MS-POINT: MULTI-MODAL HIERARCHICAL SPECTRAL LEARNING FOR POINT CLOUD UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Object-level point–image alignment leaves a structural observability gap: agreement between pooled descriptors does not directly constrain intermediate representations or distinguish neighborhood arrangements that produce the same readout. We introduce MS-POINT, a cross-modal pretraining framework that makes internal structure an explicit target of representation learning. At selected encoder stages, complementary neighborhood-consensus and deviation responses expose distinct structural statistics before pooling. Branch-specific visual targets, cross-level coordination, and adaptive fusion jointly preserve shared object semantics and encourage useful specialization. A common interface of intermediate features and spatial neighborhoods supports both graph encoders and point Transformers. Evaluations across object classification, few-shot adaptation, dense prediction, and low-label scene learning demonstrate the transfer value of this design. Same-backbone interventions and capacity- and computation-matched controls connect these improvements to the structure of supervision. Together, our findings establish supervision of intermediate structural states as a concrete design principle for learning transferable, geometry-aware representations across modalities. Code is available at `https://anonymous.4open.science/r/MS-Point-9602`.

1 INTRODUCTION

A point cloud and an image observe the same object through different sensing processes. Learning their agreement provides a useful source of supervision when a pretrained 3D encoder must transfer to tasks with limited labels. CrossPoint learns this agreement through object-level contrast; MM-Point enriches it with multiple views and projection spaces (Afham et al., 2022; Yu & Song, 2024). Yet successful object matching leaves a central representation-learning question unresolved: which internal structures does the alignment objective actually constrain?

The supervision interface has two distinct bottlenecks. **Its location:** a loss on the final object descriptor updates the entire encoder, but supplies no independent visual target to an intermediate state. **Its observability:** pooling is a many-to-one map. Feature configurations with different neighborhood organization can produce the same descriptor, so any deterministic projection after pooling remains unable to distinguish them. More views or larger projection heads do not, by themselves, change these properties of the readout. The challenge is therefore to expose useful structural distinctions to supervision before aggregation.

The distinction matters most when the downstream task changes. Object identity can be recognized from a few dominant cues, whereas a part label or a rare scene category may depend on weak but spatially organized evidence. We therefore seek representations whose internal states remain useful beyond the pretraining readout. Supervising both neighborhood agreement and deviation provides a way to test this requirement directly, while matched-capacity controls distinguish structural information from the benefit of additional heads.

We introduce MS-POINT around two coupled decisions: *where supervision acts and what it observes*. Selected encoder states receive direct visual targets, while complementary neighborhood-consensus and deviation responses expose local agreement and variation within each state. Semantic-

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.

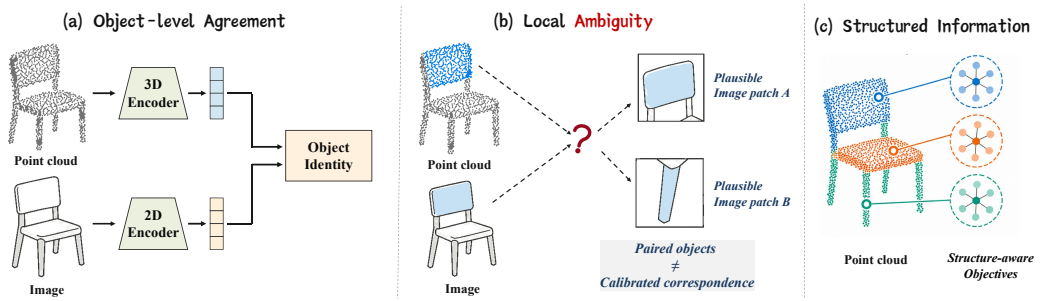

Figure 1: **From object agreement to structural supervision.** Object identity leaves ambiguous local matches; intermediate states and neighborhood responses expose additional structure before pooling.

depth matching anchors these branches to image features; adaptive fusion learns their contribution to the object representation. The same interface operates on graph features and Transformer tokens without a reconstruction decoder.

This formulation shifts the design of cross-modal learning from enriching a final target to specifying the structural states that must support it. Hierarchical masking, image-guided reconstruction, and local distillation provide established routes to structure-aware learning (Zhang et al., 2022; 2023; Sautier et al., 2022). MS-Point supplies a direct contrastive route: intermediate geometry becomes an explicit training target, with structural responses and visual semantics coordinated in one objective. A complete object and a single image share only partial information, so visibility-aware local alignment further extends this principle to dense prediction.

Our contributions are:

- A structured supervision interface that exposes encoder depth and neighborhood variation to point–image learning before object aggregation.
- Complementary graph responses, semantic-depth targets, and joint branch/fusion objectives that transfer across graph encoders and point Transformers.
- Evidence from classification, few-shot adaptation, segmentation, and low-label scenes, supported by same-backbone and capacity- and computation-matched interventions.

2 Related Work

Hierarchical representations and prediction targets. PointNet++ aggregates nested neighborhoods, while DGCNN constructs dynamic feature-space relations (Qi et al., 2017b; Wang et al., 2019). Thus, locality and hierarchy are already present in many point encoders. Point-BERT and Point-MAE introduce masked point-token learning, and Point-M2AE extends it across scales (Yu et al., 2022; Pang et al., 2022; Zhang et al., 2022). These methods make clear that the prediction target, not only the backbone, shapes the learned representation. MS-Point asks how to expose intermediate states and neighborhood responses directly to a cross-modal contrastive target.

Beyond a single reconstruction objective. PointGPT studies autoregressive point-patch generation (Chen et al., 2023); PointDif uses diffusion-based denoising (Zheng et al., 2024). PCP-MAE examines the shortcut introduced by supplying masked patch centers to a decoder and instead predicts centers (Zhang et al., 2024). That distinction is relevant to our motivation: success on a pretext objective need not certify every property required for transfer. TPM expands masked learning through multiple masks (Liu et al., 2025), while Point-PQAE uses cross reconstruction between decoupled views (Zhang et al., 2025). MS-Point contributes neither masking nor generation; it changes the states and neighborhood statistics observed by point–image alignment.

Point–image supervision. CrossPoint and MM-Point provide the closest object-level contrastive references (Afham et al., 2022; Yu & Song, 2024). I2P-MAE and Inter-MAE exploit images within masked learning, and ReCon combines generative and contrastive signals (Zhang et al., 2023; Liu et al., 2024; Qi et al., 2023). Visual guidance and multi-level learning therefore precede our work.

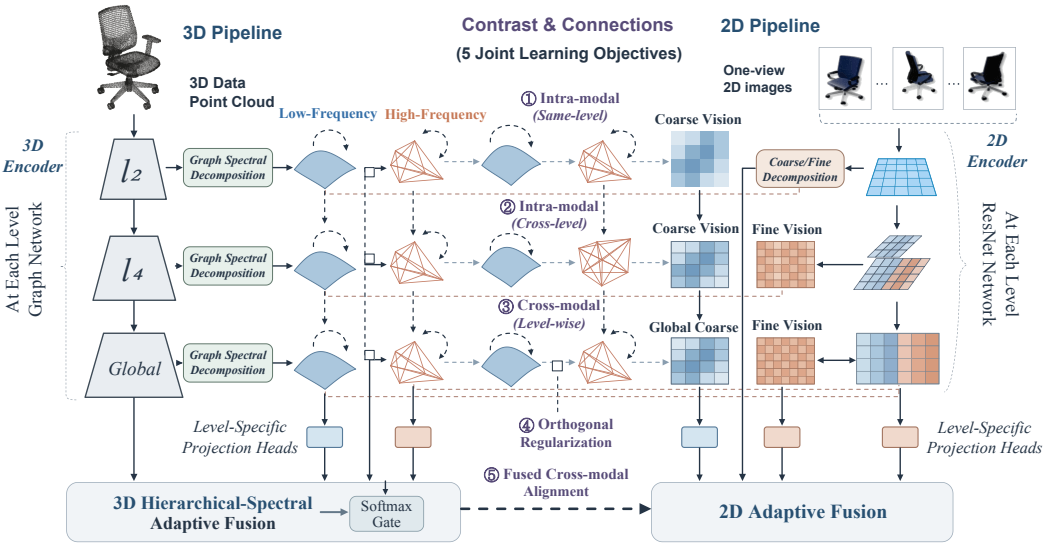

Figure 2: **Structured point–image pretraining.** Intermediate states and complementary responses receive visual targets before adaptive fusion. Low/high and orthogonal labels denote first-order responses and projected cosine regularization.

Our distinction is to organize explicit contrastive branches by semantic depth and complementary neighborhood responses, reuse the corresponding encoder tensors, and learn how their pooled descriptors contribute to the fused representation.

OpenShape and ULIP-2 extend multimodal 3D learning through larger paired collections and richer visual supervision (Liu et al., 2023a; Xue et al., 2024). We study a complementary question under a fixed view budget: how to use the structure already present inside the point encoder.

3 Method

3.1 From Object Readouts to Hierarchical Supervision

Given a point cloud and its paired image, we learn an encoder whose intermediate states support transfer beyond object matching. Let $H^{(l)} \in \mathbb{R}^{N_l \times C_l}$ denote a spatial feature map at depth l, and let Φ be an object readout. A global contrastive target observes $q(\Phi(H^{(l)}))$. The non-injectivity of pooling permits distinct feature organizations with the same descriptor:

$$H \neq H', \quad \Phi(H) = \Phi(H') \quad \Longrightarrow \quad q(\Phi(H)) = q(\Phi(H')). \tag{1}$$

The target cannot distinguish these organizations through this readout. MS-Point exposes selected states and neighborhood-dependent responses before aggregation, so their structural statistics enter the objective explicitly (Figure 2).

We describe graph and attention encoders through a common hierarchy $\{H^{(l)}\}_{l \in \mathcal{D}}$. Each state records a different history of relational aggregation. DGCNN provides evolving point-feature maps (Wang et al., 2019); point Transformers provide spatially indexed patch tokens (Yu et al., 2022; Pang et al., 2022; Zhang et al., 2022). Feature depth defines the hierarchy in both cases, while staged token pooling also changes spatial resolution. We read the global-stage map before pooling, retaining its spatial support for response construction.

For each branch map, max–mean pooling combines selective channel evidence and distributed activation statistics:

$$\Phi(H) = \left[\max_i H_i,\ N^{-1} \textstyle\sum_i H_i\right], \qquad z_m = q_m(\Phi(H_m))\,. \tag{2}$$

The map q_m places the descriptor in a common contrastive space. Intermediate states receive their own targets instead of relying exclusively on gradients through the final descriptor (Figure 3). The same backbone evaluation supplies every selected state; branch selection determines supervision coverage rather than a new encoder architecture.

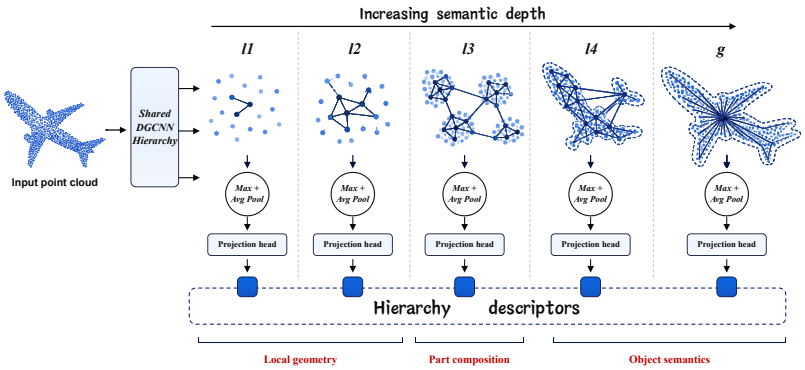

Figure 3: **Supervising the feature hierarchy.** Intermediate states supply independent descriptors; the global source is read before pooling.

3.2 Complementary Responses on Point and Token Graphs

For a response source $l \in \mathcal{S} \subseteq \mathcal{D}$, let $W^{(l)}$ encode its receiving neighborhoods and $P^{(l)} = D^{-1}W^{(l)}$, with $D_{ii} = \sum_j W^{(l)}_{ij}$. We compute

$$H^{(l)}_{\mathrm{s}} = P^{(l)}H^{(l)}, \qquad H^{(l)}_{\mathrm{d}} = (I - P^{(l)})H^{(l)}, \qquad H^{(l)}_{\mathrm{s}} + H^{(l)}_{\mathrm{d}} = H^{(l)}. \tag{3}$$

Graph features use their feature-space neighborhood relation. For MS-Point-T, token centers $c^{(l)}_i \in \mathbb{R}^3$ determine a geometric graph independently of attention:

$$P^{(l)}_{ij} = \frac{\mathbf{1}\{j \in \mathcal{N}_k(c^{(l)}_i)\}}{|\mathcal{N}_k(c^{(l)}_i)|}, \qquad \mathcal{A} = (\mathcal{D} \times \{\mathrm{base}\}) \cup (\mathcal{S} \times \{\mathrm{s}, \mathrm{d}\}). \tag{4}$$

This interface leaves the backbone operations unchanged. Fixed-resolution Transformers reuse token centers across selected blocks; hierarchical Transformers use the centers associated with each stage.

The consensus branch aggregates features shared by connected regions, providing a local structural reference. The deviation branch exposes departures from this reference, including transitions at part junctions and boundaries. Since both responses originate from one state and one relation, their distinction is introduced before the object readout (Figure 4). Independent heads then allow visual supervision to select useful neighborhood agreement and local contrast separately.

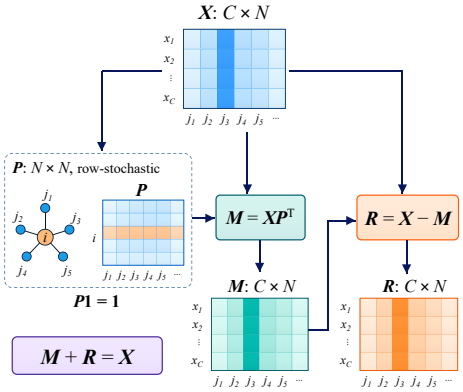

Figure 4: **Responses before pooling.** In channel-first notation, $M = XP^\top$ and $R = X - M$ satisfy $M + R = X$.

For a symmetric reference graph, normalized adjacency S and Laplacian $L = I - S = U\Lambda U^\top$ give responses $SH = U(I - \Lambda)U^\top H$ and $LH = U\Lambda U^\top H$. They are first-order polynomial graph filters (Shuman et al., 2013; Defferrard et al., 2016); deviation gives greater gain to modes of larger graph variation. The directed implementation retains the complementary graph-shift pair, while the symmetric reference defines its spectral interpretation (Appendix U).

The structural gain is in what reaches the readout. For a permutation Π that preserves pooled channel statistics, $\Phi(\Pi H) = \Phi(H)$, whereas a fixed neighborhood relation generally gives $P\Pi H \neq \Pi P H$. Response descriptors can therefore distinguish arrangements that the base descriptor cannot. A joint relabeling of features and graph remains equivariant, preserving the point-set symmetry. This separates sensitivity to the relation between features and their neighbors from sensitivity to arbitrary point ordering (Appendix A).

Complementarity holds on the raw maps, before independent adaptation and pooling. The training objective exploits their distinct statistics rather than requiring disjoint frequency bands. Source-

selection experiments test where the responses help, and duplicated-feature controls test whether their neighborhood information contributes beyond branch capacity.

3.3 Visual Anchors and Instance-Aware Fusion

The image pathway provides multi-granularity semantic anchors. A stage feature F_j supplies a base descriptor and spatial-average/residual counterparts, with $\Gamma_{\text{base}}(F) = F$, $\Gamma_{\text{s}}(F) = B(F)$ and $\Gamma_{\text{d}}(F) = F - B(F)$. For a point branch $m = (l, b)$, the depth map $\rho(m) = (r(l), b)$ selects its visual target:

$$h_{\rho(m)} = q_m^{\text{img}}\big(\Phi_{\text{img}}(\Gamma_b(F_{r(l)}))\big), \qquad u_f = \sum_{m \in \mathcal{A}} \alpha_m z_m, \quad z_f = q_f(u_f). \tag{5}$$

Here B is spatial averaging and Φ_{img} is a spatial readout. Gates predict normalized nonnegative scores α_m from the branch embeddings, with $\sum_m \alpha_m = 1$; image embeddings are fused separately to obtain h_f.

The correspondence follows semantic depth, not a literal layer number (Figure 5). Shallower point states draw on intermediate image features, while deeper states use more abstract targets. ResNet, ConvNeXt, and spatial ViT features can supply these anchors through the same interface. Independent nonlinear heads accommodate different feature widths and let branches sharing an image source specialize in their projected objectives.

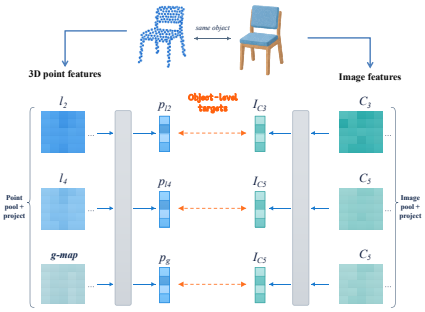

Figure 5: **Semantic-depth targets.** Branch-specific projections pair intermediate and deep point states with visual anchors.

Object identity supplies the positive cross-modal relation without requiring point–pixel registration. Instance-aware routing complements this fixed branch correspondence: the targets define each branch's role, while the gates adapt their contribution to the current object. The fused objective trains their collective readout, and branch objectives retain direct supervision of every selected structural state. Full stage maps, dimensions, graph conventions, and implementation variants are specified in Appendix R.

3.4 Joint Consistency, Alignment, and Diversity

Two point augmentations yield embeddings $z_m^{(1)}, z_m^{(2)}$; the paired image supplies $h_{\rho(m)}$. Let ℓ denote NT-Xent (Chen et al., 2020), $\mathcal{E}$ the selected cross-level pairs, and $\bar{z}_m = (z_m^{(1)} + z_m^{(2)})/2$. For paired batches a, b of size B, its directional term is

$$\ell_{a\to b} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(s(a_i, b_i)/\tau)}{\sum_{j\neq i}\exp(s(a_i, a_j)/\tau) + \sum_{j=1}^{B}\exp(s(a_i, b_j)/\tau)}, \qquad \ell(a, b) = \tfrac{1}{2}(\ell_{a\to b} + \ell_{b\to a}) \tag{6}$$

where s is cosine similarity and τ is temperature. Corresponding observations of the same object are positives; the remaining batch instances provide negatives. The branch map changes which representation enters this objective, while the instance discrimination rule stays shared. We define $\bar{z}_f$ by the same view average. Branch stability and hierarchy coordination are

$$\mathcal{L}_{\text{same}} = |\mathcal{A}|^{-1} \sum_{m \in \mathcal{A}} \ell(z_m^{(1)}, z_m^{(2)}), \tag{7}$$

$$\mathcal{L}_{\text{cross}} = |\mathcal{E}|^{-1} \sum_{(m,n)\in\mathcal{E}} \sum_{v=1}^{2} \ell(z_m^{(v)}, z_n^{(v)}). \tag{8}$$

The first term regularizes each branch across input perturbations; the second coordinates semantic states across depth. Direct visual supervision and fused alignment are

$$\mathcal{L}_{\text{level}} = |\mathcal{A}|^{-1} \sum_{m} \ell(\bar{z}_m, h_{\rho(m)}), \qquad \mathcal{L}_{\text{fused}} = \ell(\bar{z}_f, h_f). \tag{9}$$

These terms anchor structural descriptors and their mixture to the same object semantics. To discourage redundant response directions, we add

$$\mathcal{L}_{\text{band}} = |\mathcal{S}|^{-1} \sum_{l \in \mathcal{S}} \cos^2(z_{\text{s},l}, z_{\text{d},l}), \qquad \mathcal{L} = \sum_{r \in \mathcal{R}} \lambda_r \mathcal{L}_r, \tag{10}$$

where $\mathcal{R} = \{\text{same}, \text{cross}, \text{level}, \text{fused}, \text{band}\}$ and the diversity term is averaged over point views and samples. The finite cosine penalty encourages specialization while the alignment terms preserve common semantics. All terms update the shared encoder jointly, balancing agreement across observations with differentiation across structural responses. Loss-removal and weight-transfer experiments test this balance; exact reductions and optimization settings remain in Appendix G.4.

The shared parameters θ receive gradients from both agreement and specialization:

$$\nabla_\theta \mathcal{L} = \sum_{r \in \mathcal{R}} \lambda_r \nabla_\theta \mathcal{L}_r. \tag{11}$$

Branch-level alignment directly constrains states that the global loss reaches only through subsequent aggregation. Fused alignment instead rewards their collective prediction, while the diversity term discourages both response heads from choosing the same direction. Their weights control the balance of these training signals. We test their contributions by removing one objective at a time under the complete architecture, rather than attributing the result to the number of losses alone.

4 Experiments

4.1 Training and Evaluation Protocols

We pretrain on ShapeNet point–image pairs (Chang et al., 2015) with two augmented point views and one encoded image per iteration. DGCNN and ResNet-50 form the default backbone pair; point Transformers and modern image encoders test portability (Wang et al., 2019; He et al., 2016). Optimization settings are deferred to Appendix G.3.

Our controlled reruns fix the pretraining collection, 1,024 input points, ScanObjectNN OBJ-BG, a frozen-feature linear SVM, and no voting. Same-backbone interventions isolate the objective; capacity- and computation-matched branches test the structural inputs. Published results retain their native protocols, while few-shot and dense tasks test broader transfer (Appendix Z).

4.2 Classification and Recent Baselines

Tables 1–2 compare synthetic ModelNet objects (Wu et al., 2015) and real ScanObjectNN scans (Uy et al., 2019). DGCNN-based MS-Point reaches 93.6/96.5 on ModelNet40/10 and 89.6 on ScanObjectNN, exceeding published MM-Point by 1.2/1.1/1.8 points. Controlled ModelNet40/ScanObjectNN gains are 1.1/1.7 points. MS-Point-T reaches 94.3/97.2 on ModelNet40/10 and 90.4 on ScanObjectNN, improving over the evaluated masked-learning systems, including Inter-MAE and Point-MAE with TPM. These results connect structured targets to transferable features across both clean and scanned objects.

4.3 Backbone Generality and Structural Controls

With the point Transformer and ResNet-50 fixed, hierarchy, responses, and fusion progressively improve three downstream tasks (Table 8). The full objective gains 1.1 ModelNet40 points, 2.1 ScanObjectNN points, and 1.2 ShapeNetPart mIoU over global-only alignment. The benefit therefore extends to token features beyond EdgeConv.

Keeping heads, gates, losses, parameters, and computation fixed, real-neighborhood responses outperform duplicated features by 0.7/1.4 classification points and 0.6 part mIoU. Random neighborhoods and shuffled features are also weaker (Table 9). Thus, additional capacity alone does not explain the full-training gains; the structural inputs to supervision matter.

Table 1: ModelNet40/10 accuracy (%). Frozen-feature results share one block; * marks our controlled linear-SVM evaluations with 1,024 points and no voting. A dash indicates an unavailable result.

Method	MN40	MN10
Supervised references		
PointNet (Qi et al., 2017a)	89.2	–
GIFT (L_5) (Bai et al., 2017)	89.5	91.5
MVCNN (Su et al., 2015)	89.7	–
Frozen-feature linear evaluation		
SwAV (Caron et al., 2020)	90.3	93.5
OcCo (Wang et al., 2021)	89.2	92.7
STRL (Huang et al., 2021)	90.9	–
Point-BERT (Yu et al., 2022)	87.4	–
Point-MAE (Pang et al., 2022)	91.0	–
Point-MAE*	91.4	–
Point-M2AE (Zhang et al., 2022)	92.9	–
CrossPoint (Afham et al., 2022)	91.2	–
I2P-MAE (Zhang et al., 2023)	93.4	–
Inter-MAE* (Liu et al., 2024)	92.1	–
MM-Point (Yu & Song, 2024)	92.4	95.4
MM-Point*	92.5	–
Point-MAE + TPM* (Liu et al., 2025)	91.8	–
MS-Point (DGCNN)*	93.6	96.5
MS-Point-T*	**94.3**	**97.2**

Table 2: ScanObjectNN accuracy (%). Native-protocol literature results and controlled OBJ-BG results are grouped separately. Supervised and full-fine-tuning results provide additional references.

Method	Acc.
Supervised / full fine-tuning	
GBNet (Qiu et al., 2022)	80.5
PointMLP (Ma et al., 2022)	85.2
Point-BERT (Yu et al., 2022)	83.1
MaskPoint (Liu et al., 2022a)	84.3
Point-MAE (Pang et al., 2022)	85.2
Frozen features: native protocols	
Point-MAE (Pang et al., 2022)	77.7
Point-M2AE (Zhang et al., 2022)	84.1
CrossPoint (Afham et al., 2022)	81.7
I2P-MAE (Zhang et al., 2023)	87.1
MM-Point (Yu & Song, 2024)	87.8
Point-MAE + TPM (Liu et al., 2025)	78.2
Point-M2AE + TPM (Liu et al., 2025)	84.0
MS-Point (DGCNN)	89.6
Controlled OBJ-BG linear SVM	
Point-MAE (Pang et al., 2022)	78.1
Inter-MAE (Liu et al., 2024)	84.8
MM-Point (DGCNN)	87.9
Point-MAE + TPM (Liu et al., 2025)	79.2
MS-Point (DGCNN)	89.6
MS-Point-T (Transformer)	**90.4**

4.4 Few-Shot Classification

Table 3 evaluates 5-way and 10-way transfer. MS-Point leads the listed methods across both datasets, including generative and supervised references (Wu et al., 2016; Li et al., 2018; Liu et al., 2019). On ScanObjectNN 5-way 10-shot, it reaches 89.8%, compared with 88.0% for MM-Point and 74.8% for CrossPoint. The consistent ranking across label budgets supports reuse of the pretrained representation when task labels are scarce.

Table 3: Few-shot classification on ModelNet40 and ScanObjectNN (mean±standard deviation).

