Hyper-PCN: Hypergraph-Based Point Cloud Completion via High-Order Correlation Modeling

Linfei Li[1,2] Pei Tan[3] Siqi Li[1,2*] Changqing Zou[4] Yue Gao[1,2]

[1]{BNRist, THUIBCS, BLBCI, School of Software}, Tsinghua University [2]Yangtze Delta Region Institute, Tsinghua University

[3]School of Information and Software Engineering, University of Electronic Science and Technology of China

[4]State Key Lab of CAD&CG, Zhejiang University

lilf25@mails.tsinghua.edu.cn, tanpei@std.uestc.edu.cn, lisiqi19971013@gmail.com, changqing.zou@zju.edu.cn, gaoyue@tsinghua.edu.cn

Abstract

Point cloud completion is an important yet challenging problem in 3D computer vision, which aims to reconstruct complete and dense 3D shapes from partial point clouds. Although transformer-based and geometry-based approaches have made significant progress, they often struggle to capture the complex, high-order correlations inherent in point clouds. To address this limitation, we propose Hyper-PCN, a point cloud completion framework that leverages hypergraphs to explicitly model complex, higher-order correlations within incomplete inputs for more accurate completion. It comprises two key modules: Hyper Refinement Stack, designed to progressively capture coarse-to-fine high-order correlations through a series of hypergraph learning stages, and Anchor-based Hypergraph Neural Network, which employs a two-stage sampling strategy to construct collaborative hypergraphs, ensuring robust modeling of global structures. Extensive experiments on multiple datasets demonstrate that our approach consistently outperforms state-of-the-art methods. Code is available at https://github.com/Rinfly/Hyper-PCN.

1. Introduction

Point cloud, as a pivotal representation of 3D geometry, has garnered significant attention in applications ranging from autonomous driving [8, 22] and robotic manipulation [27, 35] to augmented reality. However, point clouds acquired from real-world scanners are often incomplete due to issues such as occlusion and limited sensor resolution [10]. This incompletion severely hinders downstream tasks, including 3D reconstruction [25], point-cloud classification, and segmentation. Point cloud completion addresses this issue by reconstructing complete shapes from incomplete data, playing a pivotal role in enhancing data quality and facilitating subsequent 3D vision applications.

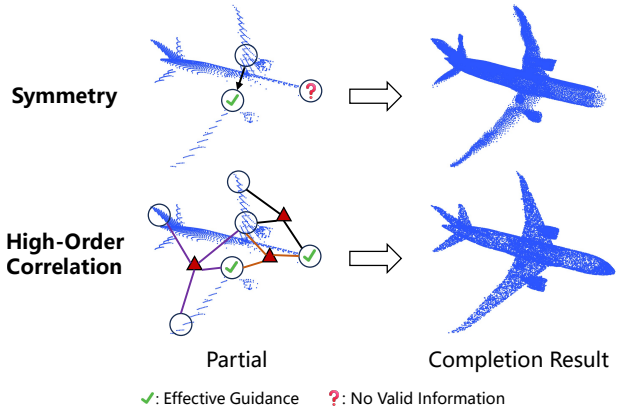

Figure 1. Performance comparison between symmetry-based methods and our approach, which utilizes higher-order correlations. When no symmetry exists or both symmetrical regions are incomplete, higher-order correlation guidance yields superior completion results.

Point cloud completion has emerged as a research hotspot in recent years, with most approaches focusing on enhancing the ability to capture correlations within point clouds. Early methods primarily relied on max-pooling feature extraction architectures like PointNet [28] and PointNet++ [29] to perform completion based on global features. PoinTr [44] pioneered the use of transformer [36] for this task, partitioning point clouds into patches and applying self-attention to capture local and global dependencies. Subsequent studies have advanced along two main directions: (1) devising novel transformer-based architectures to better capture correlations and thus produce more fine-grained results [6, 21, 30, 38, 43, 45, 50, 51], and (2) in-

*Corresponding author

troducing symmetry or learnable geometric priors to inject global shape-level context into the process [3, 42].

Despite these advances, existing methods often struggle with complex geometries and fine structural details, especially in challenging scenarios where common priors like symmetry are absent, as illustrated in Figure 1. This limitation stems from a fundamental challenge: the insufficient modeling of high-order correlations within point clouds. Real-world structures exhibit rich, multi-faceted semantic relationships beyond pairwise similarities or symmetry. For example, an aircraft's components such as wings, tail fins, and fuselage together form the aerodynamic shape, which must be organized in an ideal manner to alter airflow and enable flight [32]. Such multi-component, synergistic relationships cannot be adequately captured by pair-wise models, necessitating a framework capable of modeling multi-to-multi, high-order correlations.

Although transformer has gained popularity in point cloud completion, its reliance on pair-wise query-key interactions limits the capacity to model high-order correlations. **Hypergraph**, on the contrary, provides a natural and powerful way to represent such complex relationships, offering a promising direction for completion tasks. Although some studies have attempted to employ hypergraph for processing point clouds, their methods are primarily designed for complete point clouds. When dealing with incomplete point clouds, the application of hypergraph faces two specific challenges: (1) it is inherently more difficult to extract reliable correlation information from sparse and incomplete structures, and simple one-shot hypergraph construction may only capture limited relationships; (2) in terms of graph construction, common strategies like random point sampling [31] or voxel-based partitioning [18] tend to bias the computational focus towards the intact regions of the point cloud, which consequently constrains the prediction and recovery of the missing parts.

In this paper, we address the modeling of high-order correlations in incomplete point clouds for the first time and propose Hyper-PCN, a novel point cloud completion method based on hypergraph neural networks. To tackle the challenges mentioned above, we design two key components: (1) Hyper Refinement Stack (HyperRS), which progressively mines coarse-to-fine high-order correlation information from incomplete point clouds through a series of hypergraph modeling and convolution operations; and (2) Anchor-based Hypergraph Neural Network (A-HGNN), which employs a two-stage sampling strategy to construct hypergraphs through the collaboration between key points and anchors, guiding the hypergraph learning towards more comprehensive global features. Experimental results demonstrate that Hyper-PCN effectively models high-order correlations in point clouds and achieves superior performance on multiple public datasets, significantly outperforming existing point cloud completion methods. Overall, the contributions are summarized as follows:

- We first model high-order correlations in incomplete point clouds and propose Hyper-PCN, a novel hypergraph-based method for point cloud completion.
- We propose HyperRS, a novel architecture that progressively captures coarse-to-fine high-order correlations in incomplete point clouds, and A-HGNN, which leverages two-stage sampling to guide hypergraph construction.
- Extensive experiments demonstrate that our method effectively models high-order correlations in point clouds and outperforms state-of-the-art methods.

2. Related Works

2.1. Point Cloud Completion

Early one-stage methods like TopNet [33] struggled to capture fine geometric details, prompting a shift toward two-stage frameworks. PCN [46] first extracts global features with PointNet [28], then uses MLPs to produce a coarse point cloud, and finally upsamples it via folding. Despite its influence, PCN's reliance on simple MLPs hinders the handling of complex structures. Subsequent works like SnowFlake [38] and FBNet [43] addressed this by introducing dedicated upsampling modules, such as the snowflake point deconvolution and feedback-aware completion blocks. As the point set can be viewed as a token sequence, PoinTr [44] and its following works [6, 21, 30, 45, 50, 51] employ the transformer architecture [36] to predict the missing point proxies. Recent studies [3, 42] leverage geometric features such as symmetry or learnable priors to provide contextual information at the global shape-level.

Other related works include view-guided completion, self-supervised completion and 3D generation. Pioneered by ViPC [49], view-guided methods [1, 9, 23, 40, 52] fuse 2D visual cues with 3D data to enhance reconstruction. Self-supervised methods [5, 7, 15] reduce the dependency on ground-truth by leveraging proxy tasks, such as mask-reconstruction or cross-view consistency. Some controllable 3D generation methods [34] can also perform completion by using incomplete point clouds as control conditions, but they require substantial computational resources.

2.2. Hypergraph Learning

Hypergraph is a set of vertices and hyperedges that encodes high-order correlations beyond pairwise graphs [12]. Unlike a conventional graph where an edge connects only two vertices, a hyperedge can link an arbitrary number of vertices, forming a subset and thus providing a natural and flexible framework for modeling complex group-wise relationships. Built upon this structure, Hypergraph Neural Network (HGNN) [11, 13, 20] has emerged as powerful tool for deep learning on hypergraphs.

