branch.

As a complement to the Contextual Branch, the Detail-oriented Branch focuses on extracting significant local details, such as edges, sharp corners, and other geometrically complex regions, denoted by L and l. This branch employs a stacked-layer approach to combine local perception techniques, such as MLPs and local max pooling, to extract rich local features. These local features are then aggregated and fused with the global features to enhance the overall representation of the point cloud. For clarity in the subsequent feature fusion stage, we designate the final global feature as the contextual feature ($\mathcal{F}_{ctx}$) and the aggregated output of the Detail-oriented Branch as the fine-grained feature ($\mathcal{F}_{fine}$).

To enhance the expressive capacity of point cloud features, we propose a novel module called the Wavelet-based Point Cloud Enhancement Module (WTEM). WTEM integrates wavelet-based multi-resolution analysis into the feature extraction pipeline to better capture both fine-grained local geometry and broader global context. This design is particularly effective in improving feature quality for downstream tasks such as point cloud completion.

Wavelet-based Point Cloud Enhancement Module (WTEM). Let the input point cloud be $P \in \mathbb{R}^{B\times N\times 3}$ and its associated per-point features after a shared MLP be $X \in \mathbb{R}^{B\times C\times N}$. To enable wavelet-based multi-resolution analysis, we reshape X into a pseudo-2D tensor

$$X' \in \mathbb{R}^{B\times C\times H_0\times W_0}, \quad H_0 = N,\ W_0 = 1. \tag{1}$$

A one-level discrete wavelet transform (DWT) with fixed analysis filters W_{DWT} decomposes $X^{(l-1)}$ into four subbands,

$$Y^{(l)} = \text{DWT}(X^{(l-1)}; W_{\text{DWT}}), \quad Y^{(l)} \in \mathbb{R}^{B\times C\times 4\times H_l\times W_l}, \tag{2}$$

where $H_l = \lfloor H_{l-1}/2 \rfloor$ and $W_l = \lfloor W_{l-1}/2 \rfloor$ ($W_l = 1$ in practice). The four subbands (LL, LH, HL, HH) are rearranged into a depth layout for independent processing by depthwise separable convolutions,

$$\tilde{Z}^{(l)} = \text{DWConv2d}(Y^{(l)}), \quad \hat{Y}^{(l)} = \gamma^{(l)} \cdot \tilde{Z}^{(l)}, \tag{3}$$

where $\gamma^{(l)} \in \mathbb{R}^{1\times 4C\times 1\times 1}$ denotes a learnable scaling factor initialized to 0.1. The enhanced subbands are recursively fused through the inverse wavelet transform (IWT) using fixed synthesis filters W_{IWT},

$$X^{(l-1)} = \text{IWT}(\hat{Y}^{(l)}; W_{\text{IWT}}). \tag{4}$$

After L levels of decomposition and reconstruction, the final reconstructed signal $X^{(0)} \in \mathbb{R}^{B\times C\times H_0\times W_0}$ is obtained. In parallel, a depthwise baseline convolution is applied to the original feature map, and the outputs are fused as

$$F_{\text{out}} = \mathrm{DWConv2d}_{\text{base}}(X') + X^{(0)}. \tag{5}$$

Permutation invariance. In the proposed pipeline, WTEM is applied to per-point features organized along the point index. While the DWT operation itself is order-sensitive, the final descriptors are aggregated by symmetric pooling (adaptive max-pooling along the point dimension), thereby restoring permutation invariance at the network output. Alternatively, if strict invariance at the module level is required, WTEM can be applied within local patches $\mathcal{N}(i)$ constructed by kNN or ball query; the outputs from each patch are subsequently aggregated by symmetric reduction functions.

Hyperparameters. WTEM employs fixed (non-trainable) wavelet filters, defaulting to Haar (db1). The number of decomposition levels L is set to 1 by default. All convolutions are depthwise with channel grouping $= 4C$. Scaling factors $\gamma^{(l)}$ are initialized to 0.1. The neighborhood size N_p for patch-based construction, when used, is set to 16 or 32.

WTEM modules are inserted after key convolutional layers within the Detail-oriented Branch, acting as plug-and-play feature refiners. By capturing multi-scale structural patterns, WTEM significantly improves the network's ability to reconstruct both detailed and globally consistent point cloud shapes.

The WTEM module can be summarized as:

$$F_{\text{enh}} = \mathrm{WTEM}(F_{\text{in}}) \tag{6}$$

The **Contextual Branch** ($\mathcal{B}_{ctx}$) is designed to perceive the global shape and overall structure of the point cloud. This process yields a comprehensive contextual feature, denoted as $\mathcal{F}_{ctx} \in \mathbb{R}^{C_1}$, which encapsulates the extensive properties of the shape.

$$\mathcal{F}_{ctx} = \mathcal{B}_{ctx}(X) \tag{7}$$

Concurrently, the **Detail-oriented Branch** ($\mathcal{B}_{detail}$) is specifically engineered to capture salient local geometric patterns and high-frequency details. This branch produces a fine-grained feature, $\mathcal{F}_{fine} \in \mathbb{R}^{C_1}$, that highlights the intricate and discriminative regions of the input.

$$\mathcal{F}_{fine} = \mathcal{B}_{detail}(X) \tag{8}$$

To synergize these two distinct yet complementary features, we employ a **Cross-Attention** mechanism for adaptive feature fusion. Instead of simple concatenation, this mechanism allows the features to dynamically interact. Specifically, one feature (e.g., $\mathcal{F}_{ctx}$) serves as the query to attend to the other ($\mathcal{F}_{fine}$), which acts as the key and value. This enables the model to selectively emphasize detailed information guided by the global context, or conversely, to refine the contextual understanding using salient local cues. The resulting fused feature, termed the adaptive feature $\mathcal{F}_{adapt} \in \mathbb{R}^{C_2}$, is a robust representation that is both globally aware and locally precise.

$$\mathcal{F}_{adapt} = \text{CrossAttn}(\text{Query} = \mathcal{F}_{ctx}, \text{Key/Value} = \mathcal{F}_{fine}) \tag{9}$$

This adaptively fused representation provides a high-quality foundation for the subsequent coarse point cloud generation task.