Method	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
Results on ModelNet40				
3D-GAN	55.8±3.4	65.8±3.1	40.3±2.1	48.4±1.8
PointCNN	65.4±2.8	68.6±2.2	46.6±1.5	50.0±2.3
RSCNN	65.4±8.9	68.6±7.0	46.6±4.8	50.0±7.2
Jigsaw3D	34.3±1.3	42.2±3.5	26.0±2.4	29.9±2.6
OcCo	90.6±2.8	92.5±1.9	82.9±1.3	86.5±2.2
CrossPoint	92.5±3.0	94.9±2.1	83.6±5.3	87.9±4.2
MM-Point	96.5±2.8	97.2±1.4	90.3±2.1	94.1±1.9
MS-Point (Ours)	**97.4±2.5**	**98.1±1.2**	**91.6±1.8**	**95.3±1.5**
Results on ScanObjectNN				
Jigsaw3D	65.2±3.8	72.2±2.7	45.6±3.1	48.2±2.8
OcCo	72.4±1.4	77.2±1.4	57.0±1.3	61.6±1.2
CrossPoint	74.8±1.5	79.0±1.2	62.9±1.7	73.9±2.2
MM-Point	88.0±6.5	90.7±5.1	76.7±1.9	83.9±4.2
MS-Point (Ours)	**89.8±5.8**	**92.4±4.5**	**78.5±1.7**	**85.8±3.8**

4.5 Segmentation Transfer

We fine-tune the pretrained encoder for point-wise prediction on ShapeNet-Part (Yi et al., 2016) and S3DIS (Armeni et al., 2016). MS-Point attains 86.0% and 59.8% mIoU, improving over MM-Point by 0.3 and 0.7 points (Tables 4–5). Although pretraining uses object-level targets, the learned features also benefit dense prediction. This transfer motivates the local extension, which adds spatial correspondence to the same structured interface.

4.6 Ablations and Statistical Reliability

Intermediate supervision, cross-level coordination, and fusion progressively improve classification. Using both responses at l_2, l_4, g reaches 89.6%, versus 87.9% for smoothing and 88.1% for deviation alone (Tables 6–7). The fixed-capacity controls above complement this source-selection study.

Five-seed full-training comparisons give positive paired 95% intervals against the global baseline and MM-Point on both classification datasets, supporting reproducible margins (Appendix B). The shorter-budget pilot is analyzed separately in Appendix Q.

Table 4: ShapeNetPart: OA and mIoU (%).

Method	OA	mIoU
PointNet (Qi et al., 2017a)	–	83.7
PointNet++ (Qi et al., 2017b)	–	85.1
DGCNN (Wang et al., 2019)	–	85.1
OcCo (Wang et al., 2021)	94.4	85.0
CrossPoint (Afham et al., 2022)	94.4	85.3
MM-Point (Yu & Song, 2024)	94.5	85.7
MS-Point	**94.7**	**86.0**

Table 5: S3DIS: OA and mIoU over 13 classes (Armeni et al., 2016).

Method	OA	mIoU
Jigsaw3D (Sauder & Sievers, 2019)	84.1	55.6
OcCo (Wang et al., 2021)	84.6	58.0
CrossPoint (Afham et al., 2022)	86.7	58.4
MM-Point (Yu & Song, 2024)	88.7	59.1
MS-Point	**89.4**	**59.8**

4.7 Efficiency and Dense Transfer

MS-Point reuses intermediate states rather than re-encoding each branch. Compared with the one-image global baseline, it adds 2.13% parameters and 23.98% GPU time; compared with four-image MM-Point, it reduces GPU time by 17.34% and peak memory by 21.64% (Table 11). MS-Point-T takes 30.4 H100 hours versus 72.0–76.5 for the evaluated MAE systems.

Geometry-preserving local alignment increases ShapeNetPart instance mIoU from 86.1 to 86.8 and S3DIS mIoU from 67.5 to 69.0 in its evaluation protocol (Table 46). On SemanticKITTI (Behley et al., 2019), the local model obtains 47.3 mIoU with 1% labels, compared with 34.2 from scratch; the full-label comparison is 65.2 versus 63.4 (Table 12). The larger low-label margin highlights the value of reusable features for dense annotation.

4.8 Representation Diagnostics

Figure 6 shows how activation patterns change across point and image stages. Intermediate states emphasize different spatial regions, providing distinct inputs to the branch objectives. Color ranks compare locations within a stage rather than absolute activation across layers; point sizes are display choices.

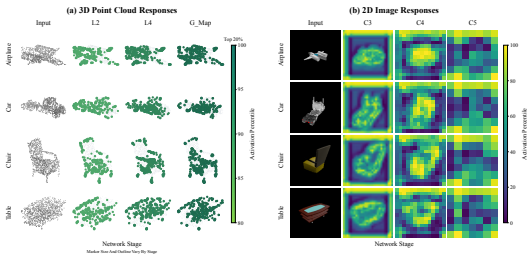

Figure 6: **Depth-dependent responses.** Colors show within-stage feature-norm ranks for four objects.

We further inspect 52 ShapeNet objects across two validation-selected runs, with and without pooled local matching. These diagnostics examine the internal state underlying the transfer results, using the same object set for each comparison. Full sampling and checkpoint details are retained in Appendix L.4.

From l_4 to the pre-pooling state, neighborhood reciprocity rises and in-degree variation falls (Figure 7). Deep aggregation thus changes the organization of the feature graph. Non-reciprocal edges persist, supporting a directed graph-shift formulation for the implementation and a symmetric reference for spectral analysis.

These measurements connect the abstract response operator to the learned neighborhoods: the source relation evolves with representation depth, so branch inputs contain more than repeated projections of a final descriptor. Reciprocity characterizes this relation, rather than serving as a downstream accuracy metric.

Local-token activity varies substantially between runs (Figure 8). Entire zero vectors cannot supply discriminative dot-product scores; a larger target set therefore needs informative descriptors as well as spatial coverage. The point bins and image grid make this failure mode visible within the pooled local extension.

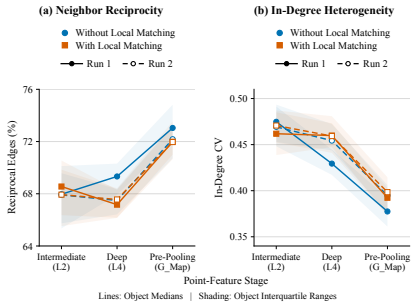

Figure 7: **Directed neighborhoods.** Reciprocity and in-degree variation over 52 objects; bands show interquartile ranges.

This distinction guides geometry-preserving alignment: token construction should maintain useful spatial support, and visibility should determine which observations share evidence. Local supervision complements object identity by specifying where a cross-modal match is supported.

Figure 9 examines whether the learned branch mixture repeats across runs. Local matching improves point-side consistency but not image-side consistency. Routing behavior is therefore modality dependent, motivating inspection of both pathways rather than interpreting a gate weight as a universal measure of importance.

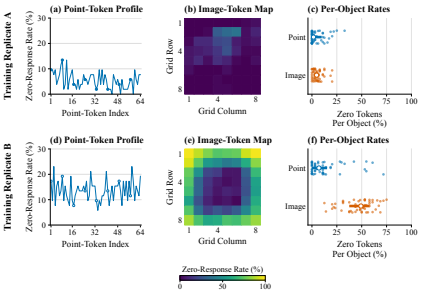

Figure 8: **Local-token activity.** Zero-vector rates for point bins and image grids in two local-matching runs.

Together, the diagnostics show that supervision acts on states with distinct spatial responses, evolving neighborhood structure, and unequal branch activity. Their downstream value is established by the controlled evaluations; the internal measurements identify where the structured interface can be refined further.

5 Conclusion

MS-Point makes internal structure an explicit interface for cross-modal learning. Encoder depth determines where supervision acts, and complementary neighborhood responses determine what it observes. Controlled comparisons and transfer evaluations support this design across graph and token encoders. The resulting principle extends naturally toward local objectives in which geometry and visibility identify the information shared across modalities.

The observability perspective locates a limitation of global alignment in the statistics visible to its loss, beyond the choice of backbone. It also clarifies why object-level transfer and spatial correspondence require complementary evidence. Selecting local targets according to shared visibility offers a direction for strengthening informative geometric distinctions while allowing unobserved and modality-specific structure to retain their own roles in the representation.

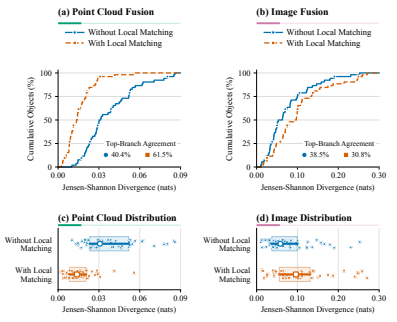

Figure 9: **Fusion consistency.** Cross-run gate divergence and top-branch agreement, measured separately in each modality.

Reproducibility Statement

The branch sets, operators, pooling, and losses are defined in Section 3. Dataset and training settings, the training algorithm, and extended ablations are provided in Appendices G–J. Appendix P states the graph-filter assumptions, and Appendix Q separates exploratory runs from the full-model results. We provide the source package, configuration definitions, and extended experimental analyses.

Ethics Statement

This work studies representations learned from existing point-cloud benchmarks and paired images. Performance on these datasets does not establish reliability for deployment across sensors, environments, or demographic contexts. Any reuse of the data must follow the original dataset terms. Dataset-specific biases and the limitations of rendered views should be considered when transferring the models to real scenes.

AI Use Statement

We used generative AI to assist with organization, language editing, literature and citation checks, mathematical exposition, experimental-design feedback, analysis of existing run records, figure and table preparation, and LaTeX formatting. We checked the resulting text and numerical comparisons against our experiment summaries and the available records. AI assistance did not generate synthetic benchmark measurements or perform model training. We take responsibility for the final claims and results.

References

Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. CrossPoint: Self-Supervised Cross-Modal Contrastive Learning for 3D Point Cloud Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9902–9912, 2022.

Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1534–1543, 2016.

Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, Qi Tian, and Longin Jan Latecki. GIFT: Towards Scalable 3D Shape Retrieval. *IEEE Transactions on Multimedia*, 19(6):1257–1271, 2017. doi: 10.1109/tmm.2017.2652071.

Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924. Curran Associates, Inc., 2020.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012*, 2015. URL `https://arxiv.org/abs/1512.03012`.

Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. PointGPT: Auto-regressively Generative Pre-training from Point Clouds. In *Advances in Neural Information Processing Systems*, volume 36, pp. 29667–29679. Curran Associates, Inc., 2023. doi: 10.52202/075280-1291.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-Temporal Self-Supervised Representation Learning for 3D Point Clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6535–6545, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution On X-Transformed Points. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Haotian Liu, Mu Cai, and Yong Jae Lee. Masked Discrimination forăSelf-supervised Learning onăPoint Clouds. In *European Conference on Computer Vision*, pp. 657–675, 2022a. doi: 10.1007/978-3-031-20086-1_38.

Jiaming Liu, Yue Wu, Maoguo Gong, Zhixiao Liu, Qiguang Miao, and Wenping Ma. Inter-Modal Masked Autoencoder for Self-Supervised Learning on Point Clouds. *IEEE Transactions on Multimedia*, 26:3897–3908, 2024. doi: 10.1109/tmm.2023.3317998.

Jiaming Liu, Linghe Kong, Yue Wu, Maoguo Gong, Hao Li, Qiguang Miao, Wenping Ma, and Can Qin. Triple Point Masking. *IEEE Transactions on Circuits and Systems for Video Technology*, 35 (9):8466–8477, 2025. doi: 10.1109/tcsvt.2025.3549786.

Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. OpenShape: Scaling Up 3D Shape Representation Towards Open-World Understanding. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44860–44879. Curran Associates, Inc., 2023a. doi: 10.52202/075280-1944.

Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-Shape Convolutional Neural Network for Point Cloud Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8895–8904, 2019.

Youquan Liu, Lingdong Kong, Jun CEN, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment Any Point Cloud Sequences by Distilling Vision Foundation Models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 37193–37229. Curran Associates, Inc., 2023b. doi: 10.52202/075280-1617.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, 2022b.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. In *International Conference on Learning Representations (ICLR)*, 2022.

Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked Autoencoders for Point Cloud Self-supervised Learning. In *European Conference on Computer Vision*, pp. 604–621, 2022. doi: 10.1007/978-3-031-20086-1_35.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017b.

Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with Reconstruct: Contrastive 3D Representation Learning Guided by Generative Pretraining. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28223–28243. PMLR, 2023.

Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric Back-Projection Network for Point Cloud Classification. *IEEE Transactions on Multimedia*, 24:1943–1955, 2022. doi: 10.1109/tmm.2021.3074240.

Jonathan Sauder and Bjarne Sievers. Self-Supervised Deep Learning on Point Clouds by Reconstructing Space. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-Lidar Self-Supervised Distillation for Autonomous Driving Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9891–9901, 2022.

D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013. doi: 10.1109/msp.2012.2235192.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 945–953, 2015.

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1588–1597, 2019.

Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J. Kusner. Unsupervised Point Cloud Pre-Training via Occlusion Completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9782–9792, 2021.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics (TOG)*, 38 (5):1–12, 2019. doi: 10.1145/3326362.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, 2015.

Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding. In *European Conference on Computer Vision (ECCV)*, pp. 574–591, 2020. doi: 10.1007/978-3-030-58580-8_34.

Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP-2: Towards Scalable Multimodal Pre-training for 3D Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27091–27101, 2024.

Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016. doi: 10.1145/2980179.2980238.

Hai-Tao Yu and Mofei Song. MM-Point: Multi-View Information-Enhanced Multi-Modal Self-Supervised 3D Point Cloud Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7):6773–6781, 2024. doi: 10.1609/aaai.v38i7.28501.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-Training 3D Point Cloud Transformers With Masked Point Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19313–19322, 2022.

Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-M2AE: Multi-scale Masked Autoencoders for Hierarchical Point Cloud Pre-training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27061–27074. Curran Associates, Inc., 2022. doi: 10.52202/068431-1962.

Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3D Representations From 2D Pre-Trained Models via Image-to-Point Masked Autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21769–21780, 2023.

Xiangdong Zhang, Shaofeng Zhang, and Junchi Yan. PCP-MAE: Learning to Predict Centers for Point Masked Autoencoders. In *Advances in Neural Information Processing Systems*, volume 37, pp. 80303–80327. Curran Associates, Inc., 2024. doi: 10.52202/079017-2553.

Xiangdong Zhang, Shaofeng Zhang, and Junchi Yan. Towards More Diverse and Challenging Pre-training for Point Cloud Learning: Self-Supervised Cross Reconstruction with Decoupled Views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28696–28706, 2025.

Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point Cloud Pre-training with Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22935–22945, 2024.

Table 6: Hierarchy and fusion on ScanObjectNN.

Cumulative change	Acc.
Global only	81.7
+ Intermediate targets	84.5
+ Cross-level coordination	85.8
+ Adaptive fusion	**86.5**

Table 7: Response sources on ScanObjectNN. Branch counts vary.

Response	Sources	Acc.
Hierarchy	–	86.5
Both	l_2	87.4
Both	l_2, l_4	88.3
Smoothing	l_2, l_4, g	87.9
Deviation	l_2, l_4, g	88.1
Both	l_2, l_4, g	**89.6**

A Analysis of Structured Supervision

MS-Point changes both the location and the inputs of cross-modal supervision. We analyze these choices separately from backbone expressiveness and additional head capacity, then describe how the joint losses and fusion affect their interpretation.

A.1 Supervision Location and Observable Statistics

Let $H_l(P;\theta)$ be the feature map at stage l, Φ an object-pooling operator, and q_l a projection head. Final-only alignment evaluates $q_g\Phi(H_g)$, whereas a hierarchy objective evaluates a family $\{q_l\Phi(H_l)\}$. The final loss already updates earlier stages through the chain rule. Direct targets add stage-specific readouts and gradient paths. A backbone that concatenates earlier maps before final pooling also retains some hierarchy information in its global descriptor.

A supervision-location control should therefore attach the same number of independent heads to the final map and match active capacity. This tests where the heads read features. Allocating unused parameters is insufficient, because inactive heads do not contribute to optimization.

Pooling can hide different neighborhood arrangements. For a fixed pooling operator Φ and deterministic projection q, $\Phi(H) = \Phi(H')$ implies that $q \circ \Phi$ and its loss against any image embedding assign the same value to both feature maps. The structural distinction is absent from the objective's current observations. A neighborhood-dependent readout can separate this equivalence class, giving the loss direct access to a difference that additional post-pooling heads cannot recover.

Consider a three-node path with self-loops. For this analytical example, let P denote its row-normalized graph operator, rather than the input point cloud:

$$P = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/3 & 1/3 & 1/3 \\ 0 & 1/2 & 1/2 \end{bmatrix}, \quad h = (1,0,0)^\top, \quad h' = (0,1,0)^\top. \tag{12}$$

Both signals have max–mean descriptor $(1, 1/3)$. However,

$$Ph = (1/2, 1/3, 0)^\top, \qquad Ph' = (1/2, 1/3, 1/2)^\top, \tag{13}$$

whose means are $5/18$ and $4/9$. The deviation maxima also differ: $1/2$ versus $2/3$. Responses before pooling can therefore distinguish these two arrangements. The example is an exact algebraic diagnostic, not a point-cloud benchmark or evidence of an accuracy gain.

When a response can disappear under pooling. There are also cases where filtering adds no information to a chosen readout. If P is doubly stochastic, then $\mathbf{1}^\top PH = \mathbf{1}^\top H$ and mean pooling gives zero on $(I - P)H$. For a row-stochastic random walk on an undirected graph, the corresponding conserved mean is weighted by its stationary distribution. Nonlinear adaptation, max pooling, and nonuniform degrees can change what remains observable. These facts explain why operator choice and pooling must be specified together, and why complementarity before pooling alone is insufficient to justify a representation-learning claim.

A.2 Interactions among the Training Objectives

The five objectives act simultaneously on shared encoder parameters, so their effects depend on their interactions. Same-branch contrast rewards consistency across augmentations; it may also

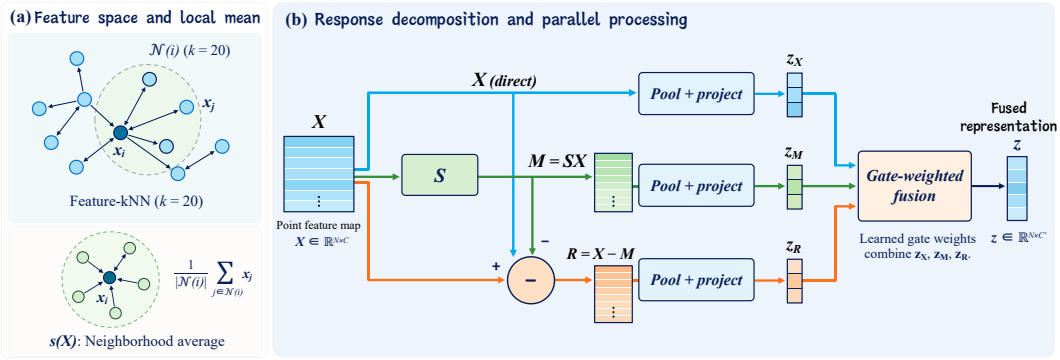

Figure 10: **From a local neighborhood to independent response paths.** A feature-space neighborhood defines an average and a node-wise deviation. The direct, mean, and residual paths are pooled and projected independently before gated fusion. The illustrated neighborhood uses $k = 20$. Pooling produces an object embedding; the final $N \times C'$ annotation should be read at the pre-pooling feature level, while our fused descriptor has width 256.

suppress a response that varies strongly with an augmentation. Cross-level contrast promotes shared information across depths; excessive weight can reduce useful specialization. Branch-level image alignment supplies direct visual targets, while fused alignment constrains the aggregate. Removing branch alignment still leaves an image signal through fused alignment and must not be described as removing all visual supervision.

For objective r, let $g_r = \nabla_\theta \mathcal{L}_r$. The joint gradient is

$$g = \sum_r \lambda_r g_r, \qquad \chi_{r,s} = \frac{g_r^\top g_s}{\|g_r\|_2 \|g_s\|_2 + \epsilon}. \tag{14}$$

The squared norm exposes the interaction explicitly:

$$\|g\|_2^2 = \sum_r \lambda_r^2 \|g_r\|_2^2 + 2\sum_{r<s} \lambda_r \lambda_s \langle g_r, g_s \rangle. \tag{15}$$

Positive cross terms reinforce a shared update direction; negative terms create competition. This gives a concrete optimization interpretation of our loss design: alignment preserves shared semantics, while response regularization promotes specialization. Fixed weights determine their relative contribution but do not by themselves make the cross terms positive. Stage-wise gradient norms and $\chi_{r,s}$ are useful diagnostics for testing this interaction during training; we propose them as follow-up measurements, separately from the loss-removal and weight-transfer results reported here.

Soft directional decoupling acts in the projected space, where each response can retain object identity while expressing a different direction. Appendix P derives its tangential gradient. We evaluate this mechanism through transfer accuracy and response similarity together: the desired outcome is useful specialization under a shared semantic target, rather than the smallest cosine alone.

A.3 Fusion and the Deployed Representation

For branch embeddings z_m, the gates produce

$$\alpha_m = \frac{\exp(a_m(z_m))}{\sum_{n\in\mathcal{A}} \exp(a_n(z_n))}, \qquad z_f = q_f\left(\sum_{m\in\mathcal{A}} \alpha_m z_m\right). \tag{16}$$

A large α_m is a property of this parameterization, not a causal estimate of a branch's usefulness. An input branch with a small weight can still affect training through its direct loss. In addition, gate weights and embedding norms jointly determine the weighted sum. Useful diagnostics include gate entropy, branch norms, retrieval quality per branch, and controlled branch interventions.

The pretraining fusion output and the downstream feature readout must also be distinguished. If a frozen-SVM evaluator reads the raw global backbone descriptor, the branch heads and fusion are auxiliary training modules. The test-time cost then differs from the pretraining parameter count. If

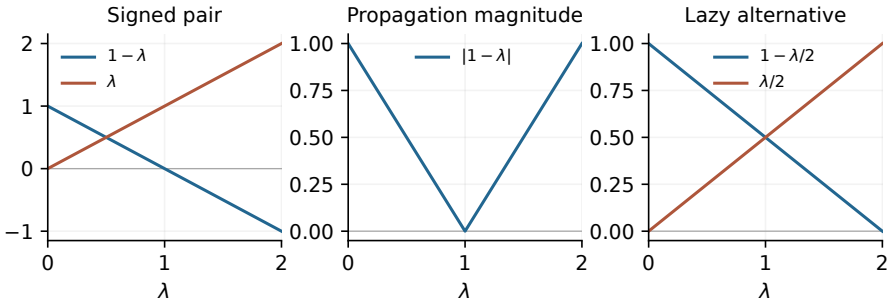

Figure 11: Analytic response functions, not measured model spectra. Left: the standard signed pair $1 - \lambda$ and λ. Center: the magnitude $|1 - \lambda|$ is not monotone on $[0, 2]$. Right: a proposed lazy pair with $\eta = 1/2$ has nonnegative monotone responses. The horizontal coordinate refers to a symmetric normalized Laplacian; this plot does not assign a real orthogonal Fourier basis to the directed implementation.

an evaluator instead concatenates branches or reads the fused embedding, that is a separate readout protocol. Each accuracy table should identify the actual readout rather than infer it from the training diagram.

B Expanded Evaluation of Structured Supervision

We evaluate recent baselines, modern encoders, controlled response inputs, repeated runs, and label-efficient scene learning. The complete results below supplement the main comparisons. Full-training experiments and exploratory runs are analyzed separately because they use different budgets and evaluation endpoints.

B.1 Recent Baselines under a Common Evaluator

Our reruns use the same ShapeNet subset or point–image pairs, 1,024 points, ScanObjectNN OBJ-BG, no voting, and a frozen linear SVM. Optimizer-search scope, augmentation settings, and seed count are held common in this comparison. TPM is attached to Point-MAE as a masking strategy; it is not a separate backbone. Access to images still differs between point-only and multimodal methods, so this is a comparison of pretraining systems, not an isolation of the objective across every row.

MS-Point improves over the controlled MM-Point rerun by 1.1 and 1.7 percentage points. These differ from the 1.2/1.8-point gaps to the published MM-Point row because the reference is different. The Transformer instance reaches 94.3/90.4. Neither this table nor the published benchmark table is an exhaustive ranking of modern 3D foundation models. Appendix Z separates the literature compilation from these reruns.

B.2 Changing the Backbone while Keeping the Objective Testable

Table 8 changes the supervision on the same point Transformer. Adding hierarchy gives 0.4/0.7 points; adding responses gives a further 0.4/0.8 points; fusion adds 0.3/0.6 points on ModelNet40/ScanObjectNN. This progression supports the compatibility of the interface with token features. It does not assert that Transformer blocks and EdgeConv stages have identical internal semantics. The token-graph construction and modern image-encoder interface are detailed in Appendix Y.

B.3 Do Neighborhood Inputs Matter beyond Additional Heads?

Table 9 retains the adaptors, projections, gates, fusion, and loss structure while changing the response inputs. We match parameter count and GFLOPs across these controls. Compared with duplicated features, the complementary responses improve accuracy by 0.7 and 1.4 points and part mIoU

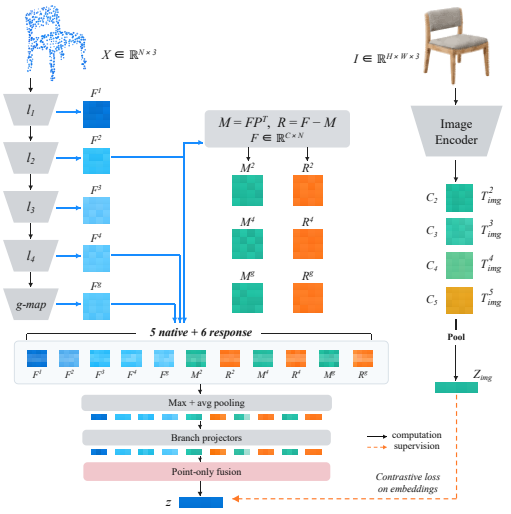

Figure 12: **Eleven point branches.** Five native maps and six responses at l_2, l_4, g enter pooling, independent projectors, and point-side fusion. Blue lines show computation; orange dashed lines denote embedding supervision. Image stages provide training targets.