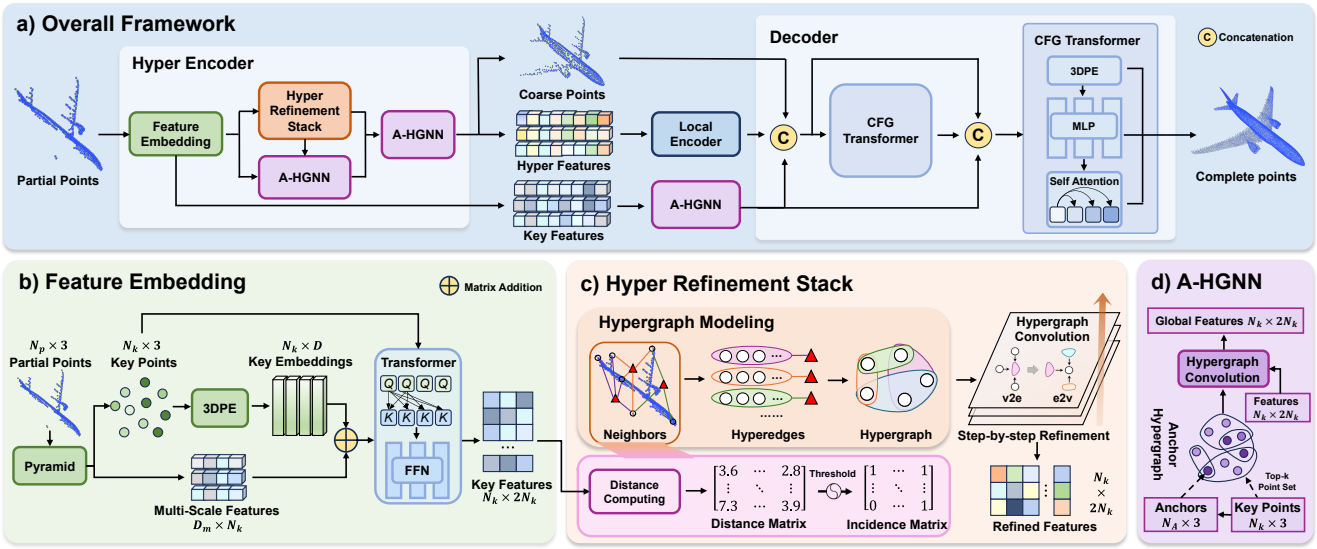

Figure 2. **a)** The overall *Hyper-PCN* framework, consisting of a hypergraph encoder and a two-stage decoder. **b)** The detailed architecture of the *Feature Embedding*. **c)** The detailed architecture of the *Hyper Refinement Stack (HyperRS)*. **d)** The detailed architecture of the *Anchor-based Hypergraph Neural Network (A-HGNN)*.

Due to their advantages, hypergraph and HGNN have been applied to many tasks in computer vision, including image classification [14] and detection [19], depth estimation [2] and pose estimation [41]. In the domain of 3D point clouds, hypergraph has also demonstrated its utility in tasks such as 3D object detection [31], semantic segmentation [47], and quality assessment [4]. However, these existing works primarily focus on analyzing complete point clouds.

3. Method

3.1. Overall Framework

Hyper-PCN operates through a streamlined encoder-decoder process, and the overall framework is shown in Figure 2. The hyper encoder first takes a partial point cloud and generates a set of coarse points along with their deep feature embeddings. The decoder then leverages these features to refine the coarse points, ultimately reconstructing a complete and detailed point cloud.

Hyper Encoder The hyper encoder takes a partial point cloud as input and selects N_k key points through feature embedding. These key points are then processed by Hyper Refinement Stack (HyperRS) and Anchor-based Hypergraph Neural Network (A-HGNN). HyperRS constructs distance based hypergraphs and performs progressive refinement to yield hyper features and a coarse shape, while A-HGNN build collaborative hypergraphs through key points and anchors, effectively extracting global higher-order relations. The encoder outputs coarse points together with hyper features and key features.

Decoder The decoder refines the coarse prediction using both global and local cues. It concatenates the coarse features from the encoder with the features refined by the local encoder and A-HGNN, and feeds the combined representation into the two stage Cross Fusion Geometry (CFG) Transformer. Then the decoder integrates these features to produce the final dense completion.

Loss Function We train our model by minimizing the Chamfer Distance (CD) between predictions and the ground truth at three stages: the coarse points, the intermediate points from the first CFG Transformer, and the complete points. The overall loss is their weighted sum.

3.2. Hypergraph Neural Network

Hypergraph Neural Network (HGNN) extends traditional graph neural networks from pairwise graphs to hypergraphs, where a single hyperedge can connect more than two vertices and thus capture higher-order relations.

Let the hypergraph $\mathcal{G}$ be denoted as

$$\mathcal{G} = (V, E, W), \tag{1}$$

where V is the set of vertices with $|V| = n$, E is the set of hyperedges with $|E| = m$, and $W = \mathrm{diag}(w_e) \in \mathbb{R}^{m \times m}$ is the diagonal matrix of hyperedge weights.

The incidence matrix $H \in \{0, 1\}^{n \times m}$ is defined as

$$H_{ve} = \begin{cases} 1, & \text{if vertex } v \in e, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

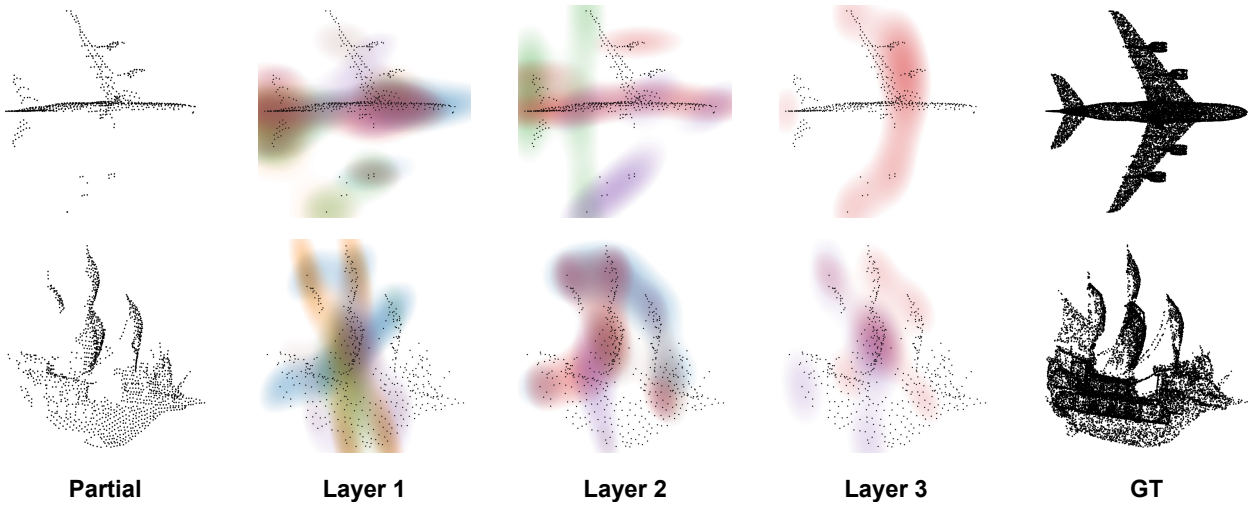

Figure 3. Visualization of hyperedges in *Hyper Refinement Stack* layers for airplane and watercraft samples. Regions with the same color indicate point sets connected by the same hyperedge. As the layer depth increases, the distance threshold τ decreases, which gradually reduces the number of hyperedges and allows them to capture progressively finer (more deeply related) correlations. Extracted relations include (1) low-order traits such as symmetry (e.g., airplane wings), (2) explicit high-order similarities (e.g., repeated sails on a watercraft), and (3) semantic high-order relations (e.g., the aerodynamic coupling between an airplane's wing and tail).

The vertex degree $d(v)$ and hyperedge degree $\delta(e)$ are given by:

$$d(v) = \sum_{e \in E} w_e H_{ve}, \quad \delta(e) = \sum_{v \in V} H_{ve}, \tag{3}$$

and their corresponding diagonal matrices are

$$D_v = \mathrm{diag}(d(v)) \in \mathbb{R}^{n \times n}, \quad D_e = \mathrm{diag}(\delta(e)) \in \mathbb{R}^{m \times m}. \tag{4}$$

The hypergraph convolution [13] is a two-stage vertex $\rightarrow$ hyperedge $\rightarrow$ vertex information flow. Specifically, each vertex first sends its feature information to the incident hyperedges, where the features of connected vertices are aggregated to update the hyperedge embeddings. Then, each vertex collects messages from its neighboring hyperedges, weighted by the corresponding hyperedge importance, to refine its own representation.

The two-stage process can be expressed in matrix as:

$$X^{(t+1)} = \sigma\big(D_v^{-1} H W D_e^{-1} H^\top X^{(t)} Q_t\big), \tag{5}$$

where $X^{(t)} \in \mathbb{R}^{n \times d_t}$ (d_t denotes the feature dimension at layer t) denotes vertex features at layer t, and Q_t is the learnable transformation matrix.

3.3. Feature Embedding

As a preparatory step for hypergraph computation, the feature embedding (based on PointNet) first builds a multiscale feature pyramid to get N_k key points $P_k \in \mathbb{R}^{N_k \times 3}$ and multi-scale features $F_m \in \mathbb{R}^{D_m \times N_k}$, where D_m is the aggregated channel dimension. Then it applies a 3D positional encoding to each key point and uses a transformer block to obtain key embeddings $E_k \in \mathbb{R}^{N_k \times D}$ that aggregate multi-scale context, where D is the channel dimension. The transformer output is fused with the pyramid descriptors by addition and a linear projection, which yields discriminative key features $F_k \in \mathbb{R}^{N_k \times 2N_k}$ that serve as the shared input to the two paths in the encoder.

3.4. Hyper Refinement Stack

This module addresses the difficulty of extracting higher-order relations from partial point clouds. To mitigate the limitation that single-pass modeling yields only limited higher-order cues, we adopt a threshold-annealed stack of hypergraph computation layers.

In detail, we stack L layers, each of which constructs a hypergraph based on feature space and performs hypergraph convolution. Let the feature map at layer ℓ be $X^{(\ell)}$, we construct a binary incidence matrix $H^{(\ell)} \in \{0,1\}^{N_\ell \times N_\ell}$, where N_ℓ is the number of vertices at layer ℓ. Taking each vertex as the centroid, the incidence is defined as:

$$H_{i,j}^{(\ell)} = \begin{cases} 1, & \big\|X_i^{(\ell)} - X_j^{(\ell)}\big\|_2 \le \tau_\ell, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The hyperedge set is formed by grouping each vertex's neighbors whose distances fall within a threshold τ_ℓ, which is linearly annealed from τ_{start} to τ_{end} ($\tau_{\text{start}} > \tau_{\text{end}}$) across the L layers:

$$\tau_\ell = \tau_{\text{start}} + \frac{\ell - 1}{L - 1}\big(\tau_{\text{end}} - \tau_{\text{start}}\big), \ell \in \{1, \ldots, L\}. \tag{7}$$

In this way, the hyperedges contain diverse vertices and are more numerous (as low-degree hyperedges are meaningless) in shallow layers, capturing broad contextual cues, while deeper layers focus on closer neighbors to get fine-grained high-order relations.