3.2.2. Refinement with Cascaded SDG Blocks

To transform the coarsely generated point cloud into a complete and detailed shape, we employ a two-stage cascaded refinement network built upon Self-structure Dual-Generator (SDG) blocks. This progressive strategy aims to first establish a structurally sound geometric foundation and then enrich it with fine-grained surface details.

The core of this stage, the SDG block, operates on a dual-branch architecture: **Structure Analysis** and **Similarity Alignment**. Let the input to an SDG block consist of a coarse point cloud P_{in}, a global feature vector f_{g}, a set of local geometric features f_{local}, and the original partial point cloud P_{partial}.

1. **Structure Analysis Branch:** This branch utilizes a self-attention mechanism to analyze the internal structural relationships of P_{in}. Critically, to make the model aware of under-reconstructed areas, we introduce a geometric-aware positional embedding. We compute the Chamfer Distance between the current coarse prediction P_{in} and the initial partial input P_{partial}. This distance, which quantifies the reconstruction error, is then transformed into a sinusoidal positional embedding. By injecting this embedding into the self-attention module, we guide the model to focus its representational power on the most geometrically inconsistent or missing regions, thereby ensuring global structural integrity.
2. **Similarity Alignment Branch:** This branch employs a cross-attention mechanism to fuse the global structure with local details. It uses the structurally-aware features from the first branch as queries to attend to the rich, fine-grained information contained within the local features f_{local}. This process

effectively aligns the high-level structural context with the corresponding low-level geometric details.

The outputs from these two branches are then fused and processed through a residual learning module, which predicts point-wise displacement vectors. These vectors are added to the input points P_{in} to produce the refined output P_{out}.

Our refinement process proceeds in two stages. The first SDG block takes the initial coarse prediction P_{coarse} and generates an intermediate, structurally improved point cloud P_{mid}:

$$P_{\text{mid}} = \text{SDG}_1(P_{\text{coarse}}, f_{\text{g}}, f_{\text{local}}, P_{\text{partial}}) \tag{10}$$

Subsequently, the second SDG block takes this intermediate result P_{mid} as input and performs a second round of refinement to produce the final high-fidelity output P_{fine}. Note that the same global and local features are reused to maintain context.

$$P_{\text{fine}} = \text{SDG}_2(P_{\text{mid}}, f_{\text{g}}, f_{\text{local}}, P_{\text{partial}}) \tag{11}$$

This cascaded, dual-branch refinement framework enables a coarse-to-fine enhancement of the point cloud. The first stage focuses on correcting major structural errors, while the second stage adds intricate details, significantly improving both the geometric completeness and visual realism of the final completed shape.

3.3. Dense Refiner

Given the coarse point cloud $\mathcal{P}_{\text{coarse}} \in \mathbb{R}^{B\times N_c\times 3}$ generated by the Coarse Generator, the Dense Refiner progressively upsamples it into a dense, high-resolution shape. This is achieved through a carefully designed encoder-seed-decoder architecture, where each component is engineered to simultaneously preserve fine-grained local details and enforce global shape consistency.

1. Hierarchical Feature Encoder. The encoder processes the input point cloud $\mathcal{P}_{\text{coarse}}$ to extract a powerful global shape feature $\mathbf{g} \in \mathbb{R}^{B\times 512\times 1}$ and a set of patch-level features $(\mathbf{P}_{\text{patch}}, \mathbf{F}_{\text{patch}})$, where $\mathbf{P}_{\text{patch}} \in \mathbb{R}^{B\times N_p\times 3}$ and $\mathbf{F}_{\text{patch}} \in \mathbb{R}^{B\times 256\times N_p}$ ($N_p = 256$). Our encoder architecture is built upon PointNet++ but enhances its representational power by **interleaving** Set Abstraction (SA) modules with our proposed Multi-Scale Cross-Attention (MSCA) blocks. This **alternating sequence** facilitates feature refinement at each hierarchical level. Specifically, after the first two SA layers, an MSCA block is applied in a **self-attention** configuration ($\mathbf{F}_i = \text{MSCA}(\mathbf{P}_i, \mathbf{F}_i, \mathbf{P}_i, \mathbf{F}_i)$) to refine the sampled features by capturing rich, multi-scale contextual dependencies. The final SA layer performs a global grouping to produce the holistic shape feature $\mathbf{g}$.

2. Multi-Scale Cross-Attention (MSCA). MSCA is the core feature refinement module, designed to explicitly capture both local geometric patterns and long-range dependencies across multiple scales. Given a query set $(\mathbf{P}_q, \mathbf{F}_q)$ and a support set $(\mathbf{P}_s, \mathbf{F}_s)$, MSCA operates in a top-down fashion over L scales defined by downsampling rates $\{d_l\}_{l=1}^{L}$ and neighbor counts $\{k_l\}_{l=1}^{L}$.