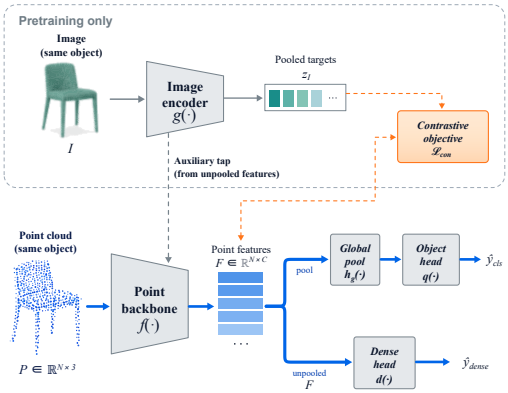

Figure 13: **Training and transfer paths.** Image features provide auxiliary supervision during pretraining. The point encoder then supplies a pooled descriptor to an object head or unpooled features to a dense head. Downstream point-only inference does not require the image pathway.

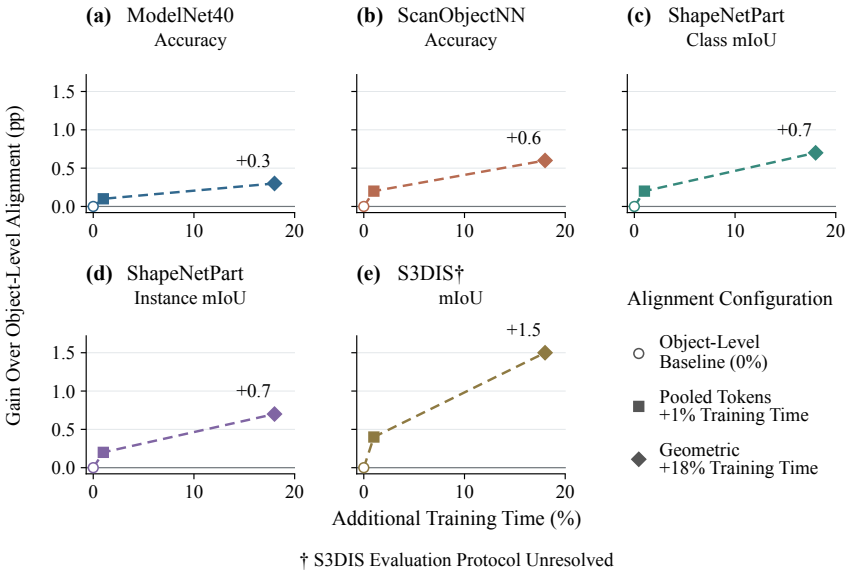

Figure 14: **Local targets and training cost.** Gains are relative to object-level alignment within each task (Table 46). Pooled tokens add 1% training time and geometry-preserving targets add 18%. ShapeNetPart class and instance mIoU are separate metrics. The S3DIS panel uses the local-study 67.5 baseline; its protocol note distinguishes this series from the other S3DIS evaluations.

Table 8: Same point Transformer and ResNet-50 image encoder. Part denotes ShapeNetPart mIoU as reported.

Objective	MN40	Scan	Part	H100 h
Global only	93.2	88.3	85.7	27.4
+ Hierarchy	93.6	89.0	86.1	28.5
+ Responses	94.0	89.8	86.5	29.6
+ Fusion (MS-Point-T)	94.3	90.4	86.9	30.4

Table 9: Matched controls: 73.37M parameters and 15.2 GFLOPs for all rows.

Response input	MN40	Scan	Part
Duplicated features	92.9	88.2	85.5
Random neighbors	93.0	88.3	85.6
Shuffled features	93.1	88.5	85.7
Smoothing + deviation	93.6	89.6	86.1

Table 10: Five pretraining seeds (mean±SD). Part: instance mIoU; S3DIS: Area-5 mIoU.

Metric	Global	MM-Point	MS-Point
n	5	5	5
MN40	92.80 ± 0.18	92.50 ± 0.20	93.60 ± 0.12
Scan	87.90 ± 0.32	87.90 ± 0.35	89.60 ± 0.24
Part	85.45 ± 0.16	85.30 ± 0.18	86.10 ± 0.13
S3DIS	66.95 ± 0.27	66.80 ± 0.29	68.20 ± 0.22

Table 11: Accuracy and pretraining cost. Views: encoded point/image inputs per iteration.

Metric	Global	MM-Point	MS-Point
Views	2/1	2/4	2/1
Params. (M)	71.84	73.10	73.37
GFLOPs	13.1	21.8	15.2
GiB	9.72	12.80	10.03
GPU h	24.6	36.9	30.5
MN40	92.8	92.5	93.6
Scan	87.9	87.9	89.6

by 0.6 points. Random neighbors and shuffled features also perform below the real-neighborhood configuration. Thus, in this full-training setting, branch duplication alone does not recover the performance of the structured inputs. The result tests the complete branch intervention; a separate point-response/image-response factorial control is needed to assign the benefit uniquely to the 3D operator.

The ten-pair archived pilot remains a separate result: its mean gain is 0.1337 points and its 95% interval is $[-0.1334, +0.4008]$. A longer-training gain does not retrospectively make that pilot statistically conclusive. Keeping both studies shows how the evidence changes with the training regime.

B.4 Repeated Runs and Practical Efficiency

We use five independent pretraining seeds, select checkpoints on validation data, and evaluate the test set once after selection. Table 10 gives the mean and standard deviation. Paired ModelNet40 intervals are $[+0.62, +0.98]$ against the global baseline and $[+0.85, +1.35]$ against MM-Point. The corresponding ScanObjectNN intervals are $[+1.38, +2.02]$ and $[+1.34, +2.06]$. All four intervals are above zero. Pairing is performed across runs; marginal standard deviations alone do not determine the paired uncertainty.

Relative to the one-image global baseline, MS-Point adds 2.13% parameters and 23.98% GPU time while improving both classification metrics. Relative to four-image MM-Point, it uses 17.34% less reported GPU time and 21.64% less peak memory. These are different reference points, and both matter. Reusing intermediate maps avoids branch-specific backbone passes, but two augmented point views still require two point-encoder evaluations. Absence of an MAE decoder explains an architectural difference; the measured system costs, not that fact alone, establish the reported speed difference.

B.5 Local Alignment and Low-Label Outdoor Transfer

The completed local comparison reports ShapeNetPart instance mIoU of 86.1, 86.3, and 86.8 for object-level, pooled multi-token, and geometry-preserving alignment, respectively. S3DIS mIoU is 67.5, 67.9, and 69.0, with relative times 1.00, 1.01, and 1.18 (Table 46). These are within-study dense-transfer gains. They do not share the DGCNN benchmark baseline of 59.8 or the repeated-run baseline of 68.20, so cross-table subtraction would be inappropriate.

For outdoor evaluation, we use a scene-level MS-Point instance with MinkUNet-34C and the SemanticKITTI training sequences 00–07 and 09–10; sequence 08 supplies validation mIoU over 19 classes (Behley et al., 2019). This is adaptation of the objective to voxel-stage features, not direct

loading of DGCNN weights into a sparse-convolution backbone. The local instance improves over

Table 12: SemanticKITTI sequence-08 mIoU by label budget. Missing results are not interpolated. The supplied combined SLidR/Seal reference is retained separately in the appendix.

Pretraining / initialization	1%	5%	10%	100%
Random initialization	34.2	48.8	53.7	63.4
Scene-only SSL	40.8	–	–	–
Global MS-Point	44.6	53.9	58.1	64.7
MS-Point + local	47.3	55.9	59.8	65.2

random initialization by 13.1 points at the reported 1% budget and 1.8 points at full labels. This pattern supports label-efficient representation reuse in this scene setting. It does not quantify annotation hours or establish superiority to every image–LiDAR baseline. The supplied "SLidR / Seal" row does not identify one unique method and is therefore excluded from the main ranking. Label-sampling units and the membership of the frequency groups are among the metadata still required for complete reproduction.

C Controlled Experiments and Further Analysis

We use benchmark evaluations, full-training controls, and short-run pilots to test complementary aspects of the method. Appendix B gives the full-training comparisons. Here we examine which factors each control isolates, how hierarchy and responses interact, and how training budget changes the evidence. Appendices S and Z provide the corresponding configurations and task results.

C.1 Parameter-Matched Paired Runs

The ten-seed experiment in Table 31 compares two implementations with exactly 73,367,890 trainable parameters. Both branches, projection heads, adaptors, gates, and losses remain active in the duplicated-feature control. This is more informative than comparing the response model with a single smaller global head. The mean terminal ModelNet40 SVM difference is $+0.1337$ percentage points, with a 95% paired interval of $[-0.1334, +0.4008]$. The interval includes zero, so these short runs do not establish a stable mean improvement. They also do not estimate the effect of the full-training configuration.

The intervention changes both modalities. The saved code shows that the duplicated-feature variant changes both sides of the alignment. In this variant, both the point response inputs and the image response inputs are copies of the unfiltered source. In the smoothing/deviation variant, point-neighborhood responses and image smoothing/residual targets are both enabled. Thus, this comparison tests the joint structured-response interface. It does not isolate the 3D graph operator from the changed 2D targets. A stronger factorial control would independently enable point responses and image responses in four cells while retaining the same trainable paths.

Equal parameter counts do not give equal runtime: recorded mean duration increases from 442.78 to 551.73 seconds, or 24.6%. Peak memory changes little in this pilot, but that does not imply equal compute. The completed controls in Table 9 report matched GFLOPs and close epoch times (Table 75). They address a longer-training setting; retaining the actual kernels and checking active-head gradients remain important reproducibility details.

C.2 Hierarchy–Response Interaction

For binary hierarchy and response factors, let Y_{ab} be the metric when hierarchy is $a \in \{0, 1\}$ and responses are $b \in \{0, 1\}$. We define

$$\Delta_{\text{hier}|0} = Y_{10} - Y_{00}, \quad \Delta_{\text{resp}|0} = Y_{01} - Y_{00}, \quad \delta_{\text{int}} = Y_{11} - Y_{10} - Y_{01} + Y_{00}. \tag{17}$$

The four-cell means in Table 38 give $\delta_{\text{int}} = -0.1$ percentage points on ModelNet40 and -0.5 on ScanObjectNN. Although the combined model exceeds either single-axis model in that table, these values indicate sub-additive gains. They do not support a claim of super-additive synergy. Fusion is a fifth configuration and must be analyzed as an additional intervention.

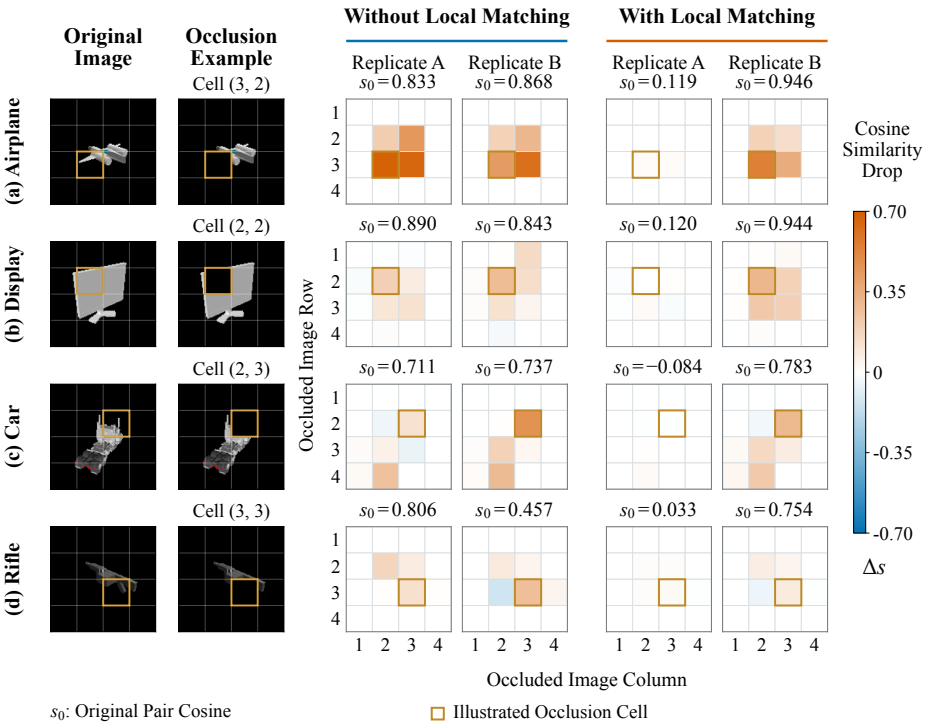

Figure 15: **Spatial sensitivity to image occlusion.** One 56×56 cell of a 224×224 image is replaced by border-median RGB while the point cloud is fixed. Each heatmap shows the cosine-similarity drop $\Delta s = s_0 - s_{\text{occ}}$ across a 4×4 grid; s_0 is printed above it. Four objects and two training replicates compare configurations with and without local matching. Orange outlines identify the displayed intervention.

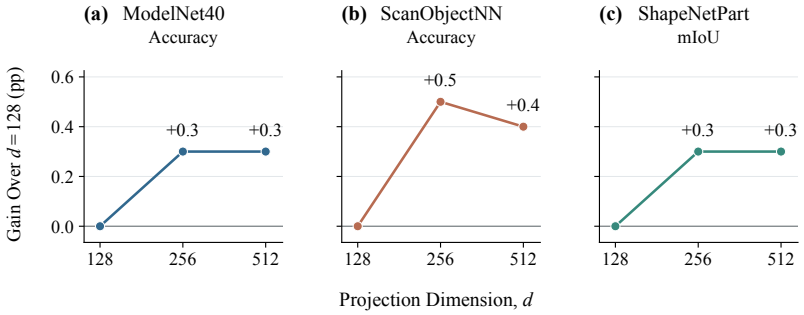

Figure 16: **Projection-width sensitivity.** Gains are percentage-point differences from $d = 128$ for each task (Table 51). At the displayed precision, increasing width from 256 to 512 leaves ModelNet40 and ShapeNetPart unchanged and reduces ScanObjectNN by 0.1 point. Lines connect the three evaluated widths.

The four cells differ in parameter count and runtime, so the contrast describes these configurations rather than isolating an interaction at fixed capacity. Paired seed-wise interactions, δ_i, would permit an uncertainty interval for the interaction itself. Computing an interval from four isolated means or borrowing the ten-seed pilot variance would not be valid.

C.3 Graph Structure, Shuffling, and Response Ablations

The graph controls in Table 39 are designed to distinguish useful neighborhood structure from merely duplicating heads. Their interpretation depends on the exact intervention. Let Π be a node

permutation. Jointly relabeling features and graph gives

$$(\Pi P \Pi^\top)(\Pi H) = \Pi(PH). \tag{18}$$

With permutation-invariant pooling, this relabeling need not change the descriptor. It is not a disrupted-neighborhood control. Holding P fixed while shuffling H changes the graph–signal association and addresses the intended question. Any unshuffling applied later must be specified so that other paths retain their intended identities.

A random-neighbor control should match the number of edges and the self-neighbor policy. Matching mean degree alone does not match connectivity, edge lengths, or computational sparsity. Single-response rows should report whether the unused head is removed, inactive, or replaced by an active capacity control. The different smooth-only and deviation-only values reported in Tables 39, 40, and 42 remain separate because their full configurations have not been reconciled.

C.4 Filter Flexibility and View Budget

The two-seed filter pilot in Table 32 identifies compact learnable filters as a promising accuracy–cost trade-off. Learnable first-order and polynomial variants improve mean best-validation accuracy by approximately 0.15 and 0.25 percentage points over the fixed pair. In contrast, the three-response variant adds over 14 million parameters for a mean change of approximately 0.03 points. These observations favor increasing filter flexibility before expanding the response bank. Repeated full-budget comparisons can determine how consistently the small pilot gains transfer.

The modest gains and added cost make a small number of learnable response coefficients a useful next test before adding full-width branches. Section D specifies the actual saved parameterizations. In particular, the learned first-order coefficient is a self/neighborhood mixture, not a learned hard Fourier cutoff. A coefficient near one makes the deviation branch small, which also provides a concrete collapse diagnostic.

The view pilot in Table 33 shows no mean validation benefit from additional encoded images at its short-run budget. The separate full-training table, Table 44, describes small gains from one to two images and no further gain from two to three. The two studies must not be pooled because their endpoints and training budgets differ. Equal-update and equal-GPU-time comparisons are both useful: the first measures information per update, while the second measures what can be learned within a compute budget.

C.5 Loss Deletion and Hyperparameter Transfer

Our loss-deletion comparison places the largest accuracy decrease at removal of branch-level alignment. In the completed comparison, this supports direct branch supervision as a useful training signal. It would not establish that every other loss is necessary, or that these five terms are optimal. Each deletion should retain the architecture and train from the same initialization distribution. A loss removed only during evaluation tests a different intervention.

Defaults should also be evaluated as a transfer rule. Selecting one configuration on pretraining validation and reusing it across downstream tasks tests practical reuse. Selecting different weights for each downstream task tests attainable task-specific accuracy at a larger search cost. The reported tuning differences range from 0.1 to 0.8 percentage points. Their practical size should be judged against a predeclared threshold and uncertainty estimates. A sweep of the cross-level coefficient alone cannot establish stability of all five weights.

C.6 Longer Runs of a Pooled Set-Matching Extension

Appendix W reports additional archived five-, 25-, and 100-epoch classification experiments. The implemented extension matches sets of index-pooled point bins and image-grid tokens; it does not use geometric patch correspondences. The two complete paired 100-epoch seeds give opposite differences, -0.8509 and $+0.7293$ percentage points, and a third local run has no terminal summary. These records do not establish a consistent gain from pooled set matching. They motivate retaining spatially defined patches and visibility in the proposed local extension, while leaving the archived index-pooled classification study distinct from the completed geometry-preserving dense-transfer comparison.

D EXTENSIONS OF THE STRUCTURED OBJECTIVE

The objective requires intermediate node or token features, a neighborhood relation, and image targets. This interface supports alternative encoders, learned responses, and local alignment. Appendix B reports the completed backbone and scene studies; the following formulation explains their common interface. Appendix Y gives an explicit token algorithm and develops visibility-aware local targets, distinguishing general design choices from unspecified implementation settings.

D.1 TOKEN BACKBONES AND STRONGER IMAGE ENCODERS

At selected block or stage d, a point-token encoder supplies $H^{(d)} \in \mathbb{R}^{M_d \times C_d}$ and centers $c_i^{(d)} \in \mathbb{R}^3$. A center-space k-NN or radius graph defines $P^{(d)}$, followed by the same propagation, deviation, pooling, and projection interface. Class tokens without geometric centers are excluded from this graph. Token merging must update features, centers, and membership information together. Fixed-resolution blocks give a semantic-depth axis; a spatial-scale axis is present only when sampling or pooling changes the support.

Patch-based masked encoders provide relevant token interfaces (Yu et al., 2022; Pang et al., 2022; Zhang et al., 2022). Their masking, tokenization, and pretraining initialization remain part of the controlled comparison. Adding structured targets to a randomly initialized transformer and comparing it with a differently pretrained transformer would confound the objective with initialization. To test generality, each named encoder should have both a global-objective row and a structured-objective row under the same data and evaluation settings.

For the image encoder, ConvNeXt stages provide convolutional stage outputs (Liu et al., 2022b); image-transformer patch tokens require a stated choice of blocks and spatial layout. Each 3D branch has an image-stage identity, response identity, and independent projection-head identity. Sharing one raw image stage across two point levels need not share the heads. Extracting C2–C5 is not equivalent to using all four stages in active losses: the actual mapping and fusion inputs must be listed.

Table 13: A controlled backbone evaluation design. Every cell requires both objectives; no prospective accuracy values are assigned.

Point encoder	Image encoder	Within-cell comparison
DGCNN	ResNet-50	Global versus structured
Named point transformer	ResNet-50	Global versus structured
DGCNN	ConvNeXt variant	Global versus structured
Same point transformer	Same ConvNeXt variant	Global versus structured

The graph-type observations in Appendix S compare attention affinities, feature neighborhoods, and geometric neighborhoods. Feature-space k-NN does not read attention weights directly, but its features are still produced by the attention encoder. Only a center-space graph is determined by the supplied geometry alone. Compatibility establishes how to attach the objective. The completed same-Transformer comparison tests its transfer benefit, while representation diagnostics remain needed to explain which token properties change.

Figures 17 and 18 illustrate two aspects of target routing: the image encoder may supply frozen teacher features, and one deep image stage may serve multiple point states through independent projections. The frozen-teacher diagram describes an interface option rather than a change to every experiment.

D.2 IMPLEMENTED LEARNABLE AND MULTI-RESPONSE FILTERS

The saved first-order extension learns one self-weight $\alpha = \sigma(a)$ per source and modality:

$$H_{\mathrm{s},\alpha} = \alpha H + (1-\alpha)PH, \qquad H_{\mathrm{d},\alpha} = H - H_{\mathrm{s},\alpha}. \tag{19}$$

The initial self-weight is 0.1 in the saved class definition. With three sources in each modality, the variant adds six scalar parameters. Under the symmetric construction and degree-coordinate convention, its signed responses are $1-(1-\alpha)\lambda$ and $(1-\alpha)\lambda$. Learning α changes the balance between self and neighborhood information; it does not introduce a sharp band boundary. As $\alpha \to 1$, the deviation tends to zero, so branch norms should be monitored. Figure 19 visualizes this self/neighborhood trade-off on a symmetric reference graph.

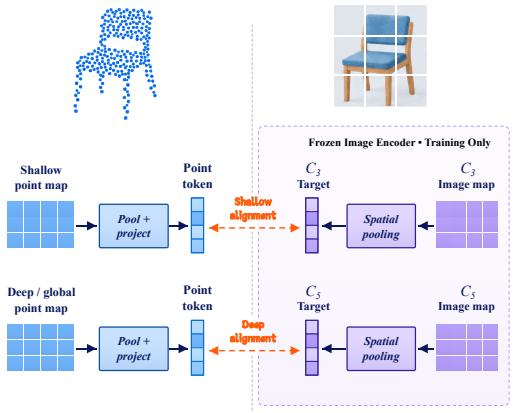

Figure 17: **Frozen visual-target variant.** Shallow and deep point descriptors use C3 and C5 targets. The illustrated frozen image encoder is a training-time teacher configuration; the stage interface also supports a trainable image encoder.

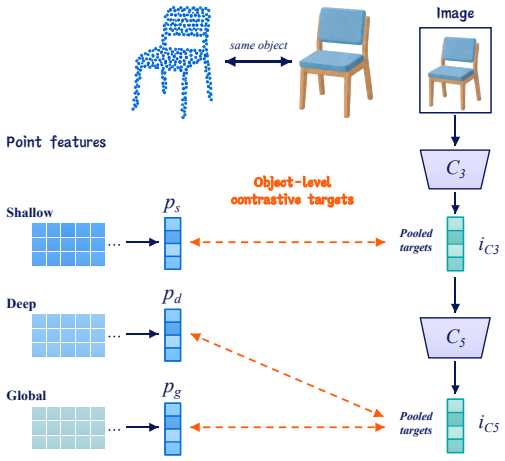

Figure 18: **Compact target routing.** A shared image hierarchy supplies C3 to the shallow point branch and C5 to the deep and global branches. Each target link compares object-level descriptors from the same object.

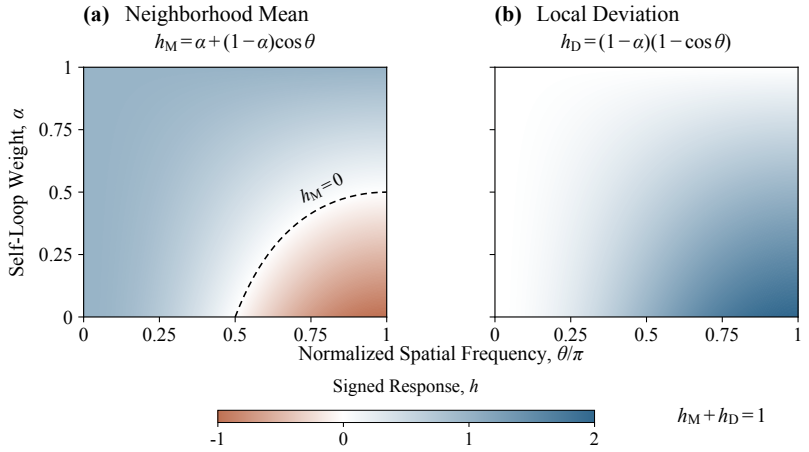

Figure 19: **Self-weight controls the complementary responses.** On a one-dimensional symmetric nearest-neighbor reference, the propagation response is $h_M = \alpha + (1-\alpha)\cos\theta$ and the deviation response is $h_D = (1-\alpha)(1-\cos\theta)$. Colors encode signed gain; the dashed curve marks $h_M = 0$. Their sum is one throughout.

The polynomial variant uses a convex combination

$$H_{\mathrm{s},\beta} = \beta_0 H + \beta_1 PH + \beta_2 P^2 H, \quad \beta = \mathrm{softmax}(b), \quad H_{\mathrm{d},\beta} = H - H_{\mathrm{s},\beta}. \tag{20}$$

Three coefficients per source and modality add 18 parameters. The graph indices are fixed within one forward application, so the second propagation is genuinely $P^2 H$. Rebuilding a feature graph after the first propagation would instead give $P_1(P_0 H)$, which is not a polynomial of one fixed shift. The code initializes the coefficient logits to $(-2, 0, 2)$; this differs from a uniform mixture and must be included when reproducing the pilot.