We then apply hypergraph convolution as Equation (5) with binary weights to aggregate higher-order relations and get updated features $\widetilde{X}^{(\ell)}$. After that, we apply a residual update with SiLU activation ($\mathrm{SiLU}(x) = x \cdot \sigma(x)$, where σ is the sigmoid function) and batch normalization (BN) to obtain the output features of layer $\ell + 1$:

$$X^{(\ell+1)} = X^{(\ell)} + \mathrm{BN}\Big(\mathrm{SiLU}\big(\widetilde{X}^{(\ell)}\big)\Big). \tag{8}$$

Through layer-by-layer progressive hypergraph modeling and convolution, HyperRS effectively excavates coarse-to-fine high-order correlations in incomplete point clouds, as verified in Section 4.3 and Figure 3. The high-order correlation features from each layer are progressively integrated to form a deep understanding of the point-cloud structure, which helps improve completion accuracy and the preservation of fine details.

3.5. Anchor-based Hypergraph Neural Network

To effectively model global higher-order relations in incomplete point clouds, this module adopts a two-stage sampling strategy to establish robust information pathways between key points and anchors. The core idea is that the synergy between key points and anchors, as opposed to relying on either alone, mitigates structural bias and enables the construction of a hypergraph that more comprehensively captures global dependencies across the entire shape.

Given the key points $P_k = \{p_{k,i}\}_{i=1}^{N_k}$ from the feature embedding module, the anchors $P_a = \{p_{a,j}\}_{j=1}^{N_a}$ are obtained by deterministic uniform subsampling of P_k under a budget N_a. Then we compute the Euclidean Distance (ED) between each key point and each anchor:

$$ED_{i,j} = \|p_{k,i} - p_{a,j}\|_2, \qquad ED \in \mathbb{R}^{N_k \times N_a}, \tag{9}$$

where $ED_{i,j}$ denotes the distance from key point $p_{k,i}$ to anchor $p_{a,j}$.

For each point $p_{k,i}$, we select the index set of its α nearest anchors $\mathcal{N}(i)$:

$$\mathcal{N}(i) = \underset{j \in \{1,\dots,N_a\}}{\arg \text{Top-}\alpha}\big(-ED_{i,j}\big). \tag{10}$$

Then we build the hypergraph incidence matrix $H^{(A)} \in \{0,1\}^{N \times M}$ with entries

$$H^{(A)}_{i,j} = \begin{cases} 1, & j \in \mathcal{N}(i), \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where $H^{(A)}_{i,j} = 1$ means point i is incident to anchor j.

After that, we use the input features $F_{in} \in \mathbb{R}^{N_k \times 2N_k}$ as the hypergraph vertex features and apply hypergraph convolution (Equation (5)) to obtain the final global high-order correlation features $F_{out} \in \mathbb{R}^{N_k \times 2N_k}$.

3.6. CFG Transformer

CFG Transformer is a two-level architecture designed to fuse geometric information with higher-order relations. It begins by augmenting the coarse input points P_c with 3D positional encoding, concatenating these encodings with the raw coordinates to form a PE-augmented representation $Z_{\mathrm{PE}} \in \mathbb{R}^{N_c \times D_{\mathrm{PE}}}$. This representation is then processed by a self-attention block to extract geometry-aware features $F_g \in \mathbb{R}^{N_c \times D_g}$, which align local geometry with structural relations captured at a higher order. These features are further transformed into a fused refinement representation $F_{\mathrm{ref}} \in \mathbb{R}^{N_c \times D_{\mathrm{ref}}}$, which is regressed into point-wise offsets. The final output is a completed point cloud $Y \in \mathbb{R}^{N_o \times 3}$, where N_o denotes the number of output points. Here, D_{PE}, D_g, and D_{ref} represent the channel dimensions of the PE-augmented features, geometry-aware features, and refinement features, respectively.

4. Experiment

4.1. Experiment Settings

Datasets and Evaluation Metric We use three widely adopted datasets for training and evaluation, including the PCN dataset[46], MVP dataset[26], and ShapeNet55/34 dataset[35]. Additionally, we test our method on the KITTI dataset to evaluate the generalization ability in real-world scenarios. We adopt three standard metrics for quantitative evaluation: CD-L1, CD-L2, and F1-Score@1%. For the CD metric, lower values indicate better reconstruction, while for the F1 score, the larger value is better.

Implementation Details Our model is trained on NVIDIA RTX 3090 GPUs for 420 epochs using the AdamW optimizer (batch size 64) with an initial learning rate of 2×10^{-4} and weight decay of 5×10^{-4}. A 20-epoch warm-up linearly increases the learning rate from 1×10^{-5} to 2×10^{-4}. We set the HyperRS depth to $L = 6$, linearly annealing the distance threshold from $\tau_{\text{start}} = 0.20$ to $\tau_{\text{end}} = 0.16$. A-HGNN comprises two stages with anchors/top-k pairs of $(N_{a1} = 128, k_1 = 24)$ and $(N_{a2} = 192, k_2 = 32)$.

4.2. Comparison to the State-of-the-art

Evaluation on PCN Dataset In this section, we evaluate the proposed network on the PCN dataset and compare it with previous methods. We present the quantitative results of the CD-L1 and F1-score on the PCN dataset in Table 1.

Table 1. Quantitative results on the PCN dataset. (CD-L1$\times 10^{-3}$ and F1-Score@1%)

Methods	Venue	CD-Avg (↓)	F1 (↑)	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
PCN [46]	3DV'18	9.64	0.695	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59
PoinTr [44]	ICCV'21	8.38	0.745	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29
FBNet [43]	ECCV'22	6.94	–	3.99	9.05	7.90	7.38	5.82	8.85	6.35	6.18
SeedFormer [50]	ECCV'22	6.74	0.818	3.85	9.05	8.06	7.06	5.21	8.85	6.05	5.58
SnowflakeNet [38]	TPAMI'23	7.21	0.801	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40
GTNet [48]	IJCV'23	7.15	–	4.17	9.33	8.38	7.66	5.49	9.44	6.69	6.07
AnchorFormer [6]	CVPR'23	6.59	–	3.70	8.94	7.57	7.05	5.21	8.40	6.03	5.81
SVDFormer [51]	ICCV'23	6.54	0.841	3.62	8.79	7.46	6.91	5.33	8.49	5.90	5.83
AdaPoinTr [45]	TPAMI'23	6.53	0.845	3.68	8.82	7.47	6.85	5.47	8.35	5.80	5.76
ODGNet [3]	AAAI'24	6.51	0.833	3.78	8.78	7.57	6.83	5.12	8.46	5.86	5.66
CRA-PCN [30]	AAAI'24	6.39	–	3.59	8.70	7.50	6.70	5.06	8.24	5.72	5.64
SymmCompletion [42]	AAAI'25	6.28	0.853	3.53	8.49	7.30	6.52	5.06	8.23	5.64	5.49
PointMAC [17]	NIPS'25	6.33	–	3.54	8.66	7.44	6.65	4.98	8.19	5.64	5.57
Ours	–	**6.20**	**0.858**	**3.46**	**8.48**	**7.23**	**6.49**	**4.83**	**8.10**	**5.57**	**5.48**