- **Hierarchical Sampling.** For each scale l, farthest-point sampling (FPS) is used to subsample the points, creating a multi-scale pyramid of point sets.
- **Cross-Scale Feature Propagation.** The process begins at the coarsest scale L by applying VectorAttention (VA). For finer scales $l < L$, features from the coarser level $(l+1)$ are propagated to the current level l using **three-nearest-neighbor (3-NN) interpolation**:

$$\tilde{\mathbf{F}}_q^{(l)} = \text{3-Interpolate}\big(\mathbf{F}_q^{(l+1)}, \mathbf{P}_q^{(l+1)}, \mathbf{P}_q^{(l)}\big) \tag{12}$$

  These propagated features are then fused with the native features at scale l via a residual MLP:

$$\mathbf{F}_q^{(l)} \leftarrow \text{MLP}_{\text{fuse}}\Big(\Big[\mathbf{F}_{\text{native}}^{(l)}, \tilde{\mathbf{F}}_q^{(l)}\Big]\Big) \tag{13}$$

The final aggregated feature $\mathbf{F}_{\text{agg}}$ is produced by applying VectorAttention at each scale on the fused features, ensuring both local geometric consistency and global shape coherence.

3. Seed Generator. The Seed Generator bridges the encoder and decoder by transforming the global feature $\mathbf{g}$ and patch features $\mathbf{F}_{\text{patch}}$ into a set of 512 high-quality seed points $\mathcal{P}_0 \in \mathbb{R}^{B\times 512\times 3}$. These seeds serve as a robust, topologically complete skeleton for subsequent upsampling. We first upsamples the patch features $\mathbf{F}_{\text{patch}}$ by a factor of 2, enriching feature density via learned attention over $k = 16$ neighbors. Subsequent MLPs then fuse these upsampled features with the global feature $\mathbf{g}$ to predict the 3D coordinates of an initial seed cloud. Finally, to ensure full coverage of the original object's geometry, these predicted seeds are concatenated with the partial input, and a final FPS step selects the ultimate seed points $\mathcal{P}_0$.

4. Hierarchical Decoder. The decoder reconstructs the detailed, dense point cloud by progressively upsampling the seed points through a cascade of three Upsample Block modules, with respective upsampling factors of $1\times$, $4\times$, and $8\times$. This results in a sequence of point clouds with increasing resolution.

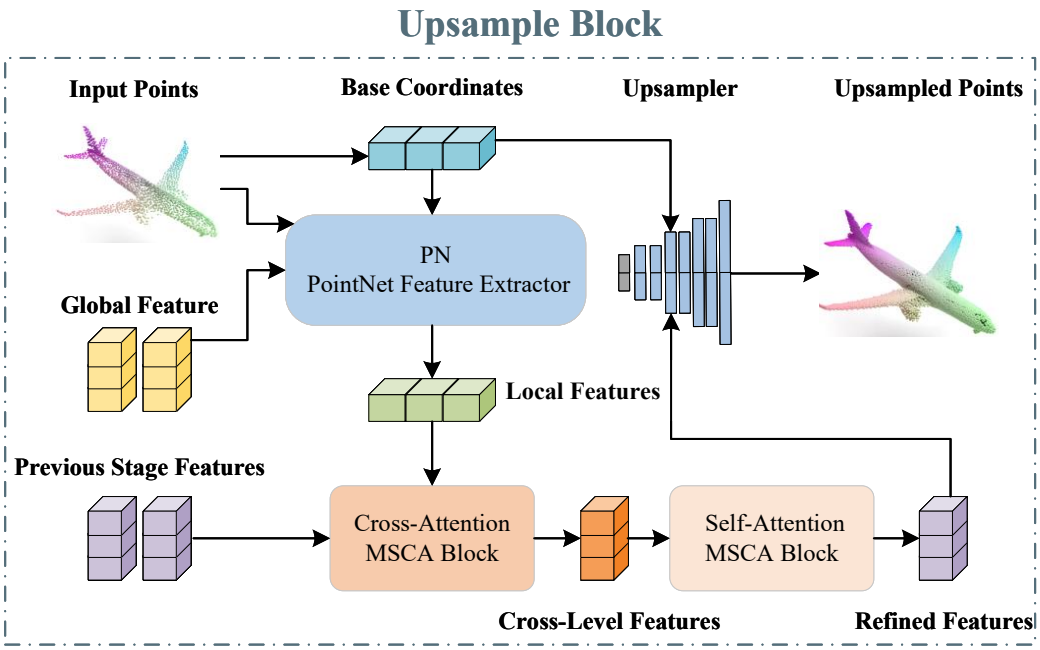


Figure 3: Framework Overview of our Upsample Block

Upsample Block Module. Each `Upsample Block` refines and densifies the point cloud from the previous stage $(\mathcal{P}^{(\ell)}, \mathbf{F}^{(\ell)})$ by executing a four-step process, as illustrated in Figure 3:

1. **Local Feature Extraction:** A PointNet-like module (PN) first extracts per-point features $\mathbf{h}^{(\ell)}$ for the current point set $\mathcal{P}^{(\ell)}$, conditioned on the global feature $\mathbf{g}$.
$$\mathbf{h}^{(\ell)} = \mathrm{PN}\big(\mathcal{P}^{(\ell)}, \mathbf{g}\big) \tag{14}$$
2. **Cross-Level Attention:** An MSCA module performs cross-attention between the current point set and the output of the previous decoder stage $(\mathcal{P}^{(\ell-1)}, \mathbf{F}^{(\ell-1)})$. This step effectively injects coarser-level geometric context into the current refinement level.
$$\mathbf{g}^{(\ell)} = \mathrm{MSCA}_{\mathrm{cross}}\big((\mathcal{P}^{(\ell)}, \mathbf{h}^{(\ell)}), (\mathcal{P}^{(\ell-1)}, \mathbf{F}^{(\ell-1)})\big) \tag{15}$$
3. **Intra-Level Self-Attention:** A second MSCA module is applied to the newly aggregated features $\mathbf{g}^{(\ell)}$ in a self-attention manner. This further refines the features by modeling complex dependencies within the current scale.
$$\mathbf{f}^{(\ell)} = \mathrm{MSCA}_{\mathrm{self}}\big((\mathcal{P}^{(\ell)}, \mathbf{g}^{(\ell)}), (\mathcal{P}^{(\ell)}, \mathbf{g}^{(\ell)})\big) \tag{16}$$

4. **Point Cloud Upsampling:** Finally, a deconvolutional module (`DeConv`) takes the highly refined features $\mathbf{f}^{(\ell)}$ to predict 3D coordinate offsets, which are added to an upsampled version of $\mathcal{P}^{(\ell)}$, yielding the next-level, denser point cloud $\mathcal{P}^{(\ell+1)}$.

$$\mathcal{P}^{(\ell+1)} = \text{DeConv}\big(\mathcal{P}^{(\ell)}, \mathbf{f}^{(\ell)}\big) \tag{17}$$

By cascading these blocks, our decoder generates three levels of detailed point clouds, where the final output $\mathcal{P}_{\text{detail}} = \mathcal{P}^{(3)}$ is used for loss computation.

4. Experiment

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel
PCN	12.11	14.53	11.50	13.32	**6.90**	12.01	11.37	14.44	12.80
ODGNet	8.87	5.44	10.75	9.30	10.10	8.10	11.34	8.22	7.75
FSC	7.98	4.53	9.77	8.32	9.08	7.47	10.08	7.11	7.45
SeedFormer	7.79	4.79	9.67	8.63	8.66	6.65	10.01	6.96	6.96
GeoFormer	7.74	4.46	9.86	8.58	8.54	6.58	10.01	7.09	6.80
Ours	**7.20**	**4.15**	**9.39**	**8.09**	7.76	**6.06**	**9.28**	**6.48**	**6.38**
Improvement	↓7.0%	↓7.0%	↓2.9%	↓2.8%	x	↓7.9%	↓7.3%	↓3.1%	↓6.2%

Table 1: Quantitative comparison of point-cloud completion results on the PCN dataset (64-point input) measured by the CD-ℓ_1 distance ($\times 10^{-3}$).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel
PCN	10.60	6.09	11.55	9.51	12.33	12.57	12.87	9.87	9.98
ODGNet	6.50	3.77	8.77	7.56	6.84	5.09	8.47	5.84	5.66
FSC	7.02	4.07	9.12	8.1	7.21	5.88	9.30	6.26	6.25
SeedFormer	6.74	3.85	9.05	8.06	7.06	5.21	8.85	6.05	5.85
GeoFormer	6.42	3.60	8.69	7.46	6.71	5.15	8.28	5.84	5.63
Ours	**6.30**	**3.54**	**8.65**	**7.33**	**6.57**	**4.93**	**8.16**	**5.66**	**5.57**
Improvement	↓1.9%	↓1.7%	↓0.5%	↓1.7%	↓2.1%	↓4.3%	↓1.4%	↓3.1%	↓1.1%

Table 2: Quantitative comparison of point-cloud completion results on the PCN dataset (2048-point input) measured by the CD-ℓ_1 distance ($\times 10^{-3}$).

4.1. Experiments on PCN Dataset

The PCN[1] dataset is one of the most widely used benchmark datasets for point cloud completion. It is a subset of ShapeNet[2] with shapes from 8 categories. The incomplete point clouds are generated by back-projecting 2.5D depth images from 8 viewpoints in order to simulate real-world sensor data. For each shape, 16,384 points are uniformly sampled from the mesh surfaces as complete ground-truth, and 2,048 points are sampled as partial input. Our task focuses on the completion of sparse point clouds. Therefore, we downsample all the partial point clouds in PCN to 64 points, while the complete point clouds remain unchanged. The experimental results with sparse point clouds containing 64 points as input are presented in Table 1.

Although our model design is primarily targeted at the sparse point cloud completion task, we still evaluated its completion performance on standard point cloud inputs. The results are presented in Table 2 and Figure 5. When processing the 2048 partial point clouds natively provided by the PCN dataset, our model still maintains state-of-the-art performance.