The three-response extension is a telescoping bank:

$$H_0 = P^2 H, \qquad H_1 = PH - P^2 H, \qquad H_2 = H - PH, \qquad H_0 + H_1 + H_2 = H. \tag{21}$$

In symmetric spectral coordinates, its responses are $(1-\lambda)^2$, $\lambda(1-\lambda)$, and λ. The middle response can change sign, and the first can retain high-variation modes. The responses overlap in frequency, and the additional heads substantially increase capacity, as the pilot parameter counts show.

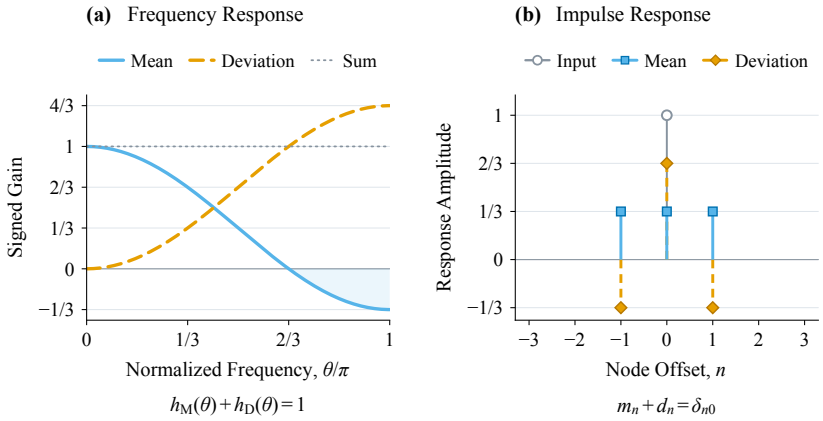

Figure 20: **Frequency and impulse responses of a three-tap reference.** Uniform averaging over offsets $-1, 0, 1$ gives $h_M(\theta) = (1 + 2\cos\theta)/3$ and $h_D = 1 - h_M$. Their impulse responses sum to the input impulse. The negative high-frequency gain of the mean response illustrates the distinction between signed complementarity and disjoint frequency bands.

D.3 Geometry-Preserving Local Alignment

The global objective pairs an image and a point cloud by object identity. It does not require a point-to-pixel calibration matrix. The saved code also contains a correspondence-light pooled-set objective, detailed with its actual outcomes in Appendix W. Geometry-preserving local alignment has a stronger requirement: local targets must represent corresponding visible content. Region-level distillation is an established route for image–LiDAR learning (Sautier et al., 2022); here we specify how a local objective could complement the structured object targets.

Let point patch i have center c_i, membership set $\mathcal{N}_i$, and descriptor u_i. Let image patch j have center p_j and descriptor v_j. Given projection π, valid target set $\mathcal{V}_i$, and visibility weight $\nu_{ij} \geq 0$, define

$$w_{ij} = \frac{\nu_{ij} \exp(-\|\pi(c_i) - p_j\|_2^2/(2\sigma^2))}{\sum_{k\in\mathcal{V}_i} \nu_{ik} \exp(-\|\pi(c_i) - p_k\|_2^2/(2\sigma^2))}. \tag{22}$$

Only patches with a nonzero denominator are used. Point transformations, camera projection, and image crop transforms must be composed consistently. Centers behind the camera or outside the valid image region are excluded from local positive pairs.

A candidate soft-target objective is

$$\mathcal{L}_{\text{local}} = -\frac{1}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{V}_i} w_{ij} \log \frac{\exp(\text{sim}(u_i, v_j)/\tau_{\text{loc}})}{\sum_{k\in\mathcal{C}_i} \exp(\text{sim}(u_i, v_k)/\tau_{\text{loc}})}, \tag{23}$$

where $\mathcal{C}_i$ specifies the candidate positive and negative set. Its construction must avoid treating alternate visible positives as unquestioned negatives. A symmetric image-to-point term is optional and constitutes a further design choice. We would train with $\mathcal{L} + \lambda_{\text{loc}}\mathcal{L}_{\text{local}}$, choosing λ_{loc} on validation data. This is an explicit extension, not a replacement definition of the object-level method.

Without calibration, learned attention or transport can define a set-matching objective, but the learned weights do not automatically become geometric correspondences. Adaptive pooling of an arbitrary point ordering into 64 bins is similarly not a geometric patch construction. Local experiments should retain patch centers and memberships and compare against equally sized correspondence-light heads. Boundary-sensitive metrics, visibility perturbations, and calibration perturbations would determine when local supervision helps or becomes unreliable.

D.4 Scene-Level and Low-Label Transfer

A sparse-voxel backbone can expose intermediate voxel features and spatial centers for the same structured objective. This is a new scene-level instantiation. A DGCNN checkpoint cannot be

loaded directly into a sparse convolutional network with unrelated parameter shapes. The transfer path must specify whether the method trains a scene encoder from scratch with structured SSL, distills from a point encoder, or transfers compatible weights.

For SemanticKITTI, the standard single-scan setting separates training sequences 00–07 and 09–10 from validation sequence 08 and evaluates 19 semantic classes (Behley et al., 2019). The completed low-label study still needs to specify whether 1%, 5%, or 10% labels refer to frames, points, or another sampling unit. The same sampled subsets must be used across initializations. Validation-sequence mIoU is not a test-server result, and single-scan and multi-scan settings should remain separate.

Pretraining camera use, downstream camera use, and deployed camera use are distinct fields. A point-only deployed encoder may still benefit from privileged visual pretraining information. Comparisons should state the visual teacher, its prior training data, and any cached-feature cost. SLidR and other region-distillation methods must be evaluated separately rather than combined into one baseline row. The completed outdoor results are reported in Table 12; the combined SLidR/Seal reference is kept separately because it does not identify a unique baseline.

E Evaluation, Efficiency, and Reproducibility

E.1 Comparable Evaluation Protocols

A comparable result requires the data, model, and evaluator to be specified together. We represent these choices by a protocol tuple

$$\mathcal{P} = (D_{\text{pre}}, T_{\text{prior}}, E, N, D_{\text{down}}, s, r, v, c), \tag{24}$$

where the entries denote pretraining data, prior teacher information, encoder and initialization, input point count, downstream dataset, split, feature readout, voting policy, and checkpoint/hyperparameter selection rule. Reporting $\mathcal{P}$ makes clear which contrasts isolate an objective and which compare complete systems.

Frozen linear evaluation and full fine-tuning answer different questions. The former tests the selected feature readout with the encoder fixed. The latter also measures how initialization interacts with downstream optimization. ScanObjectNN variants differ in background and perturbations; a score without the split cannot establish a controlled advantage. Likewise, ShapeNetPart class mIoU and instance mIoU are different summaries, and S3DIS Area-5 and cross-validation settings should not be combined into one ranking.

We separate the DGCNN benchmark and local-alignment configurations. The DGCNN benchmark yields ShapeNetPart mIoU 86.0 and S3DIS mIoU 59.8. The local-alignment observations use a baseline of 86.1 instance mIoU and 67.5 S3DIS mIoU. These are not interchangeable denominators for a gain. Their splits, models, and fine-tuning settings must be matched before a common gain can be computed.

Table 14: Minimum fields needed to make supplementary comparisons reproducible. This is a reporting schema, not a claim that every archived result contains these fields.

Scope	Required fields
Pretraining	Data subset hash, teacher source, checkpoint initialization, augmentation and view counts
Architecture	Backbone identity, branch set, graph rule, image map, projection widths, active parameters
Optimization	Batch size, updates, optimizer, schedule, temperature, exact loss reductions, precision
Selection	Validation split, selection metric, checkpoint rule, hyperparameter search budget
Classification	Dataset variant, input sampling, SVM settings, feature readout, voting
Segmentation	Split/fold, output metric, fine-tuning head and schedule, sampling, voting
Cost	Device count/type, timing boundaries, rendering/cache/validation costs, memory metric
Provenance	Run identifier, source/configuration hashes, seed, logs and checkpoint identifier

E.2 Independent Runs, Paired Contrasts, and Selection

For n independent pretraining seeds, we report the mean and sample standard deviation. Pairing is meaningful when two variants share a planned randomization scheme, such as the same initialization

seed and sampled data subset. It does not make the observations identical or remove every source of training variance. The uncertainty for a mean paired effect follows the variance of $d_i = a_i - b_i$:

$$\overline{d} = \frac{1}{n}\sum_i d_i, \qquad s_d^2 = \frac{1}{n-1}\sum_i (d_i - \overline{d})^2, \qquad \mathrm{CI}_{95} = \overline{d} \pm t_{0.975,n-1}\frac{s_d}{\sqrt{n}}. \tag{25}$$

The interval cannot be recovered by subtracting marginal intervals for two method means. Few-shot episodes characterize sampling variability conditional on the pretrained encoder; hundreds of episodes do not substitute for multiple independently pretrained encoders. For low-label studies, both the labeled-subset variability and pretraining variability should be distinguished.

The exact sign-flip calculation used for the ten pairs evaluates all 2^{10} assignments to the signs of d_i and compares the resulting mean magnitude with the observed value. Its interpretation requires the corresponding symmetry or exchangeability assumption under the null. Neither a nonsignificant paired test nor a near-zero mean proves equivalence. An equivalence or noninferiority claim needs a prespecified practical margin and an appropriate interval-based decision.

Checkpoint selection uses validation data. A final test evaluation follows the selected checkpoint and hyperparameters, rather than choosing the largest test value across epochs. The filter pilot uses best validation values, so it should be treated as exploratory model selection. Two seeds do not support a stable ranking among these alternatives. A confirmatory study should specify primary comparisons, stopping rules, and treatment of multiple tests, and retain every seed-level outcome.

E.3 Cost Accounting at Two Reference Points

Structured supervision adds graph operations, projections, fusion, and batch contrastive similarities. With V_p point views and V_i encoded images, a useful accounting decomposition is

$$C_{\mathrm{step}} = V_p C_{\mathrm{point}} + V_i C_{\mathrm{image}} + C_{\mathrm{graph}} + C_{\mathrm{response}} + C_{\mathrm{heads}} + C_{\mathrm{contrast}}, \tag{26}$$

$$C_{\mathrm{response}} = O\left(\sum_{s\in\mathcal{S}} N_s k_s C_s\right), \quad C_{\mathrm{contrast}} = O(TB^2 d). \tag{27}$$

Here T counts evaluated contrastive terms, B is batch size, and d is embedding width. The graph-construction cost is separate from propagation; brute-force feature-space search can have quadratic dependence on point count. Channel-chunked gathering changes peak temporary memory but does not eliminate the associated arithmetic. Two point views require two point-encoder evaluations even when all branches reuse their intermediate maps.

Two comparisons answer different cost questions. Against a one-image global baseline, the structured objective adds work. Against a four-image multimodal configuration, reduced image encoding can offset that work. Table 11 reports 2.13% more parameters, 16.03% more GFLOPs, and 23.98% more GPU hours than the global row; relative to four-image MM-Point, it reports 17.34% fewer GPU hours and 21.64% less peak memory. We compute these percentages from the matched-hardware resource study. The analytical estimates in Table 19 retain their separate scope.

A final profile should identify hardware, device count, precision, batch size, warm-up, synchronization, and stopping rule. Rendering, cached teacher features, validation, SVM fitting, and checkpoint I/O should either be included or reported separately. Total GPU hours and epoch time must agree under the stated boundaries. For example, 30.5 hours divided by 50 minutes per epoch implies 36.6 single-GPU epochs; the schedule and timing boundaries are needed to reconcile those entries. Absence of a reconstruction decoder similarly does not guarantee that a two-view, image-guided system is faster than an MAE.

E.4 Label Efficiency and Reuse

The practical use case is pretraining a reusable point encoder when downstream labels are limited. An evaluation should fix the labeled subsets and compare random initialization, point-only SSL, global point–image SSL, and structured point–image SSL under the same downstream capacity and budget. Accuracy or mIoU should be reported as a function of the label fraction, together with uncertainty and per-class behavior where class imbalance matters.

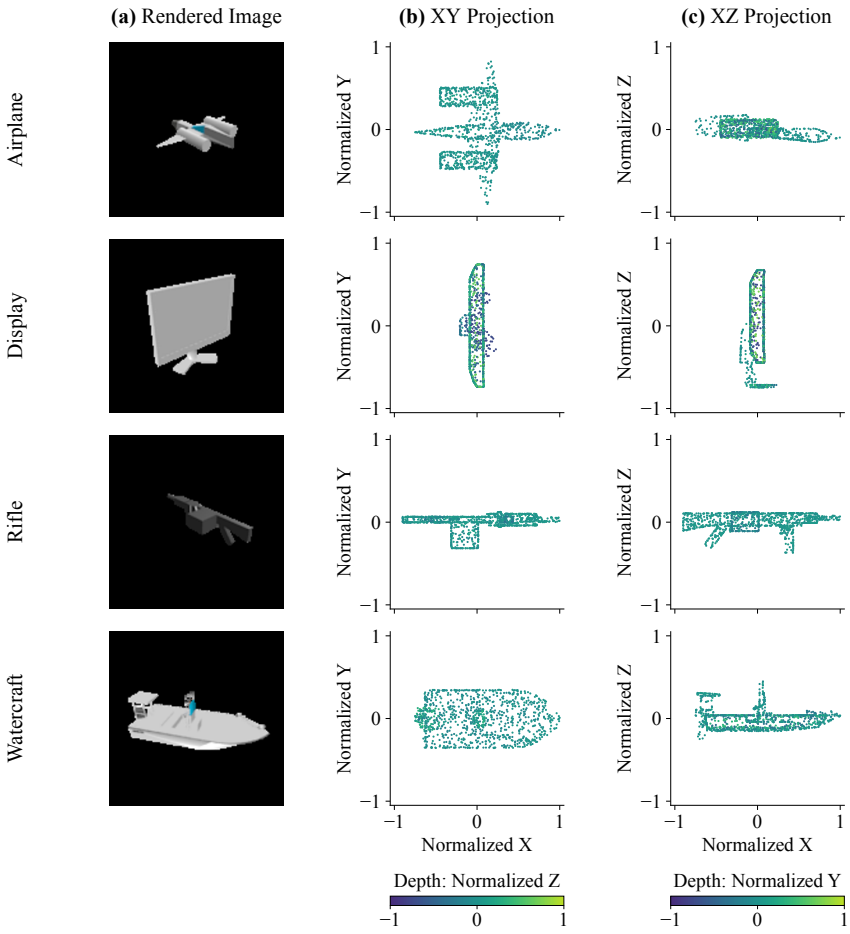

Figure 21: **Paired inputs and geometric projections.** Each row shows a fixed ShapeNet rendering and XY/XZ projections of the same object's 1,024 points. Coordinates are centered and normalized to the unit sphere. Color encodes omitted Z or Y depth on $[-1, 1]$. Pairing is by object identity; the orthographic views are not calibrated to the rendering camera.

For transfer to J tasks, total cost is

$$C_{\text{reuse}} = C_{\text{pre}} + \sum_{j=1}^{J} C_{\text{transfer},j}, \qquad C_{\text{scratch}} = \sum_{j=1}^{J} C_{\text{scratch},j}. \tag{28}$$

A claim of amortized savings requires comparable target quality and all relevant costs. Reusing one fixed pretrained encoder differs from tuning a separate pretraining configuration for each task. Pointwise labels and object categories impose different annotation work, but no annotation-time saving is quantified here because it has not been measured.

E.5 Additional Benchmark Comparisons

F Discussion and Conclusion

Object-level agreement is a useful training signal, but it does not fully specify which internal states should support transfer or which neighborhood statistics the objective should observe. MS-Point addresses these two questions through hierarchical targets and complementary responses computed before pooling. The formulation adds explicit supervision to existing encoder features, with independent image targets and adaptive fusion.

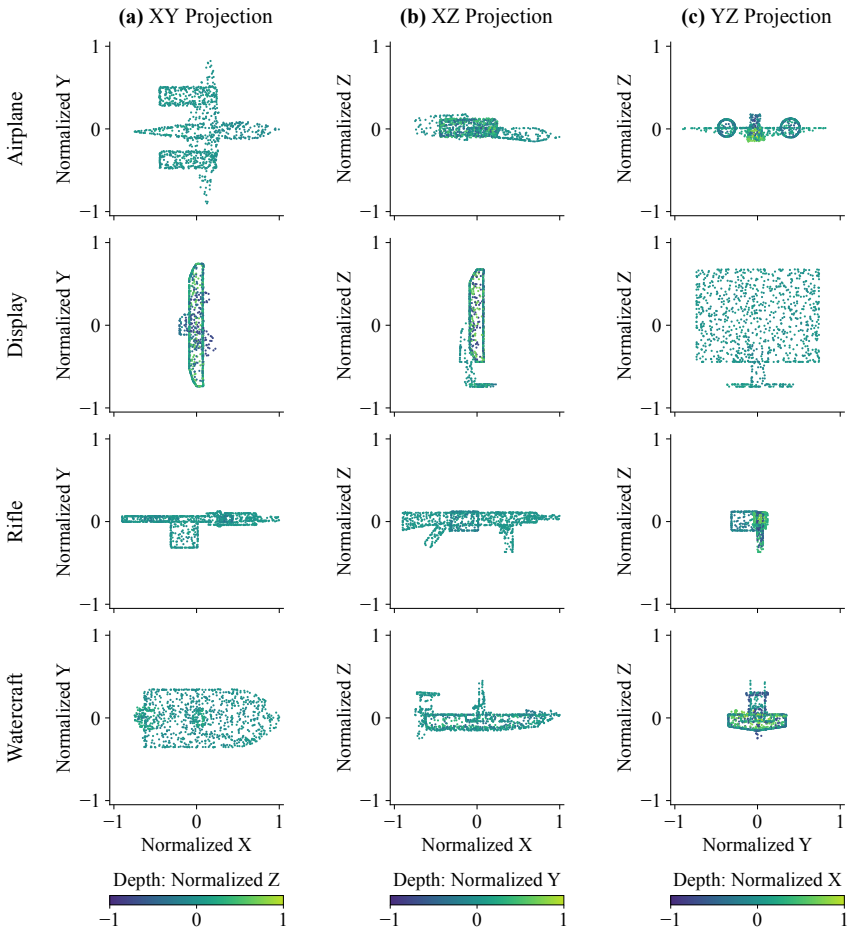

Figure 22: **View-dependent geometric overlap.** XY, XZ, and YZ views project the same 1,024 points; color encodes the omitted Z, Y, and X coordinate, respectively. Equal axis scales expose changes in apparent extent and overlap across projections. These geometric views motivate explicit visibility and camera information for local matching.

Table 15: **ScanObjectNN benchmark references.** The literature rows use different evaluation settings, including fine-tuning, and do not constitute a uniform frozen-SVM ranking.

Method	Accuracy (%)
GBNet (Qiu et al., 2022)	80.5
PointMLP (Ma et al., 2022)	85.2
Point-BERT (Yu et al., 2022)	83.1
MaskPoint (Liu et al., 2022a)	84.3
Point-MAE (Pang et al., 2022)	85.2
OcCo (Wang et al., 2021)	78.3
STRL (Huang et al., 2021)	77.9
CrossPoint (Afham et al., 2022)	81.7
MM-Point (Yu & Song, 2024)	87.8
MS-POINT (Ours)	**89.6**
Improvement	*+1.8*

Our experiments connect the supervision design to transfer across tasks and encoders. Same-backbone Transformer comparisons support the portability of the objective, and parameter- and computation-matched controls show that duplicated projections alone do not reproduce the full-training gains. Local alignment improves dense transfer, while low-label SemanticKITTI experiments evaluate a scene-level instance. These results complement classification and few-shot transfer. The shorter five-epoch pilot remains statistically inconclusive, and the longer index-pooled set extension has mixed outcomes, indicating that the benefit depends on the training regime and on how local support is defined.

The next step is selective alignment under partial observation. A complete point cloud and a single image share only the visible surface, while neighborhood deviations contain both useful detail and sensor noise. Geometry, visibility, and shared semantic evidence can guide which responses receive a cross-modal target. This extends our central principle—making structure observable to supervision—toward targets that also distinguish shared content from modality-specific information.

F.1 Further Context on Graph Responses and Local Supervision

Graph responses as a supervision interface. Graph signal processing expresses filtering as a function of a graph operator (Shuman et al., 2013). Polynomial filters can be evaluated through local propagation without computing eigenvectors (Defferrard et al., 2016). Our first-order pair follows this established framework. The contribution is not a new graph Fourier transform. It is the separate supervision of neighborhood prediction and deviation, together with hierarchy-aware branch targets. Spectral identities justify the operator interpretation under stated assumptions; controls are needed to establish any representation-learning benefit.

Local alignment and partial observation. PointContrast uses point-level consistency for dense transfer (Xie et al., 2020). SLidR transfers image information to LiDAR regions, and Seal distills vision foundation models to point-cloud sequences (Sautier et al., 2022; Liu et al., 2023b). Such methods explicitly address spatially localized supervision. Our object-level branches require shared object identity, not exact point–pixel calibration. Their extension to regions requires support geometry and visibility. Neighborhood organization can help resolve semantic ambiguity between candidate regions, but cannot supply the unseen surface of a single view.

Scale and evaluation protocol. OpenShape and ULIP-2 expand the data and teacher information available to multimodal 3D learning (Liu et al., 2023a; Xue et al., 2024). These developments are relevant context, but comparing their zero-shot results with a frozen linear SVM confounds the evaluator, training data, and supervision source. Published comparisons establish context; controlled reruns and same-backbone interventions evaluate the structured objective.

G Additional Implementation Details

G.1 Dataset Usage and Evaluation Protocols

We evaluate MS-Point on widely used point cloud representation learning benchmarks. ShapeNet is used as the primary pretraining corpus, following prior cross-modal self-supervised protocols. ModelNet40 and ModelNet10 are used for clean object-level linear classification. ScanObjectNN is used to evaluate robustness under real-world scanning noise, partiality, clutter, and background contamination. ShapeNetPart and S3DIS are used for dense prediction transfer.

Figure 21 illustrates the object-paired inputs. Geometric depth colors describe the input coordinates, whereas Figure 6 visualizes learned responses.

For linear evaluation, the pretrained point encoder is frozen and only a linear SVM classifier is trained on the extracted object-level representation. This protocol isolates the representation quality from task-specific nonlinear adaptation. For few-shot classification, we follow the common N-way K-shot evaluation protocol and report the mean and standard deviation over repeated trials. For segmentation fine-tuning, the pretrained backbone is used as initialization and then optimized with the downstream segmentation head.

Table 16: **Summary of datasets and evaluation protocols.** We follow standard protocols used in prior point cloud self-supervised and cross-modal pretraining works.

Dataset	Usage	Task	Evaluation Protocol	Main Metric
ShapeNet (Chang et al., 2015)	Pretraining	Self-supervised 2D–3D alignment	Two point views + rendered image	Pretraining loss
ModelNet40 (Wu et al., 2015)	Downstream	Object classification	Frozen backbone + linear SVM	Accuracy
ModelNet10 (Wu et al., 2015)	Downstream	Object classification	Frozen backbone + linear SVM	Accuracy
ScanObjectNN (Uy et al., 2019)	Downstream / ablation	Real-world object classification	Frozen backbone + linear SVM	Accuracy
ShapeNetPart (Yi et al., 2016)	Downstream	Part segmentation	Fine-tuning	OA / mIoU
S3DIS (Armeni et al., 2016)	Downstream	Indoor semantic segmentation	Fine-tuning	OA / mIoU

G.2 Architecture Details

The default 3D encoder is DGCNN (Wang et al., 2019). Let $X^{(l)} \in \mathbb{R}^{C_l \times N}$ denote the feature map at hierarchy level l. We extract intermediate representations from four EdgeConv stages and the final global stage:

$$\mathcal{D} = \{l_1, l_2, l_3, l_4, g\}. \tag{29}$$

For each level, an object-level descriptor is produced by max-average pooling:

$$f^{(l)} = \left[\mathrm{MaxPool}(X^{(l)}), \mathrm{AvgPool}(X^{(l)})\right], \qquad l \in \mathcal{D}. \tag{30}$$

The descriptor concatenates the channel maxima and means.

We construct graph responses at selected levels

$$\mathcal{S}_{\mathrm{spec}} = \{l_2, l_4, g\}, \tag{31}$$

which span intermediate and deep feature maps. Each selected level supplies a neighborhood prediction and its deviation:

$$X^{(s)}_{\mathrm{smooth}} = X^{(s)}\tilde{A}^{\top}, \qquad X^{(s)}_{\mathrm{dev}} = X^{(s)}(I - \tilde{A})^{\top}, \qquad s \in \mathcal{S}_{\mathrm{spec}}, \tag{32}$$

where $\tilde{A} = D^{-1}(W + I)$ uses the receiving-row convention from Section 3. The two raw responses are complementary before independent branch processing.

The image encoder is a ResNet-50 (He et al., 2016). We extract stage-wise feature maps from four image stages:

$$\mathcal{M} = \{2, 3, 4, 5\}. \tag{33}$$

Each stage produces a semantic image descriptor, and each stage is further decomposed into coarse/fine branches through blur-residual filtering:

$$F^{\mathrm{smooth}}_m = \mathrm{Blur}(F_m), \qquad F^{\mathrm{dev}}_m = F_m - F^{\mathrm{smooth}}_m. \tag{34}$$

The branch map selects image targets with corresponding base or response roles. This pairing does not assume an exact correspondence between the modalities' spatial or spectral coordinates.

G.3 Training Configuration

We optimize with Adam (Kingma & Ba, 2015) and cosine learning-rate decay, using the loss weights listed below.

The reported configuration follows the data augmentation pipeline and contrastive temperature of CrossPoint (Afham et al., 2022). Additional settings include the source levels, branch map, and weights for cross-level coordination, fused alignment, and response diversity. Matching these choices alone does not establish a fully controlled comparison; Section E lists the remaining protocol factors.

The nominal settings are $\lambda_{\mathrm{cross}} = 0.4$ and $\lambda_{\mathrm{band}} = 0.05$. The former weights agreement across selected levels; the latter penalizes squared cosine between projected responses. Appendix K reports the local sweeps around these settings.