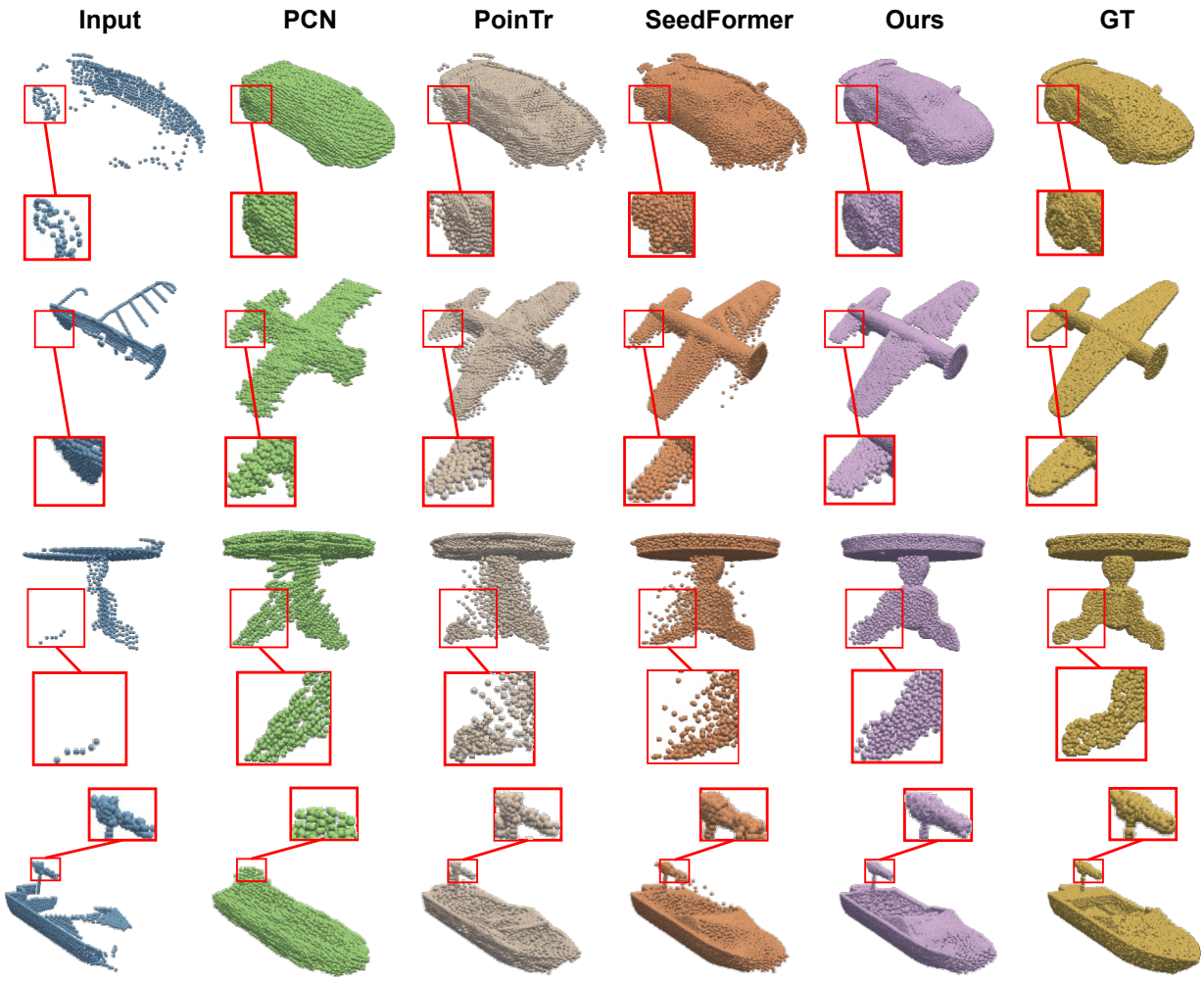

Figure 4. Visualization results on the PCN dataset. As emphasized by the zoom-in views, our Hyper-PCN preserves fine-grained structures such as the complex geometry of cars and tables, airplane propellers and tails, and watercrafts, demonstrating strong completion quality and generalization across diverse categories.

Table 2. Results on Seen ShapeNet-34 and Unseen ShapeNet-21 test set. CD-S, CD-M, and CD-H stand for CD-L2 values under the simple, median, and hard difficulty levels, respectively. (CD-L2$\times 10^{-3}$ and F1-Score@1%)

Method	Venue	Seen ShapeNet-34					Unseen ShapeNet-21				
		CD-S	CD-M	CD-H	CD-Avg (↓)	F1 (↑)	CD-S	CD-M	CD-H	CD-Avg (↓)	F1 (↑)
PCN [46]	3DV'18	1.87	1.81	2.97	2.22	0.154	3.17	3.08	5.29	3.85	0.101
TopNet [33]	CVPR'19	1.77	1.61	3.54	2.31	0.171	2.62	2.43	5.44	3.50	0.121
PFNet [16]	CVPR'20	3.16	3.19	7.71	4.68	0.347	5.29	5.87	13.33	8.16	0.322
GRNet [39]	ECCV'20	1.26	1.39	2.57	1.74	0.251	1.85	2.25	4.87	2.99	0.216
SnowflakeNet [38]	ICCV'21	0.60	0.86	1.50	0.99	0.422	0.88	1.46	2.92	1.75	0.388
PoinTr [44]	ICCV'21	0.76	1.05	1.88	1.23	0.421	1.04	1.67	3.44	2.05	0.384
SeedFormer [50]	ECCV'22	0.48	0.70	1.30	0.83	0.452	0.61	1.07	2.35	1.34	0.402
ProxyFormer [21]	CVPR'23	0.44	0.67	1.33	0.81	0.466	0.60	1.13	2.54	1.42	0.415
AdaPoinTr [45]	TPAMI'23	0.51	0.67	1.17	0.78	0.399	0.61	0.96	2.11	1.23	0.416
ODGNet [3]	AAAI'24	0.44	0.64	1.14	0.75	0.451	0.59	1.01	2.26	1.29	0.415
CRA-PCN [30]	AAAI'24	0.45	0.65	1.18	0.76	–	0.55	0.97	2.19	1.24	–
SymmCompletion [42]	AAAI'25	0.33	0.48	1.00	0.60	–	0.39	0.70	1.83	0.97	–
PointMAC [17]	NIPS'25	0.44	0.64	1.14	0.75	–	0.53	0.96	2.16	1.22	–
Ours	–	**0.32**	**0.46**	**0.98**	**0.58**	**0.571**	**0.38**	**0.68**	**1.77**	**0.94**	**0.550**

Table 3. Results on the ShapeNet55 dataset. CD-S, CD-M, and CD-H stand for CD-L2 values under the simple, median, and hard difficulty levels, respectively. (CD-L2$\times 10^{-3}$ and F1-Score@1%)

Method	CD-S	CD-M	CD-H	CD-Avg (↓)	F1 (↑)
PCN [46] (2018)	1.94	1.96	4.08	2.66	0.133
PoinTr [44] (2021)	0.58	0.88	1.79	1.09	0.464
SeedFormer [50] (2022)	0.50	0.77	1.49	0.92	0.472
SnowflakeNet [38] (2023)	0.70	1.06	1.96	1.24	0.398
ProxyFormer [21] (2023)	0.49	0.75	1.55	0.93	0.483
AdaPoinTr [45] (2023)	0.49	0.69	1.24	0.81	0.503
ODGNet [3] (2024)	0.47	0.70	1.32	0.83	0.437
CRA-PCN [30] (2024)	0.48	0.71	1.37	0.85	–
SymmCompletion [42] (2025)	0.34	0.54	1.18	0.69	–
PointMAC [17] (2025)	0.47	0.69	1.34	0.83	0.490
Ours	**0.33**	**0.51**	**1.12**	**0.65**	**0.565**

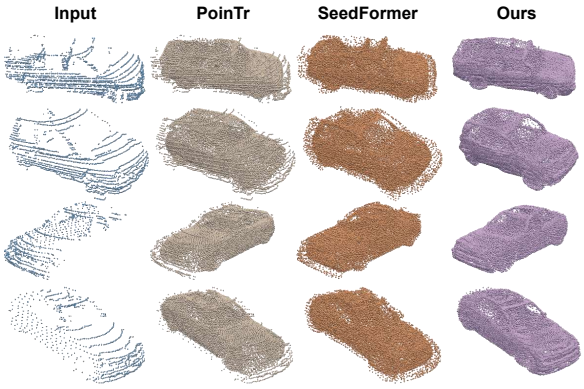

Figure 5. Qualitative results on the KITTI dataset.

Our method achieves the best CD performance in each category and the best average values for both CD and F1-score across all categories.

In addition to quantitative comparisons, we also provide visualized results in Figure 4. As shown in the figure, qualitative comparisons further highlight Hyper-PCN's advantages over PCN, PoinTr, and SeedFormer. Across cars, airplanes, tables, and watercraft, our completions exhibit cleaner topology, sharper edges, and more uniform point density, while spurious points in empty regions are markedly reduced. The zoomed insets show that Hyper-PCN preserves the structures present in the input and extends them into missing areas in a geometry-consistent manner, including wheel arches, wing tips and tail surfaces, slender table legs, and the bow and stern outlines. Moreover, our method does so without fragmentation or over-smoothing. Overall, Hyper-PCN delivers high fidelity and high consistency that align closely with the ground truth while maintaining the original geometry of the partial scan.

Evaluation on ShapeNet-55/34 Dataset Furthermore, we evaluate our method on the ShapeNet-55/34 dataset, including the ShapeNet55 dataset as well as the more challenging ShapeNet-34 seen and ShapeNet-21 unseen subsets. The test samples are categorized into three difficulty levels—easy, median, and hard—by retaining 75%, 50%, and 25% of the partial point cloud, respectively. As shown in Table 3 and Table 2, we report CD-L2 values for each method at easy (CD-S), median (CD-M), and hard (CD-H) levels. The results show that Hyper-PCN consistently outperforms previous approaches across all difficulty levels, demonstrating its effectiveness and strong generalization capability.

Evaluation on KITTI Dataset We further evaluate on real world data using the KITTI dataset. Following prior practice, the model trained on the PCN dataset is directly tested on the KITTI. The qualitative comparison results in

Table 4. Results on the MVP dataset. (CD-L2$\times 10^{-4}$ and F1-Score@1%)

Methods	PCN [46]	CRN [37]	VRCNet [26]	PDR [24]	CRA-PCN [30]	SymmCompletion [42]	PointMAC [17]	**Ours**
CD (↓)	8.65	7.25	5.82	5.66	5.33	4.89	5.24	**4.76**
F1 (↑)	0.342	0.434	0.495	0.499	0.529	0.534	0.537	**0.558**

Table 5. Quantitative results of ablation study on key components HyperRS and A-HGNN using the PCN dataset. (CD-L1$\times 10^{-3}$ and F1-Score@1%)

HyperRS	A-HGNN	CD (↓)	F1 (↑)
		6.43	0.844
✓		6.36	0.848
	✓	6.32	0.851
✓	✓	**6.20**	**0.858**

Figure 5 shows that Hyper-PCN reconstructs a more complete car geometry, maintains sharper boundaries and more uniform point density with noticeably fewer outliers. Besides, it recovers occluded regions better while preserving the visible structures in the input. Please refer to Appendix for quantitative results and analysis.

Evaluation on MVP Dataset Table 4 provides a quantitative comparison of our method against others on the MVP dataset, indicating that our Hyper-PCN can generate high-quality complete point clouds.

4.3. Visualization of HyperRS

To understand how HyperRS groups points during refinement, we visualized the hyperedges in each HyperRS layer. Figure 3 shows hyperedge connections for airplane and watercraft samples across three HyperRS layers. This progressive, layer-by-layer hypergraph modeling enables HyperRS to effectively mine coarse-to-fine high-order correlations in incomplete point clouds.

