Geometry-Guided Semantic Reconstruction for 3D Instance Segmentation

Yuanhao Su[1,2], **Shaofeng Zhang**[1*]
[1]University of Science and Technology of China, [2]Fuzhou University
yharoldsu0627@gmail.com, sfzhang@ustc.edu.cn

Abstract

3D instance segmentation requires representations that are both geometrically structured and semantically discriminative. Pure 3D encoders model spatial structure well but often lack strong appearance semantics, while directly lifting 2D priors into 3D by concatenation injects rich semantics without ensuring that they align well with the underlying 3D structure. In this paper, we address this problem by formulating 2D-to-3D semantic transfer as a **masked semantic reconstruction** task. Specifically, we mask a subset of background lifted 2D priors and train the 3D encoder to recover the missing semantic priors, providing explicit supervision for distilling rich 2D semantics into the 3D backbone. We further propose **geometry-modulated semantic injection**, which uses raw 3D geometry to modulate lifted 2D priors before sparse encoding, so that geometry explicitly controls semantic injection and alleviates the distribution gap between 2D semantic features and 3D geometric features. Experiments on two baseline frameworks demonstrate that the proposed method achieves state-of-the-art results in both full-data and few-shot settings on ScanNetV2 and ScanNet200. Code is publicly available at https://github.com/yharoldsu0627/Geometry_Guided_Semantic_Reconstruction

1 Introduction

3D scene understanding from point clouds is a fundamental capability for applications such as robotics [35]. As a central problem in point cloud learning [7], 3D instance segmentation aims to identify and delineate individual object instances in complex scenes, providing the geometric and semantic information needed for downstream reasoning and decision-making [10, 33]. Pure 3D encoders usually model spatial structure well, but their appearance-level semantics are still limited [2, 40]. At the same time, the high cost of 3D annotation makes it difficult to learn strong semantic representations directly in 3D space [3, 36, 46]. In contrast, priors from 2D foundation models often contain much richer semantics learned from large-scale image data, yet they lack inherent 3D geometric awareness [28]. This creates a central question for 3D instance segmentation: **how can we inject the rich semantics from 2D priors into the 3D encoding process while ensuring that the resulting representations remain aligned with the underlying 3D structure?**

Existing methods mainly follow two directions, but neither fully resolves this geometry-semantic tension. One line takes only 3D geometric features as encoder input and learns scene representations directly in 3D [10, 33, 18]. These methods preserve structure well, but their semantic expressiveness is still bounded by what can be learned from limited 3D supervision (see Figure 1(a)). Another line transfers semantics from 2D priors by lifting them into 3D and fusing them with 3D features [24, 1, 27]. However, when this fusion is implemented by direct concatenation or similarly lightweight injection,

*Shaofeng Zhang is the corresponding author. This work was supported by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China and NSFC Youth Project 62606504.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

the encoder must discover cross-modal interactions from concatenated features without explicit guidance on how geometry should regulate semantic injection [27, 15] (see Figure 1(b)). This is particularly difficult because 2D semantic features and 3D geometric features follow markedly different distributions, emphasizing appearance and structure respectively. Direct concatenation often fails to produce stable joint representations, as shown in prior work [15, 44, 45]. Moreover, simply providing extra 2D cues introduces no explicit supervision that encourages the encoder to exploit geometry and context to absorb them [24]. As a result, semantically rich but geometrically misaligned 2D priors are difficult to transform into representations that are well aligned with the underlying 3D structure [5]. Empirically, direct concatenation remains consistently weaker than our proposed design across both benchmarks.

To address these limitations, we move beyond both pure 3D encoding and direct concatenation by formulating 2D-to-3D semantic transfer as a masked semantic reconstruction problem with geometry-modulated semantic injection (see Figure 1(c)). Specifically, we mask a subset of background 2D priors and train the encoder to recover the missing semantic priors, which provides explicit supervision for distilling rich 2D semantics into the 3D backbone. Masking is restricted to background priors so that the auxiliary objective does not directly disturb foreground instance cues. At the same time, it forces the encoder to infer missing semantics from context. To make this reconstruction objective more effective, we further use raw 3D geometry to modulate the lifted 2D priors before they enter the encoder. In this way, geometry explicitly controls semantic injection and helps alleviate the distribution gap between 2D semantic features and 3D geometric features, as supported by Table 4 and Table 6. Under this design, the encoder is encouraged to combine 3D structure with surrounding semantic context to recover the missing priors instead of relying directly on local and potentially geometrically inconsistent 2D cues, as supported by Table 4 and Table 7. Extensive experiments on two baseline frameworks further demonstrate consistent improvements across both benchmarks, and Table 1 and Table 2 show that the full design consistently outperforms direct concatenation. Our main contributions are summarized as follows:

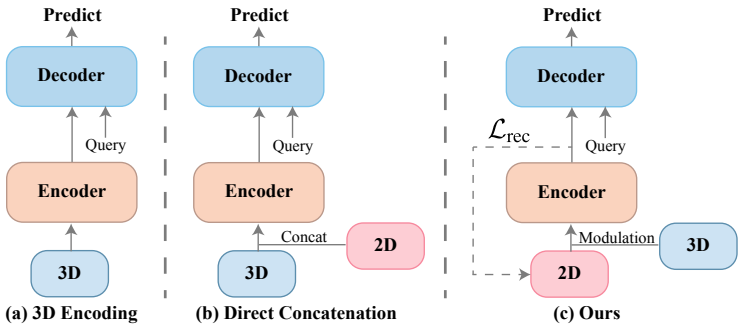

Figure 1: Comparison of three paradigms for 3D instance segmentation with 2D priors. (a) Pure 3D encoding preserves geometry but lacks rich semantics. (b) Direct concatenation injects lifted 2D priors, yet the transfer remains passive. (c) Our method combines geometry-modulated semantic injection and masked semantic reconstruction to enforce explicit semantic transfer.

1) A masked semantic reconstruction objective for 2D-to-3D transfer. By masking a subset of background 2D priors and training the encoder to recover the missing semantic priors, the method turns semantic transfer from simple feature injection into a supervised reconstruction objective.

2) A geometry-modulated semantic injection design that uses raw 3D geometry to modulate lifted 2D priors before they are fed into the encoder. This helps bridge the distribution gap between 2D semantic features and 3D geometric features, while allowing geometry to explicitly regulate how 2D semantics enter the 3D backbone.

3) Extensive experiments on two baseline frameworks showing that the proposed method achieves state-of-the-art results in both full-data and few-shot settings on ScanNetV2 and ScanNet200.

2 Related Work

3D Instance Segmentation with 2D Priors. Previous work on 3D instance segmentation has explored a broad spectrum of paradigms, including proposal-based, grouping-based, superpoint-based, kernel-based, and transformer-based methods [12, 6, 10, 8, 47, 39, 17, 33, 37, 19, 20, 13]. More

recent instance-centric designs have further advanced the benchmark frontier. Spherical Mask formulates instances with a spherical coarse-to-fine representation to improve spatial localization [34]. EASE introduces language-guided semantic priors and edge-aware supervision to better separate adjacent instances with similar appearance [31]. Relation3D strengthens query reasoning by explicitly modeling inter-instance relations in a DETR-style decoder [18]. IKNE further enhances candidate features through instance-wise knowledge aggregation and structural guidance [30]. To strengthen semantic understanding, recent methods have also introduced 2D priors into 3D pipelines. SegDINO3D concatenates lifted DINO image and object features with raw 3D inputs before the sparse encoder [27]. Open3DIS and Any3DIS use multi-view 2D masks to generate or refine 3D instance proposals [22, 21]. OpenScene, GeoPurify, and SAS transfer 2D priors through feature distillation or geometry-aware refinement [24, 5, 16]. While these approaches demonstrate the value of 2D guidance, they mainly rely on direct concatenation, mask lifting, or fixed-target distillation, all of which treat lifted 2D priors as side information rather than signals that must be actively integrated with 3D geometry. A key limitation is that such lightweight fusion strategies provide little explicit supervision for the encoder to absorb and integrate 2D semantics with 3D structure.

Reconstruction-Based 3D Representation Learning. Reconstruction has become a widely adopted strategy for learning 3D representations. Point-BERT [41], Point-MAE [23], and Point-M2AE [42] recover masked point tokens or local geometry to learn contextual 3D features. Cross-modal variants further incorporate image-derived supervision, such as ACT [4], I2P-MAE [43], and Contrast with Reconstruct [26]. These works establish reconstruction as an effective self-supervised learning signal. However, they are primarily designed for generic 3D representation learning rather than supervised instance segmentation. Bringing reconstruction into downstream instance segmentation therefore requires careful task-specific design. The reconstruction target should complement rather than compete with the primary objective, and the auxiliary supervision should enhance rather than disrupt the learned representations.

Conditional Modulation for Cross-Modal Fusion. Conditional modulation offers an adaptive mechanism for cross-modal fusion, where one signal reshapes another through learned affine parameters. Representative formulations include FiLM [25] and AdaIN [9], which have proven effective across vision and multimodal tasks. However, its use in 3D instance segmentation remains limited, especially for controlling how 2D priors are injected into sparse 3D encoders. In our context, modulation is not merely a generic fusion operator. Rather, it provides a way for geometry to regulate semantic injection so that lifted 2D priors are integrated more consistently with the underlying 3D structure.

