HIG-PCN: HIGHER-ORDER STRUCTURE-GUIDED POINT CLOUD COMPLETION

Anonymous authors
Paper under double-blind review

ABSTRACT

Point cloud completion aims to recover complete object geometry from partial observations, supporting 3D shape understanding and robotic interaction when occlusion and limited viewpoints prevent complete scans. Existing methods commonly combine local geometric features with global shape context to infer missing surfaces. However, when an entire component is unobserved, nearby points may provide insufficient evidence, while a global representation may not explicitly preserve the relationships among the visible regions that constrain its geometry. For example, the remaining frame and bars of a partially observed chair back can jointly constrain the placement and arrangement of missing supports. Recovering such structures therefore requires considering how multiple regions relate to one another, rather than treating each missing region independently. This motivates explicit modeling of higher-order relationships that capture shared geometric context among sets of points. We propose HIG-PCN, a point cloud completion framework that represents these relationships using hypergraphs, where each hyperedge associates multiple related points. Its central idea is to combine local geometric groups with learned nonlocal groups and carry their shared context from shape encoding through point generation and refinement. In this way, evidence distributed across visible regions remains available when reconstructing unobserved geometry, without requiring semantic part annotations. Experiments assess completion accuracy and surface coverage on standard benchmarks, while controlled analyses show that both local and nonlocal group correspondences influence the reconstructed geometry.

1 INTRODUCTION

Point clouds provide a direct representation of three-dimensional geometry for shape understanding and robotic interaction. In practice, depth cameras and LiDAR sensors capture only the surfaces visible from their viewpoints, leaving substantial regions unobserved under occlusion. Point cloud completion seeks to recover complete object geometry from these partial measurements. Beyond filling small gaps, it must infer surfaces and components for which little or no direct evidence is available. A useful reconstruction must therefore agree with the observations while remaining compatible with the structure of the whole object.

Existing methods typically encode an incomplete shape and decode a complete point set (Yuan et al., 2018). Progressive decoders and geometry-aware transformers combine local surface features with broader shape context, improving the organization and detail of generated points (Xiang et al., 2021; Yu et al., 2021; Zhou et al., 2022; Yu et al., 2023). These representations are effective for learning shape priors and aggregating information across the input. A difficulty remains when a missing component cannot be inferred from its immediate surroundings alone: the evidence that constrains it may be spread across several disconnected visible regions. An overall shape summary does not necessarily retain explicit associations among those regions or indicate which observations should be considered together.

A partially observed chair back illustrates this difficulty. Its remaining frame and bars provide complementary information about the extent, orientation, and spacing of missing supports. Each visible fragment is ambiguous in isolation, but their joint arrangement can narrow the plausible reconstructions. The relevant cue is therefore a relation among sets of points, not simply the presence

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

of a nearby surface. We refer to these set-level associations as *higher-order relationships*. Local groups can collect evidence about surface continuity, while nonlocal groups can combine related observations from spatially separated regions. Explicit grouping can complement local features and global shape context by recording which observations contribute to a shared representation.

Hypergraphs provide a natural representation of this idea: a hyperedge connects multiple points within one relation, with point-to-group memberships specifying which observations are aggregated together (Feng et al., 2019; Gao et al., 2022). Hypergraph-based completion has also demonstrated the relevance of group relations to this task (Li et al., 2026b). The central question here is how to make group-level evidence useful *during reconstruction*. If group features are reduced to a single global descriptor, individual group representations are no longer directly accessible during point generation. Missing geometry must be inferred and then refined using the available context, so group information should remain accessible at both stages.

We propose HIG-PCN (Figure 1) to incorporate this principle into point cloud completion. The model combines coordinate-based groups that capture local geometry with learned groups that associate feature-related points beyond spatial proximity. Its decoder uses group-level features alongside point-level features when generating an initial shape and refining the dense output. The core idea is thus to use higher-order information not only to describe the incomplete input, but also to guide the construction of unobserved geometry. The groups are learned without semantic part annotations; they provide task-dependent context rather than an assumed decomposition into named parts.

We evaluate geometric accuracy and surface coverage, and use module ablations and controlled membership perturbations to examine how group information affects the prediction. These analyses distinguish the use of higher-order features from the stronger claim that individual missing components are recovered correctly, which requires separate structural evaluation.

Our main contributions are:

- **Local and nonlocal group modeling.** A dual-scale hypergraph encoder combines spatial neighborhoods with learned feature-based groups, representing complementary sources of completion evidence without part labels.
- **Group-guided reconstruction.** The decoder uses group context in both initial point generation and dense refinement, making higher-order information available as missing geometry is constructed.
- **Accuracy and mechanism analysis.** Benchmark comparisons, module ablations, and controlled membership perturbations examine reconstruction quality and the influence of both hypergraph scales. Structural-region diagnostics identify remaining component-recovery failures.

2 Related Work

Point cloud completion. Early encoder–decoder methods learn a global representation and regress complete coordinates (Yuan et al., 2018). Structural decoders organize output points as trees (Tchapmi et al., 2019), deform grids (Yang et al., 2018), or use voxel–point transformations (Xie et al., 2020). VRCNet combines a probabilistic latent model with relational enhancement and introduces the MVP benchmark used in our experiments (Pan et al., 2021). Recent decoders use progressive splitting (Xiang et al., 2021), hierarchical folding (Wen et al., 2020), recurrent feedback (Yan et al., 2022), and transformer proxies (Yu et al., 2021; Zhou et al., 2022; Yu et al., 2023), discriminative anchors (Chen et al., 2023), missing-part-sensitive proxies (Li et al., 2023), and cross-resolution interaction (Rong et al., 2024). Other methods move observed points along learned paths (Wen et al., 2023), use self-supervised pretraining and semantic refinement (Xu et al., 2023), or introduce an observed-to-missing interface as a structural prior (Wu et al., 2025). Adaptive point graphs vary the neighborhood size with local geometric complexity (Shu et al., 2025), while prototype-based models target generalization to unseen categories (Yu et al., 2026). At the loss level, flexible weighting of Chamfer distance balances local accuracy and global coverage (Li et al., 2026a). HIG-PCN focuses on a complementary question: how to encode group-wise relations from incomplete geometry and make those representations available to both sparse and dense point generation.

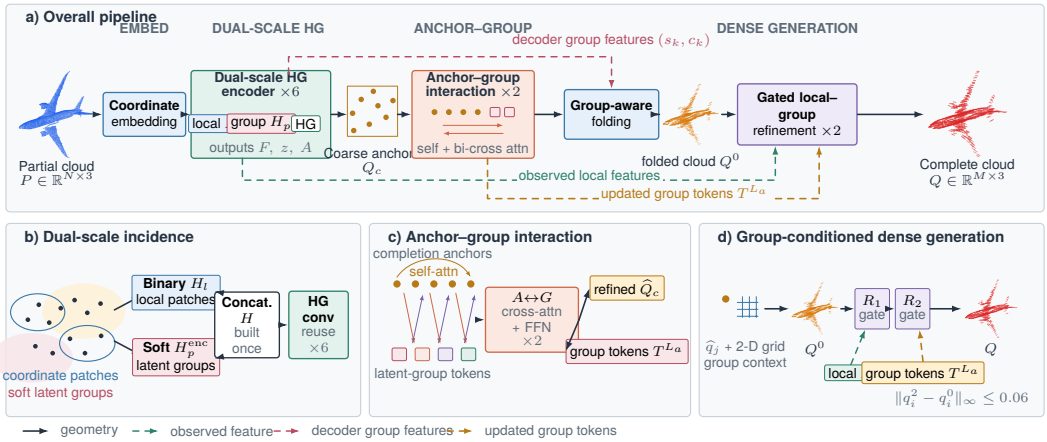

Figure 1: Overview of HIG-PCN. (a) The full pipeline uses higher-order information from dual-scale encoding through anchor–group interaction and dense completion. (b)–(d) show dual-scale incidence construction, bidirectional anchor–group interaction, and group-conditioned generation. Decoder-side group features condition folding, while the updated group tokens are reused during dense refinement. Solid arrows trace geometry; colored dashed arrows distinguish observed features, decoder-side group features, and updated group tokens.

Hypergraph learning for 3D structure. Graph neural networks aggregate pairwise edges, whereas one hyperedge can join an arbitrary point set. HGNN formalizes convolution through normalized incidence matrices (Feng et al., 2019). Geometry-aware prototype hypergraphs have also transferred relations between known and novel semantic classes for point-cloud segmentation (Zhang et al., 2026). Our groups receive no class or part supervision and are learned only through the completion objective. The closest completion method constructs threshold-annealed feature-space hypergraphs and an anchor-based hypergraph neural network (Li et al., 2026b). HIG-PCN instead combines fixed coordinate-space neighborhoods with soft latent-group incidence, lets predicted anchors and group tokens exchange information, and reuses group features in folding and point-wise dense refinement. The incidence matrices are held fixed within each forward pass, which also enables controlled row-permutation tests. The design follows the standard incidence-based formulation of hypergraph learning (Gao et al., 2022; 2023) but changes how the hyperedges are constructed and how their features are used by the decoder.

3 Method

3.1 Problem and Overview

Given a partial cloud $P = \{p_i\}_{i=1}^{N} \subset \mathbb{R}^3$, the model predicts a complete cloud $Q = \{q_j\}_{j=1}^{M}$. We use $N = M = 2048$ on MVP. A multi-frequency coordinate embedding maps each point to

$$\gamma(p) = [p, \{\sin(2^b p), \cos(2^b p)\}_{b=0}^{B-1}], \tag{1}$$

followed by an MLP to obtain $F^0 \in \mathbb{R}^{N\times C}$. HIG-PCN then encodes group-wise structure, predicts M_c coarse anchors, and generates $U = M/M_c$ points per anchor. The decoder contains anchor–group interaction, group-aware folding, and group-conditioned dense refinement. Component ablations isolate these operations, while an auxiliary loss on the folded point cloud tests whether direct supervision of the intermediate shape improves dense refinement.