4.4. Ablation Study

We investigate the effect of Hyper-PCN's key components by conducting an ablation study in which HyperRS and A-HGNN are removed from the full model, and report the CD-L1 and F1-score on the PCN dataset. As shown in Table 5 and Figure 6, each component independently brings a clear improvement over the baseline without it. When the two are combined, it achieves the best overall performance.

We also study the depth of the HyperRS by varying the number of stacked layers. Figure 7 reports the CD-L1 and F1-Score. Performance improves steadily as L increases from 1 to 6, reaching the best CD and F1 when $L = 6$, indicating that deeper refinement aggregates coarse-to-fine higher-order cues. When the stack is deeper than six layers, CD rises while F1 falls, suggesting over-smoothing and accumulated noise as annealed hypergraphs become overly sparse and message passing deepens. We therefore adopt $L = 6$ as the default.

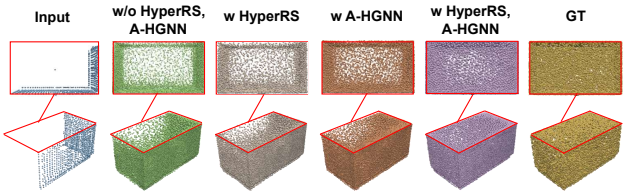

Figure 6. Qualitative results of ablation study on key components HyperRS and A-HGNN.

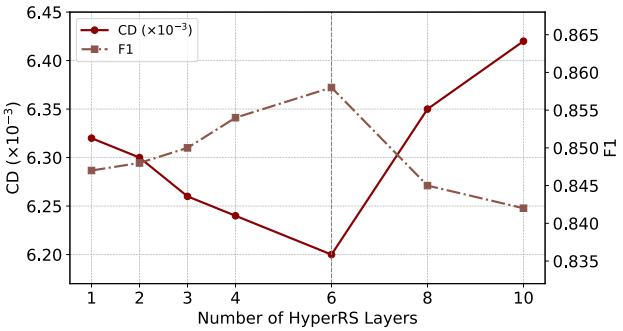

Figure 7. Ablation on number of layers in HyperRS using the PCN dataset. (CD-L1$\times 10^{-3}$ and F1-Score@1%)

5. Conclusion

In this paper, we propose Hyper-PCN, a novel point cloud completion framework that leverages hypergraph learning to model high-order correlations within incomplete point clouds. The method introduces two key components: HyperRS, which progressively captures coarse-to-fine structural relationships through layered hypergraph convolution, and A-HGNN, which employs a two-stage sampling strategy to construct collaborative hypergraphs to guide robust hypergraph construction. Extensive experiments demonstrate that Hyper-PCN consistently outperforms state-of-the-art methods across multiple benchmarks, effectively improving completion quality and preserving fine-grained details. This work highlights the importance of high-order correlation modeling and establishes hypergraphs as a powerful tool for point cloud completion.

6. Acknowledgment

This work was supported by Brain Science and Brain-like Intelligence Technology——National Science and Technology Major Project (2025ZD0217300), National Natural Science Foundation of China under grant No. 62501358.

References

[1] Emanuele Aiello, Diego Valsesia, and Enrico Magli. Cross-modal learning for image-guided point cloud shape completion. *Adv. Neural Inform. Process. Syst.*, 35:37349–37362, 2022. 2

[2] Lin Bie, Siqi Li, Yifan Feng, and Yue Gao. HyperDepth: Hypergraph-based multi-scale representation fusion for monocular depth estimation. In *Int. Conf. Comput. Vis.*, pages 5081–5090, 2025. 3

[3] Pingping Cai, Deja Scott, Xiaoguang Li, and Song Wang. Orthogonal dictionary guided shape completion network for point cloud. In *AAAI*, pages 864–872, 2024. 2, 6, 7

[4] Wu Chen, Qiuping Jiang, Wei Zhou, Long Xu, and Weisi Lin. Dynamic hypergraph convolutional network for no-reference point cloud quality assessment. *IEEE Trans. Circuit Syst. Video Technol.*, 34(10):10479–10493, 2024. 3

[5] Yang Chen, Yirun Zhou, Weizhong Zhang, and Cheng Jin. Complete structure guided point cloud completion via cluster- and instance-level contrastive learning. In *Adv. Neural Inform. Process. Syst.*, 2025. 2

[6] Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. AnchorFormer: Point cloud completion from discriminative nodes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13581–13590, 2023. 1, 2, 6

[7] Ruikai Cui, Shi Qiu, Saeed Anwar, Jiawei Liu, Chaoyue Xing, Jing Zhang, and Nick Barnes. P2C: Self-supervised point cloud completion from single partial clouds. In *Int. Conf. Comput. Vis.*, pages 14351–14360, 2023. 2

[8] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Trans. Intell. Transp. Syst.*, 23(2):722–739, 2021. 1

[9] Chenghao Fang, Jianqing Liang, Jiye Liang, Hangkun Wang, Kaixuan Yao, and Feilong Cao. Multi-modal point cloud completion with interleaved attention enhanced transformer. In *IJCAI*, 2025. 2

[10] Ben Fei, Weidong Yang, Wen-Ming Chen, Zhijun Li, Yikang Li, Tao Ma, Xing Hu, and Lipeng Ma. Comprehensive review of deep learning-based 3D point cloud completion processing and analysis. *IEEE Trans. Intell. Transp. Syst.*, 23 (12):22862–22883, 2022. 1

[11] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI*, pages 3558–3565, 2019. 2

[12] Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. Hypergraph learning: Methods and practices. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(5): 2548–2566, 2020. 2

[13] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. HGNN+: General hypergraph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3):3181–3199, 2022. 2, 4

[14] Yan Han, Peihao Wang, Souvik Kundu, Ying Ding, and Zhangyang Wang. Vision HGNN: An image is more than a graph of nodes. In *Int. Conf. Comput. Vis.*, pages 19878–19888, 2023. 3

[15] Sangmin Hong, Mohsen Yavartanoo, Reyhaneh Neshatavar, and Kyoung Mu Lee. ACL-SPC: Adaptive closed-loop system for self-supervised point cloud completion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9435–9444, 2023. 2

[16] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. PF-Net: Point fractal network for 3D point cloud completion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7662–7670, 2020. 7

[17] Linlian Jiang, Rui Ma, Li Gu, Ziqiang Wang, Xinxin Zuo, and Yang Wang. PointMAC: Meta-learned adaptation for robust test-time point cloud completion. In *Adv. Neural Inform. Process. Syst.*, 2025. 6, 7, 8

[18] Ping Jiang, Xiaoheng Deng, Leilei Wang, Zailiang Chen, and Shichao Zhang. Hypergraph representation for detecting 3D objects from noisy point clouds. *IEEE Trans. Knowl. Data. En.*, 35(7):7016–7029, 2022. 2

[19] Mengqi Lei, Siqi Li, Yihong Wu, Han Hu, You Zhou, Xinhu Zheng, Guiguang Ding, Shaoyi Du, Zongze Wu, and Yue Gao. YOLOv13: Real-time object detection with hypergraph-enhanced adaptive visual perception. *arXiv*, 2025. 3

[20] Mengqi Lei, Yihong Wu, Siqi Li, Xinhu Zheng, Juan Wang, Yue Gao, and Shaoyi Du. SoftHGNN: Soft hypergraph neural networks for general visual recognition. *arXiv*, 2025. 2

[21] Shanshan Li, Pan Gao, Xiaoyang Tan, and Mingqiang Wei. ProxyFormer: Proxy alignment assisted point cloud completion with missing part sensitive transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9466–9475, 2023. 1, 2, 7

[22] Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A Chapman, Dongpu Cao, and Jonathan Li. Deep learning for LiDAR point clouds in autonomous driving: A review. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(8):3412–3432, 2020. 1

[23] Moyun Liu, Ziheng Yang, Bing Chen, Youping Chen, Jingming Xie, Lei Yao, Lap-Pui Chau, Jiawei Du, and Joey Tianyi Zhou. MAENet: Boost image-guided point cloud completion more accurate and even. *Inf. Fusion*, 122: 103179, 2025. 2

[24] Zhaoyang Lyu, Zhifeng Kong, Liang Pan, Dahua Lin, et al. A conditional point diffusion-refinement paradigm for 3D point cloud completion. In *Int. Conf. Learn. Represent.*, 2022. 8

[25] Zhiliang Ma and Shilong Liu. A review of 3D reconstruction techniques in civil engineering and their applications. *Adv. Eng. Inform.*, 37:163–174, 2018. 1

[26] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8524–8533, 2021. 5, 8

[27] François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Found. Trends Robot.*, 4(1):1–104, 2015. 1

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 652–660, 2017. 1, 2

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inform. Process. Syst.*, 2017. 1

[30] Yi Rong, Haoran Zhou, Lixin Yuan, Cheng Mei, Jiahao Wang, and Tong Lu. CRA-PCN: Point cloud completion with intra- and inter-level cross-resolution transformers. In *AAAI*, pages 4676–4685, 2024. 1, 2, 6, 7, 8

[31] Weijing Shi and Raj Rajkumar. Point-GNN: Graph neural network for 3D object detection in a point cloud. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1711–1719, 2020. 2, 3

[32] András Sóbester and Alexander IJ Forrester. *Aircraft aerodynamic design: geometry and optimization*. John Wiley & Sons, 2014. 2

[33] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. TopNet: Structural point cloud decoder. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 383–392, 2019. 2, 7

[34] Tencent Hunyuan3D Team. Hunyuan3D-Omni: A unified framework for controllable generation of 3D assets. *arXiv*, 2025. 2

[35] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 2442–2447. IEEE, 2017. 1, 5

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 30, 2017. 1, 2

[37] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 790–799, 2020. 8

[38] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflake point deconvolution for point cloud completion and generation with skip-transformer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(5):6320–6338, 2023. 1, 2, 6, 7