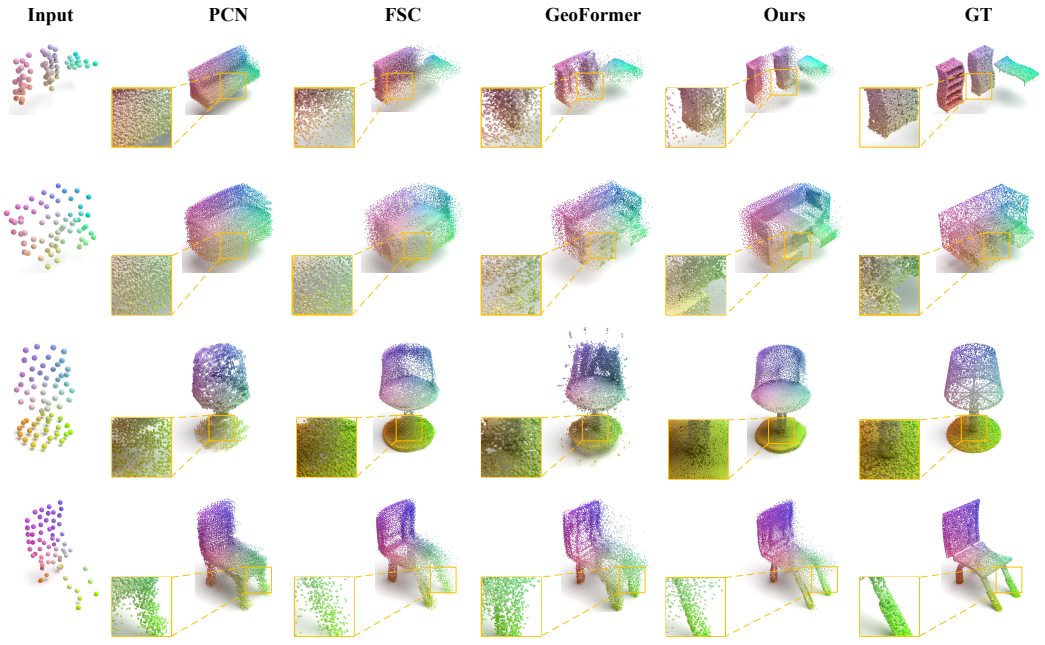


Figure 4: Visual Comparison results on the PCN dataset(64 input)

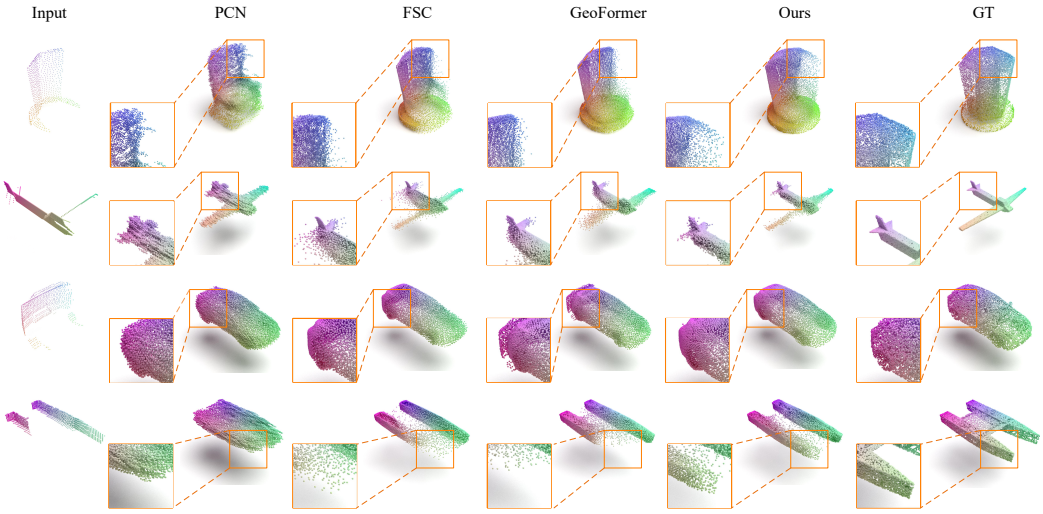


Figure 5: Visual Comparison results on the PCN dataset(2048 input)

4.1.1. Qualitative Results and Quantitative Comparisons On PCN

We conducted quantitative and qualitative experiments on 8 categories in the PCN subset, with the experimental results shown in Figure 2 and Table 1.

Our comparison baselines include: 1) The early learning-based point cloud completion method PCN[1], which is also the designer of the PCN dataset; 2) The state-of-the-art point cloud completion model ODGNet[17] 3) FSC[9], the pioneer of few point cloud completion; 4) The advanced multi-modal point cloud completion method GeoFormer[34]; 5) The point-based completion method Seedformer[10].

The results of both quantitative and qualitative experiments demonstrate the superiority of our method.

As shown in Figure 4 and Figure 5, our method produces more uniform and structurally reliable point clouds, significantly reducing geometric structure collapse and the generation of outliers.

4.2. Experiments on ShapeNet55/34 Dataset

To comprehensively assess the generalization capability of our model, we extend the experiments to ShapeNet-55 and ShapeNet-34, two subsets that jointly span the entire ShapeNet repository across 55 distinct object categories. ShapeNet-55 comprises 41,952 training shapes and 10,518 testing shapes, while ShapeNet-34 includes 46,765 training shapes drawn from 34 categories. For ShapeNet-34, the test split is further divided into 3,400 shapes belonging to the 34 seen cat-

egories and 2,305 shapes from the remaining 21 unseen categories, enabling an evaluation of both within- and cross-category generalization.

Following the protocol established in previous works [31, 34, 10, 17], partial observations are synthesized during the training stage.

During training, random viewpoints are selected, and the $n\%$ farthest points from these viewpoints are discarded to simulate partial scans. At test time, the viewpoints are fixed, and n is systematically varied to three difficulty levels: 25% (simple), 50% (moderate), and 75% (hard). Since this work targets sparse point cloud completion, we further tailor the data generation pipeline. After discarding 25% of the points to create a partial observation, we uniformly subsample the remaining points to exactly 64 via farthest-point sampling (FPS), yielding the final sparse input for both training and evaluation.