3.2 Dual-Scale Hypergraph Encoder

Geometric patch hyperedges. We sample E_l centers by farthest-point sampling and connect each center to its k nearest input points. This gives a binary incidence matrix $H_l \in \{0,1\}^{N\times E_l}$. The first FPS center is selected randomly during training. At inference, it is the point farthest from the cloud centroid, which makes hyperedge construction deterministic.

Encoder latent-group hyperedges. Let $\mathcal{R} = \{r_s\}_{s=1}^{K}$ be learnable group prototypes. Each normalized feature $\bar{f}_i$ receives a soft membership over the prototypes:

$$H_p^{\text{enc}}(i,s) = \frac{\exp(\bar{f}_i^\top \bar{r}_s/\tau)}{\sum_{t=1}^{K} \exp(\bar{f}_i^\top \bar{r}_t/\tau)}. \tag{2}$$

The encoder incidence matrix is $H = [H_l, H_p^{\text{enc}}]$. It is constructed once per input and reused across the encoder blocks, so the hyperedge memberships do not change within a forward pass. Unlike hard clustering, Equation 2 remains differentiable and assigns nonzero membership weights to multiple groups.

Message passing. For vertex and hyperedge degree matrices D_v and D_e, one block computes

$$\begin{aligned} \mathcal{P}_H &= D_v^{-1/2} H D_e^{-1} H^\top D_v^{-1/2}, \\ G &= \mathcal{P}_H F W, \\ \widetilde{F} &= \text{Drop}(\text{ReLU}(\text{LN}(G))), \\ F^+ &= \text{LN}\Big(F + \text{FFN}(\widetilde{F})\Big). \end{aligned} \tag{3}$$

Because $\mathcal{P}_H$ is positive semidefinite and similar to a row-stochastic matrix, its spectrum lies in $[0,1]$. Thus incidence aggregation is non-expansive in Frobenius norm before the learned projection and residual FFN; Appendix C gives the proof. Stacking L blocks gives F; concatenated max/mean pooling yields $z \in \mathbb{R}^{2C}$. A separate group-assignment head then predicts the memberships used by the decoder:

$$A(i,k) = \text{softmax}_k(W_p F_i). \tag{4}$$

Separating A from H_p^{enc} allows encoder aggregation and decoder conditioning to learn different soft assignments. Group features and centers are

$$s_k = \sum_i \widehat{A}(i,k) F_i, \qquad c_k = \sum_i \widehat{A}(i,k) p_i, \tag{5}$$

where $\widehat{A}$ is normalized over points. A balance loss discourages the assignment of most points to a single group.

3.3 Anchor–Group Interaction

A global decoder first predicts coarse anchors $Q_c \in \mathbb{R}^{M_c \times 3}$. For anchor q_j, inverse-distance interpolation obtains local feature ℓ_j from F. Anchor and group tokens are initialized as

$$a_j^0 = \phi_a([z, \ell_j, q_j]), \qquad t_k^0 = \phi_p([z, s_k, c_k]). \tag{6}$$

Each interaction block applies anchor self-attention, anchor-to-group attention, group-to-anchor attention, and token-wise feed-forward layers. The two cross-attention directions update the anchors and group tokens using each other. After L_a blocks, the updated group tokens T^{L_a} are retained for dense refinement. A bounded update produces

$$\widehat{q}_j = q_j + \sigma(g_a([a_j^{L_a}, q_j]))\, \alpha_a \tanh(d_a([a_j^{L_a}, q_j])). \tag{7}$$

3.4 Group-Aware Folding

Each refined anchor $\widehat{q}_j$ receives a 2D folding grid point u. A base decoder predicts

$$q_{j,u}^0 = \widehat{q}_j + d_f([z, \ell_j, \widehat{q}_j, u]). \tag{8}$$

The anchor also attends to the decoder-side group features and centers from Equation 5, yielding context h_j. A gated group-conditioned residual corrects the base point:

$$\begin{aligned} q_{j,u}^1 &= q_{j,u}^0 + \sigma(g_f(x_{j,u}))\, \alpha_f \tanh(d_p(x_{j,u})), \\ x_{j,u} &= [z, \ell_j, h_j, \widehat{q}_j, u]. \end{aligned} \tag{9}$$

This separation keeps the base folding decoder trainable when the learned group assignments are not yet informative.

3.5 Progressive Local–Group Refinement

We reindex the group-aware folded points $q_{j,u}^1$ as $Q^0 = \{q_i^0\}_{i=1}^M$. HIG-PCN then applies L_r pointwise refinement stages. At stage r, each dense point queries a local observed feature ℓ_i^r and attends to the updated group tokens T^{L_a}, producing h_i^r. It is updated by

$$\begin{aligned} x_i^r &= [z, \ell_i^r, h_i^r, q_i^r], \\ q_i^{r+1} &= q_i^r + \sigma(g_r(x_i^r))\, \alpha_r \tanh(d_r(x_i^r)). \end{aligned} \tag{10}$$

Neighbor indices are found in detached chunks, while inverse-distance weights and selected features remain differentiable. This avoids retaining a $B \times M \times N$ distance tensor without changing the fixed-neighbor interpolation. Final residual layers are zero-initialized and gate biases are -2.5, so adding the refiner to the matched model without dense refinement reproduces its prediction before fine-tuning.

3.6 Bounded Refinement and Complexity

The gated residual gives an explicit limit on coordinate displacement. Since each gate lies in $[0, 1]$ and $\tanh(\cdot)$ lies in $[-1, 1]$, every coordinate obeys

$$\|q_i^{L_r} - q_i^0\|_\infty \le \sum_{r=0}^{L_r-1} \alpha_r. \tag{11}$$

With $L_r = 2$ and $\alpha_r = 0.03$, the total refinement offset is bounded by 0.06 per coordinate. Zero-initialized residual heads make the initial mapping exactly identical to the underlying folding path, not merely close in expectation. This allows the refinement stages to start from the folding solution and learn only bounded coordinate offsets.

For $E = E_l + K$ hyperedges, incidence aggregation costs $O(NEC)$ and stores $O(NE)$ incidence values; channel projections and feed-forward layers add $O(NC^2)$ work per block. Anchor–group attention costs $O(M_c^2 C + M_c K C)$. Dense group lookup costs $O(MKC)$ per stage. Local kNN search retains only $O(BcN)$ distances for chunk size c, instead of $O(BMN)$, while preserving the same $O(BMN)$ distance-computation time.

3.7 Objective

Reconstruction uses bidirectional Euclidean CD-L1:

$$\begin{aligned} \mathcal{L}_{\mathrm{CD1}}(X, Y) = \frac{1}{2}\Bigg[&\frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} \|x - y\|_2 \\ &+ \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in X} \|y - x\|_2 \Bigg]. \end{aligned} \tag{12}$$

We supervise coarse and dense outputs and add dense repulsion, group-usage balance, coarse patch compactness, coarse repulsion, and a low-weight pre-interaction coarse loss:

$$\begin{aligned} \mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_d \mathcal{L}_d + \lambda_r \mathcal{L}_{\mathrm{rep}} \\ + \lambda_b \mathcal{L}_{\mathrm{bal}} + \lambda_h \mathcal{L}_{\mathrm{hyp}} \\ + \lambda_g \mathcal{L}_{\mathrm{global}} + \lambda_{cr} \mathcal{L}_{\mathrm{crep}} + \lambda_{\mathrm{pre}} \mathcal{L}_{\mathrm{pre}}. \end{aligned} \tag{13}$$

We use $(\lambda_c, \lambda_d, \lambda_r, \lambda_b, \lambda_h) = (0.55, 1, 0.04, 0.01, 0.015)$ and $(\lambda_g, \lambda_{cr}) = (0.05, 0.003)$. The auxiliary intermediate-shape loss is $\mathcal{L}_{\mathrm{pre}} = \mathcal{L}_{\mathrm{CD1}}(Q^0, Y)$, where Q^0 is the folded cloud before Eq. (10). We set $\lambda_{\mathrm{pre}} = 0.5$ in the full model; setting it to zero defines the final-output-only ablation.

4 Experiments

4.1 Protocol and Implementation

We follow the established PCN, MVP, and seen/unseen ShapeNet evaluation protocols. PCN reports CD-L1$\times 10^3$, whereas ShapeNet reports CD-L2$\times 10^3$. MVP contains 62,400 training partial clouds

Table 1: Quantitative comparison on PCN. Published baselines are transcribed from Li et al. (2026b). CD-L1 is scaled by 10^3, and F1 uses a 1% threshold. Dashes denote unreported entries.

Method	Venue	CD-Avg ↓	F1 ↑	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
PCN	3DV'18	9.64	0.695	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59
PoinTr	ICCV'21	8.38	0.745	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29
FBNet	ECCV'22	6.94	–	3.99	9.05	7.90	7.38	5.82	8.85	6.35	6.18
SeedFormer	ECCV'22	6.74	0.818	3.85	9.05	8.06	7.06	5.21	8.85	6.05	5.58
SnowflakeNet	TPAMI'23	7.21	0.801	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40
GTNet	IJCV'23	7.15	–	4.17	9.33	8.38	7.66	5.49	9.44	6.69	6.07
AnchorFormer	CVPR'23	6.59	–	3.70	8.94	7.57	7.05	5.21	8.40	6.03	5.81
SVDFormer	ICCV'23	6.54	0.841	3.62	8.79	7.46	6.91	5.33	8.49	5.90	5.83
AdaPoinTr	TPAMI'23	6.53	0.845	3.68	8.82	7.47	6.85	5.47	8.35	5.80	5.76
ODGNet	AAAI'24	6.51	0.833	3.78	8.78	7.57	6.83	5.12	8.46	5.86	5.66
CRA-PCN	AAAI'24	6.39	–	3.59	8.70	7.50	6.70	5.06	8.24	5.72	5.64
SymmCompletion	AAAI'25	6.28	0.853	3.53	8.49	7.30	6.52	5.06	8.23	5.64	5.49
PointMAC	NeurIPS'25	6.33	–	3.54	8.66	7.44	6.65	4.98	8.19	5.64	5.57
HIG-PCN (ours)	–	**6.18**	**0.859**	**3.36**	**8.45**	**7.13**	**6.43**	**4.81**	**8.09**	**5.52**	**5.43**

and 41,600 test partial clouds, each paired with a 2,048-point complete shape (Pan et al., 2021). Following the released protocol, we retain raw coordinates without per-shape normalization and report bidirectional CD-L2$\times 10^4$ (Li et al., 2026b). F1@1% is the harmonic mean of precision and recall at a nearest-neighbor threshold of 0.01. Since MVP contains unequal numbers of views per category, *Overall* is the equal mean of the 16 category means.