3 The Proposed Method

3.1 Overall Architecture

Figure 2 illustrates the overall pipeline. Our method augments a supervised 3D instance segmentation backbone with two encoder-side components: geometry-modulated semantic injection and masked semantic reconstruction. Given multi-view RGB images and a 3D point cloud of N points, dense image features are first extracted by a pre-trained DINO-X encoder and lifted to each visible 3D point through camera projection. This yields per-point 2D feature vectors $\mathbf{P} = \{\mathbf{p}_j\}_{j=1}^{N}$, where $\mathbf{p}_j \in \mathbb{R}^d$ is the d-dimensional 2D feature of point j. The point cloud is partitioned into M superpoints $\{\mathcal{S}_i\}_{i=1}^{M}$ following [14], and the superpoint-level 2D prior $\mathbf{f}_i = \frac{1}{|\mathcal{S}_i|}\sum_{j\in\mathcal{S}_i}\mathbf{p}_j$ is obtained by averaging the per-point 2D features within each superpoint $\mathcal{S}_i$. During training, we define a masked 2D input by zeroing the lifted priors on a subset of background superpoints. The resulting masked per-point 2D features are fed into a Sparse 3D U-Net [2] as the host feature stream, while the raw 3D input geometry is transformed into affine parameters that modulate this semantic stream at the encoder input. The encoder outputs are then aggregated back to superpoints, and the masked superpoints are supervised to recover their original superpoint-level 2D priors. A DETR-style query decoder [18, 27] predicts instance masks from the aggregated superpoint features. The reconstruction branch is used only during training and discarded at inference.

3.2 Geometry-Modulated Semantic Injection

Directly concatenating lifted 2D priors with 3D features leaves the encoder to discover cross-modal interactions implicitly. To make geometry explicitly control how semantics enter the backbone, we

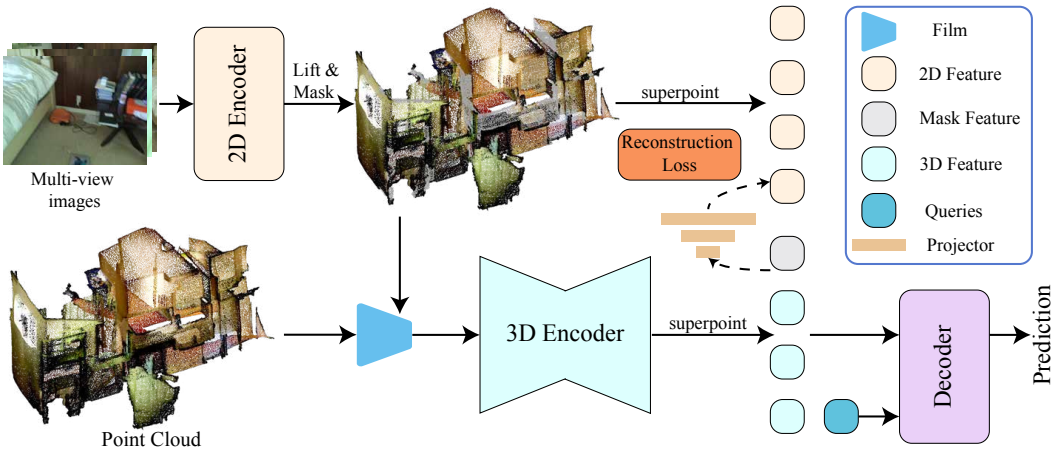

Figure 2: Overview of the proposed framework. Multi-view DINO-X features are lifted to 3D points and processed by a sparse 3D encoder. During training, a subset of background superpoints is masked in the lifted 2D stream and reconstructed from geometry-modulated encoder features. The aggregated superpoint features are finally decoded into instance masks.

modulate the lifted 2D stream with raw 3D input geometry before sparse encoding. We first define the masked 2D input shared by the semantic injection and reconstruction branches. Let $\mathbf{x}_j \in \mathbb{R}^{d_g}$ denote the raw 3D input feature of point j, and let $\pi(j)$ return the index of the superpoint that contains point j. Let $\mathcal{O}$ denote the set of superpoints assigned to ground-truth instances and $\mathcal{B} = \{1, \dots, M\} \setminus \mathcal{O}$ the background set. A masked subset $\mathcal{M} \subseteq \mathcal{B}$ is drawn uniformly at random with a background masking ratio r. Thus $|\mathcal{M}| = \lfloor r|\mathcal{B}| \rfloor$ (see Sec. 3.3 for details). The masked per-point 2D feature is

$$\tilde{\mathbf{p}}_j = \begin{cases} \mathbf{0}, & \text{if } \pi(j) \in \mathcal{M}, \\ \mathbf{p}_j, & \text{otherwise}, \end{cases} \tag{1}$$

i.e., the 2D features of all points inside the masked background superpoints are zeroed out. The masked 2D features $\{\tilde{\mathbf{p}}_j\}$ and the raw 3D features $\{\mathbf{x}_j\}$ are voxelized separately, yielding per-voxel features $\mathbf{v}_u^{(2)}$ and $\mathbf{v}_u^{(3)}$. The 2D voxel features $\mathbf{v}_u^{(2)}$ are taken as the input stream of the Sparse 3D U-Net, while a lightweight shared 3-layer MLP ψ applied to each 3D voxel feature predicts a pair of affine parameters,