Table 17: **Default training configuration.** The values are chosen to match prior cross-modal point cloud pretraining protocols while introducing minimal additional tuning.

Component	Hyperparameter	Default Value
Input	Number of points	1024
Input	Rendered views per object	1
Point augmentation	Scaling / rotation / translation / jitter / dropout	Enabled
Image augmentation	Random crop / normalization	Enabled
Optimizer	Optimizer	Adam
Scheduler	Learning-rate schedule	Cosine decay
Contrastive loss	Temperature τ	0.10
Spectral branch	Source levels $\mathcal{S}_{\text{spec}}$	$\{l_2, l_4, g\}$
Cross-branch pairing	Default pairing	Adjacent
Loss weight	λ_{same}	1.0
Loss weight	λ_{cross}	0.4
Loss weight	$\lambda_{\text{cmid}-\text{l}}$	1.0
Loss weight	$\lambda_{\text{cmid}-\text{f}}$	1.0
Loss weight	λ_{band}	0.05

G.4 Detailed Formulation and Architecture

G.4.1 Problem Definition and Overview

Given an object point cloud $P \in \mathbb{R}^{N\times 3}$ and a paired rendered image I, we construct two independently augmented point views $P^{(1)}$ and $P^{(2)}$. Let $\mathcal{D} = \{l_1, l_2, l_3, l_4, g\}$ denote the five base hierarchy levels and $\mathcal{S}_{\text{spec}} = \{l_2, l_4, g\}$ the response sources. We use the branch index set

$$\mathcal{A} = \{(l, \text{base}) : l \in \mathcal{D}\} \cup \{(s, b) : s \in \mathcal{S}_{\text{spec}},\ b \in \{\text{smooth}, \text{dev}\}\}, \qquad |\mathcal{A}| = 11. \tag{35}$$

We use smooth and dev for smoothing and deviation; the table labels "low/high" refer to these responses. The fused representation aggregates the eleven branch inputs and is not counted as another input branch. Figure 2 shows the point and image pathways and their training connections. A nine-branch implementation that retains only l_2, l_4, g as base sources is a separate model variant, documented in Appendix R.

From the two questions to the training graph. The hierarchy determines the states to supervise; responses determine the statistics presented by each state. Filtering occurs before pooling because additional heads after an identical pooled descriptor cannot recover a difference that the descriptor does not expose. The method improves how structure is represented and used; it does not reconstruct all spatial information discarded by pooling. The same interface can read point-token features and centers, as specified in Appendix Y.

G.4.2 Multi-Level Point Representations

We treat the point encoder as a sequence of spatially indexed feature states and attach supervision to these states before their final aggregation. In a graph encoder, each state is formed through relation-conditioned message passing: a node is updated using both its current feature and its contrast with neighboring features. For a neighborhood $\mathcal{N}_i^{(l)}$ at depth l, this operation takes the form

$$e_{ij}^{(l)} = \phi_l\left(h_i^{(l)}, h_j^{(l)} - h_i^{(l)}\right), \qquad h_i^{(l+1)} = \text{AGG}_{j\in\mathcal{N}_i^{(l)}}\, e_{ij}^{(l)}. \tag{36}$$

Here ϕ_l learns an edge message and AGG is a permutation-invariant neighborhood reduction. DGCNN (Wang et al., 2019) instantiates this formulation with feature-space neighborhoods and channel-wise maximum aggregation. As features evolve, the neighborhood relation adapts, and successive updates integrate context through different paths. This supplies a natural sequence of structural states for our hierarchy objective: an intermediate map contains a different history of relational aggregation from the final map.

We use these intermediate states as direct cross-modal targets. Instead of requiring the final descriptor to carry every distinction useful at earlier depths, each selected stage receives its own image-guided constraint. The graph backbone provides the states; MS-Point determines which of them are observed by the training objective. The Transformer construction in Section 3.2 uses the same supervision interface with token features and spatial centers.

The resulting hierarchy is

$$\{X^{(l)}\}_{l\in\mathcal{D}_{\mathrm{int}}}, \qquad \mathcal{D}_{\mathrm{int}} = \{l_1, l_2, l_3, l_4\}. \tag{37}$$

The stages differ in receptive field and feature depth. This motivates exposing them as separate training targets, without assuming a fixed semantic role for each layer. The global-stage map g is read before object pooling so that its nodes still admit a neighborhood relation.

We construct a common object-level interface for the selected structural states by combining two complementary readouts of each feature map $X^{(l)} \in \mathbb{R}^{C_l\times N}$. The descriptor preserves strong channel responses alongside activation statistics distributed over the point set:

$$f^{(l)} = \big[\mathrm{MaxPool}(X^{(l)}),\ \mathrm{AvgPool}(X^{(l)})\big], \qquad l \in \mathcal{D}. \tag{38}$$

This readout combines selective evidence from strongly activated locations with the aggregate response of the whole set. Each stage thus supplies a compact semantic anchor for its own visual target. Neighborhood structure enters through the source map and the pre-pooling graph responses; the joint readout then presents their statistics to the contrastive objective. The permutation-invariant construction follows the set-aggregation principle of PointNet and the global descriptor used in DGCNN (Qi et al., 2017a; Wang et al., 2019).

The base descriptor family is

$$\mathcal{F}_{\mathrm{spatial}} = \{f^{(l)} : l \in \mathcal{D}\}, \tag{39}$$

where $f^{(g)}$ is the global-stage descriptor. Each member receives its own projected cross-modal target.

Figure 3 summarizes these five independently supervised readouts.

G.4.3 Complementary Graph Responses

Figure 4 shows the graph response construction. The operator acts before pooling, so neighborhood statistics enter the branch descriptors.

Neighborhood Prediction and Deviation

Separating consensus from local contrast. We expose two structural statistics that a pooled object descriptor mixes. The smoothing branch establishes a neighborhood-consensus reference: features shared by connected regions reinforce one another through propagation. The deviation branch records how each node departs from that reference, making localized feature transitions available to the objective. Such transitions can carry discriminative cues at part junctions, creases, and region boundaries. By applying independent image-guided targets to the two responses before descriptor fusion, we let semantic supervision select useful local agreement and local contrast separately. Both branches reuse the same source map and neighborhood relation.

For a selected level $s \in \mathcal{S}_{\mathrm{spec}}$, let $X^{(s)} \in \mathbb{R}^{C_s\times N}$ be the corresponding point feature map. We construct a normalized neighborhood propagation matrix

$$\tilde{A} = D^{-1}(W + I), \tag{40}$$

where W is a feature-space k-NN adjacency, $D_{ii} = \sum_j (W+I)_{ij}$, and row i specifies the neighbors received by node i. With channels in rows, the two responses are

$$X^{(s)}_{\mathrm{smooth}} = X^{(s)}\tilde{A}^{\top}, \qquad X^{(s)}_{\mathrm{dev}} = X^{(s)}(I - \tilde{A})^{\top}, \qquad s \in \mathcal{S}_{\mathrm{spec}}. \tag{41}$$

The responses sum to $X^{(s)}$ before branch-specific processing. For an undirected reference with $W = W^{\top}$, define the symmetric normalization $S = D^{-1/2}(W+I)D^{-1/2}$ and $L = I - S = U\Lambda U^{\top}$.

In node-major notation, $SH = U(I - \Lambda)U^\top H$ and $(I - S)H = U\Lambda U^\top H$. These are first-order polynomial filters (Shuman et al., 2013; Defferrard et al., 2016). The random-walk operator $\tilde{A}$ is similar to S in degree-weighted coordinates; it is not itself generally symmetric. Appendix P states the interpretation and its limits.

We apply this decomposition only to selected source levels,

$$\mathcal{S}_{\text{spec}} = \{l_2, l_4, g\}, \tag{42}$$

These sources span intermediate and deep maps while keeping the number of response branches fixed. The source g denotes the feature map before global pooling, not a pooled object vector.

We pool each response and project it through an independent head:

$$f^{(b,s)} = \text{Pool}\left(X_b^{(s)}\right), \qquad z^{(b,s)} = q_{b,s}\left(f^{(b,s)}\right), \qquad (b,s) \in \{\text{smooth}, \text{dev}\} \times \mathcal{S}_{\text{spec}}. \tag{43}$$

Spectral Interpretation and Operator Assumptions. The two branches form complementary first-order graph filters. In the symmetric graph Fourier coordinates above, their responses are $1 - \lambda$ and λ: propagation supplies a neighborhood reference, and deviation gives greater relative gain to modes with larger graph variation. Their sum recovers the input before branch adaptation. For directed feature neighborhoods, the same construction is evaluated as PH and $(I - P)H$. We use this operator pair to expose different structural statistics to semantic supervision. The spectral assumptions and response-energy analysis are developed in Appendices P and U.

The response descriptor family is

$$\mathcal{F}_{\text{spectral}} = \{f^{(b,s)} : (b,s) \in \{\text{smooth}, \text{dev}\} \times \mathcal{S}_{\text{spec}}\}. \tag{44}$$

The response branches expose neighborhood-shared features and local deviations as distinct contrastive inputs. Whether these responses provide useful information beyond duplicated unfiltered features is assessed separately from their algebraic complementarity.

G.4.4 Adaptive Branch Fusion

Different branches may contribute unequally to the training objective. We aggregate their descriptors through learned branch scores over the full set:

$$\mathcal{F}_{\text{all}} = \mathcal{F}_{\text{spatial}} \cup \mathcal{F}_{\text{spectral}}. \tag{45}$$

The fusion module adapts each descriptor, computes branch scores, and forms a weighted sum:

$$u_f = \sum_{m \in \mathcal{F}_{\text{all}}} \alpha_m a_m(f^{(m)}), \qquad \alpha_m = \frac{\exp\left(g_m(a_m(f^{(m)}))\right)}{\sum_{n \in \mathcal{F}_{\text{all}}} \exp\left(g_n(a_n(f^{(n)}))\right)}, \qquad z_f = q_f(u_f). \tag{46}$$

Here a_m maps each descriptor to a common width, g_m predicts a scalar score, and the softmax normalizes scores across branches. The head q_f projects the weighted sum. Scores depend on the current object, allowing the mixture to vary across inputs.

G.4.5 Multi-Level Image Targets

Explicit correspondence in the compact implementation. For $\mathcal{S} = \{l_2, l_4, g\}$ and $\mathcal{B} = \{\text{smooth}, \text{dev}\}$, our compact variant uses $\mathcal{A}_{\text{R1}} = \{(s, \text{base}) : s \in \mathcal{S}\} \cup (\mathcal{S} \times \mathcal{B})$, containing nine branches. Its stage map is $r(l_2) = C3$, $r(l_4) = C5$, and $r(g) = C5$, with $\rho(s,b) = (r(s), b)$. Every branch has an independent 256-dimensional projection; the two C5 consumers do not share heads. The eleven-branch formulation additionally includes base sources l_1, l_3. We specify the active branch set separately for each variant. Exact dimensions, adaptors, gates, and reduction conventions are given in Appendix R.

ResNet stages C2–C5 correspond to `layer1`–`layer4`, with 256, 512, 1024, and 2048 channels. These are residual-stage endpoints, not literal convolutional layer indices. For a 224×224 image, their usual output strides are 4, 8, 16, and 32. The map ρ selects a target stage and response type for each point branch; sharing a raw stage does not require sharing a projection head.

Let a ResNet-style backbone produce stage-wise feature maps

$$\{F_m\}_{m \in \mathcal{M}}, \qquad \mathcal{M} = \{2, 3, 4, 5\}. \tag{47}$$

For each stage, we construct a base descriptor and an averaging–residual pair:

$$h_m^{\mathrm{img}} = a_m^{\mathrm{img}}(\mathrm{Pool}(F_m)), \qquad h_{b,m}^{\mathrm{img}} = a_{b,m}^{\mathrm{img}}(\mathrm{Pool}(\Gamma_b(F_m))), \quad b \in \{\mathrm{smooth}, \mathrm{dev}\}, \tag{48}$$

Here $\Gamma_{\mathrm{smooth}}(F_m) = \mathrm{Blur}(F_m)$ applies spatial averaging, and $\Gamma_{\mathrm{dev}}(F_m) = F_m - \mathrm{Blur}(F_m)$ forms its residual. Each a^{img} projects a pooled descriptor. Appendix R specifies our implementation's kernel and padding. The candidate image-target family is

$$\mathcal{H}_{\mathrm{img}}^{\mathrm{struct}} = \{h_m^{\mathrm{img}}\}_{m\in\mathcal{M}} \cup \{h_{b,m}^{\mathrm{img}} : b \in \{\mathrm{smooth}, \mathrm{dev}\}, m \in \mathcal{M}\}. \tag{49}$$

The map ρ selects the active targets from this family. Point and image responses are paired by their training roles; they need not share a Fourier basis or a spatial resolution.

Figure 5 makes the selected stage map explicit. The two C5 links use independent projection heads, allowing the deep and global point states to share an image source while retaining distinct objectives.

G.4.6 Transformer and Modern Image Encoders

The supervision interface requires spatially indexed intermediate features, not an EdgeConv operation. We instantiate MS-Point-T with point/patch tokens. At a selected block or stage d, let $H^{(d)} \in \mathbb{R}^{M_d \times C_d}$ contain the token features and let $c_i^{(d)} \in \mathbb{R}^3$ be token i's spatial center. We construct a geometric k-NN relation over these centers and normalize each receiving neighborhood:

$$S_{ij}^{(d)} = \frac{\mathbf{1}\{j \in \mathcal{N}_k(c_i^{(d)})\}}{|\mathcal{N}_k(c_i^{(d)})|}, \qquad H_{\mathrm{smooth}}^{(d)} = S^{(d)}H^{(d)}, \qquad H_{\mathrm{dev}}^{(d)} = (I - S^{(d)})H^{(d)}. \tag{50}$$

This graph is determined by token geometry and is independent of the attention weights. Spatial tokens enter the response branches; a global class token has no local support and is excluded from the graph. For a fixed-resolution Transformer, we reuse the centers across selected blocks. For a hierarchical Transformer, we use the centers associated with each stage after token pooling. The former supplies a hierarchy in feature depth, while the latter additionally supplies a hierarchy in spatial resolution. This interface is compatible with the token representations used by Point-BERT, Point-MAE, and Point-M2AE (Yu et al., 2022; Pang et al., 2022; Zhang et al., 2022).

We pool the base and response maps separately, project each descriptor to 256 dimensions, and apply the same branch targets, gates, fusion, and joint objective as in the point-graph setting. Each head is independent, so two branches can share an image stage without sharing the projected target. Constructing responses adds a neighborhood operation; it does not modify the backbone's attention mechanism or require a reconstruction decoder. The same-backbone comparison in Section 4.3 therefore tests the structured objective rather than attributing a backbone-capacity increase to the method.

For the image encoder, ConvNeXt stages or spatial ViT patch tokens can replace ResNet C2–C5. We select intermediate and deep features by semantic depth, preserve the patch grid for the image averaging–residual operation, and use independent 256-dimensional projection heads. Class tokens are not treated as an image grid. These choices retain the distinction between feature depth and spatial support instead of equating literal layer numbers across architectures. Appendix Y gives the token interface and local-alignment extension in detail.

G.4.7 Joint Training Objective

Let $z_m^{(1)}$ and $z_m^{(2)}$ denote the embeddings of point branch m under the two augmented views. Let $h_{\rho(m)}^{\mathrm{img}}$ be the corresponding image-side target determined by the branch matching map ρ. We use NT-Xent as the base contrastive loss, denoted by $\mathrm{NTXent}(\cdot,\cdot)$.

Structured Intra-Modal and Cross-Modal Consistency The first three terms act on the same branch family. Same-branch consistency encourages invariance between point augmentations:

$$\mathcal{L}_{\mathrm{same}} = \frac{1}{|\mathcal{A}|}\sum_{m\in\mathcal{A}} \mathrm{NTXent}\Big(z_m^{(1)}, z_m^{(2)}\Big), \tag{51}$$

where $\mathcal{A}$ contains all base and response branches. The loss applies point-view consistency to each branch separately.

The second term encourages agreement between selected levels:

$$\mathcal{L}_{\text{cross}} = \frac{1}{|\mathcal{P}(\mathcal{A})|} \sum_{(m,n)\in\mathcal{P}(\mathcal{A})} \sum_{a=1}^{2} \text{NTXent}\left(z_m^{(a)}, z_n^{(a)}\right), \tag{52}$$

where $\mathcal{P}(\mathcal{A})$ is a selected cross-branch pairing set.

The third term aligns each point branch with its image target:

$$\mathcal{L}_{\text{cmid-level}} = \frac{1}{|\mathcal{A}|} \sum_{m\in\mathcal{A}} \text{NTXent}\left(p_m,\, h^{\text{img}}_{\rho(m)}\right), \qquad p_m = \frac{1}{2}\left(z_m^{(1)} + z_m^{(2)}\right). \tag{53}$$

Fused Alignment and Response Diversity A separate term aligns the fused point representation with the fused image target:

$$\mathcal{L}_{\text{cmid-fused}} = \text{NTXent}\left(\frac{z_f^{(1)} + z_f^{(2)}}{2}, h_f^{\text{img}}\right). \tag{54}$$

We penalize squared cosine similarity between the projected smoothing and deviation embeddings at the same source:

$$\mathcal{L}_{\text{band}} = \frac{1}{|\mathcal{S}_{\text{spec}}|} \sum_{s\in\mathcal{S}_{\text{spec}}} \cos^2(z_{\text{smooth},s}, z_{\text{dev},s}). \tag{55}$$

Here $z_{\text{smooth},s}$ and $z_{\text{dev},s}$ are the projected embeddings of the two responses at source s. We use a finite squared-cosine penalty for *soft directional decoupling*: it discourages the two heads from using the same embedding direction while the contrastive terms retain their common object semantics. This separates their representational roles without imposing a fixed partition of the feature space. The penalty weight controls the trade-off between response diversity and semantic alignment. Consistent with this design, our diagnostic reaches its best transfer accuracy at an intermediate penalty, rather than at the smallest response cosine (Appendix L.2).

Overall Objective The final objective is compactly written as

$$\mathcal{L} = \sum_{r\in\mathcal{R}} \lambda_r \mathcal{L}_r, \qquad \mathcal{R} = \{\text{same}, \text{cross}, \text{cmid-level}, \text{cmid-fused}, \text{band}\}. \tag{56}$$

All five terms are optimized jointly on the shared encoder. Their roles define two coupled requirements: semantic agreement across observations and specialization across structural responses. Same-branch consistency regularizes sensitivity to the chosen augmentations; cross-level coordination links descriptors across depth. Branch alignment supplies direct image-guided gradients to each structural state, while fused alignment trains their collective readout. The response penalty counterbalances directional redundancy between the two response heads.

For shared parameters θ, the joint gradient is

$$\nabla_\theta \mathcal{L} = \sum_{r\in\mathcal{R}} \lambda_r g_r, \qquad g_r = \nabla_\theta \mathcal{L}_r, \tag{57}$$

so each weight controls the contribution of a task-specific gradient to a shared trajectory. We use $(1, 0.4, 1, 1, 0.05)$ in the order of Eq. 56: the two consistency terms coordinate the representation, the cross-modal terms anchor it to image semantics, and the smaller diversity term encourages specialization. The model is trained to satisfy these requirements together. Loss-removal and weight-transfer experiments examine the resulting trade-off; Appendix A develops the gradient interaction analysis, and Appendix R specifies the reduction conventions for reproduction.

Appendix A analyzes descriptor indistinguishability, the information exposed by pre-pooling responses, and the interaction among the five objectives.

H ALGORITHMIC DESCRIPTION

Algorithm 1 summarizes the full pretraining procedure. The procedure extracts intermediate point features, constructs smoothing and deviation responses at selected levels, and aligns the resulting branches with image targets. Low/high in the algorithm retains the response notation.

Algorithm 1 MS-Point pretraining.

Require: Point cloud P, rendered image I, point encoder E_{3D}, image encoder E_{img}, spectral levels $\mathcal{S}_{\text{spec}}$, loss weights $\{\lambda_r\}$.

1: Generate two augmented point views $P^{(1)}$ and $P^{(2)}$.

2: Encode each point view with the 3D backbone:

$$\{X^{(l,a)}\}_{l\in\mathcal{D}} = E_{3D}(P^{(a)}), \quad a \in \{1,2\}.$$

3: For each level $l \in \mathcal{D}$, compute object descriptors by max-average pooling:

$$f^{(l,a)} = [\text{MaxPool}(X^{(l,a)}), \text{AvgPool}(X^{(l,a)})].$$

4: For each selected spectral level $s \in \mathcal{S}_{\text{spec}}$, construct $\tilde{A}$ and compute

$$X^{(s,a)}_{\text{smooth}} = X^{(s,a)}\tilde{A}^\top, \qquad X^{(s,a)}_{\text{dev}} = X^{(s,a)}(I-\tilde{A})^\top.$$

5: Project all hierarchy and spectral descriptors with branch-specific projection heads.

6: Fuse hierarchy-spectrum descriptors using the adaptive fusion module to obtain $z_f^{(1)}$ and $z_f^{(2)}$.

7: Encode image I with the image backbone and construct hierarchical coarse/fine image descriptors.

8: Compute same-branch point-view consistency $\mathcal{L}_{\text{same}}$.

9: Compute cross-branch coordination $\mathcal{L}_{\text{cross}}$.

10: Compute branch-wise point–image alignment $\mathcal{L}_{\text{cmid}-\text{level}}$.

11: Compute fused point–image alignment $\mathcal{L}_{\text{cmid}-\text{fused}}$.

12: Compute low/high complementarity regularization $\mathcal{L}_{\text{band}}$.

13: Update the model by minimizing

$$\mathcal{L} = \sum_{r\in\mathcal{R}} \lambda_r \mathcal{L}_r.$$

Training behavior. The five terms act in the same training update. Same-branch consistency encourages agreement across point augmentations, and cross-level coordination encourages selected stages to remain compatible. Branch-level alignment supplies direct image targets; fused alignment trains their learned mixture. The response-diversity penalty discourages parallel projected responses. These roles describe the intended training pressures, rather than guarantees about the resulting representation.

I COMPLEXITY AND RESOURCE CONSUMPTION

I.1 THEORETICAL COMPLEXITY

Let N be the number of points, k the number of neighbors, C a representative feature dimension, $L = |\mathcal{A}|$ the number of hierarchy-spectrum branches, and $K = |\mathcal{P}(\mathcal{A})|$ the number of selected cross-branch pairs. The backbone complexity is dominated by DGCNN EdgeConv operations and dynamic graph construction. All base and response branches reuse features from each point-view forward pass, so they do not require an additional full encoder evaluation per branch.

The response propagation costs approximately $O(|\mathcal{S}_{\text{spec}}|NkC)$ once neighborhoods have been built. Graph construction, branch pooling, projections, and contrastive similarity matrices add separate costs. The branch-count expressions in Table 18 summarize this structure rather than a complete end-

to-end FLOP count. In particular, batch contrastive costs also depend on batch size and embedding width.

Table 18: **Simplified complexity by method family.** Backbone passes are counted per point view. The table does not include all costs of graph construction or batch contrastive similarities.

Method Family	Passes	Main Extra Cost	Scaling Behavior
Global-only 2D–3D alignment	1	Global projection + contrastive loss	$O(\text{Backbone})$
Input-space multi-scale encoding	$R > 1$	Repeated point encoding	$O(R \cdot \text{Backbone})$
Hierarchical-only supervision	1	Multi-level projection + losses	$O(\text{Backbone} + LCd)$
MS-POINT	1	Spectral filtering + branch losses	$O(\text{Backbone} + \lvert\mathcal{S}_{\text{spec}}\rvert NkC + LCd)$

The branches reuse intermediate states from each point-view forward pass. With two point views, this still entails two encoder evaluations per update; it avoids an additional full encoder evaluation for every branch. The gain in supervision must be weighed against the cost of graph construction, heads, and losses.

I.2 EMPIRICAL RESOURCE CONSUMPTION

Table 19 preserves the analytical resource estimates for 1,024 points, DGCNN, ResNet-50, batch size 64, and mixed precision. These are illustrative estimates, not a matched-hardware profiling result. Parameter scope, device count, and timing boundaries must be fixed before they are compared with later implementations.

Table 19: **Analytical resource estimates.** Values are retained for the stated DGCNN–ResNet-50 setting. They are not a matched-hardware profiling measurement.

Method	Params (M)	Memory (GB)	Time/ epoch	Pretrain time	Scan Acc.
Global-only CrossPoint-style baseline	13.8	8.1	11.2 min	18.7 h	81.7
Hierarchical only	15.2	8.9	12.6 min	21.0 h	86.5
Hierarchical + spectral branches	16.1	9.8	14.1 min	23.5 h	89.6
MS-POINT full	16.4	10.2	14.6 min	24.3 h	**89.6**

In the estimated table, the full configuration increases accuracy by 7.9 percentage points over the listed global baseline while increasing estimated pretraining time from 18.7 to 24.3 hours. This arithmetic describes the table; it does not establish measured throughput or a general efficiency advantage.

I.3 ACCURACY–EFFICIENCY TRADE-OFF

We combine accuracy, memory, and epoch time into the following descriptive score:

$$\text{EffScore} = \frac{\text{Acc.}}{\text{GPU Mem.} \times \text{Time/Epoch}}, \tag{58}$$

with accuracy, memory, and time expressed in the units used in the table. This ratio does not account for training duration, deployment cost, or the relative value of accuracy and resources.

Table 20: **Resource-normalized score.** The calculation is retained for completeness; accuracy, memory, and time should also be assessed separately.

Method	Acc.	GPU Mem.	Time / Epoch	EffScore
Global-only baseline	81.7	8.1	11.2	0.900
Hierarchical only	86.5	8.9	12.6	0.771
Spectral only variant	88.1	9.4	13.8	0.679
MS-POINT full	**89.6**	10.2	14.6	0.602

Table 21: **Ablation on cross-level pairing strategy.** Adjacent pairing gives the highest accuracy among the reported settings.