The encoder uses $C = 128$, six blocks, 192 local hyperedges with $k = 32$, and 16 latent-group hyperedges. HIG-PCN predicts 256 anchors, folds eight points per anchor, and applies two four-head interaction blocks followed by two dense refinement stages. The complete network has 3.13M parameters, of which 0.46M belong to dense refinement. End-to-end models are optimized for 420 epochs with AdamW, global batch size 64, and weight decay 5×10^{-4}. The learning rate warms from 10^{-5} to 2×10^{-4} over 20 epochs and then follows cosine decay to 2×10^{-6}. Controlled component studies start from the same anchor–group checkpoint and use an identical 60-epoch schedule.

We fix seed 11 and construct inference-time hypergraphs deterministically. MVP has 26 views per complete shape; consequently, we first average views within each shape, then average shapes per category. Confidence intervals use 10,000 category-stratified bootstrap samples, and paired category-level Wilcoxon tests are Holm corrected when several permutations are tested. Each completed configuration is one training run, so intervals quantify paired test-shape variation rather than variation across training seeds. Additional environment, checkpoint, and provenance details are provided in Appendices D and H.

4.2 Completion Accuracy

We compare against representative folding-, transformer-, and symmetry-based completion methods. Tables 1 and 2 follow the established reporting layouts, while Table 3 reports the complete 41,600-view MVP test set. Published baseline entries are transcribed from their cited sources; all HIG-PCN entries are obtained with the corresponding dataset protocol. Best results are shown in bold.

PCN results. HIG-PCN obtains 6.18 CD-L1 and 0.859 F1 in Table 1. Compared with the strongest listed baseline, this reduces average CD from 6.28 to 6.18 (1.6% relative) while increasing F1 by 0.006. The gain is not driven by one category: HIG-PCN gives the lowest CD in all eight, spanning objects with thin structures, such as airplanes and lamps, and objects dominated by broad surfaces, such as cabinets and sofas. Thus, the improvement is not limited to a single type of object geometry.

Seen and unseen ShapeNet results. Table 2 shows a larger gain under varying observation difficulty. On seen ShapeNet-34, HIG-PCN lowers average CD from 0.60 to 0.53, an 11.7% relative reduction, and reaches the highest reported F1 of 0.577. On the 21 unseen categories, it reduces CD from 0.97 to 0.91 (6.2%) and achieves 0.581 F1. Improvements remain visible on the hard subsets (1.00→0.96 for seen classes and 1.83→1.73 for unseen classes). The unseen-category result shows

Table 2: Seen ShapeNet-34 and unseen ShapeNet-21 comparison. Published baselines are transcribed from Li et al. (2026b). CD-S/M/H denote CD-L2 on simple, median, and hard subsets; CD-L2 is scaled by 10^3, and F1 uses a 1% threshold.

		Seen ShapeNet-34					Unseen ShapeNet-21				
Method	Venue	CD-S	CD-M	CD-H	CD-Avg ↓	F1 ↑	CD-S	CD-M	CD-H	CD-Avg ↓	F1 ↑
PCN	3DV'18	1.87	1.81	2.97	2.22	.154	3.17	3.08	5.29	3.85	0.101
TopNet	CVPR'19	1.77	1.61	3.54	2.31	0.171	2.62	2.43	5.44	3.50	0.121
PFNet	CVPR'20	3.16	3.19	7.71	4.68	0.347	5.29	5.87	13.33	8.16	0.322
GRNet	ECCV'20	1.26	1.39	2.57	1.74	0.251	1.85	2.25	4.87	2.99	0.216
SnowflakeNet	ICCV'21	0.60	.86	1.50	0.99	0.422	0.88	1.46	2.92	1.75	.388
PoinTr	ICCV'21	.76	1.05	1.88	1.23	0.421	1.04	1.67	3.44	2.05	0.384
SeedFormer	ECCV'22	0.48	0.70	1.30	0.83	0.452	0.61	1.07	2.35	1.34	0.402
ProxyFormer	CVPR'23	0.44	0.67	1.33	0.81	0.466	0.60	1.13	2.54	1.42	0.415
AdaPoinTr	TPAMI'23	0.51	0.67	1.17	0.78	0.399	0.61	0.96	2.11	1.23	0.416
ODGNet	AAAI'24	0.44	0.64	1.14	0.75	0.451	0.59	1.01	2.26	1.29	0.415
CRA-PCN	AAAI'24	0.45	0.65	1.18	0.76	–	0.55	0.97	2.19	1.24	–
SymmCompletion	AAAI'25	0.33	0.48	1.00	0.60	–	0.39	0.70	1.83	0.97	–
PointMAC	NeurIPS'25	0.44	0.64	1.14	0.75	–	0.53	0.96	2.16	1.22	–
HIG-PCN (ours)	–	**0.31**	**0.45**	**0.96**	**0.53**	**0.577**	**0.35**	**0.65**	**1.73**	**0.91**	**0.581**

that the complete model transfers beyond the categories used for training rather than improving only easy partial views from known classes.

Table 3: Category-macro results on all 41,600 MVP test views. Published rows are transcribed from Li et al. (2026b); CD-L2 is scaled by 10^4, and F1 uses a 1% threshold.

Method	CD-L2 ↓	F1 ↑
PCN (Yuan et al., 2018)	8.65	0.342
CRN (Wang et al., 2020)	7.25	0.434
VRCNet (Pan et al., 2021)	5.82	0.495
PDR (Lyu et al., 2022)	5.66	0.499
CRA-PCN (Rong et al., 2024)	5.33	0.529
SymmCompletion (Yan et al., 2025)	4.89	0.534
PointMAC (Jiang et al., 2025)	5.24	0.537
Hyper-PCN (Li et al., 2026b)	4.76	0.558
HIG-PCN (ours)	**4.68**	**0.568**

MVP results. On the largest benchmark, HIG-PCN achieves 4.68 CD-L2 and 0.568 F1 (Table 3), improving the previous best values of 4.76 and 0.558. This corresponds to a 1.7% relative CD reduction and a 0.010 absolute F1 gain over the closest method. The joint CD and F1 improvement indicates lower nearest-neighbor error without reducing thresholded surface coverage.

4.3 Qualitative Comparison

CD and F1 summarize point-wise error and coverage but do not isolate individual thin structures. Figure 2 provides four fixed examples selected before the external baseline outputs were added. All methods recover the overall airplane shape, whereas the chair and table expose incomplete supports. Visible bars alone do not establish successful completion: some may already be present in the input, and a plausible silhouette may omit unobserved surfaces. We therefore add a structural-region diagnostic in Appendix K, with identical ground-truth-defined crops, regional F1, and recall restricted to unobserved target points. In the fixed chair-backrest example, HIG-PCN obtains 0.479 missing-region recall at distance 0.02, compared with 0.705 for SymmCompletion and 0.927 for the PCN-trained PoinTr reference; the table lower frame shows the same ordering. These two examples identify remaining failures, not a general ranking. They do not support a claim that the current qualitative outputs demonstrate superior component recovery.



Figure 2: Qualitative comparison on MVP. From left: partial input, SymmCompletion (Yan et al., 2025), PoinTr (Yu et al., 2021), HIG-PCN, and ground truth. SymmCompletion uses its official MVP-2K checkpoint; because PoinTr provides no MVP checkpoint, its released PCN-new model is shown only as a cross-domain reference. Predictions use the dataset coordinates and 2,048 points; annotations report CD-L2$\times 10^4$ ($\downarrow$) and F-score@1% ($\uparrow$). Samples and views were fixed before the external baselines were evaluated.

4.4 Ablation and Mechanism Analysis

Table 4: Controlled full-MVP component study. All rows share initialization and a 60-epoch budget; each successive row adds only the named component or objective term.

Setting	CD-L2 $\downarrow$	F1 $\uparrow$
Anchor–group interaction only	7.31	0.409
+ Group-token dense refinement	7.29	0.407
+ Intermediate-shape supervision	7.25	0.409

Component contributions. Table 4 isolates the two additions after anchor–group interaction. Reusing the updated group tokens during dense refinement changes CD from 7.31 to 7.29 and F1 from 0.409 to 0.407 in this controlled run. Adding direct supervision to the folded point cloud lowers CD further to 7.25 and restores F1 to 0.409. Relative to anchor–group interaction alone, the complete configuration improves CD by 0.06 without reducing F1 at the reported precision. This result motivates supervising the intermediate shape when the dense refiner and folding decoder are optimized jointly.

4.5 Do Both Hypergraph Scales Affect the Prediction?

We use a preselected, class-balanced subset of 32 shapes and all 832 associated views. Keeping point features fixed, we permute the point rows of the local incidence matrix, the latent-group incidence matrix, or both. The permutation preserves the incidence values and hyperedge degree or mass statistics, but breaks the correspondence between each point and its hyperedge memberships. As shown in Table 5, permuting local incidence produces the largest single-scale change (+6.5133 CD

and -0.15289 F1). Permuting latent-group incidence also degrades the result by $+0.2615$ CD and -0.00788 F1, and permuting both is most damaging overall. All six metric tests remain significant after Holm correction ($p \leq 0.0153$). Thus, the prediction depends on both incidence matrices, although their effects are not equal. This experiment tests model dependence on the incidence structure; it does not claim that the learned groups are semantic parts. Full intervals and invariant checks are provided in Appendix G.

Table 5: Incidence-row permutation tests. Positive ΔCD and negative ΔF1 indicate worse completion; p gives Holm-adjusted CD/F1 tests.