[39] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. GRNet: Gridding residual network for dense point cloud completion. In *Eur. Conf. Comput. Vis.*, pages 365–381. Springer, 2020. 7

[40] Hang Xu, Chen Long, Wenxiao Zhang, Yuan Liu, Zhen Cao, Zhen Dong, and Bisheng Yang. Explicitly guided information interaction network for cross-modal point cloud completion. In *Eur. Conf. Comput. Vis.*, pages 414–432, 2024. 2

[41] Xixia Xu, Qi Zou, and Xue Lin. Adaptive hypergraph neural network for multi-person pose estimation. In *AAAI*, pages 2955–2963, 2022. 3

[42] Hongyu Yan, Zijun Li, Kunming Luo, Li Lu, and Ping Tan. SymmCompletion: High-fidelity and high-consistency point cloud completion with symmetry guidance. In *AAAI*, pages 9094–9102, 2025. 2, 6, 7, 8

[43] Xuejun Yan, Hongyu Yan, Jingjing Wang, Hang Du, Zhihong Wu, Di Xie, Shiliang Pu, and Li Lu. FBNet: Feedback network for point cloud completion. In *Eur. Conf. Comput. Vis.*, pages 676–693. Springer, 2022. 1, 2, 6

[44] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PoinTr: Diverse point cloud completion with geometry-aware transformers. In *Int. Conf. Comput. Vis.*, pages 12498–12507, 2021. 1, 2, 6, 7

[45] Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou. AdaPoinTr: Diverse point cloud completion with adaptive geometry-aware transformers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12):14114–14130, 2023. 1, 2, 6, 7

[46] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In *Int. Conf. 3D Vis.*, pages 728–737. IEEE, 2018. 2, 5, 6, 7, 8

[47] Songyang Zhang, Shuguang Cui, and Zhi Ding. Hypergraph spectral clustering for point cloud segmentation. *IEEE Sign. Process. Letters*, 27:1655–1659, 2020. 3

[48] Shengping Zhang, Xianzhu Liu, Haozhe Xie, Liqiang Nie, Huiyu Zhou, Dacheng Tao, and Xuelong Li. Learning geometric transformation for point cloud completion. *Int. J. Comput. Vis.*, 131(9):2425–2445, 2023. 6

[49] Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. View-guided point cloud completion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15890–15899, 2021. 2

[50] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. SeedFormer: Patch seeds based point cloud completion with upsample transformer. In *Eur. Conf. Comput. Vis.*, pages 416–432. Springer, 2022. 1, 2, 6, 7

[51] Zhe Zhu, Honghua Chen, Xing He, Weiming Wang, Jing Qin, and Mingqiang Wei. SVDFormer: Complementing point cloud via self-view augmentation and self-structure dual-generator. In *Int. Conf. Comput. Vis.*, pages 14508–14518, 2023. 1, 2, 6

[52] Zhe Zhu, Liangliang Nan, Haoran Xie, Honghua Chen, Jun Wang, Mingqiang Wei, and Jing Qin. CSDN: Cross-modal shape-transfer dual-refinement network for point cloud completion. *IEEE Trans. Vis. Comput. Graph.*, 30(7):3545–3563, 2023. 2

Hyper-PCN: Hypergraph-Based Point Cloud Completion via High-Order Correlation Modeling

Supplementary Material

7. More Results on KITTI

Following previous works, we report the quantitative results of MMD and FD on the KITTI dataset in Table 6. Although our method does not achieve the best numerical performance, it produces better visual results; additional visualizations are shown in Figure 8. It is worth noting that the KITTI dataset lacks ground truth, therefore the quantitative metrics are computed using reference point clouds sampled from the PCN dataset, and distribution differences between the two datasets may affect the accuracy of these metrics. Consequently, we place more emphasis on visual evaluation and consider the results on the ShapeNet-21 unseen subset to better reflect the model's generalization ability.

Table 6. Quantitative comparison on the KITTI dataset in terms of Fidelity Distance (FD) and Minimal Matching Distance (MMD)

Methods	PoinTr	SeedFormer	SVDFormer	Ours
FD (↓)	**0.0**	1.45	11.3	1.25
MMD (↓)	8.21	1.09	**0.97**	3.42

8. Additional Ablation Studies

Additional Ablation Studies on HyperRS To investigate the sensitivity of HyperRS to the filtering thresholds, we ablate the start and end thresholds $(\tau_{\text{start}}, \tau_{\text{end}})$. As reported in Table 7, setting $(\tau_{\text{start}}, \tau_{\text{end}}) = (0.2, 0.16)$ achieves the best trade-off among all configurations, with a CD of 6.20 and an F1 score of 0.858. A smaller threshold pair $(0.1, 0.06)$ is less effective at suppressing noisy responses, while larger thresholds such as $(0.3, 0.26)$ and $(0.4, 0.26)$ tend to over-filter fine structures, both leading to performance degradation. These results indicate that using a moderate level of thresholding in HyperRS is crucial for balancing noise suppression and the preservation of geometric details.

Table 7. Ablation studies on the HyperRS thresholds $(\tau_{\text{start}}, \tau_{\text{end}})$.

$(\tau_{\text{start}}, \tau_{\text{end}})$	CD (↓)	F1 (↑)
(0.1, 0.06)	6.22	0.855
(0.2, 0.16)	**6.20**	**0.858**
(0.3, 0.26)	6.24	0.853
(0.4, 0.26)	6.25	0.851

Additional Ablation Studies on A-HGNN We further ablate the number of neighbors used in the two A-HGNNs, denoted as $(\text{Topk}_1, \text{Topk}_2)$. As shown in the Table 8, setting $(\text{Topk}_1, \text{Topk}_2) = (24, 36)$ yields the best performance, achieving a CD of 6.20 and an F1 score of 0.858. A smaller neighborhood size $(12, 18)$ is insufficient to capture rich high-order relations, leading to slightly worse reconstruction quality. In contrast, larger settings such as $(36, 48)$ and $(48, 72)$ introduce redundant or noisy connections, which also cause mild performance degradation. These results suggest that choosing a moderate number of neighbors for the two anchor hypergraphs strikes a better balance between contextual aggregation and noise suppression.

Table 8. Ablation studies on the top-k neighbors of A-HGNNs.

$(Topk_1, Topk_2)$	CD (↓)	F1 (↑)
(12, 18)	6.23	0.853
(24, 36)	**6.20**	**0.858**
(36, 48)	6.22	0.854
(48, 72)	6.24	0.851

9. Discussion on the Advantage of Hypergraph

Conceptual Motivation Conventional graph convolution models relationships through pair-wise interactions. In point cloud completion, missing regions frequently break such pair-wise connectivity, limiting reliable information propagation. Hypergraph learning, in contrast, explicitly models group-wise, high-order correlations by connecting a set of vertices within a single hyperedge. Even if parts of a structure are missing, the remaining pieces can still form a hyperedge, allowing the network to infer the properties of the missing geometry from the group context.

Ablation Design and Empirical Validation We conduct a comparative study by replacing HyperRS and A-HGNN modules with standard, widely-used counterparts, while maintaining a comparable computational budget to ensure fairness:

- GCN-based: We replace hypergraph convolution with graph convolution in these modules to represent pair-wise correlations. To ensure a fair comparison, we employ the same graph construction strategy as our method.
- Set Abstraction-based: We replace the modules with PointNet or PointNeXt.

The results on PCN dataset are shown in Table 9. Despite using comparable or larger model capacity, both GCN- and SA-based variants underperform the proposed hypergraph-based design. We further increase the stacking depth of GCN layers within HyperRS and observe a bit improvement when L=12. Nevertheless, the GCN-based variant still exhibits a clear gap compared to the hypergraph-based design, suggesting that explicit group-wise modeling is more effective than deep stacks of pair-wise convolutions.

Table 9. Ablation of core modules by replacement.

(a) HyperRS.

Base	FLOPs(G)	Params(M)	CD(↓)
GCN (L=6)	6.47	12.61	6.28
GCN (L=12)	12.94	25.22	6.27
GCN (L=18)	19.41	37.83	6.33
PointNet	17.24	2.11	6.29
PointNeXt	146.42	17.85	6.35
Ours	**3.23**	**6.31**	**6.20**

(b) A-HGNN.

Base	FLOPs(G)	Params(M)	CD(↓)
GCN	25.82	2.10	6.26
PointNet	12.93	1.06	6.36
PointNeXt	13.84	1.12	6.39
Ours	**0.54**	**1.05**	**6.20**

10. Additional Visualizations

Additional Visualizations on PCN dataset We provide some additional visualization results on the PCN dataset (Figure 9 and Figure 10). The results show that HyperPCN can generate high-quality completed point clouds in various complex scenarios, accurately restoring details and structure.

Additional Visualizations of HyperRS In addition, we present more visualizations of the hyperedges in the HyperRS across different objects. Figure 11 illustrates HyperRS's hyperedge connections on different objects, further validating HyperRS's effectiveness in mining high-order correlation information from incomplete point clouds.

Additional Visualizations of A-HGNN To validate A-HGNN's effectiveness in capturing global high-order correlations, we visualized the hyperedges formed by anchor connections in the two A-HGNN modules in the encoder, as shown in Figure 12. Regions of the same color indicate the point sets connected by the same anchor. It can be observed that A-HGNN's hyperedges mainly connect the edge parts of neighboring missing regions, and the connected regions of the two A-HGNN modules are partly complementary; together they cover the edge information of missing regions across the whole shape. The edges of missing regions typically contain rich structural cues and semantic information that are crucial for inferring the shape and details of the missing parts. Through anchor-guided hypergraph construction, A-HGNN can effectively capture these global high-order correlations, thereby improving the quality and consistency of point cloud completion.