Pairing	λ_{cross}	Acc.
None	0.0	84.5
Adjacent	0.4	**85.8**
All pairs	0.4	85.2
All pairs	0.8	84.6

Table 22: **Ablation on image-side supervision structure.** The comparison varies the active image targets within the reported configuration.

Design	Stages	Pair	Acc.
Global	×	×	87.1
Stages	✓	×	88.4
Stages + responses	✓	✓	**89.6**

Table 23: **Ablation on fusion strategy.** The adaptive variant has the highest reported accuracy; these rows do not isolate the effect of sample-dependent weights.

Fusion	Learned	Acc.
No fusion branch	×	85.8
Uniform average	×	86.0
Concatenation + MLP	✓	86.2
Adaptive fusion	✓	**86.5**

Table 24: **Ablation on spectral source levels.** The selected intermediate and deep sources give the highest reported accuracy, with more active response branches.

Sources	Branches	Acc.
None	0	86.5
$\{l_2\}$	2	87.4
$\{l_4\}$	2	87.2
$\{g\}$	2	87.0
$\{l_2, l_4\}$	4	88.3
$\{l_2, l_4, g\}$	6	**89.6**

The score decreases from 0.900 to 0.602 in the retained table. Accuracy divided by memory and epoch time is sensitive to units and does not define a general utility measure. We therefore treat it as a descriptive calculation and prefer separate accuracy, memory, and time reporting for future controlled comparisons.

J Additional Ablation Studies

J.1 Effect of Cross-Level Pairing Strategy

The cross-level loss depends on the selected branch pairs. Adjacent pairing connects neighboring levels and related responses; all-pairs pairing adds constraints across the full family. Adjacent pairing gives the highest reported accuracy in Table 21. This comparison favors the sparser pairing under the reported setting, but does not identify oversmoothing as the cause of the all-pairs result.

J.2 Effect of Structured Image-Side Supervision

We compare a global image target, multi-stage targets, and multi-stage targets with averaging and residual responses. This tests how image-side target construction affects the full system.

The reported accuracy rises with multi-stage targets and again with image responses. These observations are consistent with the target-design hypothesis. A matched active-head control is still needed to distinguish the target construction from changes in capacity or the number of losses.

J.3 Effect of Fusion Strategy

We compare averaging, concatenation, and adaptive branch fusion. Adaptive fusion gives the highest reported accuracy in Table 23. A learned but input-independent gate would provide a closer control for whether the benefit specifically depends on changing weights across samples.

J.4 Effect of Response Source Levels

Table 24 compares where responses are constructed. The source set $\{l_2, l_4, g\}$ gives the highest reported accuracy among these choices. It samples intermediate and deep maps, but also changes the number of active response branches relative to smaller source sets. Equal-count source comparisons would isolate depth selection more directly.

Table 25: **Sensitivity to λ_{cross}.** Moderate cross-level coordination performs best.

λ_{cross}	0.0	0.2	0.4	0.6	0.8
Acc.	88.2	88.9	**89.6**	89.1	88.4

Table 26: **Sensitivity to λ_{band}.** The intermediate weight gives the highest reported accuracy; the penalty acts on embedding directions.

λ_{band}	0.00	0.01	0.05	0.10	0.50
Acc.	88.2	89.0	**89.6**	89.2	88.5

K Hyperparameter Sensitivity

K.1 Sensitivity to Cross-Level Coordination Weight

The coefficient λ_{cross} weights agreement across selected branches relative to the other objectives. The following sweep varies this coefficient within the reported configuration.

The highest reported accuracy occurs at $\lambda_{\text{cross}} = 0.4$. Accuracy alone does not show whether other weights make the branch representations more homogeneous; that interpretation requires a separate representation diagnostic.

K.2 Sensitivity to Response-Diversity Weight

The coefficient λ_{band} controls the squared-cosine penalty on projected response pairs. Table 26 reports the highest accuracy at 0.05. The lower accuracy at larger weights suggests that stronger directional separation is not always helpful; it does not establish a change in statistical independence or semantic structure.

K.3 Sensitivity to Contrastive Temperature

The NT-Xent temperature controls the sharpness of contrastive similarity distributions. A very small temperature may overemphasize hard negatives, while a very large temperature may weaken discriminability.

Table 27: **Sensitivity to contrastive temperature τ.** The default value $\tau = 0.10$ gives the highest accuracy in this sweep.

τ	0.03	0.05	0.10	0.20	0.30
Acc.	88.1	88.8	**89.6**	89.0	88.3

The reported values vary modestly over $\tau \in [0.05, 0.20]$. This is a local sensitivity result for one configuration, rather than evidence that the same temperature range transfers across backbones or data scales.

L Qualitative and Diagnostic Analyses

L.1 Branch Utilization

The fusion module predicts a weight for each branch and object. Table 28 summarizes these weights over the validation set. Aggregate weights describe the learned mixture; they do not show how it changes with object category or geometric complexity.

The average weights are distributed across the listed branch groups. Their magnitudes should be read together with adaptor-output norms, because a large gate alone does not establish a large contribution to the fused vector. Per-object weights and controlled branch interventions would give more direct evidence of utilization.

L.2 Projected Response Similarity

We measure average cosine similarity between projected smoothing and deviation embeddings. The table reports lower similarity as $\mathcal{L}_{\text{band}}$ receives more weight, with the highest accuracy at an intermediate setting. The diagnostic describes embedding directions, not a decomposition of information into independent components.

Table 28: Mean fusion weights over the validation set, grouped by depth and response.

Group	Weight
Shallow	0.18
Middle	0.21
Deep/global	0.24
Smoothing	0.20
Deviation	0.17

Table 29: Soft directional regularization and transfer. Mean cosine, squared cosine, and ScanObjectNN accuracy (%).

Penalty	cos	$\cos^2$	Acc.
None	0.61	0.37	88.2
Weak	0.42	0.18	89.0
Default	0.31	0.10	**89.6**
Strong	0.08	0.01	88.5

Default regularization attains 89.6% accuracy at a mean cosine of 0.31; stronger regularization reduces cosine to 0.08 but lowers accuracy to 88.5%. The result identifies a useful balance between specialization and shared semantic alignment. The active-response controls further test whether the structured inputs improve learning beyond the projection heads themselves.

L.3 Image-Occlusion Sensitivity

Figure 15 intervenes on one image region at a time while holding the point input fixed. Reading each response map together with its original pair similarity distinguishes spatial sensitivity from an already weak alignment score. The local-matching examples vary across training replicates: a nearly white response map alone therefore does not establish occlusion robustness. These cases make the partial-visibility problem concrete and motivate jointly reporting local activity, alignment quality, and perturbation response.

L.4 Representation Diagnostics: Protocols and Interpretation

We examine how the supervision interface is expressed inside the trained model. Figure 6 shows feature-norm ranks for four ShapeNet objects from the local-matching checkpoint at seed 6201. The spatial pattern changes across stages in both modalities, supporting stage-specific inspection rather than reading all structure from the final descriptor. Within-stage percentile normalization makes locations comparable in a map, but does not compare absolute activation magnitudes between layers.

For the remaining diagnostics, we use the same 52 in-corpus objects, with four objects from each of 13 classes and 1,024 points per object. Seeds 6201 and 6203 provide validation-selected checkpoints of configurations with and without pooled local matching. These archived diagnostic configurations use nine branches and learned fusion in both modalities; their original pretraining uses 2,048 points. They complement the downstream evaluations with measurements of internal organization.

Figure 7 examines non-self edges. From l_4 to the pre-pooling map, reciprocity increases and in-degree variation decreases in both configurations and runs. Nevertheless, reciprocal edges remain well below 100%. The learned neighborhoods therefore retain directionality, which motivates the graph-shift formulation and the separate symmetric reference analysis in Section 3.2.

Local supervision also warrants an activity check. Figure 8 measures entire zero-valued token vectors in the pooled-set extension. Mean zero-token rates are 5.41% and 13.55% on the point side, and 5.47% and 46.84% on the image side, for the two runs. A zero token supplies no discriminative dot-product score. The result identifies a concrete diagnostic for local objectives: adding more targets is useful only when their descriptors remain informative. Point-bin indices are not geometric patch coordinates, and image-grid activity alone does not identify semantic foreground.

Finally, Figure 9 compares branch weights across the two runs. With local matching, point-side median Jensen–Shannon divergence falls from 0.03074 to 0.01380 nats and top-branch agreement rises from 40.4% to 61.5%. Image-side divergence instead increases from 0.05822 to 0.09616, with agreement declining from 38.5% to 30.8%. The modality-specific pattern argues for inspecting both pathways. Gate consistency describes the learned mixture, while transfer quality is established by the downstream evaluations. Appendix L adds occlusion diagnostics, and Appendix W details the pooled-set experiments.

M LIMITATIONS

First, the responses are complementary first-order filters, not disjoint Fourier projectors. Symmetric spectral analysis requires the stated graph assumptions; directed feature neighborhoods provide a more limited graph-shift interpretation. A deviation can reflect a useful boundary, an outlier, or an unstable neighborhood. The small differences among learnable filters require uncertainty estimates before a general ranking is justified.

Second, the object-level objective pools its branches and does not define point–pixel correspondence. The completed local experiments broaden dense evaluation, but do not make the object-level loss visibility-aware. Learned region associations need explicit support and unmatched handling. The archived index-pooled set extension remains a separate experiment with mixed long-run classification outcomes.

Third, structured targets add cost and configuration choices. The completed compute controls address capacity and reported FLOPs in one setting, while the five-epoch paired pilot remains inconclusive. Neither establishes that every branch, loss, or source is necessary across all encoders. Point-only and image-guided methods also differ in information access, which must be considered alongside computational cost.

Fourth, our evaluation covers specific backbone, dataset, and view configurations. The low-label scene experiments instantiate the objective on a scene encoder; they do not test direct transfer of DGCNN weights. We analyze the DGCNN benchmark, local-alignment study, and five-seed study within their respective protocols. Wider evaluation across sensors, graph choices, and label-sampling strategies is needed to establish how broadly the improvements transfer.

N BROADER IMPACTS

Reusable point encoders may reduce annotation requirements for 3D perception, particularly when dense labels are expensive. Potential uses include robotics, indoor mapping, inspection, and geometric analysis. The benchmark results do not establish benefits in these applications; deployment would require evaluation under the intended sensors and operating conditions.

The same perception capabilities can support surveillance or tracking. This study does not introduce a deployment system or collect new human-subject data. Applications involving human environments should account for privacy and consent. Public benchmarks also reflect particular object categories, viewpoints, and scene layouts, so their coverage may limit transfer. Future evaluations should document these acquisition conditions and test performance across them.

O DATASET LICENSES AND TERMS OF USE

We use publicly available datasets that are standard in the point cloud learning literature. Table 30 summarizes their sources and license or usage information. When official dataset terms are more restrictive than redistribution licenses from third-party mirrors, we follow the official terms.

All datasets are used only for academic research and evaluation. We do not redistribute the original datasets. Users who wish to reproduce the experiments should download the datasets from their official sources and comply with the corresponding terms of use.

P COMPLEMENTARY GRAPH FILTERING AND SOFT RESPONSE DECOUPLING

Symmetric construction. Let $W = W^\top \geq 0$, $\widetilde{W} = W + I$, $D_{ii} = \sum_j \widetilde{W}_{ij}$, and $S = D^{-1/2}\widetilde{W}D^{-1/2}$. The normalized Laplacian $L = I - S$ is symmetric, so $L = U\Lambda U^\top$ with an orthonormal basis U. For a node-major feature matrix H, the complementary responses are

$$H_{\mathrm{s}} = SH = U(I - \Lambda)U^\top H, \tag{59}$$

$$H_{\mathrm{d}} = (I - S)H = U\Lambda U^\top H. \tag{60}$$

Table 30: **Dataset license and usage information.** For datasets with unclear or mirror-dependent licensing, we follow the original official dataset terms and use the data only for research purposes.

Dataset	Source / Owner	License or Terms	Usage in This Work
ShapeNet (Chang et al., 2015)	Princeton / Stanford	Non-commercial research and educational use under official ShapeNet terms	Pretraining
ModelNet40 / ModelNet10 (Wu et al., 2015)	Princeton ModelNet	Public research benchmark; follow official ModelNet terms	Linear classification
ScanObjectNN (Uy et al., 2019)	HKUST VGD	Repository released under MIT License; dataset used under project terms	Robust classification / ablation
ShapeNetPart (Yi et al., 2016)	ShapeNet annotation benchmark	Follow ShapeNet / ShapeNetPart research-use terms	Part segmentation
S3DIS (Armeni et al., 2016)	Stanford 2D-3D-S / CVGL	Follow official Stanford dataset terms; some mirrors provide permissive metadata licenses	Indoor semantic segmentation

These identities provide a polynomial-filter interpretation without requiring eigendecomposition during training (Shuman et al., 2013; Defferrard et al., 2016). Substitution into $S = I - L$ gives the response pair directly. The graph variation of a channel h is

$$h^\top Lh = \frac{1}{2}\sum_{i,j}\widetilde{W}_{ij}\left(\frac{h_i}{\sqrt{D_{ii}}} - \frac{h_j}{\sqrt{D_{jj}}}\right)^2 = \sum_i \lambda_i |(U^\top h)_i|^2. \tag{61}$$

The deviation response weights each graph mode by λ_i, linking local feature contrast to graph variation. Propagation has signed response $1 - \lambda_i$. On the normalized-Laplacian domain $[0, 2]$, its magnitude decreases to zero at one and then rises again. These are soft polynomial responses rather than ideal frequency cutoffs; their learned usefulness is determined by the cross-modal targets. Appendix U derives a lazy variant with a monotone nonnegative propagation response.

Response overlap and soft directional decoupling. The identity $H_\mathrm{s} + H_\mathrm{d} = H$ holds before adaptation, pooling, and projection. In general,

$$\langle H_\mathrm{s}, H_\mathrm{d}\rangle_F = \sum_i (1-\lambda_i)\lambda_i \|\widehat{H}_{i,:}\|_2^2 \neq 0, \qquad \widehat{H} = U^\top H. \tag{62}$$

The overlap term quantifies how the two responses share graph modes. This is compatible with our objective: both branches should retain object identity while contributing different structural statistics. We regulate their projected directions with a finite penalty rather than enforcing an algebraic partition of the raw signal. For nonzero embeddings a, b, the squared-cosine term has gradient

$$u = \frac{a}{\|a\|_2}, \quad v = \frac{b}{\|b\|_2}, \quad c = u^\top v, \qquad \nabla_a c^2 = \frac{2c}{\|a\|_2}(v - cu), \quad u^\top \nabla_a c^2 = 0. \tag{63}$$

The gradient is tangent to the normalized embedding sphere: it changes relative direction to first order. Combined with contrastive alignment, it encourages specialization while the visual targets maintain shared semantics. The weight determines the strength of this competition. Our diagnostic in Appendix L.2 finds the best transfer at an intermediate weight, supporting soft regularization rather than minimizing cosine in isolation. This is an angular objective; statistical independence would require a separate information-theoretic criterion.

Random-walk and directed operators. For the undirected construction, $P = D^{-1}\widetilde{W} = D^{-1/2}SD^{1/2}$. Therefore

$$D^{1/2}PH = S(D^{1/2}H). \tag{64}$$

The degree-weighted coordinate change provides the symmetric spectral interpretation for an undirected random walk. Under the channel-first convention $X = H^\top$, propagation is $XP^\top$. Our directed feature-space k-NN implementation uses the more general graph-shift formulation: PH and $(I - P)H$ are complementary first-order polynomials of the same shift. This construction requires a neighborhood operator, whereas the orthonormal Fourier analysis additionally requires the symmetric assumptions above.

What the objective can observe. Filtering before pooling changes which neighborhood statistics reach the loss. A permutation of node features preserves channel-wise maximum and mean pooling, yet can alter the pooled graph responses when the graph is held fixed. This makes structural organization observable through an additional readout. The duplicated-feature, degree-matched random-graph, and feature-shuffle controls test its value for learning. Coordinate-space graphs tie this organization to input geometry; feature-space graphs let it evolve with the representation.

Q Exploratory Controls and Extensions

Q.1 Parameter-Matched Short Runs

The following analysis uses archived implementation-level runs, not a new full reproduction of the eleven-branch manuscript configuration. Both variants have 73,367,890 trainable parameters and use the same warm-start setting. Ten paired seeds (3401–3410) were run for five epochs with at most 2,048 training objects and batch size four. The endpoint is the terminal ModelNet40 SVM accuracy. The duplicated-feature control replaces the response inputs with two copies of the unfiltered features while retaining the heads and losses.

Table 31: Parameter-matched exploratory runs. Runtime is the mean recorded short-run duration. These values are not full-pretraining benchmark results.

Variant	Accuracy (%)	Runtime (s)	Peak memory (GiB)
Duplicated-feature heads	91.4222	442.78	10.026
Smoothing/deviation	91.5559	551.73	10.028

For paired differences $d_i = a_i^{\text{response}} - a_i^{\text{control}}$, the mean difference is 0.1337 percentage points. The paired Student-t interval,

$$\overline{d} \pm t_{0.975,9} \frac{s_d}{\sqrt{10}}, \tag{65}$$

is $[-0.1334, 0.4008]$ percentage points. The two-sided paired test gives $p = 0.2867$, and an exact sign-flip test gives $p = 0.3047$. Six of ten paired differences are positive. Thus, the pilot controls parameter count but does not establish a reliable improvement. Its mean runtime rises by 24.6%; it is not compute-matched. A full experiment should preserve the graph-building and aggregation path in the control and report multiple independent pretraining seeds.

Q.2 Response and View Pilots

The response pilot uses two seeds (5101 and 5102), five warm-start epochs, at most 2,048 objects, and batch size 32. Table 32 reports the best validation accuracy from each run, averaged across seeds. This endpoint differs from the terminal SVM endpoint above. With two seeds, the numerical ordering does not justify a claim that one response family is consistently better.

The view pilot uses seeds 4401–4403, five warm-start epochs, at most 1,024 objects, and batch size two. Its validation results show increased runtime without a mean accuracy gain from additional encoded views (Table 33). This finding is specific to the pilot budget. Candidate renderings stored on disk and images encoded per update are separate quantities.

The response parameterizations and token/local extensions are specified in Section D. Appendix R distinguishes the saved implementation from the main formulation, and Appendix V gives the individual paired outcomes.

Table 32: Two-seed response pilot. Parameter counts refer to the archived implementation, not the analytical resource estimates in Table 19.

Response	Val. (%)	Parameters	Min.
Fixed pair	90.930	73,367,890	7.36
Learnable	91.082	73,367,896	7.63
Polynomial I, P, P^2	91.184	73,367,908	8.58
Three-way	90.955	87,574,488	8.72

Table 33: Three-seed encoded-view pilot. The endpoint and training budget are specific to this experiment.

Views	Val. (%)	Sec.
1	91.277	425.12
2	91.277	469.06
3	90.938	525.29

R Implementation Variants and Module Specifications

Table 34: Notation for the supervision interface. The branch set changes with the explicitly named implementation variant.

Symbol	Meaning
$X^{(s)} \in \mathbb{R}^{C_s \times N_s}$	Channel-first stage map; $H^{(s)} = (X^{(s)})^\top$ is node-major.
$\mathcal{D}$, $\mathcal{S}_{\text{spec}}$	Base stages and the selected response-source subset.
$\mathcal{B} = \{\text{smooth}, \text{dev}\}$	Neighborhood prediction and deviation response identities.
$\mathcal{A}$, ρ	Active branch set and its image-target map.
$z_m^{(v)}, h_m^{(u)}$	Point/image branch embeddings under views v/u.
z_f, h_f	Fused point and image embeddings.

Figures 12 and 13 separate branch computation from downstream use. The former expands the five-base/six-response construction; the latter shows how pooled and unpooled point features serve different prediction heads.

R.1 The Eleven-Branch Model and the Saved Nine-Branch Variant

The main configuration uses five base stages and six response branches. The archived exploratory implementation instead uses three base stages, $\{l_2, l_4, g\}$, plus their six responses. We denote it by R1 solely to keep its implementation scope explicit. The corresponding branch count changes from eleven to nine, and the branch-wise loss averages and fusion inputs change with it. Its results describe the nine-branch variant and remain separate from the eleven-branch results.

Table 35: Version-specific definitions. A field requiring recovery is left explicit rather than inferred from another variant.

Field	Main eleven-branch formulation	Saved R1 implementation
Base sources	l_1, l_2, l_3, l_4, g	l_2, l_4, g
Response sources	l_2, l_4, g	l_2, l_4, g
Response count	6	6
Total input branches	11	9
Graph description	Symmetrized adjacency and row normalization	Directed feature-space k-NN mean
Image correspondence	Full map, including l_1, l_3, requires run/configuration recovery	C3/C5/C5 for $l_2/l_4/g$
Cross-level reduction	Mean over selected pairs, sum over two views	Mean over pair–view terms
Deployment readout	Must accompany each result	Evaluator-specific; not inferred from fusion

The code constructs point neighborhoods from detached features. Gradients pass through gathered feature values, but not through discrete neighbor selection. The class default neighborhood size is 20; a default of 16 described in a supplementary sensitivity table cannot be substituted for a run's actual argument. Self-neighbor inclusion depends on the selection implementation and tie handling. It should be recorded separately from explicit self-loop insertion, which is absent in this code path.

R.2 Branch Correspondence and Dimensions in R1

The map is $r(l_2) = C3$, $r(l_4) = C5$, and $r(g) = C5$. Each base branch uses its base image target, while its smoothing/deviation branches use the corresponding image response. The two C5 consumers have independent projection heads. The saved image-fusion module fuses the projected branch family; in this variant, merely extracting C2 and C4 does not make them active branch targets.

Each projection uses $h = \max(256, \min(d_{\text{in}}, 1024))$ as hidden width and outputs 256 dimensions. Thus, the image-side C5 head consumes 4096 pooled dimensions, not the 2048 channels of the unpooled feature map. This distinction is important when checking parameter counts.

The image filter uses stride one, padding one, and the framework's default handling of padded zeros. A constant image map can therefore have a nonzero residual at its boundary. A proposed mechanism

Table 36: Verified R1 source and projection dimensions. Both modalities concatenate max and mean pooling. These dimensions do not fill the unspecified l_1/l_3 heads of the main variant.

Point source	Channels	Pooled dim.	Image stage	Image channels	Pooled dim.
l_2	64	128	C3	512	1024
l_4	256	512	C5	2048	4096
g_{map}	1024	2048	C5	2048	4096

Table 37: Verified trainable paths in R1. The point-response adaptor has no image-side convolutional counterpart in this saved implementation.

Module	Structure	Sharing and normalization
Point response adaptor	Conv1d$(C, C, 1)$, BN, LeakyReLU(0.2)	Independent per source/response; convolution has no bias
Projection head	Linear(d, h), BN, ReLU, Linear$(h, 256)$	Both linear biases disabled; independent heads
Branch gate	Linear$(256, 1)$	Bias enabled; softmax across branches
Fusion head	Linear$(256, 256)$, BN, ReLU, Linear$(256, 256)$	Biases disabled; separate point/image modules
Image response	3×3 average pooling and residual	No learned filter; padding convention affects border responses
Contrastive input	ℓ_2 normalization	Applied inside the contrastive criterion

visualization should distinguish this padding effect from learned fine structure. Changing padding or excluding padded pixels would be a new filter setting requiring its own validation.

R.3 Exact Contrastive and Reduction Conventions

The saved symmetric criterion concatenates two batches into $2B$ normalized embeddings z_i. For positive partner $p(i)$, it computes

$$\ell = -\frac{1}{2B} \sum_{i=1}^{2B} \log \frac{\exp(z_i^\top z_{p(i)}/\tau)}{\sum_{j \neq i} \exp(z_i^\top z_j/\tau)}. \tag{66}$$

The self logit is excluded; the positive remains in the denominator. This is the criterion used in the archived pilot. Variants with a memory bank, distributed negatives, or a modified similarity rule require separate specifications.

In R1, same-branch terms are averaged over branches and comparisons between the first point view and each other view. Cross-level terms use the two base pairs (l_2, l_4) and (l_4, g), averaged over pairs and point views. Point and image prototypes are arithmetic means across their views before normalization inside the contrastive criterion. Averaging individually normalized embeddings would produce a different objective.

The diversity term averages squared cosine over source levels, point views, and batch elements. It does not apply an additional image-side diversity loss in the inspected function. For two views and the same pair set, the main manuscript's cross-level sum is twice the R1 mean. Keeping a coefficient of 0.4 while changing the reduction therefore changes the effective regularization. A faithful comparison must either retain the reduction or explicitly rescale the coefficient.

R.4 Implementation-Level Checks

The complementarity identity should be checked on raw responses before adaptors. Shape assertions should distinguish node-major mathematics from channel-first tensors. A graph-permutation test should jointly permute graph and signal only when testing equivariance; it must not be reused as the disruption control. Parameter counting should deduplicate shared tensors and report trainable, total, and active parameters separately. Captured stage maps, target heads, and fusion inputs should be checked by whether they influence a loss, not by whether a tensor appears in the forward function.

S Supplementary Experiments and Analysis

We examine hierarchy, response construction, image-view budget, local alignment, and backbone adaptation through the following experiments. We keep the full-training comparisons separate from the shorter pilot runs because their budgets and evaluation endpoints differ. Appendix Z provides additional task results and resource details.

We distinguish benchmark results, short-run pilots, and the following controlled studies. MN40 and Scan denote reported ModelNet40 and ScanObjectNN accuracy in percent. Part and S3DIS denote reported mIoU in percent, with class or instance averaging shown where supplied. A Part column without an averaging label remains underspecified. Hours and parameter counts retain the reporting scope of the supplied table. Within-table contrasts below are arithmetic descriptions, not significance claims.

S.1 Hierarchy, responses, and fusion

The four unfused rows give the factorial contrast analyzed in Section C. The full row adds fusion and should not replace the hierarchy-plus-responses cell when estimating the interaction. Each row has a different parameter count, so a capacity-matched factorial experiment is still needed.