Misaligned incidence	ΔCD	ΔF1	Holm p
Local patch	+6.5133	-0.15289	$1.83 \times 10^{-4}/1.83 \times 10^{-4}$
Latent group	+0.2615	-0.00788	$0.0153/0.0153$
Both scales	+8.6315	-0.17141	$1.83 \times 10^{-4}/1.83 \times 10^{-4}$

4.6 Why Supervise the Intermediate Shape?

The group-token refinement row in Table 4 shows that a dense refiner can reduce average point error while slightly reducing thresholded coverage. A frozen-backbone control clarifies this behavior: the refiner lowers CD by 0.0594 relative to its input folded cloud, yet its final CD remains 0.0635 worse than the anchor–group control because the folded input itself has degraded by 0.1230. Thus, training only the final output allows the folding decoder and refinement module to co-adapt to a poorer intermediate point cloud.

Intermediate-shape supervision changes only λ_{pre}. With this term, the final output improves over its folded input by 0.05419 CD-L2 (95% interval $[0.03939, 0.07036]$ in magnitude), while the F1 change is only -0.00036 and remains within the predefined 0.002 non-inferiority margin. A gradient-transfer diagnostic measures how strongly the final-output gradient projects onto the intermediate-shape gradient direction. After final-output-only training, this projection ratio is 0.420 on training shapes and 0.457 on test shapes; intermediate-shape supervision raises the ratios to 0.920 and 0.957. The auxiliary term therefore helps retain the optimization signal that improves the folded point cloud while the dense refiner is trained. Full intervals and controls are reported in Appendix G.

Scope of the evidence. The learned groups provide completion features without part annotations. Permutation tests establish dependence on their memberships, but do not identify semantic parts or prove topological correctness. Our experiments cover synthetic completion protocols; robustness to sensor noise remains untested. Appendix J discusses this scope and the distinction between encoder and decoder assignments.

5 Conclusion

HIG-PCN encodes a partial point cloud with local and latent-group hypergraphs and passes their features to coarse generation and dense refinement. It improves completion on PCN, seen and unseen ShapeNet categories, and MVP. Incidence-row permutation tests show that predictions depend on both scales; component and gradient analyses explain how intermediate-shape supervision stabilizes dense refinement. These results support carrying higher-order relations from encoding into point generation. Region-level coverage checks also identify incomplete thin structures, emphasizing that aggregate accuracy and membership sensitivity should not be equated with verified component reconstruction.

AI USE STATEMENT

Generative AI tools, including OpenAI Codex, assisted with developing and refining the model design and experimental methodology, implementing methods and evaluation code, mathematical reasoning and derivations, and interpreting experimental results. They also assisted with literature search, preparing and revising figures, drafting and editing the manuscript, reference formatting, and adapting the LaTeX source to the conference template. The authors are responsible for the final manuscript, including its claims, results, code, citations, and all AI-assisted content.

REPRODUCIBILITY STATEMENT

Section 3 specifies the model and objective, and Section 4.1 summarizes training and evaluation. Appendices C and D provide tensor dimensions, initialization, normalization, and optimization details. Appendices E, G, and H describe statistical tests and checkpoint-selection sensitivity. Appendices I and L document preprocessing and reproduction commands. Confidence intervals quantify variation over test shapes; the reported studies do not establish robustness across training seeds.

REFERENCES

Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. Anchorformer: Point cloud completion from discriminative nodes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13581–13590, 2023.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 3558–3565, 2019.

Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2548–2566, 2022.

Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. Hgnn+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2023.

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 *Advances in Neural Information Processing Systems*, volume 38, 2025.

Da-Yeong Kim and Yeong-Jun Cho. Dance: Density-agnostic and class-aware network for point cloud completion. In *AAAI Conference on Artificial Intelligence*, volume 40, pp. 5648–5655, 2026.

Jie Li, Shengwei Tian, Long Yu, and Xin Ning. Flexible-weighted chamfer distance: Enhanced objective function for point cloud completion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(7):8106–8119, 2026a. doi: 10.1109/TPAMI.2026.3669003.

Linfei Li, Pei Tan, Siqi Li, Changqing Zou, and Yue Gao. Hyper-pcn: Hypergraph-based point cloud completion via high-order correlation modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39121–39130, 2026b.

Shanshan Li, Pan Gao, Xiaoyang Tan, and Mingqiang Wei. Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9466–9475, 2023.

Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. In *International Conference on Learning Representations*, 2022.

Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8524–8533, 2021.

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 Conference on Artificial Intelligence*, volume 38, pp. 4676–4685, 2024.

Zhenyu Shu, Jian Yao, and Shiqing Xin. Dfg-pcn: Point cloud completion with degree-flexible point graph. *IEEE Transactions on Visualization and Computer Graphics*, 31(12):10698–10710, 2025. doi: 10.1109/TVCG.2025.3612379.

Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 383–392, 2019.

Xiaogang Wang, Marcelo H. Ang, Jr., and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 790–799, 2020.

Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1939–1948, 2020.

Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):852–867, 2023. doi: 10.1109/TPAMI.2022.3159003.

Zizhao Wu, Jian Shi, Xuan Deng, Cheng Zhang, Genfu Yang, Ming Zeng, and Yunhai Wang. Spacnet: Rethinking point cloud completion with structural prior. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):6268–6279, 2025. doi: 10.1109/TVCG.2024.3507013.

Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *IEEE/CVF International Conference on Computer Vision*, pp. 5499–5509, 2021.

Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *European Conference on Computer Vision*, pp. 365–381, 2020.

Mingye Xu, Yali Wang, Yihao Liu, Tong He, and Yu Qiao. Cp3: Unifying point cloud completion by pretrain-prompt-predict paradigm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9583–9594, 2023. doi: 10.1109/TPAMI.2023.3257026.

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 Conference on Artificial Intelligence*, volume 39, pp. 9094–9102, 2025.

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 *European Conference on Computer Vision*, pp. 676–693, 2022.

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 206–215, 2018.

Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 12498–12507, 2021.

Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou. Adapointr: Diverse point cloud completion with adaptive geometry-aware transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14114–14130, 2023.

Xumin Yu, Zuyan Liu, Yanbo Wang, Jie Zhou, and Jiwen Lu. Protocomp++: Diverse point cloud completion with controllable prototype. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(7):8710–8725, 2026. doi: 10.1109/TPAMI.2026.3674197.

Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *International Conference on 3D Vision*, pp. 728–737, 2018.

Zihao Zhang, Aming Wu, Yang Li, Yahong Han, and Jialie Shen. Geometric-aware hypergraph reasoning for novel class discovery in point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10006–10015, 2026.

Yi Zhong, Weize Quan, Dong-Ming Yan, Jie Jiang, and Yingmei Wei. Pointcformer: A relation-based progressive feature extraction network for point cloud completion. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 10689–10697, 2025.

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 *European Conference on Computer Vision*, pp. 416–432, 2022.

Zhensheng Zhou, Jianqing Liang, Jiye Liang, Zijin Du, and Chenghao Fang. Semantic guided part relation-aware network for point cloud completion. In *AAAI Conference on Artificial Intelligence*, volume 40, pp. 13871–13879, 2026.

A SCOPE

This supplement records implementation details, exact statistical decisions, and completed mechanism diagnostics that support the main paper. It introduces no claim required to understand the main contribution. All paths below are relative to the anonymized code-and-data package; no external repository or mutable web artifact is required.

B CLOSEST-METHOD BOUNDARY

Table 6 contrasts HIG-PCN with four recent relation-, group-, and refinement-based completion systems.

Table 6 fixes the novelty boundary used in the main paper. HIG-PCN does not claim the first use of hypergraphs, coarse-to-fine generation, anchors, group relations, or iterative correction. The evaluated combination consists of local and latent-group incidence, bidirectional anchor–group interaction, reuse of updated group tokens during dense refinement, and controlled incidence-row permutations. The incidence matrices remain fixed within a forward pass. We have not completed a comparison against layer-wise incidence reconstruction and therefore claim no advantage over that alternative.

C ARCHITECTURE DETAILS

C.1 ENCODER AND HYPERGRAPH NORMALIZATION

For an input cloud $X \in \mathbb{R}^{N\times 3}$, $N = 2048$, multi-frequency coordinate features are mapped to $F \in \mathbb{R}^{N\times C}$ with $C = 128$. We build $E_g = 192$ geometric hyperedges and $E_p = 16$ latent-group hyperedges. Geometric centers are selected by farthest-point sampling and each center connects its $k = 32$ nearest input points. Latent incidence is soft:

$$H^p_{ij} = \frac{\exp(\bar{f}_i^\top \bar{u}_j/\tau)}{\sum_{\ell=1}^{E_p} \exp(\bar{f}_i^\top \bar{u}_\ell/\tau)}, \qquad \tau = 0.2, \tag{14}$$

where bars denote ℓ_2 normalization and u_j is a learned prototype. Concatenating binary geometric incidence and soft latent incidence gives $H \in \mathbb{R}^{N\times 208}$.

Each of six residual hypergraph blocks applies the implemented normalization

$$\widetilde{F} = D_v^{-1/2} H D_e^{-1} H^\top D_v^{-1/2} F, \tag{15}$$

followed by a linear projection, LayerNorm, ReLU, dropout, and a two-layer feed-forward residual update. Evaluation fixes the first farthest-point sampling choice, so rebuilding the hypergraph is deterministic.

Non-expansive incidence aggregation. Let $B = D_v^{-1/2} H D_e^{-1/2}$, so the propagation matrix is $\mathcal{P}_H = BB^\top$ and is positive semidefinite. Moreover, $S = D_v^{-1} H D_e^{-1} H^\top$ is nonnegative and row-stochastic because $S\mathbf{1} = \mathbf{1}$, while $\mathcal{P}_H = D_v^{1/2} S D_v^{-1/2}$. Similarity and positive semidefiniteness therefore place every eigenvalue of $\mathcal{P}_H$ in $[0, 1]$, implying $\|\mathcal{P}_H F\|_F \le \|F\|_F$. This guarantee applies to incidence aggregation itself; the learned projection and residual FFN can still scale features.