Our experimental evaluation results on ShapeNet55/34 are presented in Tables 3 .We compare our method with existing point cloud completion baselines, using $\mathrm{CD}{-}\ell_1$, $\mathrm{CD}{-}\ell_2$, and F-Score as evaluation metrics, the unit for $\mathrm{CD}{-}\ell_1$ is 1×10^{-3}, and the unit for $\mathrm{CD}{-}\ell_2$ is 1×10^{-4}. Our sparse point cloud completion method achieves significantly superior performance compared to existing baselines on both seen and unseen categories, which substantiates the generalization ability of our model.

Evaluation	ShapeNet 34 seen categories			ShapeNet 21 unseen categories		
	CD-ℓ_1	CD-ℓ_2	F-Score	CD-ℓ_1	CD-ℓ_2	F-Score
PCN	1.7747	0.6734	0.2039	2.2422	0.6948	0.1486
ODGNet	0.7652	0.6966	0.4199	0.8713	0.6985	0.3832
FSC	1.5469	0.6673	0.2543	1.8649	0.6912	0.2109
SeedFormer	0.8476	0.7144	0.3994	1.0835	0.7232	0.3483
Geoformer	1.4839	0.7109	0.1665	1.6947	0.7184	0.1491
Ours	**0.5929**	**0.5610**	**0.4312**	**0.7180**	**0.5636**	**0.3925**
Improvement	↓22.5%	↓15.9%	↑2.7%	↓17.6%	↓18.5%	↑2.4%

Table 3: Results on ShapeNet seen and unseen categories. The values in the *Improvement* row indicate the performance improvement over the previous best method. For F-Score, higher is better (↑), while for CD, lower is better (↓).

4.3. Training Details

The loss function is used to measure the difference between the output point cloud and the ground-truth point cloud. Similar to previous completion methods,

our loss function $\mathcal{L}$ is defined as follows:

$$\mathcal{L}(P_{\text{coarse}}, P_{\text{detail}}, P_{\text{gt}}) = l_1(P_{\text{coarse}}, \hat{P}_{\text{gt}}) + \alpha l_2(P_{\text{detail}}, P_{\text{gt}})$$

where the above loss function consists of two components, l_1 and l_2, weighted by the factor α. Specifically, l_1 and l_2 represent the distances between the predicted point cloud and the ground truth at the coarse level and the detailed level, respectively.

To ensure that the final output point cloud aligns with the ground truth in terms of both density distribution and overall structure while minimizing computational cost, we employ two distance metrics. For l_1, we use the Earth Mover's Distance (EMD)[35], which prioritizes density consistency and is computationally intensive but accurate. For l_2, we use the Chamfer Distance (CD), which provides structural consistency with lower computational complexity. Specifically,

$$\text{EMD}\left(P_{\text{coarse}}, \tilde{P}_{\text{gt}}\right) = \min_{\phi: P_{\text{coarse}} \to \tilde{P}_{\text{gt}}} \frac{1}{|P_{\text{coarse}}|} \sum_{p \in P_{\text{coarse}}} \|p - \phi(p)\|_2,$$

where ϕ represents the bijection from P_{coarse} to $\tilde{P}_{\text{gt}}$, minimizing the average distance between corresponding points in the two sets.

For l_2, we use the Chamfer Distance defined as follows:

$$\begin{aligned}\text{CD}\left(P_{\text{detail}}, P_{\text{gt}}\right) = &\frac{1}{|P_{\text{detail}}|} \sum_{p \in P_{\text{detail}}} \min_{q \in P_{\text{gt}}} \|p - q\|_2 \\ &+ \frac{1}{|P_{\text{gt}}|} \sum_{q \in P_{\text{gt}}} \min_{p \in P_{\text{detail}}} \|q - p\|_2.\end{aligned}$$

In this formulation, l_1 ensures density alignment for the coarse-level point cloud, while l_2 focuses on structural alignment for the detailed point cloud. The combination of these two loss components allows us to achieve a balance between global structural consistency and local geometric precision.

4.4. Experiments on KITTI Dataset

Since the previous two datasets are synthetic, generated from CAD models or meshes, they may not fully reflect the characteristics of real scanned point clouds. To address this gap, we further incorporate the KITTI dataset [36], which is collected from an autonomous driving platform and serves as a challenging real-world benchmark. Following prior works, we extract sequences of Velodyne

scans from KITTI and retain only the points within the annotated bounding boxes of cars.

In total, this yields 2483 partial point clouds without corresponding ground truth. Since the point clouds in the KITTI dataset are obtained from real LiDAR scans and contain no ground-truth shapes, we only evaluate the completion performance on this dataset.

The point cloud completion results on the KITTI dataset are illustrated in Fig. 6. Evidently, our approach substantially suppresses the spurious points around the vehicle and delivers more accurate completion, thereby underscoring its practical value in real-world scenarios.