$$[\boldsymbol{\gamma}_u, \boldsymbol{\beta}_u] = \psi(\mathbf{v}_u^{(3)}), \tag{2}$$

with $\boldsymbol{\gamma}_u, \boldsymbol{\beta}_u \in \mathbb{R}^d$ matching the lifted 2D feature dimension. The input 2D voxel feature is then modulated by

$$\mathbf{v}_u^{(2)} \leftarrow (1 + \boldsymbol{\gamma}_u) \odot \mathbf{v}_u^{(2)} + \boldsymbol{\beta}_u, \tag{3}$$

where $\odot$ denotes element-wise multiplication. The modulated feature is then fed into the sparse encoder. This design lets geometry shape how the lifted 2D priors enter and propagate through the encoder, rather than leaving semantic injection to passive concatenation.

3.3 Masked Semantic Reconstruction

Lifted 2D priors alone do not provide explicit supervision for semantic transfer, so we introduce a masked semantic reconstruction objective on top of the encoder features. A natural question is *which regions to mask*. Masking foreground superpoints would directly corrupt the instance-level cues that the downstream decoder relies on, creating a conflict between the reconstruction and segmentation objectives. We therefore restrict masking to background superpoints $\mathcal{M} \subseteq \mathcal{B}$ as defined in Sec. 3.2. This design creates a complementary learning signal. The encoder must infer missing background semantics from surrounding context and geometry, strengthening its contextual reasoning without disturbing the foreground features that drive instance prediction (validated in Table 7). Masking is performed independently for each scene in the batch, so that the masked subset varies across iterations. The original per-point 2D features $\{\mathbf{p}_j\}$ are kept aside as reconstruction targets.

After the Sparse 3D U-Net, point-level outputs are aggregated to superpoint features $\mathbf{g}_i \in \mathbb{R}^D$ by averaging encoder features within each superpoint partition [18]. A 3-layer MLP projector ϕ maps each aggregated superpoint feature back to the 2D feature space. Let $\mathbf{W}_1 \in \mathbb{R}^{D\times D_h}$, $\mathbf{W}_2 \in \mathbb{R}^{D_h\times D_h}$ and $\mathbf{W}_3 \in \mathbb{R}^{D_h\times d}$ denote the linear weights, and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{D_h}$, $\mathbf{b}_3 \in \mathbb{R}^d$ the corresponding biases, where D_h is the hidden dimension. For each masked superpoint $i \in \mathcal{M}$, the projector is written as

$$\mathbf{z}_i^{(1)} = \sigma(\mathbf{W}_1^\top \mathbf{g}_i + \mathbf{b}_1), \tag{4}$$

$$\mathbf{z}_i^{(2)} = \sigma(\mathbf{W}_2^\top \mathbf{z}_i^{(1)} + \mathbf{b}_2), \tag{5}$$

$$\hat{\mathbf{f}}_i = \mathbf{W}_3^\top \mathbf{z}_i^{(2)} + \mathbf{b}_3 = \phi(\mathbf{g}_i), \tag{6}$$

where $\sigma(\cdot)$ is an element-wise non-linearity. The reconstruction target is the superpoint-averaged 2D feature $\mathbf{f}_i = \frac{1}{|\mathcal{S}_i|}\sum_{j\in\mathcal{S}_i} \mathbf{p}_j$ obtained from the unmasked per-point 2D features. Since the inputs at masked locations are zeroed before the encoder, $\hat{\mathbf{f}}_i$ has to be inferred from the geometry-modulated representation and its surrounding context.

Both feature magnitude and direction are constrained by a reconstruction loss on the masked superpoints,

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \Big[\mathrm{sL1}(\hat{\mathbf{f}}_i, \mathbf{f}_i) + \big(1 - \cos(\hat{\mathbf{f}}_i, \mathbf{f}_i)\big)\Big], \tag{7}$$

where $\mathrm{sL1}(\cdot,\cdot)$ is the smooth ℓ_1 distance and $\cos(\cdot,\cdot)$ is the cosine similarity. This objective forces the encoder to infer missing 2D semantics from geometry-modulated context rather than relying only on visible local cues. The projector ϕ is used only during training and discarded at inference, so no extra cost is incurred for downstream prediction.

3.4 Training Objective

The aggregated superpoint features $\{\mathbf{g}_i\}$ are projected to the decoder channel and fed into the inherited DETR-style query decoder [18, 27]. The base set-prediction loss $\mathcal{L}_{\text{base}}$ is inherited unchanged from the underlying base framework.