Table 38: Hierarchy, responses, and fusion.

Configuration	Hierarchy	Responses	Params.	Hours	MN40	Scan
Global	–	–	71.84M	24.6	92.8	87.9
Hierarchy	✓	–	72.45M	26.5	93.1	88.7
Responses	–	✓	72.76M	27.8	93.2	88.9
Hierarchy + responses	✓	✓	73.12M	29.3	93.4	89.2
Full + fusion	✓	✓	73.37M	30.5	93.6	89.6

S.2 Neighborhood and signal controls

We hold the branch architecture and computational path fixed while changing the relation between the graph and its feature signal. A feature shuffle must disrupt the graph–signal association rather than jointly relabeling both. The one-active-response rows change the number of active pathways and should be interpreted separately from the duplicated-feature comparison.

Table 39: Neighborhood and signal controls.

Construction	Same graph cost	Same branches	MN40	Scan
Duplicate	✓	✓	92.9	88.2
Random graph	✓	✓	93.0	88.3
Shuffled features	✓	✓	93.1	88.5
Smooth only	✓	One active	93.2	88.8
Deviation only	✓	One active	93.3	89.0
Pair	✓	✓	93.6	89.6

S.3 Progressive structural supervision

The hierarchy-only ModelNet40 value here is 93.2, whereas the factorial table reports 93.1. These observations cannot be silently combined as one run. Their configuration, seed, and checkpoint identities must be reconciled before estimating effects across the two tables.

S.4 Separate response pathways

The reported smooth-only and deviation-only rows differ from the neighborhood-control table. This may reflect different active hierarchy branches or training settings, but the numerical table alone cannot resolve the cause. The pair is numerically strongest within this report; statistical reliability remains unknown.

Table 40: Progressive structural supervision.

Supervision	MN40	Scan
Global	92.8	87.9
Hierarchy	93.2	88.7
Hierarchy + smooth	93.3	88.9
Hierarchy + deviation	93.2	88.8
Both	93.4	89.2
Fusion full	93.6	89.6

Table 41: Loss deletion.

Training objective	MN40	Scan
Global	92.8	87.9
Without $\mathcal{L}_{\text{same}}$	93.2	88.9
Without $\mathcal{L}_{\text{cross}}$	93.1	88.7
Without $\mathcal{L}_{\text{cmid}-\text{level}}$	92.9	88.3
Without $\mathcal{L}_{\text{cmid}-\text{fused}}$	93.2	88.9
Without $\mathcal{L}_{\text{band}}$	93.3	89.1
Full	93.6	89.6

Table 42: Separate response pathways.

Configuration	Smooth	Deviation	MN40	Scan	Part
Global	–	–	92.8	87.9	85.45
Smooth only	✓	–	93.1	88.6	85.7
Dev only	–	✓	93.2	88.9	85.8
Pair	✓	✓	93.6	89.6	86.1

Table 43: Full-training response variants.

Response	MN40	Scan	Part	Params.	Hours
Fixed pair	93.6	89.2	86.1	73.37M	30.5
Learnable first order	93.7	89.8	86.2	73.38M	31.1
Second-order polynomial	93.7	89.7	86.2	73.39M	33.4
Three-response	93.5	89.4	86.0	87.57M	38.8

S.5 Loss deletion

Relative to the reported full row, deleting branch-level cross-modal alignment gives decreases of 0.7 and 1.3 percentage points. This is the largest deletion effect in this table. The comparison concerns the role of a term within the full system; it does not prove that the five-loss system is the minimal sufficient objective. Loss reductions and retained active heads need to be identical across rows.

S.6 Full-training response variants

The fixed-pair ScanObjectNN value is 89.2 here, versus 89.6 in several other reports. We retain the supplied value rather than harmonizing it by assumption. These full-training observations are separate from the archived two-seed response pilot. The learnable and second-order rows differ by only 0.1 point on ScanObjectNN, which requires repeated runs to interpret.

S.7 Encoded image-view budget

The reported one-to-two-view changes are +0.1 and +0.2 percentage points, accompanied by an 11% time increase. The third view adds reported time without a further mean gain. This suggests a cost-sensitive view study, but does not establish statistical equivalence or that one view is sufficient for every task. The short-run view pilot has a different budget and endpoint.

S.8 Global and local cross-modal targets

Our geometry-preserving configuration exceeds object-level alignment by 0.7 instance mIoU points on ShapeNetPart and 1.5 mIoU points on S3DIS, with an 18% time increase. These gains are computed within the local-alignment study. Its S3DIS baseline is 67.5; the DGCNN benchmark baseline of 59.8 and the repeated-run result of 68.20 belong to separate configurations and are not used to compute this gain.

Figure 14 plots these within-study differences against additional training time.

Table 44: Encoded image-view budget.

Images/update	MN40	Scan	Relative time
1	93.6	89.6	1.00
2	93.7	89.8	1.11
3	93.7	89.8	1.24

Table 45: Parameter-accounting breakdown.

Added component	Cumulative (M)	Increment (M)
Global	71.84	–
Adaptors	72.38	0.54
Branch heads	73.06	0.68
Gates + fusion	73.37	0.31

Table 46: Global and local cross-modal targets.

Alignment	MN40	Scan	Part class	Part inst.	S3DIS	Rel. time
Object-level	93.6	89.6	84.7	86.1	67.5	1.00
Pooled multi-token	93.7	89.8	84.9	86.3	67.9	1.01
Geometry-preserving	93.9	90.2	85.4	86.8	69.0	1.18

S.9 Transformer token relations

The center-neighborhood row is numerically strongest and fastest in the supplied report. This does not identify a generally optimal graph. The exact transformer, token count, graph construction, and initialization must be reported. Feature neighborhoods do not directly use attention matrices, but remain functions of features produced by the transformer.

Table 47: Transformer token relations.

Token relation	Uses attention	MN40	Scan	Part	Hours
Attention-affinity	Yes	93.8	89.7	86.4	31.6
Feature kNN	No	94.0	90.0	86.6	31.3
Radius	No	94.1	90.1	86.7	30.8
Center kNN	No	94.3	90.4	86.9	30.4

S.10 Transformer depth selection

Three selected levels and all candidate blocks have the same reported part mIoU, while the latter costs 3.4 additional hours. This motivates sparse source selection as a candidate design. A shared backbone definition, block indices, equal training budget, and repeated seeds are needed to assess it.

S.11 Transformer neighborhood sensitivity

The reported range is small around the selected neighborhood size. A degree-matched radius graph changes support in a density-dependent way and is not identical to fixed-degree nearest neighbors. Sensitivity should therefore include density and point-count perturbations rather than only a clean-data accuracy sweep.

S.12 Point-to-image branch mapping

The default C3/C5/C5 mapping exceeds the reversed map by 0.9 and 1.6 points in this report. Because the stage channels differ, independent head widths and parameter counts should also be checked. A monotone mapping is a reproducible design choice; it is not proof that every paired stage shares the same semantics.

Table 48: Transformer depth selection.

Sources	MN40	Scan	Part	Hours
Final	93.2	88.3	85.7	27.4
Intermediate + final	93.8	89.4	86.3	28.9
Three sparse	94.3	90.4	86.9	30.4
All blocks	94.2	90.3	86.9	33.8

Table 49: Transformer neighborhood sensitivity.

Graph	MN40	Scan	Part
$k = 8$	94.1	90.1	86.7
$k = 16$	94.3	90.4	86.9
$k = 24$	94.2	90.3	86.8
Radius matched degree	94.2	90.2	86.8

Table 50: Point-to-image branch mapping.

Map for l2/l4/g	Mapping	MN40	Scan	Part
C5/C5/C5	High-level	93.1	88.6	85.7
C3/C4/C5	Strict stage	93.4	89.1	85.9
C3/C5/C5	Default	93.6	89.6	86.1
C5/C3/C3	Reversed	92.7	88.0	85.3

Table 51: Shared projection dimension.

Width	Params. (M)	MN40	Scan	Part
128	72.62	93.3	89.1	85.8
256	73.37	93.6	89.6	86.1
512	75.10	93.6	89.5	86.1

Table 52: Default versus task-specific tuning.

Backbone / points	Weights	MN40	Scan
DGCNN / 1,024	Default	93.6	89.6
DGCNN / 1,024	Retuned	93.8	90.4
Transformer / 1,024	Default	94.3	90.4
Transformer / 1,024	Retuned	94.4	90.7
DGCNN / 2,048	Default	93.8	90.0
DGCNN / 2,048	Retuned	93.9	90.2

S.13 Shared projection dimension

The reported 512-dimensional heads do not improve ModelNet40 or part mIoU over width 256 and slightly reduce ScanObjectNN accuracy. This favors testing a compact shared space. It does not replace an exact capacity-matched control, because width changes both parameter count and optimization.

Figure 16 expresses the projection-width comparison as per-task gains over 128 dimensions.

S.14 Source, neighborhood, and diversity settings

This collection includes changes in source selection, neighborhood size, and one diversity coefficient. Each must be a separate one-factor intervention relative to the same default. The saved class definition uses a neighborhood default of 20, whereas this report labels 16 as the default; the launch configuration is needed to reconcile them. These rows do not cover all five loss weights.

Table 53: Source, neighborhood, and diversity settings.

Controlled change	MN40	Scan	Part
l4/g	93.2	88.9	85.7
l2/l4/g (default)	93.6	89.6	86.1
$k = 8$	93.4	89.3	86.0
$k = 16$	93.6	89.6	86.1
$k = 24$	93.5	89.5	86.1
$w_{\text{band}} = 0.02$	93.5	89.4	86.0
$w_{\text{band}} = 0.05$	93.6	89.6	86.1
$w_{\text{band}} = 0.10$	93.5	89.5	86.0

S.15 Default versus task-specific tuning

The largest reported tuning difference is 0.8 points on ScanObjectNN for DGCNN with 1,024 points. The comparison addresses the cost of reusing one setting, not a guarantee of universal hyperparameter stability. Search ranges, validation sets, and total search cost must accompany any tuned upper bound.

S.16 Parameter-accounting breakdown

The supplied increments sum to 1.53 million parameters, about 2.13% of the reported global baseline. This arithmetic does not verify the model inventories. A full count should distinguish point and image encoders, response modules, projection heads, gates, trainable teacher components, and the smaller deployed representation.

T Mechanism Diagnostics and Reproduction Protocol

T.1 A Measurement Plan for the Proposed Mechanism

The proposed diagnostics test whether neighborhood responses retain useful structure and whether that structure explains transfer. Accuracy and visualizations alone do not establish that smoothing represents shape or deviation represents boundaries. The measurements below use fixed checkpoints and held-out objects; their outcomes have not yet been established.

For a symmetric analysis graph with Laplacian L, define the normalized variation of a node-major feature map by

$$\mathcal{V}(H; L) = \frac{\operatorname{tr}(H^\top L H)}{\|H\|_F^2 + \epsilon}. \tag{67}$$

The normalization separates graph variation from a change in feature magnitude. If an explicitly directed operator is used for training, the analysis graph must be described as a separate construction. Replacing it by a symmetrized graph during analysis changes the reference against which variation is measured. It does not retrospectively make the training operator symmetric.

Table 54: Mechanism diagnostics and alternative explanations. These are proposed measurements, not completed experimental results.

Claim to examine	Measurement	Alternative to exclude
Distinct neighborhood responses	Variation, raw norms, retained energy, probe accuracy per branch	A larger norm or projection head explains the difference
Useful depth-specific targets	Equal-head final-only control; stage-wise probes	More heads alone improve optimization
Complementary information	Cross-branch prediction and conditional probes	Orthogonal heads encode the same information
Useful adaptive fusion	Gate entropy, contribution norms, retrained branch controls	Gate scale or branch norm drives apparent importance
Geometric deviations aid transfer	Boundary and corruption tests; graph perturbations	The response amplifies noise or sampling artifacts
Losses coordinate training	Gradient norms and angles at shared stages	One loss dominates after normalization

A point-level probe must specify its target: part boundaries, surface curvature, and sensor discontinuities measure different properties. Labels used to fit that probe do not become permissible inputs to pretraining. For each target, compare raw backbone features, response features before projection, and projected embeddings. A supervised probe that can extract a boundary signal establishes accessibility of that signal, not necessarily its use by the downstream task.

Noise and geometry should be varied separately. A deviation is large whenever a feature differs from its neighborhood prediction. It may encode a meaningful boundary, but also an outlier, a density transition, or a bad neighbor. Test coordinate noise, point dropout, occlusion, and density shifts as separate perturbations, with severities fixed before evaluation. A helpful branch should retain task-relevant variation without disproportionately amplifying nuisance variation. Feature-space graphs may change abruptly under perturbation, so graph-neighbor overlap is a useful companion measurement.

Fusion intervention requires care. Setting a gate to zero only at evaluation creates an input distribution the fusion head may not have seen. Report this as sensitivity to an intervention, not as the training-time contribution of the branch. Retraining a matched model with the branch replaced by an active control answers a different and stronger question. Both may be informative when labeled correctly.

T.2 Four-Cell Point/Image Response Control

The archived duplicated-feature pilot changes response construction in both modalities. A focused next experiment holds every trainable path fixed and crosses two factors: point responses off/on and image responses off/on. An off factor supplies duplicated base features through the same heads; an

on factor supplies the corresponding neighborhood or image responses. Candidate graphs and blur kernels should still be executed in controls if wall-clock matching is intended, while ensuring that unused computation does not accidentally contribute to the loss.

Table 55: Point/image response factorial design. All four cells retain independent branch heads, gates, fusion, and the same losses. Results are to be measured.

Cell	Point inputs	Image targets	Main contrast
00	Duplicated base	Duplicated base	Capacity reference
10	Smooth/deviation	Duplicated base	Point response effect
01	Duplicated base	Blur/residual	Image target effect
11	Smooth/deviation	Blur/residual	Joint response effect

A positive $Y_{11}-Y_{00}$ would show that the joint interface helps under this experiment. A positive $Y_{10}-Y_{00}$ would support a point-response effect with the image targets fixed. The interaction $Y_{11}-Y_{10}-Y_{01}+Y_{00}$ would measure whether the joint effect departs from additivity. These are distinct claims and require seed-wise estimates. This control complements, rather than replaces, the hierarchy-by-response factorial design.

T.3 Checkpoint and Seed Accounting

Each run should record its source revision, complete configuration, input-data manifest, random seed, initialization checkpoint, validation schedule, and selected checkpoint. A historical warm start changes the interpretation of a five-epoch run: it measures a local continuation from that initialization, not training efficiency from scratch. If several variants share one warm-start checkpoint, uncertainty across their continuation seeds does not measure uncertainty over independently pretrained starting representations.

The paired pilot uses a terminal SVM endpoint, whereas the response and view pilots use best observed validation accuracy. Selecting the best validation observation is appropriate for model selection only when final evaluation is separated from the selection data. Its numerical value should not be relabeled as an unbiased test estimate. A full reproduction should state how often each dataset split was accessed and which decisions were made after each access.

For few-shot classification, save sampled support/query episodes so competing models are evaluated on the same episodes. Report uncertainty over pretraining seeds separately from uncertainty over episodes conditional on one pretrained encoder. For segmentation, report class and instance averaging separately and identify the evaluation fold or split. An aggregate standard deviation cannot be reconstructed from rounded table means.

T.4 Minimal Release Artifacts

A reproducible record contains the configuration and evaluator needed to regenerate each metric, the training log needed to identify its checkpoint, and the code needed to reconstruct each branch. Full model weights are useful but do not replace those records. Costs should include the device model, device count, numerical precision, effective batch size, number of updates, candidate and encoded image counts, and inclusion or exclusion of rendering, feature caching, validation, and downstream evaluation.

A result manifest can use the tuple

$$(\text{run ID}, \text{code hash}, \text{config hash}, \text{checkpoint}, \text{split}, \text{readout}, \text{metric}, \text{value}). \tag{68}$$

The manuscript table is generated from this manifest after protocol grouping, rather than assembled by choosing the best number across incompatible runs. The table register accompanying this expanded draft preserves all supplied observations and records which ones still need that linkage.

U Graph-Shift Design and Spectral Properties

U.1 Complementarity Is Not an Energy Partition

For any linear shift P and a fixed signal H, let $H_s = PH$ and $H_d = (I - P)H$. Their sum is H, independently of symmetry or stochasticity. Their squared norms, however, satisfy

$$\|H\|_F^2 = \|H_s\|_F^2 + \|H_d\|_F^2 + 2\langle H_s, H_d\rangle_F. \tag{69}$$

The cross term does not generally vanish. For a symmetric shift $S = I - L$ with $L = U\Lambda U^\top$,

$$\langle SH, LH\rangle_F = \sum_j (1 - \lambda_j)\lambda_j \|\widehat{H}_j\|_2^2. \tag{70}$$

Both responses retain a mode for $\lambda_j \notin \{0, 1\}$. At $\lambda_j = 1/2$, they even give identical contributions for that mode. Thus, the identity of their sum neither separates all frequencies nor removes redundant information. Once separate nonlinear adaptors, pooling, and heads are applied, even the reconstruction identity is no longer a property of the resulting embeddings.

U.2 Signed Response Versus Magnitude Response

A symmetric normalized Laplacian has eigenvalues in $[0, 2]$. The signed response $1 - \lambda$ decreases on this interval, but its magnitude $|1 - \lambda|$ decreases only up to one and then increases. In particular, the endpoint $\lambda = 2$ has magnitude one. Adding self-loops does not by itself guarantee that every normalized-adjacency eigenvalue is nonnegative. Thus, neighborhood averaging does not guarantee suppression of every high-frequency mode.

The lazy family $S_\eta = I - \eta L$, $0 \le \eta \le 1/2$, gives a controlled spectral design: its propagation response is nonnegative and nonincreasing on $[0, 2]$, and its complementary deviation is ηLH. We use this family to separate a monotone spectral specification from the neighborhood-adaptive interface used in MS-Point. The directed feature-space implementation preserves each node's selected neighbors and updates their relation as features evolve. Its design invariant is the complementary pair of first-order graph-shift polynomials. Thus, symmetric spectral analysis specifies the frequency response under a chosen graph model, while directed propagation instantiates the same supervision interface on adaptive neighborhoods. Their relative robustness is an empirical question addressed through graph and filter controls.

U.3 What Stochastic Averaging Guarantees

If $P_{ij} \ge 0$ and $\sum_j P_{ij} = 1$, each channel of $(PH)_i$ is a convex combination of neighboring values. Consequently,

$$\|PH\|_{\max} \le \|H\|_{\max}, \qquad \|(I - P)H\|_{\max} \le 2\|H\|_{\max}, \tag{71}$$

where $\|H\|_{\max} = \max_{i,c} |H_{ic}|$. This is an amplitude bound. It is not a guarantee that normalized graph variation decreases, that the deviation is denoised, or that Euclidean energy is contractive for a nonsymmetric shift. Large deviations can reflect either signal structure or nuisance variation.

For an undirected nonnegative adjacency A with positive degrees, $P = D^{-1}A$ and $S = D^{-1/2}AD^{-1/2}$ obey

$$P = D^{-1/2}SD^{1/2}, \qquad D^{1/2}PH = S(D^{1/2}H). \tag{72}$$

The symmetric spectral interpretation therefore applies after the degree-coordinate transformation. On a general directed nearest-neighbor graph, a real orthogonal eigenbasis and real nonnegative Laplacian eigenvalues cannot simply be assumed. The implemented responses remain first-order polynomials of the same shift; branch construction requires only this operator relation.

U.4 No Shared Fourier Basis Is Required Across Modalities

The point operator acts on a geometric or feature-neighborhood graph, while image averaging acts on a grid with a padding convention. Their eigenvectors, eigenvalues, and boundary effects generally differ. Pairing their branch embeddings identifies a correspondence between training roles, not

between individual spectral coordinates. The image residual can also be nonzero at the edge of a constant feature map under zero-padded averaging. The cross-modal objective must learn whether these role-based targets are useful for the data; algebraic similarity alone does not establish that they describe the same object information.

U.5 A Spatial Reference for Complementarity

Figure 20 provides a concrete three-tap example. Its mean kernel assigns one third to the center and each adjacent node; the deviation kernel assigns two thirds to the center and minus one third to each neighbor. Adding the kernels recovers the identity impulse. The frequency view and spatial view describe the same operator pair, making the signed-response interpretation explicit.

V Seed-Level Exploratory Results

The entries below are transcribed from the archived paired run records and retain the terminal SVM endpoint described in Appendix Q. Accuracy and differences are in percent and percentage points, respectively. The source run IDs are preserved in the accompanying evidence manifest. Neither the ten-seed mean nor its interval should be substituted for a full-training comparison of the eleven-branch configuration.

Table 56: All ten paired terminal outcomes. Both variants have 73,367,890 trainable parameters; the response intervention changes both point and image inputs.

Seed	Duplicated features	Responses	Difference
3401	91.0454	91.3290	+0.2836
3402	91.2075	91.6937	+0.4862
3403	91.2480	91.7342	+0.4862
3404	91.4100	91.6937	+0.2836
3405	91.5721	91.4100	-0.1621
3406	91.1264	90.9643	-0.1621
3407	91.7342	91.3695	-0.3647
3408	91.7747	91.4911	-0.2836
3409	91.2480	91.9773	+0.7293
3410	91.8558	91.8963	+0.0405
Mean	91.4222	91.5559	+0.1337

The two-sided sign-flip calculation enumerates all 2^{10} sign assignments to the observed paired differences and compares their absolute means with the observed absolute mean. Its value is 0.3047. This calculation relies on sign-exchangeability under the null; it does not create independent runs or correct a mismatch in training budgets. Together with the paired Student-t interval, it supports an exploratory interpretation of the observed positive mean.

Table 57: Per-seed best validation accuracy for the response pilot. Only two seeds are available; these values are not terminal test accuracies.

Response	Seed 5101	Seed 5102	Mean
Fixed pair	90.7012	91.1585	90.9299
Learnable first order	91.0061	91.1585	91.0823
Polynomial	91.4634	90.9045	91.1839
Three responses	90.8028	91.1077	90.9553

The fixed and three-response variants select epoch zero for both seeds, while the polynomial selects epoch four. The learned first-order variant selects different epochs across the two seeds. The reported endpoint therefore depends on checkpoint selection as well as response design. These runs support exploration but do not establish a stable ranking of filter families.

W Pooled Set Matching and Longer Archived Runs

W.1 The Implemented Objective

The saved extension contains a pooled set-matching objective in addition to the response variants. A point-side head reads the l_4 map and adaptive average pooling produces 64 index bins. An image-side head reads C4 and pools it to an 8×8 grid. Features are normalized after these heads and pooling. This is a correspondence-light object-paired objective: consecutive bins in an arbitrary point ordering are not coordinate-defined patches.

Let u_{ap} and v_{bq} be normalized point-bin and image-grid descriptors from objects a and b, and $s_{abpq} = u_{ap}^{\top} v_{bq}$. The saved object-pair score is

$$S_{ab} = \frac{1}{2}\left[\frac{1}{P_t}\sum_p \tau_m \log \sum_q e^{s_{abpq}/\tau_m} + \frac{1}{Q_t}\sum_q \tau_m \log \sum_p e^{s_{abpq}/\tau_m}\right]. \tag{73}$$

A symmetric cross-entropy classifies the diagonal object pairs from S/τ_c. The matching temperature τ_m and contrastive temperature τ_c play different roles. The log-sum-exp terms provide soft best-match aggregation without a one-to-one correspondence constraint or explicit visibility model. This is the implemented alternative to the proposed geometric objective in Section D.

The extension's `local_capacity_control` allocates the same local heads but returns a zero local loss before evaluating them. Consequently, equal trainable parameter counts do not imply equal active optimization paths. The control helps document the allocation but does not fully isolate spatial matching from adding an auxiliary loss through previously inactive heads. A better active control would train equally sized heads on shuffled or object-pooled targets while retaining the same computational path.

W.2 Two-Seed Short Runs

The following runs use seeds 5101 and 5102, five warm-start epochs, at most 2,048 objects, and batch size 32. They use the same best-validation endpoint as the response pilot. Values were checked against the archived terminal summaries.

Table 58: Archived pooled-set pilot. Times are means in minutes; the allocated-head control is inactive in the local objective.

Variant	Seed 5101	Seed 5102	Parameters	Minutes
Inactive local-head control	90.8537	90.8028	73,450,066	7.23
Pooled set matching	91.0061	91.1077	73,450,066	7.26
Three responses + set matching	91.2602	91.2602	87,656,664	8.76

The pooled-set mean exceeds the inactive-control mean by 0.2287 percentage points. This is a two-seed exploration with an imperfect active-capacity control, not evidence that geometric point–pixel correspondence improves segmentation. No segmentation metric is supplied by these runs. The three-response combination changes both capacity and the response bank, so its mean cannot isolate a matching effect.

W.3 Twenty-Five-Epoch Warm-Start Runs

A longer continuation is available for seed 6101 with no training-object cap and batch size 32. Each run has 25 recorded epochs and one recorded terminal test access. Best validation and terminal SVM test accuracy remain separate columns. These runs still start from a historical checkpoint and are not independent full pretraining from scratch.

The best-validation ordering and terminal-test ordering differ. For example, the fixed-response run has a higher best validation value than pooled set matching, while pooled set matching has a higher terminal test value. This illustrates why the checkpoint-selection rule and evaluation endpoint must be preserved when moving numbers into the paper. A favorable single terminal outcome does not establish robustness across initializations.

Table 59: Archived 25-epoch continuations, seed 6101. Accuracy is in percent and time in recorded hours. One seed is insufficient for a reliable method ranking.

Variant	Best validation	Terminal test	Hours
Fixed responses	91.8191	90.9238	9.63
Polynomial responses	91.3618	90.4781	12.55
Pooled set matching	91.5142	91.8152	9.67
Three responses + set matching	91.4126	91.2480	12.91

W.4 One-Hundred-Epoch Runs from Fresh Initialization

The archive also contains fresh-initialization runs with a 100-epoch plan, no object cap, and batch size 32. Fixed responses have terminal records for seeds 6201–6203. Pooled set matching has terminal records for 6201 and 6203; seed 6202 stops at recorded epoch 89 and has no terminal summary in the available archive. A missing endpoint is not a zero score and is not imputed from a validation observation.