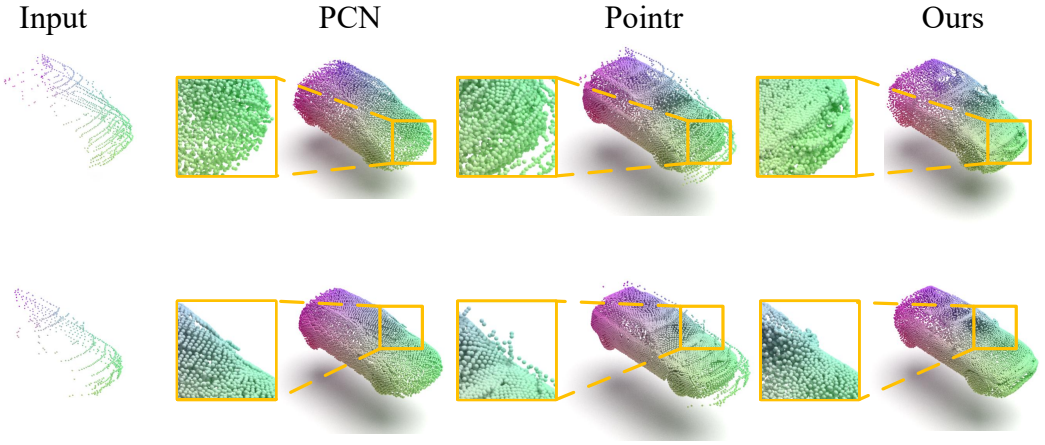


Figure 6: Visual results on KITTI Dataset

4.5. Ablation studies

Model Configuration	**WTEM**	Detail Branch	Dense Refiner	CD-ℓ_1 ↓	CD-ℓ_2 ↓	**F-Score** ↑
Full w/o WTEM		✓	✓	7.3277	2.2948	79.4393
Full w/o Detail Branch	✓		✓	7.5358	2.3829	78.3030
Full w/o Dense Refiner	✓	✓		7.4104	2.3359	78.4675
Full Model	✓	✓	✓	**7.2026**	**2.2903**	**80.3360**

Table 4: Quantitative results of ablation studies. The arrows (↑, ↓) indicate whether a higher or lower value is better.

In this section, we designed ablation studies to verify the effectiveness of our core components and architectural design. The modifications in our ablation stud-

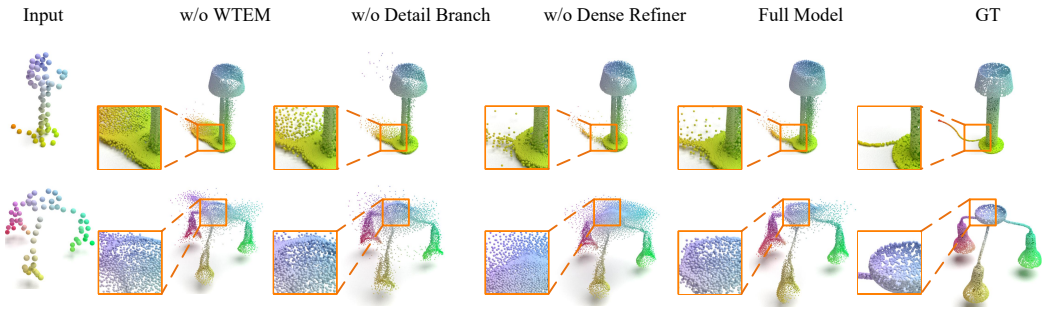


Figure 7: Visual results of the ablation study

ies mainly include the following parts: 1) To verify the effectiveness of the WTEM module, we attempted to remove this module from the detail-oriented branch. Our WTEM feature enhancement module is designed to be plug-and-play; during ablation studies it can be removed directly after the specified convolutional layers in the detail-oriented branch. 2) To verify the effectiveness of the dual-branch encoder design of the coarse generator, we removed the detail-oriented branch.

3) To verify the effectiveness of the dense refiner, we attempted to directly replace the dense refiner framework with SDG blocks of corresponding numbers and upsampling rates. Specifically, in the original Dense Refiner stage, we replace it with an SDG Block to upsample the coarse point cloud generated by the coarse generator, with an upsampling ratio configured to 8.

The results of the ablation studies are shown in Table 4 and Figure 7. The effectiveness of our core components and architectural design is validated by both quantitative evaluations and qualitative visualizations. Notably, our approach substantially mitigates artifacts and structural degradation that commonly occur during the completion of sparse point clouds.

5. Conclusion

In this paper, we introduced a novel cascaded framework that addresses the challenging task of point cloud completion from extremely sparse inputs. By decoupling the problem into two distinct stages—coarse shape generation and dense detail refinement—our method effectively overcomes the limitations of single-stage architectures. This specialized design prevents geometric collapse and allows for robust global structure modeling in the first stage, followed by high-fidelity local detail recovery in the second. Our dual-branch encoder and multi-scale refinement module work in synergy to produce complete, uniform, and accu-

rate point clouds. Extensive experiments on challenging benchmarks confirm that our approach significantly and consistently outperforms state-of-the-art methods, especially under conditions of severe data sparsity. These results not only validate the effectiveness of our decoupled design but also suggest its potential as a robust solution for broader 3D vision applications involving noisy and incomplete data.

6. Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62202151).

References

[1] W. Yuan, T. Khot, D. Held, C. Mertz, M. Hebert, Pcn: Point completion network, in: 2018 international conference on 3D vision (3DV), IEEE, 2018, pp. 728–737.

[2] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, F. Yu, ShapeNet: An Information-Rich 3D Model Repository, Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015).

[3] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, S. Savarese, Topnet: Structural point cloud decoder, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 383–392.

[4] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1912–1920.