The overall objective combines the base loss with the proposed reconstruction loss,

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}}, \tag{8}$$

Here λ_{rec} balances the two terms. $\mathcal{L}_{\text{rec}}$ is applied only at the encoder side and shares no parameters with the decoder. Thus it acts purely as an auxiliary supervision signal that shapes the encoder features while leaving the inherited matching, decoder architecture, and inference path unchanged.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate on ScanNetV2 [3] and ScanNet200 [32], two large-scale 3D indoor scene datasets with dense instance annotations. ScanNetV2 (1,201 training / 312 validation scenes, 18 categories) serves as the primary benchmark for standard 3D instance segmentation evaluation. ScanNet200 (198 instance classes) presents a more demanding evaluation scenario with a pronounced long-tail distribution, where the majority of classes are underrepresented. This imbalanced setting is particularly suited to our method, as the geometry-modulated reconstruction objective is designed to strengthen contextual reasoning and semantic transfer—capabilities that are especially valuable when discriminating rare object categories. We report mean Average Precision (mAP) by averaging precision across IoU thresholds [50%, 95%] at 5% intervals, following standard protocol. We additionally report AP@50 and AP@25 to assess performance at specific localization tolerances. All results are evaluated on the official validation sets.

Implementation Details. We instantiate our method on two baseline frameworks, Relation3D [18] and SegDINO3D [27]. The lifted 2D priors are extracted from the pre-trained DINO-X encoder [29]. For fair comparison, all variants use the same source 2D priors and inherit the original decoder, training schedule, batch size, optimizer, and remaining implementation settings from the corresponding baseline framework. For efficiency, we precompute superpoint-level 2D priors by averaging

Table 1: Comparison on ScanNetV2 validation set. Paradigm: Q = Query-based, NQ = Non-query. Cue: 3D = 3D-only, 2D+3D = 2D priors + 3D geometry. *SegDINO3D rows use identical 2D feature loading with superpoint-level aggregation (not original point-level features), enabling efficient caching. Best and second-best are marked in **bold** and underline.

Method	Year	Paradigm	Cue	mAP ↑	AP@50 ↑	AP@25 ↑
DyCo3D	CVPR21	NQ	3D	35.4	57.6	72.9
SSTNet	ICCV21	NQ	3D	49.4	64.3	74.0
MaskGroup	ICME22	NQ	3D	42.0	63.3	73.3
SoftGroup	CVPR22	NQ	3D	46.0	67.6	78.9
ISBNet	CVPR23	NQ	3D	56.8	73.3	81.3
Mask3D	ICRA23	Q	3D	55.2	73.7	82.9
SPFormer	AAAI23	Q	3D	56.3	73.9	82.9
QueryFormer	ICCV23	Q	3D	56.5	74.2	83.3
Maft	ICCV23	Q	3D	58.4	75.9	84.5
EASE	CVPR24	Q	3D	60.2	77.2	84.2
SphMask	CVPR24	NQ	3D	62.3	79.9	88.2
Relation3D	CVPR25	Q	3D	62.5	80.2	87.0
IKNE	CVPR25	NQ	3D	62.9	81.8	**88.7**
SegDINO3D*	AAAI26	Q	2D+3D	56.7	76.7	85.2
SegDINO3D(Ours)*	-	Q	2D+3D	59.2	79.1	86.2
Relation3D(Ours)	-	Q	2D+3D	**65.0**	**82.8**	88.2

lifted DINO-X features within each superpoint partition and cache them before training. During optimization, the cached superpoint features are mapped back to points according to the superpoint assignments. As analyzed in Figure 3, this design makes lifted 2D priors practical to cache by sharply reducing feature loading time and memory footprint, and all variants that use 2D priors in our experiments follow this same efficient loading protocol. In all experiments, the background masking ratio is set to $r = 0.1$ and the reconstruction loss weight is set to $\lambda_{\text{rec}} = 1.0$. All experiments are conducted on a single NVIDIA RTX 5090 GPU.

Efficiency of Superpoint-Level Representation

A key design choice in our method is to use superpoint-level 2D priors rather than point-level priors. Figure 3 compares the loading time and memory footprint of three input representations on ScanNetV2 scenes. Point-level 2D priors incur the highest overhead, with 263.3 ms loading time and 475.5 MB per scene. In contrast, superpoint-level aggregation reduces loading time to 0.4 ms and memory to 0.8 MB per scene, corresponding to about 658× faster loading and about 594× lower memory usage. This efficiency makes caching lifted 2D priors practical while retaining competitive performance in the main results.