C.2 LATENT GROUPS, ANCHORS, AND DENSE POINTS

Point-to-group logits $Z \in \mathbb{R}^{N\times 16}$ define $A = \mathrm{softmax}(Z)$. Normalizing each group over points gives

$$w_{ij} = \frac{A_{ij}}{\sum_n A_{nj}}, \quad t_j = \sum_i w_{ij} f_i, \quad c_j = \sum_i w_{ij} x_i. \tag{16}$$

Thus each latent group has a feature t_j and geometric center c_j without requiring semantic part labels.

Table 6: Mechanism boundary against PointCFormer (Zhong et al., 2025), SGPRNet (Zhou et al., 2026), DANCE (Kim & Cho, 2026), and the closest hypergraph completion baseline. This is a design comparison, not an accuracy table: datasets, modalities, output cardinalities, and evaluation protocols differ.

Method	Relation representation	Anchor or token role	Use during generation	Additional supervision
PointCFormer	Pairwise target–kNN relation scores combined with self-attention	Point proxies are corrected with denser input information	Progressive feature extraction followed by proxy correction	None beyond paired shapes
SGPRNet	Part-relation dictionary aligned with MLLM-derived text tokens	Semantic part tokens guide a joint coarse cloud	Semantic-geometric relation fusion precedes coarse-to-fine upsampling	Generated part-relation text
DANCE	Viewpoint-grouped ray candidates and a global class feature	Ray candidates are refined rather than used as hypergraph centers	Face-group attention predicts candidate offsets and opacity	Object category labels
Hyper-PCN (Li et al., 2026b)	Layer-wise binary feature-distance incidence and binary keypoint-to-anchor incidence	Two encoder A-HGNNs use deterministic uniform key-point subsamples; a pre-CFG A-HGNN uses predicted symmetry anchors	The pre-CFG A-HGNN updates partial-key features before two CFG Transformer stages	None beyond paired shapes
HIG-PCN	Stable coordinate-patch incidence plus soft prototype-induced latent incidence	Predicted output anchors and latent-group tokens update one another	Decoder-side group features condition folding; updated group tokens are reused through two pointwise refinement stages	None; no semantic part labels

The coarse decoder predicts 256 anchors. HIG-PCN forms anchor tokens from the global feature, an eight-neighbor interpolated local feature, and anchor coordinates. Group tokens combine the global feature, t_j, and c_j. Each of two four-head interaction blocks performs anchor self-attention, anchor-to-group cross-attention, group-to-anchor cross-attention, and separate feed-forward updates. A bounded coordinate head then refines each anchor.

Group-aware folding expands every anchor to eight points. A base folding head uses global, local, anchor, and 2D grid features. A second head queries the decoder-side group features and centers and adds a gated, bounded residual. After bidirectional interaction, the full model retains the updated group tokens and applies two dense refinement stages:

$$q_i^{r+1} = q_i^r + \sigma(g_r(z_i^r)) \odot 0.03 \tanh(d_r(z_i^r)), \quad r \in \{0, 1\}. \tag{17}$$

where z_i^r concatenates global context, an eight-neighbor observed-local feature, updated group-token context, and q_i^r. Neighbor lookup is query-chunked at 256 points; weights remain differentiable with respect to selected neighbor coordinates and features.

C.3 Initialization Semantics

The final linear layer of the group-conditioned folding residual head and every dense refinement coordinate head is zero-initialized. Their gates have zero weights and bias -2.5. The base folding head and anchor-refinement coordinate head are not zero-initialized. Consequently, adding the zero-initialized dense refiner to a trained anchor–group checkpoint reproduces its prediction exactly at

Table 7: Exact HIG-PCN tensor structure for MVP. ShapeNet changes only the folding factor from 8 to 24 and appends the 2,048 observations.

Stage	Input count	Output count	Width	Repeats	Context and update
Point stem	2048	2048	128	1	Multi-frequency XYZ encoding and MLP
Hypergraph encoder	2048	2048	128	6	192 local kNN hyperedges plus 16 soft latent-group hyperedges
Group pooling	2048	16	128	1	Soft point membership; feature and coordinate centroids
Anchor–group interaction	256 anchors, 16 groups	same	128	2	Anchor self-attention and bidirectional anchor–group cross-attention
Group-aware folding	256	2048	128	1	Eight grid points per anchor with local and latent-group context
Progressive refinement	2048	2048	128	2	Global, observed-local, updated-group, and current-coordinate context

Table 8: Optimization schedules. Continuations use five warm-up epochs. The paper protocol uses 20 warm-up epochs from 10^{-5} and cosine decay to 2×10^{-6}.

Protocol	Epochs	Batch	Peak LR	WD
Historical stage configurations	100	32	5×10^{-5}	0
Inherited continuation	60	32	1×10^{-5}	0
New refinement modules	60	32	5×10^{-5}	0
End-to-end models	420	64	2×10^{-4}	5×10^{-4}

continuation initialization; this statement does not apply to every coordinate-producing head in a randomly initialized model.

D Optimization Protocol

The bidirectional Euclidean Chamfer-L1 used for training is

$$\mathcal{L}_{\mathrm{CD1}}(X,Y) = \frac{1}{2}\left[\frac{1}{|X|}\sum_{x\in X}\min_{y\in Y}\|x-y\|_2 + \frac{1}{|Y|}\sum_{y\in Y}\min_{x\in X}\|y-x\|_2\right]. \tag{18}$$

For HIG-PCN with intermediate-shape supervision, the complete objective is

$$\begin{aligned}\mathcal{L} = {} & 0.55\mathcal{L}_c + \mathcal{L}_d + 0.5\mathcal{L}_{\mathrm{pre}} + 0.04\mathcal{L}_{\mathrm{rep}} \\ & + 0.01\mathcal{L}_{\mathrm{bal}} + 0.015\mathcal{L}_{\mathrm{hyp}} + 0.05\mathcal{L}_{\mathrm{global}} \\ & + 0.003\mathcal{L}_{\mathrm{crep}},\end{aligned} \tag{19}$$

where $\mathcal{L}_{\mathrm{pre}} = \mathcal{L}_{\mathrm{CD1}}(Q^0, Y)$ directly supervises the folded cloud before progressive refinement. Earlier models set its coefficient to zero. The configured MVP evaluation path is the official 41,600-view test file because the released protocol provides no separate validation split. Thus `best.pt` is explicitly test-selected at fixed intervals by the declared complete objective, and the intermediate-shape term participates in both optimization and that selection. The separately frozen epoch-420 `last.pt` comparison uses no checkpoint-selection metric.

All formal MVP comparisons use seed 11 and raw dataset coordinates. Training uses random farthest-point initialization, while validation and test use the deterministic mode. The end-to-end anchor–group control, full HIG-PCN, and intermediate-shape-supervised configurations differ only in architecture and the explicitly stated pre-refinement loss.

The completed 8.07/0.357 folding and 7.43/0.409 anchor–group rows share seed, 100-epoch schedule, and coordinate-wise L1 distance, but are historical configurations rather than a one-factor

Table 9: Frozen continuation gate. All six rows are required for full continuation promotion. The last four rows define mechanism support and are the independently frozen release condition for a full end-to-end scaffold-supervised test.

Decision component	Metric	Passing 95% bound	Interpretation
Final vs. AR	CD-L2	upper < 0	Candidate final output is superior
Final vs. AR	F1	lower ≥ -0.002	Final F1 is non-inferior
Scaffold vs. AR	CD-L2	upper $\leq 0.01\,\mathrm{CD}_{\mathrm{AR}}$	Pre-refinement scaffold remains within 1% of AR
Scaffold vs. AR	F1	lower ≥ -0.002	Scaffold F1 is non-inferior
Final vs. scaffold	CD-L2	upper < 0	Refiner provides a strict CD correction
Final vs. scaffold	F1	lower ≥ -0.002	Refiner does not reduce F1 beyond the margin

ablation. The latter also uses four multi-frequency position bands and different coarse, repulsion, hypergraph, global-coarse, and coarse-repulsion loss weights. Before rerunning this comparison, we specified a random-initialization control whose only functional configuration change is `anchor_reasoning_layers`: $2 \rightarrow 0$. Its outcome will be reported regardless of direction.

E Evaluation and Statistical Decisions

MVP has 41,600 test partial clouds but only 1,600 complete shapes; each shape has 26 views. For metric m, we first average views within a shape, then average shapes within each of 16 categories, and finally give every category equal weight. This reproduces the released category-macro *Overall*. The direct 41,600-view mean is retained only as a secondary audit.

Confidence intervals use 10,000 bootstrap draws. Each draw resamples complete shapes with replacement independently within every category and then averages the 16 category means. Category tests are two-sided Wilcoxon signed-rank tests over the 16 paired category means. For the dense-refinement continuation gate, raw p-values are Holm-adjusted across three comparisons and two metrics.

For continuity with the experiment artifact names, this supplement uses *scaffold* to denote the folded intermediate point cloud Q^0. We denote the matched anchor–group control by AR in Table 9.

The frozen-backbone control completed all 41,600 views. Relative to AR, its final CD changes from 7.3056 to 7.3691, a delta of $+0.0635$ (95% CI $[0.0377, 0.0872]$), and its F1 delta is -0.00110 ($[-0.00149, -0.00070]$). Its scaffold CD delta is $+0.1230$ ($[0.0999, 0.1469]$), which violates the 1% margin, whereas refinement changes CD by -0.0594 relative to that scaffold ($[-0.0782, -0.0437]$). All three F1 non-inferiority rows pass. Because final CD superiority and scaffold CD non-inferiority fail, the registered outcome is no mechanism support and no promotion. The final-versus-scaffold gain confirms that the new head is active; the failed scaffold and final rows instead falsify isolated refiner training as a sufficient remedy. The head is therefore active relative to its own scaffold, but this negative control rejects isolated final-only refiner training as sufficient.

This separation prevents two forms of result-dependent reasoning. A continuation can support scaffold preservation without beating AR in its final output, in which case the mechanism still receives a fair random-initialization test. Conversely, a favorable final score cannot release the scaffold-supervised run if the scaffold and refiner criteria fail. Missing fields, malformed types, or failed SHA-256 checks stop the queue rather than defaulting to release.