[5] L. Pan, X. Chen, Z. Cai, J. Zhang, H. Zhao, S. Yi, Z. Liu, Variational relational point completion network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8524–8533.

[6] L. Pan, Z. Cai, Z. Liu, Robust partial-to-partial point cloud registration in a full range, arXiv preprint arXiv:2111.15606 (2021).

[7] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, W. Sun, Grnet: Gridding residual network for dense point cloud completion, in: European conference on computer vision, Springer, 2020, pp. 365–381.

[8] Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, Pf-net: Point fractal network for 3d point cloud completion, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 7662–7670.

[9] X. Wu, X. Wu, T. Luan, Y. Bai, Z. Lai, J. Yuan, Fsc: Few-point shape completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 26077–26087.

[10] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, C. Wang, Seedformer: Patch seeds based point cloud completion with upsample transformer, in: European conference on computer vision, Springer, 2022, pp. 416–432.

[11] C. Fang, B. Yang, H. Ye, F. Cao, Fast point completion network, Neural Computing and Applications 36 (18) (2024) 10897–10913.

[12] F. Lin, Y. Xu, Z. Zhang, C. Gao, K. D. Yamada, Cosmos propagation network: Deep learning model for point cloud completion, Neurocomputing 507 (2022) 221–234.

[13] J. Li, S. Guo, L. Wang, S. Han, Completedt: Point cloud completion with information-perception transformers, Neurocomputing 592 (2024) 127790.

[14] J. Song, X. Wu, J. Yao, Q. Zhang, C. Shang, Q. Qian, J. Song, Spc: Self-supervised point cloud completion, Neural Networks (2025) 108107.

[15] B. Zhao, X. Chen, X. Hua, W. Xuan, D. D. Lichti, Completing point clouds using structural constraints for large-scale points absence in 3d building reconstruction, ISPRS Journal of Photogrammetry and Remote Sensing 204 (2023) 163–183.

[16] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, Z. Han, Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 5499–5509.

[17] P. Cai, D. Scott, X. Li, S. Wang, Orthogonal dictionary guided shape completion network for point cloud, in: AAAI, 2024.

[18] Z. Zhu, H. Chen, X. He, W. Wang, J. Qin, M. Wei, Svdformer: Complementing point cloud via self-view augmentation and self-structure dual-generator, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 14508–14518.

[19] S. Huang, Y. Xie, S.-C. Zhu, Y. Zhu, Spatio-temporal self-supervised representation learning for 3d point clouds, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 6535–6545.

[20] H. Wang, Q. Liu, X. Yue, J. Lasenby, M. J. Kusner, Unsupervised point cloud pre-training via occlusion completion, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 9782–9792.

[21] C. Sun, Z. Zheng, X. Wang, M. Xu, Y. Yang, Self-supervised point cloud representation learning via separating mixed shapes, IEEE Transactions on Multimedia 25 (2022) 6207–6218.

[22] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, L. Yuan, Masked autoencoders for point cloud self-supervised learning, in: European conference on computer vision, Springer, 2022, pp. 604–621.

[23] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, H. Li, Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training, Advances in neural information processing systems 35 (2022) 27061–27074.

[24] G. Chen, M. Wang, Y. Yang, K. Yu, L. Yuan, Y. Yue, Pointgpt: Auto-regressively generative pre-training from point clouds, Advances in Neural Information Processing Systems 36 (2023) 29667–29679.

[25] A. Sharma, O. Grau, M. Fritz, Vconv-dae: Deep volumetric shape learning without object labels, in: Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14, Springer, 2016, pp. 236–250.

[26] A. Dai, C. Ruizhongtai Qi, M. Nießner, Shape completion using 3d-encoder-predictor cnns and shape synthesis, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5868–5877.

[27] T. Hu, Z. Han, M. Zwicker, 3d shape completion with multi-view consistent inference, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 34, 2020, pp. 10997–11004.

[28] Y. Li, Z. Yu, C. Choy, C. Xiao, J. M. Alvarez, S. Fidler, C. Feng, A. Anandkumar, Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 9087–9098.

[29] B. Gong, Y. Nie, Y. Lin, X. Han, Y. Yu, Me-pcn: Point completion conditioned on mask emptiness, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 12488–12497.

[30] P. Achlioptas, O. Diamanti, I. Mitliagkas, L. Guibas, Learning representations and generative models for 3d point clouds, in: International conference on machine learning, PMLR, 2018, pp. 40–49.

[31] Y. Rong, H. Zhou, L. Yuan, C. Mei, J. Wang, T. Lu, Cra-pcn: Point cloud completion with intra-and inter-level cross-resolution transformers, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 4676–4685.

[32] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, J. Zhou, Pointr: Diverse point cloud completion with geometry-aware transformers, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 12498–12507.

[33] X. Yu, Y. Rao, Z. Wang, J. Lu, J. Zhou, Adapointr: Diverse point cloud completion with adaptive geometry-aware transformers. arxiv, arXiv preprint arXiv:2301.04545 10 (2022).

[34] J. Yu, B. Huang, Y. Zhang, H. Li, X. Tang, S. Gao, Geoformer: Learning point cloud completion with tri-plane integrated transformer, in: Proceedings of the 32nd ACM International Conference on Multimedia, 2024, pp. 8952–8961.

[35] Y. Rubner, C. Tomasi, L. J. Guibas, The earth mover's distance as a metric for image retrieval, International journal of computer vision 40 (2000) 99–121.

[36] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, The international journal of robotics research 32 (11) (2013) 1231–1237.