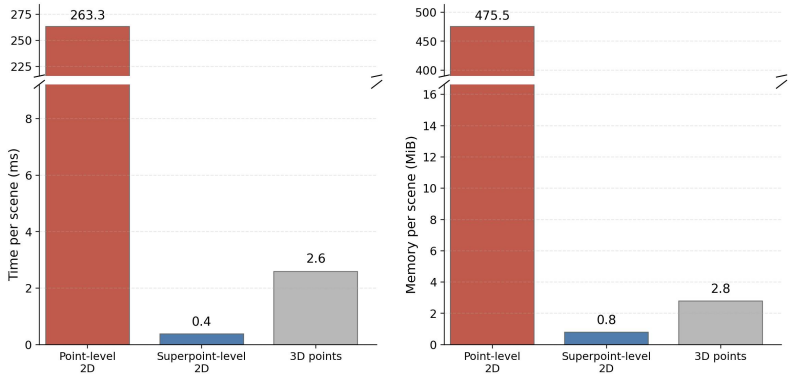

Figure 3: Efficiency of superpoint-level representation. Superpoint-level 2D priors reduce loading time from 263.3 ms to 0.4 ms and memory from 475.5 MB to 0.8 MB per scene, a 600× reduction compared to point-level priors.

4.2 Main Results

Quantitative Results. Tables 1 and 2 compare our method with representative approaches on both benchmarks, organized by paradigm and input cues. On ScanNetV2, our method achieves 65.0% mAP, 82.8% AP@50, and 88.2% AP@25, outperforming Relation3D by 2.5% in mAP and 2.6% in AP@50. The comparison with SegDINO3D is particularly informative: relative to the cached superpoint-level baseline SegDINO3D*, our full Relation3D-based model raises mAP by 8.3% and

Table 2: Comparison on ScanNet200 validation set. Paradigm and Cue definitions same as Table 1. *SegDINO3D rows use identical 2D feature loading with superpoint-level aggregation.

Method	Year	Paradigm	Cue	mAP ↑	AP@50 ↑	AP@25 ↑
SPFormer	AAAI23	Q	3D	25.2	33.8	39.6
Mask3D	ICRA23	Q	3D	27.4	37.0	42.3
QueryFormer	ICCV23	Q	3D	28.1	37.1	43.4
Maft	ICCV23	Q	3D	29.2	38.2	43.3
IKNE	CVPR25	NQ	3D	29.9	38.9	44.9
Relation3D	CVPR25	Q	3D	31.6	41.2	45.6
SegDINO3D*	AAAI26	Q	2D+3D	32.6	44.6	51.7
SegDINO3D (Ours)*	-	Q	2D+3D	<u>35.2</u>	**48.1**	**54.9**
Relation3D (Ours)	-	Q	2D+3D	**36.6**	<u>46.1</u>	<u>52.3</u>

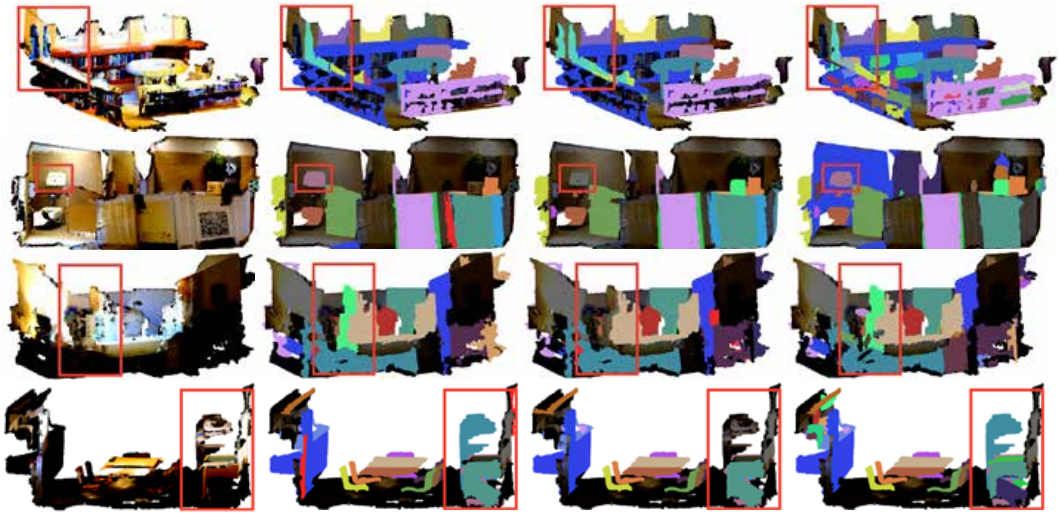

Figure 4: Qualitative comparison on the ScanNetV2 validation set. Our method yields cleaner instance boundaries and more complete instance separation in the highlighted regions, especially under occlusion and around thin or ambiguous structures.

AP@50 by 6.1%, while the SegDINO3D-based variant SegDINO3D(Ours)* also improves over SegDINO3D* by 2.5% mAP and 2.4% AP@50. This suggests that the proposed geometry-modulated injection and reconstruction objective transfers effectively across different sparse 3D backbones, even when using the same efficient superpoint-level 2D priors. Since the decoder, set-prediction losses, and training pipeline remain unchanged from Relation3D, the performance differences are concentrated on the encoder-side design. The consistent gains over both 3D-only and other 2D+3D fusion baselines suggest that geometry-modulated reconstruction is more effective than simple concatenation for integrating 2D priors. Qualitative results in Figure 4 further confirm these improvements, showing cleaner instance boundaries and better separation in challenging regions.