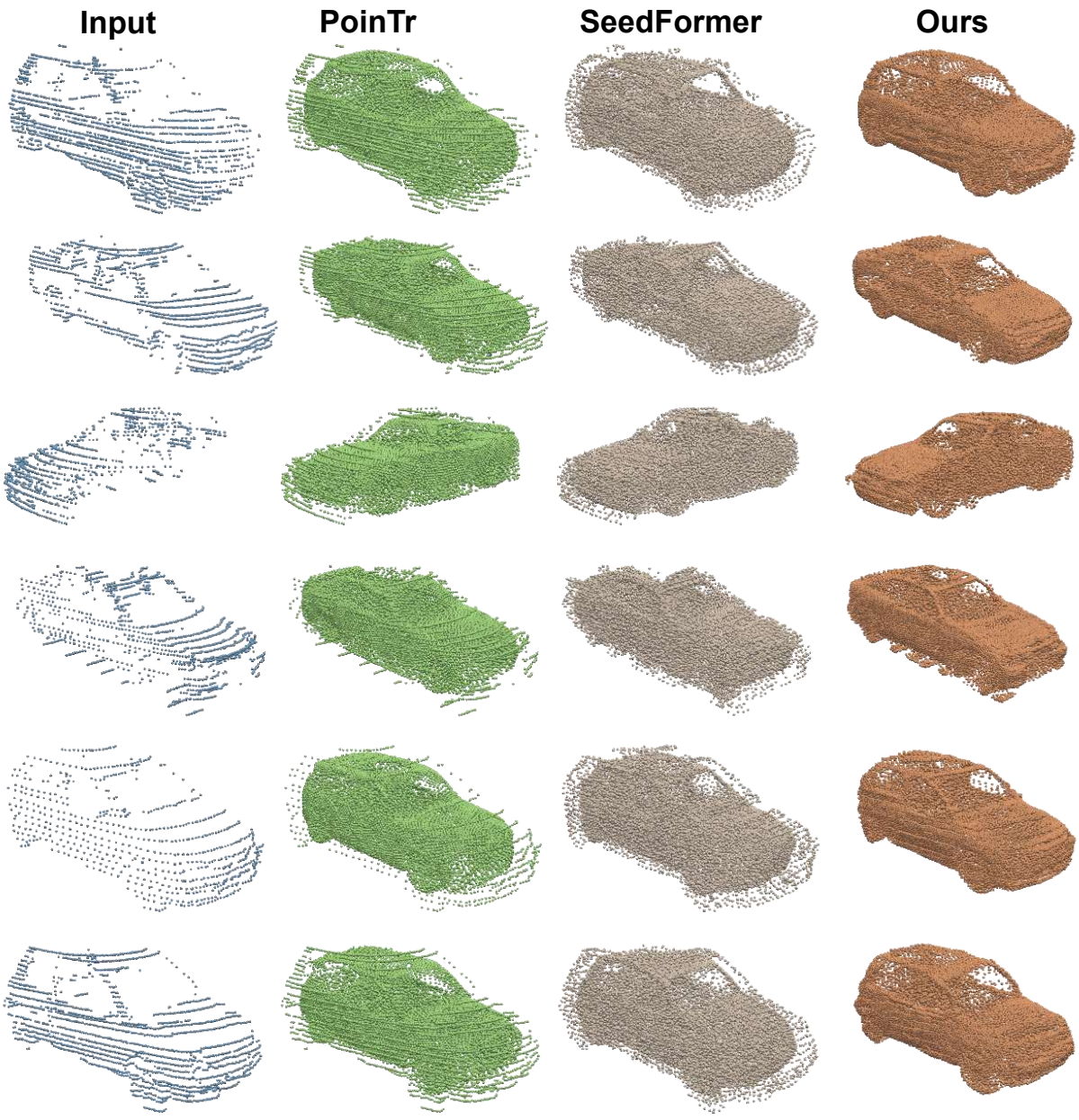

Figure 8. Additional visualizations on KITTI dataset.

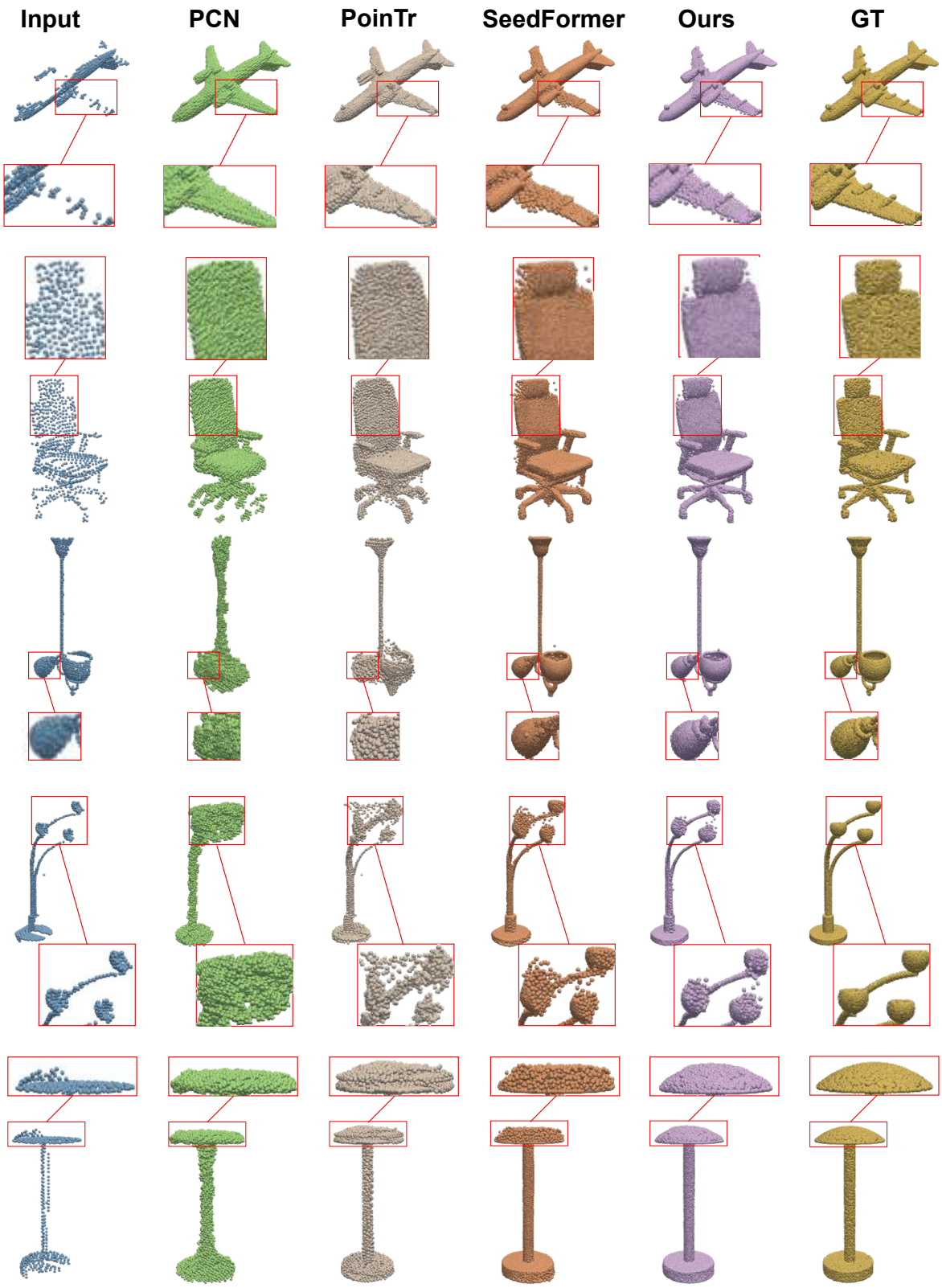

Figure 9. Additional visualizations on PCN dataset.