Table 60: Fresh-initialization long runs. The incomplete local run is explicitly retained. These are archived R1 classification experiments, not the full benchmark suite.

Variant / seed	Epochs	Best validation	Terminal test	Hours
Fixed responses / 6201	100	90.9045	91.0859	38.47
Pooled set matching / 6201	100	91.3618	90.2350	38.75
Fixed responses / 6202	100	90.6504	91.0454	38.40
Pooled set matching / 6202	90	–	–	–
Fixed responses / 6203	100	90.6504	90.2755	38.43
Pooled set matching / 6203	100	91.1585	91.0049	38.68

On the two seeds with both terminal outcomes, the pooled-set differences are -0.8509 and $+0.7293$ percentage points, averaging -0.0608. This reversal across seeds provides no support for a stable classification gain. The complete-case comparison is also subject to the unresolved missing run. It should not be pooled with the five-epoch pilot or the 25-epoch warm-start continuation. The experiment evaluates index-pooled set matching and leaves visibility-aware geometric alignment untested.

W.5 Consequences for the Next Local Experiment

The long-run record changes the priority of the next test. Increasing the training duration alone has not resolved the variability of index-pooled set matching. Before adding another response bank, the local interface should be tested with coordinate-defined patches and a control in which the same local heads receive an active auxiliary objective. This would address the specific ambiguity exposed by the saved implementation.

A compact study can fix the encoder, number of local descriptors, projection width, and candidate image grid, then compare index bins, geometric patches without calibration, and visible projected patches. The first contrast tests the construction of the point support; the second tests the additional correspondence information. A shuffled-projection condition would test dependence on the camera relation while preserving descriptor counts. All conditions should use the same validation-only checkpoint-selection rule and data splits, with sufficient independent pretraining seeds to estimate the paired differences.

Dense prediction is necessary to evaluate the stated purpose of this extension. A classification gain would not show that spatial correspondence survived pooling, and the present classification variability does not rule out a segmentation benefit. Report part or scene mIoU, boundary behavior, and transfer under limited labels alongside classification and cost. This study would connect local support and correspondence quality to the intended benefit in dense transfer.

X Connecting the Architecture Diagrams

X.1 From Object Identity to Structured Targets

Figure 1 separates object-level agreement from ambiguous local matches. The hierarchy in Figure 3 supplies stage-specific readouts; the response construction in Figure 4 adds neighborhood-dependent statistics. Figure 5 then assigns selected point stages to image targets. These diagrams describe successive parts of the same supervision interface.

X.2 Branches and Joint Training Connections

Figure 2 emphasizes the three response sources. The full point hierarchy has five base branches and six responses, expanded in Figure 12. All five objectives are optimized jointly. The low/high labels identify consensus/deviation paths; the orthogonal-regularization label denotes the squared-cosine term on projected embeddings. The standard update encodes two point augmentations and one image.

X.3 From Raw Responses to Object Embeddings

Figure 10 expands the direct, mean, and deviation paths at one source. The illustration uses node-major features for its local average, whereas Figure 4 uses channels in rows. Transposition reconciles the two conventions: $M = PH$ for node-major H, and $M^\top = H^\top P^\top$ for channel-first maps. Raw complementarity precedes adaptation and pooling. Branch projectors and fusion then produce object descriptors in the 256-dimensional alignment space.

Y Token Encoders, Partial Visibility, and Local Supervision

Y.1 A Backbone Interface with Explicit Geometric Support

The transferable component of MS-Point is its supervision interface, not a particular EdgeConv operation. For selected stage d, the point encoder supplies $(H^{(d)}, C^{(d)}, \mathcal{I}^{(d)})$: node-major features $H^{(d)} \in \mathbb{R}^{M_d \times C_d}$, centers $C^{(d)} \in \mathbb{R}^{M_d \times 3}$, and point memberships $\mathcal{I}_i^{(d)}$. Centers define a neighborhood; memberships identify the region represented by each token. A class token has no local geometric support and is excluded from the graph, although it can provide a separate object branch.

For a center-space graph, define

$$W_{ij}^{(d)} = \mathbf{1}\{j \in \mathrm{kNN}(c_i^{(d)}; C^{(d)})\}, \quad P_{ij}^{(d)} = \frac{W_{ij}^{(d)}}{\sum_j W_{ij}^{(d)}}, \tag{74}$$

$$H_s^{(d)} = P^{(d)} H^{(d)}, \quad H_r^{(d)} = H^{(d)} - H_s^{(d)}. \tag{75}$$

An implementation must specify self-neighbor inclusion and ensure nonempty neighborhoods. A radius graph can use a self-neighbor fallback. If a symmetric analysis is required, explicitly symmetrize the weights before normalization; ordinary directed k-NN is not symmetric. The center graph does not depend on the attention weights. Feature-space k-NN instead depends on learned features, even when it does not read attention matrices. These are distinct response constructions, not interchangeable names for the same graph.

Semantic depth and spatial scale. Intermediate blocks at a fixed token resolution give a depth axis. They do not automatically provide different spatial resolutions. When token pooling changes support, features, centers, and memberships must be updated together. Selecting sparse, separated blocks avoids multiplying projection costs at every depth; the reported three-level/all-block comparison gives the same part mIoU of 86.9 with 30.4 versus 33.8 H100 hours. Exact block indices, token count, and tokenizer remain configuration fields; the phrase "Point Transformer" alone is not a complete architecture specification.

Projection, targets, and fusion. Each base or response map is pooled into a descriptor and independently projected to 256 dimensions. Image ConvNeXt stages or ViT patch grids can replace ResNet stages without modifying the underlying attention mechanism. A ViT class token cannot be reshaped into a spatial grid; only patch tokens with a known layout participate in image smoothing. The map ρ records the image block/stage, response identity, and head identity. Reusing one image stage for two point stages does not imply shared projection parameters. The C3/C5/C5 map verified for the nine-branch DGCNN implementation should not be silently assigned to arbitrary Transformer block numbers.

Masking and graph scope. MS-Point-T does not require a reconstruction decoder. If a masked encoder is used as an implementation base, the graph must be defined over the tokens actually visible to that encoder, or the experiment must explicitly describe a full-token pass. Introducing hidden tokens or their target centers into the graph changes the information available during training. For a coordinate graph on unchanged centers, neighbor indices can be reused across selected blocks; feature-dependent graphs generally cannot. Brute-force center distances cost $O(M_d^2)$, response gathering costs $O(M_d k C_d)$, and storing gathered features can dominate the branch workspace. A center graph is therefore an interface with measurable cost, not a cost-free abstraction.

Algorithm 2 Structured pretraining with point tokens (interface specification)

Require: Paired object point cloud and image; selected token stages; map ρ
1: Construct two point augmentations and the selected encoded image view(s)
2: Run each encoder once per input view and retain selected feature maps
3: **for** each point view and selected token stage d **do**
4: Obtain $H^{(d)}$, centers, and memberships; construct or reuse the specified graph
5: Form $H_s^{(d)} = P^{(d)} H^{(d)}$ and $H_r^{(d)} = H^{(d)} - H_s^{(d)}$
6: Adapt, pool, and project each active base/response branch
7: **end for**
8: Construct image-stage base/response targets and apply independent heads using ρ
9: Compute branch gates and fused embeddings in each modality
10: Evaluate the five objectives with the selected branch sets and reduction conventions
11: Backpropagate their weighted sum; update trainable modules

Y.2 Which Information Should a Single View Align?

Let G denote the full object geometry, v the camera state, G_v its visible surface, B_v the background, and A_v appearance. A rendered observation can be written schematically as $I_v = \mathcal{R}(G_v, A_v, B_v)$. The point cloud may contain information about both G_v and its hidden complement. Consequently, object identity is a valid positive relation, but it does not imply that every 3D patch has a visible image counterpart. This distinction explains why a useful object-level loss can coexist with false local positives if alignment is applied indiscriminately.

A useful conceptual partition is $z^{3D} = (z^{3D}_{\text{shared}}, z^{3D}_{\text{private}})$ and $z^{2D} = (z^{2D}_{\text{shared}}, z^{2D}_{\text{private}})$. Agreement should constrain the shared factors without requiring the private factors to be identical. This is an explanatory model and a direction for future selective alignment, not a factorization already identified by the five-loss implementation. Neighborhood residuals are not automatically private, and smoothed features are not automatically shared: either can contain visible structure, hidden geometry, background, or noise.

Figure 22 illustrates how the same 3D support changes apparent extent and overlap across views. It makes explicit why an object-pair label alone cannot select a unique local image target.

Y.3 Geometry-Preserving Local Targets

A geometric local branch starts from FPS+k-NN patches, voxels, or superpoints, retaining each center and support. With camera parameters (K, R, t), project c_m to $u_m = \pi(K(Rc_m + t))$. Positive depth, image bounds, a foreground mask when available, and a depth-buffer consistency test define a visibility indicator v_m. An image patch relation can then use a normalized spatial kernel,

$$q_{mn} = \frac{v_m\, a_n \exp(-\|u_m - u_n\|^2/(2\sigma^2))}{\sum_j a_j \exp(-\|u_m - u_j\|^2/(2\sigma^2))}, \qquad m \in \mathcal{V}, \tag{76}$$

where $\mathcal{V}$ contains accepted visible supports and a_n marks valid image patches. Rows with no valid target are skipped or assigned to an explicit unmatched state; they must not be normalized as if a match existed. Camera transforms must be updated consistently with point augmentation, image cropping, and resizing. Visibility of the center alone is an approximation when a patch spans both visible and occluded surfaces.

For projected normalized features p_m and v'_n, a soft-target local objective can be specified as

$$\ell(P, V) = -\frac{1}{|\mathcal{V}|} \sum_{m \in \mathcal{V}} \sum_{n} q_{mn} \log \frac{\exp(p_m^\top v'_n/\tau)}{\sum_{j \in \mathcal{J}} \exp(p_m^\top v'_j/\tau)}, \tag{77}$$

$$\mathcal{L}_{\text{local}} = \tfrac{1}{2}\big[\ell(P, V) + \ell(V, P)\big]. \tag{78}$$

The reverse relation is normalized separately over its valid support. The candidate/negative set $\mathcal{J}$ must be specified, because treating other patches of the same object as negatives can create semantic false negatives. These equations give a reproducible visibility-aware design to accompany the reported local extension; the supplied result tables do not identify its exact kernel, threshold, or negative set. We therefore do not attribute their numerical gains to a particular choice of those unspecified settings.

Why structure can help disambiguation. Two patches can have similar pooled semantics but different neighborhood organization: for example, an elongated support and a junction may both belong to one object. Combining their base, smooth, and deviation features exposes additional evidence for matching. It does not guarantee unique correspondence, particularly for symmetric or repeated parts. A structural compatibility score can regularize an appearance-based matching matrix, while visibility and an unmatched option prevent that regularizer from forcing an absent correspondence.

When calibration is unavailable. Cross-attention or optimal transport can infer soft assignments between token sets. A useful design includes slack mass or an unmatched bin so that partial observations need not satisfy an artificial bijection. Such assignments are learned associations unless validated against geometry. Pooling an arbitrary point order into 64 bins is an even weaker spatial surrogate: it provides multiple descriptors but does not define local neighborhoods. This is why the archived pooled-set pilot and the geometry-preserving dense-transfer experiment remain separate.

Y.4 Checks that Distinguish Useful Alignment from Forced Agreement

A mechanism study should vary the overlap between visible 3D supports and image regions while holding object identity fixed. Report accepted-match coverage, retrieval precision where correspondence is known, boundary/part metrics, and downstream performance alongside the contrastive loss. Background swaps test reliance on scene context; image viewpoint changes test visibility; point reordering tests whether the local construction respects set structure. Occluded regions should either remain unmatched or be trained by a within-modality objective. Higher matching confidence alone is not evidence of correctness. These diagnostics refine the local method without retroactively claiming that the current object-level objective solves occlusion or correspondence.

Z Extended Results, Protocols, and Resource Details

We report further comparisons across backbones, graph constructions, local supervision, and label budgets. Each study fixes the comparison conditions stated in its table. We distinguish published baselines, controlled full-training experiments, and shorter exploratory runs throughout.

Z.1 Published Results and Controlled Reruns

Table 61 compares frozen representations under the evaluation settings used in the respective publications. The ScanObjectNN split and feature evaluator remain part of each result's protocol. We use Point-PQAE, PointGPT, PointDif, PCP-MAE, OpenShape, and ULIP-2 to contextualize prediction targets and supervision sources; our controlled objective comparisons use the common evaluator in Table 1.

Table 61: Published frozen-representation results. Each row retains its native evaluation setting. TPM rows use arXiv v1.

Method	Pretraining	Backbone	MN40	Scan
Point-BERT	Masked token	Transformer	87.4	–
Point-MAE	Reconstruction	Transformer	91.0	77.7
Point-M2AE	Multiscale MAE	Transformer	92.9	84.1
CrossPoint	Cross-modal	DGCNN	91.2	81.7
I2P-MAE	Image-guided MAE	Transformer	93.4	87.1
MM-Point	Multi-view contrast	DGCNN	92.4	87.8
Point-MAE + TPM	Multiple masks	Transformer	91.2	78.2
Point-M2AE + TPM	Multiple masks	Transformer	93.2	84.0
MS-Point	Structured contrast	DGCNN	93.6	89.6

Table 62: Frozen evaluation and full fine-tuning answer different questions. Literature settings are separated from our controlled OBJ-BG reruns.

Method	Evaluator	Scan setting	MN40	Scan
CrossPoint	Frozen SVM	OBJ-BG (source)	91.2	81.7
MM-Point	Frozen SVM	Published setting	92.4	87.8
Point-MAE	Frozen SVM	OBJ-BG (source)	91.0	77.7
Point-MAE	Full fine-tuning	PB-T50-RS	93.8	85.18
MS-Point	Frozen SVM	Benchmark setting	93.6	89.6

Evaluation Protocol of TPM. TPM arXiv v1 (2409.17547, Table 2) evaluates frozen linear SVM on ScanObjectNN PB-T50-RS. Its Point-MAE and Point-MAE+TPM results are 90.8/77.1 and 91.2/78.2 on ModelNet40/ScanObjectNN; Point-M2AE and Point-M2AE+TPM obtain 92.9/83.1 and 93.2/84.0. We keep these native results separate from our controlled OBJ-BG reruns, where Point-MAE+TPM obtains 91.8/79.2.

The references are Point-BERT (Yu et al., 2022), Point-MAE (Pang et al., 2022), Point-M2AE (Zhang et al., 2022), CrossPoint (Afham et al., 2022), I2P-MAE (Zhang et al., 2023), MM-Point (Yu & Song, 2024), and TPM (Liu et al., 2025). The controlled TPM row in Table 1 is 91.8/79.2, not the literature row 91.2/78.2. The two evaluations are intentionally separate.

Z.2 Cross-Backbone and Image-Stage Studies

The same-Transformer global/structured contrast in the main text isolates the objective more directly than comparing different encoders. The additional matrix tests whether stronger feature providers can coexist with the structured objective. Since it does not contain a global-only control for every ConvNeXt cell, it does not independently isolate the objective under every image encoder.

Table 63: Full structured objective with different feature providers. Part is instance mIoU.

3D encoder	2D encoder	MN40	Scan	Part	H100 h
DGCNN	ResNet-50	93.6	89.6	86.1	30.5
Point Transformer	ResNet-50	94.3	90.4	86.9	30.4
DGCNN	ConvNeXt-B	93.9	90.0	86.5	32.1
Point Transformer	ConvNeXt-B	94.6	90.8	87.2	32.3

Table 64: ResNet-50 stage endpoints for a 224×224 input. C2–C5 are stages, not literal convolution indices.

Stage	PyTorch	Standard name	Stride	Channels	Grid
C2	`layer1`	`conv2_x`	4	256	56×56
C3	`layer2`	`conv3_x`	8	512	28×28
C4	`layer3`	`conv4_x`	16	1024	14×14
C5	`layer4`	`conv5_x`	32	2048	7×7

Literal indices 10/15/20 depend on the counting convention. The stage-selection table also uses more image sources than the verified nine-branch C3/C5/C5 mapping. It is therefore a separate image-hierarchy study, not evidence that unused C2/C4 tensors affect the saved R1 loss. A reproducible configuration must name the active branch targets and fusion inputs.

Table 65: Reported image-feature selection study. Relative feature cost is the supplied quantity, not a measured total-backbone speedup.

Image features	MN40	Scan	Rel. cost
C5 only	92.9	88.1	0.82
C3 + C5	93.4	89.0	0.91
C2 + C3 + C4 + C5	93.6	89.6	1.00
Internal indices 10/15/20	93.1	88.5	0.96

Table 66: Coordination-weight sensitivity and a task-tuned reference; Part is instance mIoU.

Setting	MN40	Scan	Part
No coordination ($w = 0$)	93.1	88.7	85.6
$w = 0.2$	93.4	89.2	85.9
Default ($w = 0.4$)	93.6	89.6	86.1
$w = 0.6$	93.5	89.4	86.0
Task-specific tuning	93.8	90.4	86.4

Z.3 Loss Roles, Weights, and Extended Task Metrics

The five terms are optimized jointly, not in five chronological training stages. The description "stability–coordination–branch anchoring–fusion anchoring–diversity" states their intended roles. Their actual gradients interact, as analyzed in Appendix A.

Table 67: Five parallel objectives and default weights. Exact averaging conventions affect the effective regularization.

Term	Role	Weight
$\mathcal{L}_{\text{same}}$	Augmentation consistency	1.0
$\mathcal{L}_{\text{cross}}$	Adjacent-depth coordination	0.4
$\mathcal{L}_{\text{cmid-level}}$	Direct branch image targets	1.0
$\mathcal{L}_{\text{cmid-fused}}$	Image target for fusion	1.0
$\mathcal{L}_{\text{band}}$	Projected response diversity	0.05

Removing branch-level alignment has the largest classification reduction in the supplied deletion table. This supports its role within the complete system; the deletion still leaves image supervision through the fused target. Retuning gives up to 0.8 points on ScanObjectNN, so the default's usefulness should be described as reuse with a measurable opportunity cost, not universal optimality.

Table 68: Loss deletion including the newly supplied instance-mIoU column.

Objective	MN40	Scan	Part
Global only	92.8	87.9	85.4
Without $\mathcal{L}_{\text{same}}$	93.2	88.9	85.7
Without $\mathcal{L}_{\text{cross}}$	93.1	88.7	85.6
Without $\mathcal{L}_{\text{cmid-level}}$	92.9	88.3	85.4
Without $\mathcal{L}_{\text{cmid-fused}}$	93.2	88.9	85.7
Without $\mathcal{L}_{\text{band}}$	93.3	89.1	85.8
Full	93.6	89.6	86.1

Table 69: Neighborhood diagnostic with the newly supplied part mIoU. Single-response rows have one active response.

Construction	MN40	Scan	Part
Duplicated features	92.9	88.2	85.5
Degree-matched random	93.0	88.3	85.6
Signal shuffle	93.1	88.5	85.7
Smooth only	93.2	88.8	85.7
Deviation only	93.3	89.0	85.8
Complementary pair	93.6	89.6	86.1

Table 70: Fixed weights versus retuning, with the supplied part-segmentation values.

Encoder / points	Weights	MN40	Scan	Part
DGCNN / 1,024	Default	93.6	89.6	86.1
DGCNN / 1,024	Retuned	93.8	90.4	86.4
Transformer / 1,024	Default	94.3	90.4	86.9
Transformer / 1,024	Retuned	94.4	90.7	87.0
DGCNN / 2,048	Default	93.8	90.0	86.4
DGCNN / 2,048	Retuned	93.9	90.2	86.5

Table 71: Fusion study. Relative cost refers to the supplied fusion-cost measure.

Fusion	MN40	Scan	Part	Rel. cost
Mean	93.3	89.0	85.8	0.94
Concatenate + MLP	93.4	89.2	85.9	1.12
Adaptive gates	93.6	89.6	86.1	1.00

Table 72: Factorial comparison with additional part mIoU. Parameter counts differ between rows.

Configuration	Params. M	Hours	MN40	Scan	Part
Global	71.84	24.6	92.8	87.9	85.4
Hierarchy	72.45	26.5	93.1	88.7	85.7
Responses	72.76	27.8	93.2	88.9	85.8
Hierarchy + responses	73.12	29.3	93.4	89.2	85.9
+ Fusion	73.37	30.5	93.6	89.6	86.1

Z.4 Fusion, Factorial Effects, and Geometry Controls

The factorial gains are not super-additive: $93.4 - 93.1 - 93.2 + 92.8 = -0.1$ and $89.2 - 88.7 - 88.9 + 87.9 = -0.5$. Both axes can improve over the baseline while their benefits overlap. Similarly, the better adaptive-fusion mean supports that configuration in this study, not a causal interpretation of individual gate values. The separate single-response table reports different values and remains separately labeled in Appendix S.

Z.5 View Counts, Timing, and Optimization Progress

Candidate images stored offline and images encoded in a step are distinct quantities. The standard MS-Point setting uses two point views and one encoded image; the supplied protocol lists two candidate images. MM-Point's comparison uses 24 candidate images and four encoded images. Using a second image for a dense-task variant changes the pretraining configuration and must be recorded, rather than described as a downstream-only change.

Table 73: Candidate and encoded views in the reported resource comparison.

Method	Candidates	Images/step	Points/step	GPU h	GiB
MM-Point	24	4	2	36.9	12.80
MS-Point	2	1	2	30.5	10.03

Table 74: Full reported cost fields. Epoch minutes and total GPU hours retain their source definitions; a common epoch count is not implied.

Method	3D/2D passes	Params. M	GFLOPs	GiB	min/epoch	GPU h
Global	2/1	71.84	13.1	9.72	40.3	24.6
MM-Point	2/4	73.10	21.8	12.80	59.8	36.9
MS-Point	2/1	73.37	15.2	10.03	50.0	30.5
Point-MAE	1/0	22.1	17.9	11.40	105.0	72.0
Inter-MAE	1/0	29.6	19.6	12.10	111.0	76.5

The reported 30.5 total GPU hours and 50.0 minutes per epoch imply 36.6 epochs if both quantities cover exactly the same work on one GPU. Their boundaries therefore need to be supplied before converting one into the other. We retain both quantities without inventing an epoch count. Relative to 72.0/76.5 reported hours for Point-MAE/Inter-MAE, 30.5 hours is a reduction of 57.64%/60.13%; this is a comparison of the supplied complete systems, not an architecture-independent speed theorem.

Z.6 Paired Confidence Intervals

The full-training seed vectors are not present in the supplied summary. Hence these intervals are reported as supplied, whereas Appendix V provides the independently recomputed ten-pair pilot. A future release should include both seed identifiers and pairing rules, allowing verification of the full-training uncertainty rather than treating marginal standard deviations as sufficient.

Z.7 Cross-Modal Perturbations and Outdoor Class Groups

Table 75: Reported time per epoch for the 73.37M-parameter, 15.2-GFLOP controls in Table 9.

Control	min/epoch
Duplicated features	50.1
Random neighbors	50.3
Shuffled features	50.2
Smoothing + deviation	50.0

Table 76: ModelNet40 accuracy at reported H100 time budgets. These sampled checkpoints are not a continuous convergence curve.

Method	12 h	24 h	Final	Final h
Point-MAE	88.6	90.0	91.4	72.0
Inter-MAE	89.1	91.0	92.1	76.5
MM-Point	91.0	92.1	92.5	36.9
MS-Point	92.2	93.3	93.6	30.5
MS-Point-T	92.8	94.0	94.3	30.4

Table 77: Our paired differences and 95% intervals for five-seed full-training results, in percentage points.

Contrast	MN40 Δ	MN40 CI	Scan Δ	Scan CI
MS-Point − Global	+0.80	[+0.62,+0.98]	+1.70	[+1.38,+2.02]
MS-Point − MM-Point	+1.10	[+0.85,+1.35]	+1.70	[+1.34,+2.06]

Table 78: Our observation perturbations; Part is instance mIoU. No exact calibration is used for object-level rows.

Condition	Calibration	MN40	Scan	Part
Object identity	No	93.6	89.6	86.1
Viewpoint $\pm 15^\circ$	No	93.5	89.4	86.0
30% random occlusion	No	93.3	89.2	85.8
Asynchronous same object	No	93.2	89.0	85.7
Region soft matching	Coarse	93.8	90.1	86.7

Table 79: Full supplied 1%-label scene table, MinkUNet-34C, sequence 08. The combined reference row is not evidence for either named baseline individually. Camera denotes pretraining access.

Initialization	Camera	mIoU	Gain	GPU h
Random initialization	No	34.2	–	26.0
Scene-only SSL	No	40.8	+6.6	35.0
SLidR / Seal (unresolved identity)	Yes	42.0	+7.8	48.0
Global MS-Point	Yes	44.6	+10.4	42.0
MS-Point + local	Yes	47.3	+13.1	50.0

Table 80: Reported class-frequency groups at 1% labels: eight frequent, six medium-frequency, and five rare classes.

Initialization	Freq. (8)	Mid. (6)	Rare (5)	Overall
Random initialization	45.8	31.1	19.4	34.2
Scene-only SSL	52.4	38.8	24.7	40.8
SLidR / Seal reference	53.6	40.1	25.8	42.0
Global MS-Point	56.0	42.7	28.6	44.6
MS-Point + local	58.2	45.1	32.3	47.3

Local alignment adds 2.2/2.4/3.7 points over global MS-Point for frequent/medium/rare classes. Against random initialization, gains are 12.4/14.0/12.9 points: the medium group leads in absolute gain, while the rare group leads in relative gain (66.5%). These comparisons answer different questions. Overall class mIoU weights group means by 8/19, 6/19, and 5/19; rounding can affect the final decimal. Reproduction requires the class membership and grouping rule.