We observe a different pattern on the more challenging ScanNet200 benchmark, where the expanded semantic space places stronger demands on fine-grained appearance discrimination. On ScanNet200, our method improves mAP over Relation3D by 5.0%, reaching 36.6% mAP, 46.1% AP@50, and 52.3% AP@25. Compared with SegDINO3D* under the same superpoint-level 2D feature loading protocol, Relation3D(Ours) improves mAP by 4.0%, AP@50 by 1.5%, and AP@25 by 0.6%, while SegDINO3D(Ours)* improves by 2.6% mAP, 3.5% AP@50, and 3.2% AP@25. This comparison reveals that the proposed geometry-modulated injection and reconstruction objective can deliver consistent gains on the more challenging long-tail benchmark, even when semantic inputs are aggressively compressed for efficient caching. Our method retains a much lighter semantic input pipeline with substantially reduced feature loading time and memory footprint (Figure 3). Compared with IKNE, the method improves by 6.7% in mAP, demonstrating consistent gains across both

benchmarks. The consistent improvements across both frameworks and datasets validate the generality of the modulation-and-reconstruction paradigm across different sparse 3D backbones.

4.3 Ablation Studies

Limited-Data Training. To further validate the robustness of the proposed method, we evaluate performance in low-data regimes where training data is limited. Table 3 reports results on ScanNet200 with 5% and 10% of the training set. On both Relation3D and SegDINO3D frameworks, our method consistently outperforms the baseline across all metrics and data ratios. At 5% training data, our method improves mAP by 1.99% on Relation3D and 6.27% on SegDINO3D. At 10%, the improvements are 3.09% and 9.23% respectively. These results demonstrate that the geometry-modulated reconstruction design is effective in low-data regimes, where explicit supervision for semantic transfer becomes even more valuable.

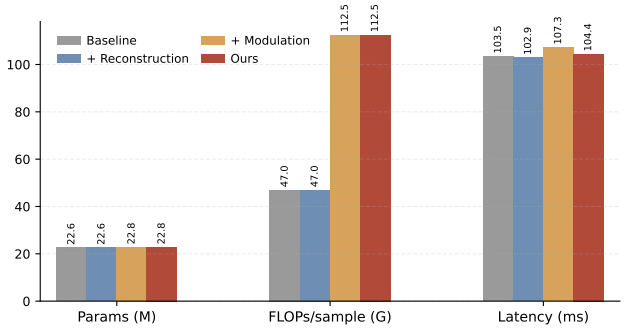

Figure 5: Efficiency comparison on ScanNetV2.

Efficiency analysis. Figure 5 analyzes the computational cost of each component. Reconstruction adds no inference-time parameters or FLOPs since it operates only during training. Modulation adds 0.20M parameters and increases FLOPs from 47.0G to 112.5G, which accounts for the latency increase from 103.5 ms to 107.3 ms. The full model inherits these costs from the modulation path, while reconstruction remains a training-only component with negligible inference overhead.

Table 3: Limited-data training results on ScanNet200. Each framework is trained with 5% or 10% of the training set and evaluated on the full validation set.

Ratio	Framework	mAP ↑	AP@50 ↑	AP@25 ↑
5%	Relation3D	10.29	15.40	18.60
	Relation3D(Ours)	**12.28**	**17.39**	**20.52**
	SegDINO3D*	10.54	16.38	20.43
	SegDINO3D(Ours)*	**16.81**	**24.98**	**29.79**
10%	Relation3D	14.80	20.80	24.40
	Relation3D(Ours)	**17.89**	**24.51**	**28.60**
	SegDINO3D*	13.70	20.57	25.06
	SegDINO3D(Ours)*	**22.93**	**32.43**	**37.09**

Mechanism decomposition. The plain baseline refers to the direct-concatenation variant without PreMod or BG Rec, which already incorporates superpoint-level 2D priors but lacks geometry modulation. PreMod applies geometry-modulated injection at the pre-input stage of the sparse encoder. Table 4 evaluates both components independently. The full configuration achieves the strongest results, raising mAP by 1.3% and AP@50 by 0.7% over the plain baseline. While PreMod alone contributes modestly, the combination with BG Rec demonstrates strong synergy, achieving 1.3% improvement that exceeds the sum of individual gains. Notably, predicting all lifted 2D priors without masking (Table 5) yields only marginal improvements (64.4% mAP vs. 63.7% baseline), confirming that auxiliary supervision alone is insufficient. The key insight is that masking forces the encoder to infer missing semantics from context rather than relying on visible local cues—the synergy between geometry-modulated injection and masked background reconstruction is essential.