F Additional Completion Analyses

F.1 Historical Configuration Comparison

Historical anchor–group interaction obtains 7.43/0.409 versus 8.07/0.357 for group-aware folding on all 41,600 MVP test views. The seed, 100-epoch schedule, and coordinate-wise L1 distance match, but multi-frequency encoding and auxiliary objectives differ; this historical comparison is therefore

Table 10: Category-wise difference between historical anchor–group interaction and earlier group-aware folding configurations. CD gain is relative reduction (%); F1 gain is absolute. Positive values favor anchor–group interaction.

	Air	Cab	Car	Chair	Lamp	Sofa	Table	Boat
CD gain	20.6	8.0	11.6	7.6	-0.7	6.7	6.9	7.4
F1 gain	.094	.031	.049	.034	.041	.037	.060	.045
	Bed	Bench	Book	Bus	Guitar	Moto	Pistol	Skate
CD gain	1.9	2.8	-1.4	18.5	23.3	18.0	18.4	19.2
F1 gain	.013	.066	.019	.088	.074	.029	.072	.065

Table 11: Aggregate values for the historical, confounded comparison. These rows are not a one-factor ablation.

Historical configuration	CD-L2	F1
Earlier group-aware folding*	8.07	0.357
Anchor–group configuration*	7.43	0.409

not a single-factor ablation. The separately specified random-initialization control changes only the number of anchor–group interaction layers from two to zero.

We first average the 26 views within each complete shape, then resample shapes within category and equally average 16 category means over 10,000 paired bootstrap draws. The historical anchor–group-minus-folding difference is -0.6437×10^{-4} in macro CD-L2 with 95% interval $[-0.7647, -0.5102] \times 10^{-4}$; the macro F1 difference is 0.05106 with interval $[0.04899, 0.05300]$. Shape win rates are 84.1%/94.7% for CD/F1. Category means favor anchor–group interaction in CD for 14/16 classes and F1 for 16/16. These intervals quantify paired test-shape variation, not training-seed variation or a causal module effect.

F.2 Updated Group Tokens in Dense Refinement

Under matched initialization and budget, anchor control obtains 7.3055/0.40854 and group-token refinement obtains 7.2907/0.40686. CD improves 0.20% while F1 falls 0.00169. Dense refinement improves category-mean CD in 10/16 classes ($p = 0.117$), but improves F1 in only 5/16 and lowers macro F1 ($p = 0.0386$, two-sided category-level Wilcoxon). This completed continuation is reported as inconclusive rather than as an accuracy gain; it motivates the intermediate-shape supervision control in the main paper.

G Mechanism Diagnostics

On the same balanced subset, increasing the trained residual multiplier from zero to one changes CD/F1 from 7.558/0.379 to 7.006/0.423. Full versus zero residual improves CD by 0.5516 and F1 by 0.04334, favoring the branch in 15/16 categories. The co-adapted shared weights have 2.52% relative L2 drift from the matched control. Difference-in-differences interactions are -0.8897×10^{-4} for CD and 0.06404 for F1, with favorable directions in 16/16 categories.

The residual-scale CD interval is $[-0.7257, -0.3757]$ and its F1 interval is $[0.03662, 0.05002]$. Under the anchor–group shared tensors, the corresponding intervals are $[0.2232, 0.4528]$ and $[-0.02476, -0.01657]$. The difference-in-differences intervals are $[-1.0390, -0.7386]$ for CD and $[0.05585, 0.07218]$ for F1. For the minimal folding graft, the F1 interval is $[0.01035, 0.01868]$; its CD interval crosses zero. Across all eight graft variants, its raw F1 Wilcoxon $p = 0.024963$ becomes 0.124815 after within-metric Holm correction (0.249630 in a joint 16-test sensitivity). No favorable individual F1 graft survives the within-metric correction. Descriptively, all four grafts that alter folding have positive F1 deltas and all four that leave folding unchanged have negative deltas; this sign separation is not an independent confirmatory test. These results explain why the follow-up directly evaluates Q^0 rather than treating the diagnostics as a completed accuracy result. At main-



Figure 3: Historical configuration comparison. For each displayed category, we select the shape nearest the median positive shape-level CD improvement and then the view nearest that shape's mean improvement; these are not best-case examples. Per-view CD-L2 ($\times 10^4$) and F1@1% appear below each prediction. The figure is diagnostic and does not substitute for a matched ablation.

Table 12: Balanced 32-shape, 832-view exploratory diagnostics. Differences are scale one minus scale zero; lower CD and higher F1 are favorable. These localization experiments are mechanism evidence, not substitutes for full-test results.

Diagnostic	Comparison	ΔCD-L2$\times 10^4$	ΔF1	Main finding
Residual scale	Co-adapted path, scale 1 minus 0	-0.5516	$+0.04334$	The trained refiner is useful relative to its folded input
Backbone transplant	Anchor–group tensors plus trained refiner, scale 1 minus 0	$+0.3380$	-0.02070	The refiner is not portable to an unchanged backbone
Interaction	Co-adapted effect minus transplanted effect	-0.8897	$+0.06404$	Backbone–refiner co-adaptation is resolved in all 16 categories
Minimal graft	Add co-adapted folding decoder to transplanted refiner	-0.0955	$+0.01452$	Adapting the folded point cloud makes the F1 effect positive
Matched control	Add coarse-anchor modules without folding decoder	$+0.3568$	-0.02287	Coarse changes alone do not repair the interaction

paper display precision, the minimal folding graft has $p = 0.02496$ and Holm value 0.12482, so no individual favorable graft is confirmatory. All four grafts that alter folding have positive F1 deltas, whereas all four that leave folding unchanged have negative deltas.

G.1 Gradient Transfer

For inherited parameters, the diagnostic computes

$$\rho = \frac{\langle g_R, g_0 \rangle}{\|g_0\|^2}, \tag{20}$$

Table 13: Registered dual-scale incidence-row permutation tests on the balanced MVP audit subset. Differences are perturbed minus baseline; CD is scaled by 10^4, so positive CD and negative F1 are adverse. Holm p pairs (CD/F1) are $1.83 \times 10^{-4}/1.83 \times 10^{-4}$, $0.0153/0.0153$, and $1.83 \times 10^{-4}/1.83 \times 10^{-4}$; adverse-category counts are 16/16, 14/13, and 16/16. Every registered outcome is reported.

Perturbation	ΔCD [95% interval]	ΔF1 [95% interval]
Local patch	$+6.5133\ [+5.6569, +7.3618]$	$-0.15289\ [-0.16665, -0.13907]$
Latent group	$+0.2615\ [+0.1021, +0.4200]$	$-0.00788\ [-0.01072, -0.00510]$
Joint encoder	$+8.6315\ [+7.4976, +9.7341]$	$-0.17141\ [-0.18653, -0.15633]$

where g_R reaches a module through final-output CD-L1 and g_0 comes from direct intermediate-shape CD-L1. Gradients are accumulated over all 26 views of each selected complete shape. The balanced audit uses two shapes per category on both train and test splits.

At continuation initialization, the zero-initialized dense refiner gives $\rho = 1$. After final-output-only refinement training, intermediate-shape ρ is 0.420 on train (95% interval $[0.376, 0.464]$) and 0.457 on test ($[0.344, 0.570]$); all 16 category means are below one on both splits. Adding the pre-registered 0.5 direct gradient moves these values to 0.920 and 0.957. The all-shared train projection changes from 0.501 to 1.001. The folded point cloud is 9.11% worse than the matched anchor–group control on train and 6.15% worse on test, while the refiner repairs it in 16/16 and 15/16 categories. This supports attenuation and rotation of direct intermediate-shape improvement gradients.

G.2 Hypergraph-Use Audit

We first test both encoder incidence scales. For each selected shape and view, a fixed point-row permutation is applied to the coordinate-defined local patches, differentiable latent groups, or both, while point features remain fixed. The local perturbation preserves binary incidence values, every local hyperedge degree, and the vertex-degree multiset. The latent perturbation preserves the membership-row and entropy multisets and every latent-hyperedge mass. Both break incidence–geometry alignment. The audit covers two complete shapes per MVP category and all 26 views, for 32 shapes and 832 partial clouds. Metrics are first averaged by complete shape and then by category; intervals use 10,000 category-stratified complete-shape bootstrap resamples, and the six permutation-test hypotheses use category-level Wilcoxon tests with Holm correction.

All six paired intervals are adverse, and the largest Holm-adjusted value is $p = 0.0153$. The local–latent interaction is $+1.8567$ CD-L2$\times 10^4$ ([+1.4590,+2.2493]) and -0.01064 F1 ([-0.01342,-0.00785]). This supports functional dependence on each encoder scale and their joint arrangement under the registered perturbations, but not semantic-part identity, uniqueness, full-test accuracy, or across-seed robustness. An independent exact rerun reproduces every statistical field and a bitwise-identical per-sample archive. The first execution failed before metric serialization because invariant reductions used float32; the amended protocol changes only those reductions to float64, with the tolerance and all scientific fields unchanged.

For the inherited latent-group characterization, define occupancy from row-normalized memberships h_{ik} as $m_k = N^{-1}\sum_i h_{ik}$ and center $c_k = \sum_i h_{ik}x_i / \sum_i h_{ik}$. With total variance V_x, our normalized compactness and effective group count are

$$V_x = N^{-1}\sum_i \|x_i - \bar{x}\|_2^2,$$

$$C = (NV_x)^{-1}\sum_{i,k} h_{ik}\|x_i - c_k\|_2^2,$$

$$K_{\text{eff}} = \exp\left(-\sum_k m_k \log m_k\right).$$

Cross-view dispersion replaces the within-view residual in C with the occupancy-weighted variation of c_k around its complete-shape mean and normalizes by mean total variance; its null in-

Table 14: Hypergraph-use audit on balanced MVP test data. H is entropy normalized by $\log 16$; Eff. is the effective group count. Compactness and view dispersion are paired with permutation nulls. CD differences are scaled by 10^4; positive CD and negative F1 indicate degradation.