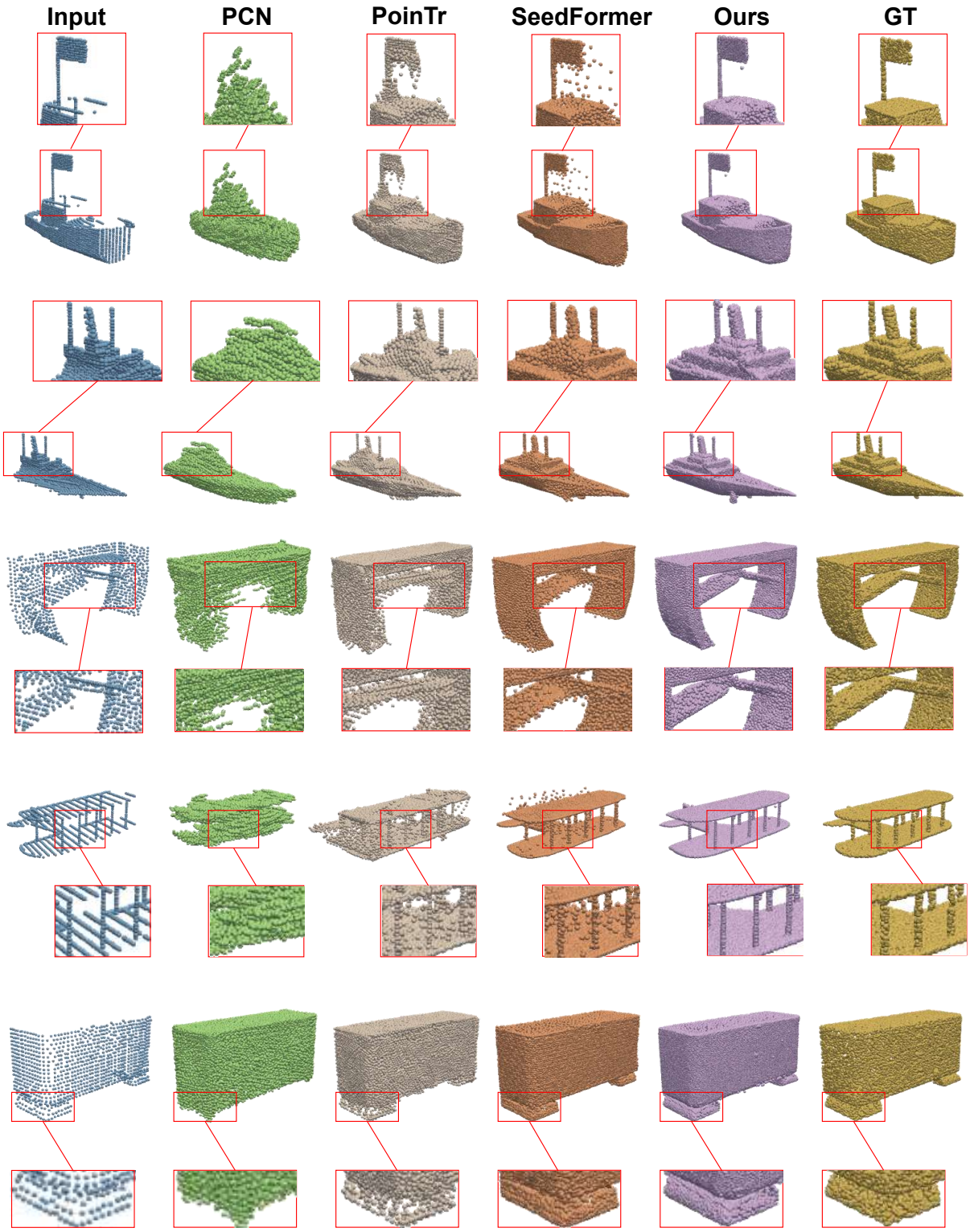

Figure 10. Additional visualizations on PCN dataset.

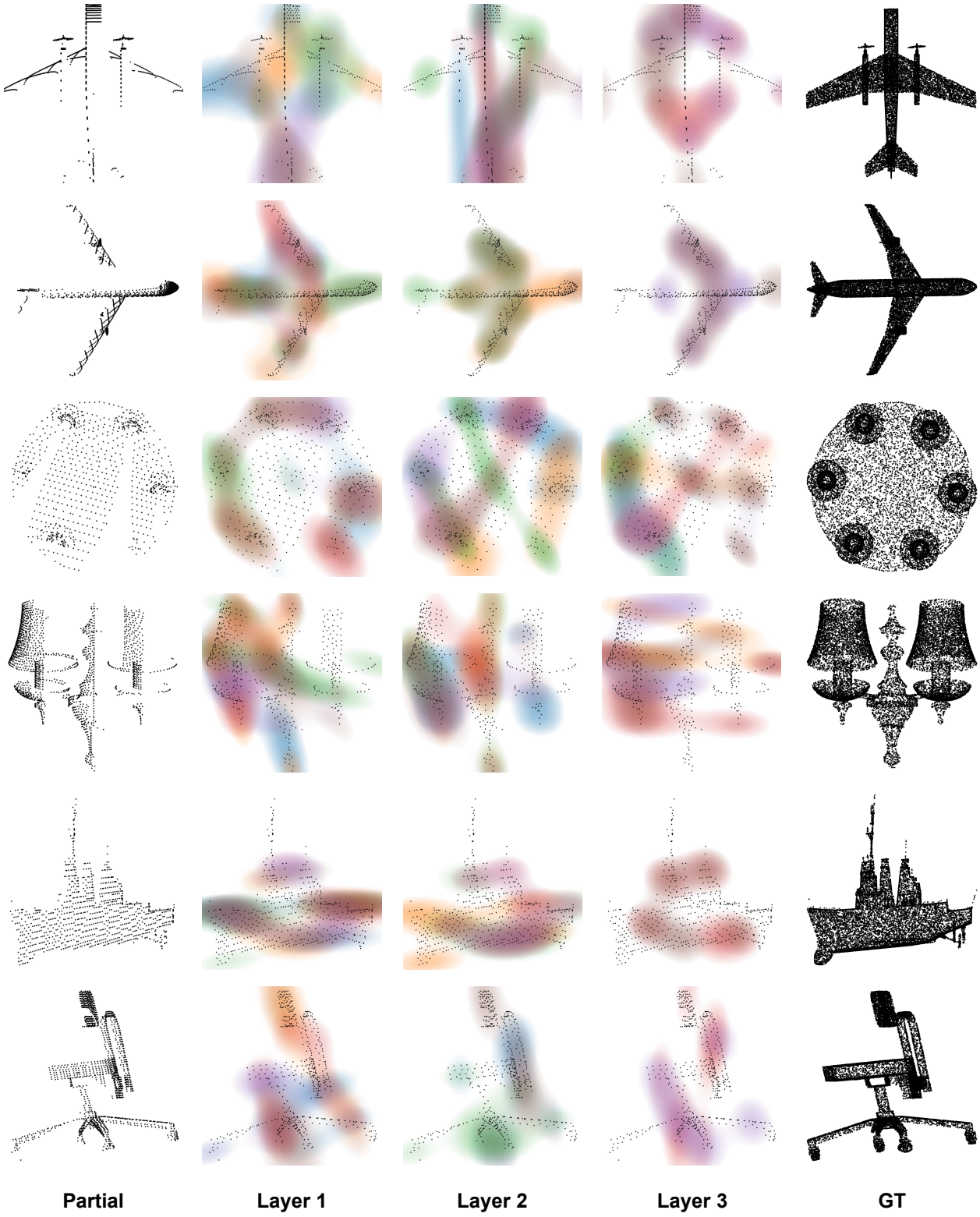

Figure 11. Additional visualizations of hyperedges in *Hyper Refinement Stack* layers.

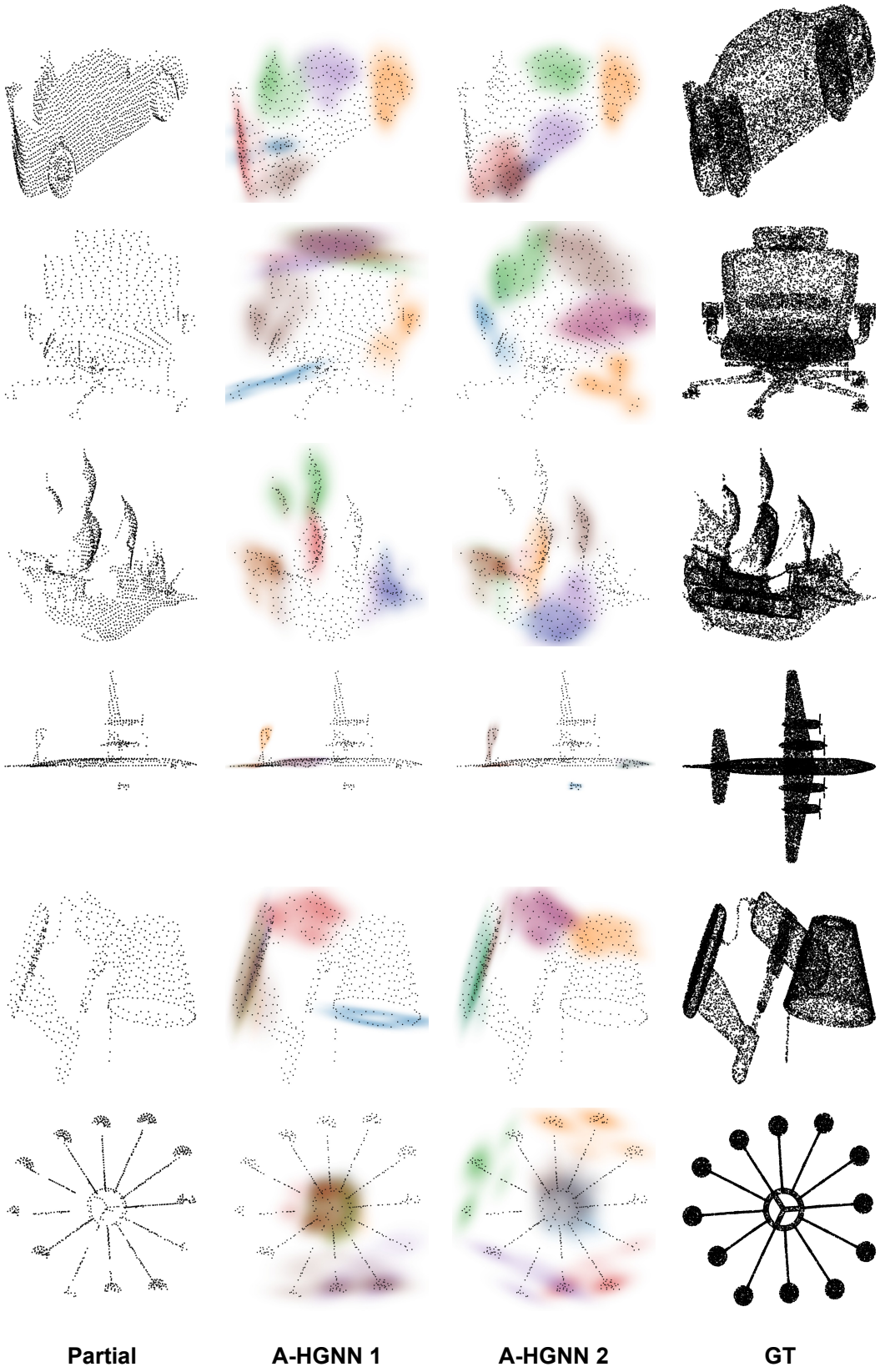

Figure 12. Additional visualizations of hyperedges in *A-HGNN*s.