Source	H / Eff.	Compact / null	View / null	ΔCD / ΔF1
Encoder	.6149 / 8.91	.7876 / .9995	.1250 / .3217	$+.2615$ / $-.00788$
Decoder	.9992 / 16.0	.9989 / 1.000	.1346 / .1356	$-.0139$ / $+.00007$

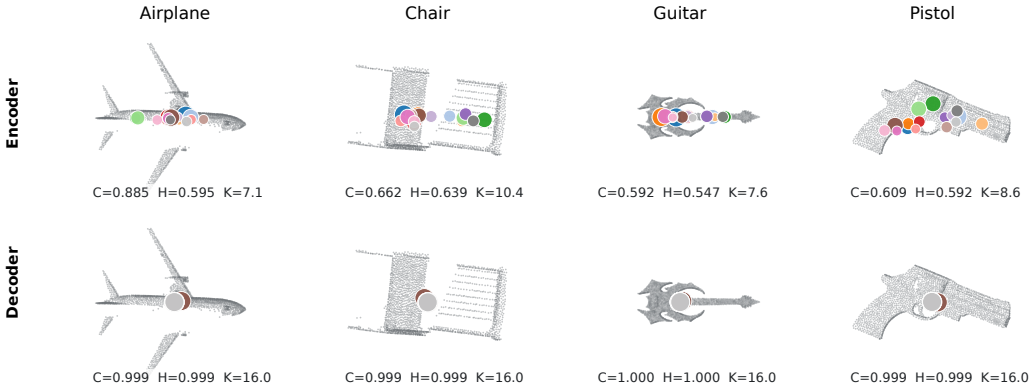

Figure 4: Spatial organization of latent hyperedge memberships on four pre-specified MVP categories. A deterministic median-distance rule selects one view per category; each cloud and both center sets share its two-component PCA projection. Gray dots are partial-cloud points; colored circles are membership-weighted centers with area encoding occupancy. Encoder centers spread over the observed geometry, while the near-uniform decoder memberships yield co-located, similarly sized centers. These examples visualize the same 832-view audit as Table 14; they are descriptive, not an independent test, and do not identify semantic parts.

dependently permutes group indices in each view. Soft NMI uses $J = N^{-1}(H_p^{\text{enc}})^\top A$ and $I(J)/\sqrt{\mathcal{H}(J\mathbf{1})\mathcal{H}(J^\top\mathbf{1})}$. Lower compactness and dispersion indicate stronger spatial organization.

Encoder memberships are more spatially compact and cross-view stable than their nulls in all 16 categories ($p = 3.05\times10^{-5}$ for each). Permuting them worsens CD and F1 (dual-scale-family Holm $p = 0.0153$ for both), resolving functional dependence on encoder-side latent hyperedges under the perturbation. In contrast, decoder memberships are nearly uniform: decoder entropy is 0.9992 of its maximum. Their permutation is unresolved in the separate encoder–decoder family (Holm $p = .8664$), and encoder–decoder soft NMI is only 0.00010.

The decoder-given-encoder interval is inside its symmetric margin for both metrics. Moreover, the latent-encoder-harm macro remains positive under every leave-one-category-out analysis: $[.1952, .3065]$ for CD and $[.00686, .00958]$ for F1 harm. Thus the failed joint-CD significance criterion is not evidence against encoder use: the joint effect is encoder-dominated, whereas the incremental decoder perturbation is practically negligible. This is a same-data secondary decomposition, not an independent replication.

We then froze an inference-only encoder-prior sweep on a disjoint balanced training subset before execution. The registered weights were $\{0, .25, .5, 1\}$. The selection rule chose zero: 0.25 was neutral, 0.5 reduced F1 by 0.00143, and 1.0 worsened CD by 0.1132×10^{-4} and F1 by 0.00410. The predefined repair instead preserves the anchor–group predictions exactly at initialization and trains the raw decoder output toward a stop-gradient encoder target with weight 0.002, plus normalized entropy with weight 0.001. We withhold final accuracy until it completes.

Table 15: Secondary 2×2 factorial decomposition on the same audit samples. Positive values denote harm. Practical-equivalence margins are inherited unchanged from the promotion protocol.

Metric	Latent-enc. harm	Joint/enc.	Decoder|enc. 95% CI	Margin
CD-L2 $\times 10^4$	.2615	96.3%	$[-.0266, .0070]$	$\pm.0698$
F1 harm	.00788	100.7%	$[-.00016, .00028]$	$\pm.002$

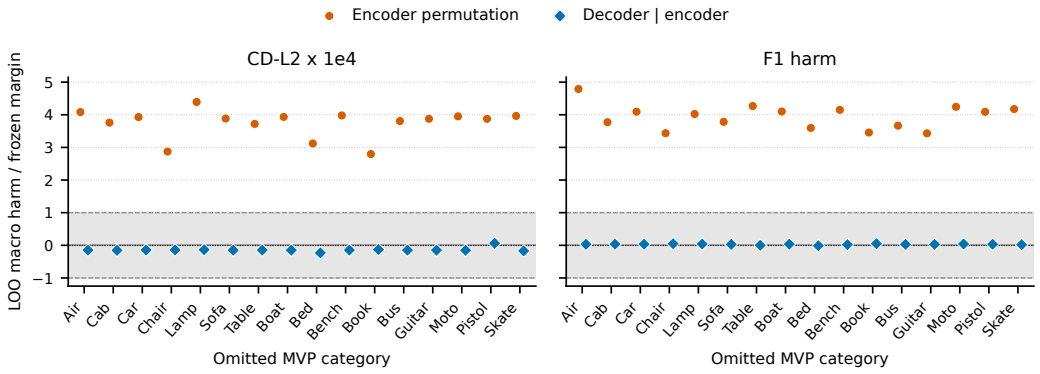

Figure 5: Leave-one-category-out sensitivity of the factorial audit. Each point omits the named MVP category, recomputes equal-category macro harm, and divides it by the frozen metric-specific practical-equivalence margin. The gray band is $[-1, 1]$. Encoder harm remains 2.80–4.79 margins above zero, whereas the conditional decoder increment remains within 0.24 margin under every omission. This is a same-data sensitivity visualization, not an independent replication.

H Checkpoint-Selection Sensitivity

The reference runner periodically evaluates its test loader and keeps `ckpt-best`. We retain the analogous selected-checkpoint result only for protocol comparability and label it as test-selected. Before either model in the formal pair completed, we additionally froze a selection-free architecture audit. It evaluates `last.pt` at epoch 420 for both the matched anchor control and HIG-PCN, then repeats the same 1,600-shape, equal-category paired analysis used for the selected checkpoints. We report all four model–checkpoint combinations regardless of outcome. The conclusion is called selection-robust only when the epoch-420 CD interval lies below zero and the F1 interval lies above the pre-registered -0.002 non-inferiority margin. Architecture-effect claims require this fixed-epoch result; the selected result remains a released-protocol comparison.

For the fixed-epoch pair, HIG-PCN minus control is -0.47 CD-L2$\times 10^4$ (95% interval [-0.64, -0.30]) and 0.027 F1 ([0.026, 0.029]). The pre-registered interpretation is *selection robust*.

A four-checkpoint audit covers the direct-scaffold continuation. Frozen at candidate epoch 27 after interim test-objective monitors, but before epoch 60 or either complete evaluation, it is not jointly blind. It reports selected `best.pt` and fixed epoch-60 `last.pt` comparisons regardless of direction. The control checkpoints are bitwise-identical over all 205 tensors, permitting hash-bound exact reuse of the 41,600-view archive. Only the fixed-epoch pair can support the direct-scaffold mechanism or full-promotion claim. The selected result remains a released-protocol diagnostic and may trigger registered follow-on compute, but cannot establish paper eligibility.

For the fixed pair, final minus control is -0.05 CD-L2$\times 10^4$ ([-0.07, -0.04]) and 0.000 F1 ([0.000, 0.001]). Scaffold minus control is 0.00 CD-L2$\times 10^4$ ([-0.01, 0.01]) and 0.001 F1 ([0.000, 0.001]); final minus scaffold is -0.05 CD-L2$\times 10^4$ ([-0.07, -0.04]) and 0.000 F1 ([-0.001, 0.000]). The fixed result passes 6/6 criteria and 4/4 mechanism criteria: it *supports the direct-scaffold mechanism gate* and *passes the six-criterion promotion gate*. Mechanism support is *supported at both checkpoints* across checkpoint roles; full promotion is *supported at both checkpoints*. Failed fixed criteria are none.

Table 16: Checkpoint-selection audit on all 41,600 MVP views. CD-L2 is scaled by 10^4; the fixed-epoch pair governs architecture-effect claims.

Checkpoint and model	CD-L2 ↓	F1 ↑
Test-selected control	7.90	0.430
Test-selected HIG-PCN	7.44	0.452
Epoch-420 control	8.76	0.429
Epoch-420 HIG-PCN	8.30	0.456

Table 17: Direct-scaffold checkpoint audit on all 41,600 MVP views. CD-L2 is scaled by 10^4. Only the fixed epoch-60 rows govern paper claims.

Checkpoint	Output	CD-L2 ↓	F1 ↑
Selected	Control	7.31	0.409
Selected	Scaffold	7.31	0.409
Selected	Final	7.25	0.409
Epoch 60	Control	7.31	0.409
Epoch 60	Scaffold	7.31	0.409
Epoch 60	Final	7.25	0.409

I SHAPENET TRANSFER PROTOCOL

Training independently samples an integer crop count corresponding to 25–75% of the complete cloud around a random unit direction, then farthest-point samples 2,048 retained points. Evaluation uses easy, median, and hard crop ratios at eight fixed diagonal centers. The model folds 24 points per anchor to generate 6,144 points and appends all 2,048 observations, giving the released 8,192-point output convention.

Exact Chamfer and repulsion are query-chunked for memory but preserve the dense forward result and gradients in unit tests. ShapeNet-55 uses two-process NCCL DDP, local batch four, and gradient accumulation to global batch 64. ShapeNet-34 uses global batch 32. Evaluation partitions indices without padding, preventing duplicate shapes in category means. The full run is released only after a real full-resolution local-batch-four training step and the MVP mechanism gate both pass.

J INTERPRETING THE LEARNED GROUPS

The latent groups are task-dependent feature groups learned from the completion objective. No annotation or constraint fixes the meaning of a group index across shapes. Their memberships collect points with related features and pass context to the decoder. Permuting incidence rows tests whether prediction depends on the point-to-group correspondence; it does not establish correspondence with named object parts. The encoder and decoder assignment heads must also be distinguished: the encoder memberships affect the audited predictions, whereas the decoder memberships are nearly uniform in the same audit. The spatial and permutation analyses in Appendix G document both observations.

The larger local-incidence perturbation effect indicates greater sensitivity to local memberships in this trained model. Comparing effect magnitudes alone does not isolate the accuracy gain attributable to each scale, because both perturbations change the inference distribution. Likewise, Chamfer distance and F1 measure nearest-neighbor error and coverage; neither verifies semantic part correctness or surface topology. The qualitative examples help inspect thin structures; Appendix K adds geometric region-level coverage scores, not semantic-part or topology verification. The current model uses a fixed neighborhood size, and the experiments cover synthetic completion protocols. Adaptive neighborhoods and evaluation on noisy real scans remain directions for future work.

Table 18: Audited 8,192-point ShapeNet files. No referenced point file is missing.

Split	Train	Test	Categories
ShapeNet-55	41,952	10,518	55
ShapeNet-34 seen	46,765	3,400	34
ShapeNet unseen	–	2,305	21

K Structural-Region Completion Diagnostic

Motivation and scope. A recognizable chair back does not necessarily imply that its missing bars have been recovered. Observed points may dominate the display, and points placed between adjacent supports may produce a plausible projection. We therefore examine structural regions using both geometric precision and coverage of unobserved surfaces. This is a descriptive pilot on two saved outputs, not an additional benchmark, a semantic-part evaluation, or a test of surface topology. No sample is substituted based on its regional score.

Fixed data and regions. We reuse the chair and table from Figure 2: MVP test indices 351 and 4485, corresponding to complete-shape indices 13 and 172 at zero-based view 13. All methods share identical 2,048-point inputs and targets. The saved HIG-PCN output is from epoch 420; the released SymmCompletion MVP-2K and PoinTr PCN-new checkpoints are from epochs 44 and 291, respectively. PoinTr remains a cross-domain reference, not a matched-training baseline; its 16,384-point output was deterministically farthest-point sampled to 2,048. The other outputs already contain 2,048 points. This diagnostic evaluates these archived predictions and does not identify them with the run underlying the main benchmark table. Using only the complete targets, before calculating regional scores, we fixed axis-aligned boxes in the original MVP coordinates: the backrest uses $x \in [-0.205, -0.10]$, $y \in [0.10, 0.43]$, $z \in [-0.20, 0.20]$; the lower frame uses $x \in [-0.20, 0.20]$, $y \in [-0.235, -0.08]$, $z \in [-0.415, 0.415]$. These contain 565 and 493 target points, respectively. The regions describe geometry and must not be interpreted as learned hyperedges or semantic labels.

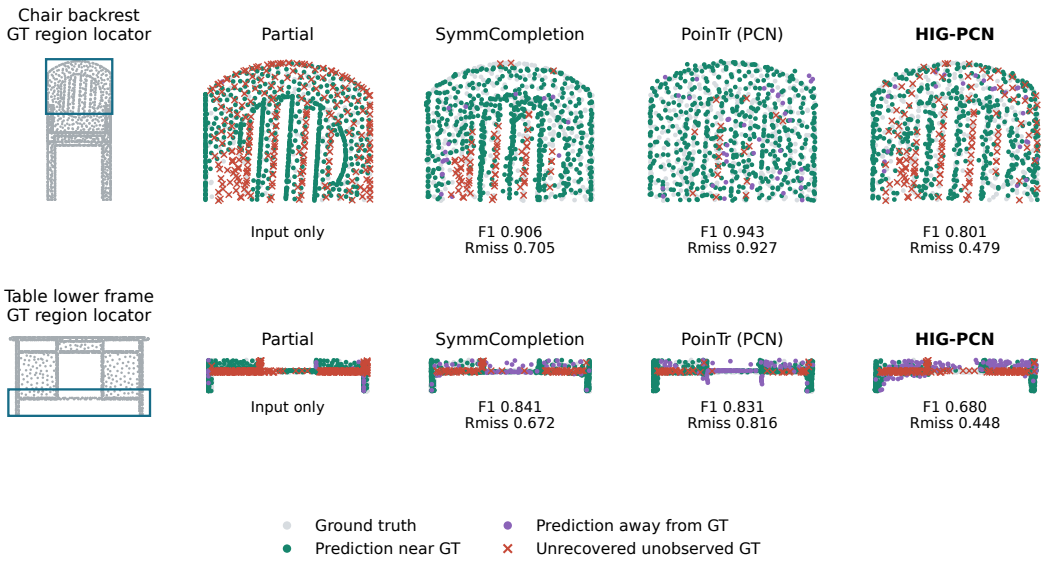

Figure 6: Structural-region diagnostic on the fixed chair and table. The first column locates each region on the complete target; the remaining columns use the same crop and projection for the input and predictions. Gray points show the target, green points lie within 0.02 of the target, and purple points lie farther away. Red crosses mark target points at least 0.02 from both the input and the prediction. All tests use three-dimensional distances, even where the projection makes points overlap. Annotations give regional F1 and unobserved target recall (R_{miss}) at 0.02.

Three-dimensional coverage measures. Let X, Y, and $\widehat{Y}$ denote the input, complete target, and prediction, and let $\mathcal{B}$ be a fixed region. Write $Y_{\mathcal{B}} = Y \cap \mathcal{B}$ and $\widehat{Y}_{\mathcal{B}} = \widehat{Y} \cap \mathcal{B}$. With $d(p, S) = \min_{s \in S} \|p - s\|_2$, define the unobserved target subset as $U_{\mathcal{B}} = \{y \in Y_{\mathcal{B}} : d(y, X) \geq 0.02\}$. This is a distance-based proxy for lack of input support, not an exact visibility label. The two regions contain 234 and 250 such points. We evaluate

$$P_{\mathcal{B}}(\tau) = \frac{1}{|\widehat{Y}_{\mathcal{B}}|} \sum_{p \in \widehat{Y}_{\mathcal{B}}} \mathbf{1}[d(p, Y) < \tau], \tag{21}$$

$$R_{\mathcal{B}}(\tau) = \frac{1}{|Y_{\mathcal{B}}|} \sum_{y \in Y_{\mathcal{B}}} \mathbf{1}[d(y, \widehat{Y}) < \tau], \tag{22}$$

$$R_{\text{miss}}(\tau) = \frac{1}{|U_{\mathcal{B}}|} \sum_{y \in U_{\mathcal{B}}} \mathbf{1}[d(y, \widehat{Y}) < \tau]. \tag{23}$$

Regional F1 is the harmonic mean of $P_{\mathcal{B}}$ and $R_{\mathcal{B}}$. Distances use the full counterpart point cloud, avoiding errors introduced by cutting off a nearby match at a box boundary. Precision penalizes off-surface points, but does not guarantee the absence of spurious connections. No alignment, scale correction, or method-specific tolerance is applied. The fixed distance thresholds are 0.01, 0.02, and 0.03; 0.02 is the display threshold, while the input-support threshold remains 0.02 in all three cases. All are distances in normalized coordinates, not percentages of a fitted bounding box. Empty target or unobserved subsets have no defined recall.

Table 19: Regional F1 / unobserved target recall at three distance thresholds. Higher is better. Each row describes one region of one shape at one view; no population confidence interval or significance claim is made.

Region	Method	$\tau = 0.01$	$\tau = 0.02$	$\tau = 0.03$
Backrest	SymmCompletion	0.378 / 0.244	0.906 / 0.705	0.978 / 0.915
	PoinTr (PCN)	0.384 / 0.389	0.943 / 0.927	0.996 / 1.000
	HIG-PCN	0.261 / 0.094	0.801 / 0.479	0.961 / 0.855
Lower frame	SymmCompletion	0.249 / 0.184	0.841 / 0.672	0.948 / 0.940
	PoinTr (PCN)	0.238 / 0.232	0.831 / 0.816	0.943 / 0.996
	HIG-PCN	0.149 / 0.072	0.680 / 0.448	0.866 / 0.764

What this diagnostic establishes. Figure 6 separates retained input detail from unrecovered target surfaces. In these two regions, HIG-PCN trails both references in regional F1 and unobserved target recall at every tested threshold (Table 19). The conclusion is not reversed by a looser tolerance, although all methods improve as the tolerance grows. These saved outputs therefore do not establish a structural-completion advantage. Conversely, two manually localized regions do not establish a population-level disadvantage. A test of the proposed high-order mechanism would require a larger independently selected set, verified component masks, and matched-training controls with and without the relevant hypergraph connections. Such a test would distinguish improved recovery of a missing component from changes in overall point density or checkpoint selection.

L Reproduction Commands

The package contains configs, model and dataset code, train/evaluate scripts, paired statistical analysis, gate logic, and immutable evidence manifests. The principal checks are:

```
python -m pytest -q
python scripts/train.py
  -config <model-config>
python scripts/evaluate.py
  -config <config> -checkpoint <best.pt>
  -split test -per-sample-output <output.npz>
```

Here `<model-config>` denotes the standard HIG-PCN MVP configuration listed in the package README; scaffold supervision is a separate diagnostic. Both Windows and Ubuntu pass all

225 package tests. The two-rank CPU/Gloo interruption test resumes an epoch-two checkpoint to epoch three and exactly matches an uninterrupted three-epoch run for model tensors, optimizer state, metrics, and rank-specific RNG state; the maximum parameter difference is zero.

Formal run directories retain the resolved configuration, environment, runtime metadata, best and last atomic checkpoints, training log, deterministic evaluation, per-sample arrays, and SHA-256 evidence manifest. The frozen-backbone continuation creates a 34-file Python-only source snapshot after resource selection. The scaffold continuation verifies and imports the identical manifest. Isolated release and rejection tests, plus a no-CUDA ShapeNet watcher lifecycle, are indexed by the immutable package evidence manifest. An immutable pipeline-verification record indexes the causal audit, prior selection, exact-initialization probe, source-snapshot dry run, and gradient checks.