Modulation direction. Table 6 compares where modulation is applied. Using 3D features as the main input yields the weakest performance. Switching to 2D priors as the main input raises mAP by 0.6% and AP@50 by 1.2%. Moving modulation to the pre-input stage further improves mAP to 65.0% and AP@50 to 82.8%. This progression reveals that 2D priors are more effective as the primary input stream, with geometry shaping the semantic stream at the earliest stage of sparse encoding.

Masking policy comparison. Table 7 compares three masking strategies. Background masking achieves 65.0% mAP, 82.8% AP@50, and 88.2% AP@25. Random masking shows marginal differences, whereas foreground masking degrades substantially, dropping mAP by 5.9% and AP@50 by 6.5%. This gap reveals that corrupting instance-level features conflicts with the primary objective,

Table 4: Mechanism ablation on ScanNetV2 with $\lambda_{\text{rec}} = 1.0$. BG Rec. denotes background-masked reconstruction with $r = 0.1$.

PreMod	BG Rec.	mAP ↑	AP@50 ↑	AP@25 ↑
✗	✗	63.7	82.1	89.3
✗	✓	64.4	81.7	89.0
✓	✗	64.1	81.3	88.5
✓	✓	65.0	82.8	88.2

Table 5: Auxiliary supervision comparison on ScanNetV2 under direct concatenation. Full reg. predicts all lifted 2D priors without masking. The auxiliary loss weight is set to 1.0.

Aux. supervision	mAP ↑	AP@50 ↑	AP@25 ↑
None	63.7	82.1	89.3
Full reg.	64.4	82.6	89.4

Table 6: Modulation position ablation on ScanNetV2 with $\lambda_{\text{rec}} = 1.0$ and $r = 0.1$.

Input	Stage	mAP ↑	AP@50 ↑	AP@25 ↑
3D	Post.	63.9	80.6	88.6
2D	Post.	64.5	81.8	88.2
2D	Pre.	65.0	82.8	88.2

Table 7: Masking policy comparison at 10% corruption on ScanNetV2 with $\lambda_{\text{rec}} = 1.0$.

Policy	mAP ↑	AP@50 ↑	AP@25 ↑
Background	65.0	82.8	88.2
Random	64.7	82.2	88.3
Foreground	59.1	76.3	84.7

while background masking creates a complementary signal that strengthens contextual reasoning without disturbing foreground cues.

Reconstruction weight sensitivity. Figure 6 (left) varies λ_{rec} with $r = 0.1$. The reconstruction weight exhibits a U-shaped trend with the minimum at 0.5, indicating that both insufficient and excessive weighting degrade performance, with the optimum at $\lambda_{\text{rec}} = 1.0$. This suggests that the reconstruction objective should be weighted comparably to the base segmentation loss.

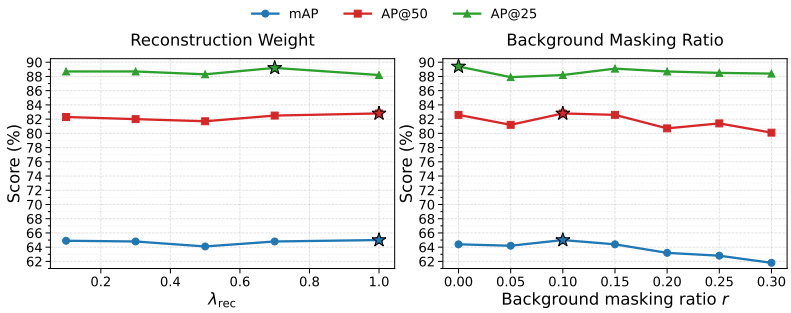

Figure 6: Sensitivity analysis on ScanNetV2. Left: reconstruction weight λ_{rec} with $r = 0.1$. Right: masking ratio r with $\lambda_{\text{rec}} = 1.0$.

Masking ratio sensitivity. Figure 6 (right) varies background masking ratio r with $\lambda_{\text{rec}} = 1.0$. The best performance occurs at $r = 0.1$. This pattern suggests that moderate background masking provides sufficient reconstruction signal to encourage contextual reasoning, while excessive masking removes too much information and degrades the learned representations.

5 Conclusion

This paper addresses the challenge of injecting rich 2D semantics into 3D instance segmentation while maintaining geometric consistency. We demonstrate that passive concatenation of lifted 2D priors is insufficient, and propose a geometry-aware framework combining masked semantic reconstruction and geometry-modulated semantic injection. The masked reconstruction objective provides explicit supervision for semantic transfer, while geometry modulation ensures that 3D structure explicitly regulates how 2D priors enter the encoder. Extensive experiments validate the generality of our approach, achieving state-of-the-art results on both ScanNetV2 and ScanNet200, with consistent improvements in few-shot settings on ScanNet200. These results establish that effective 2D-to-3D semantic transfer requires explicit, geometry-aware distillation rather than simple feature fusion.

Limitations. The method depends on the quality and availability of lifted multi-view 2D priors, and evaluation is centered on indoor RGB-D benchmarks. Extending to noisier 2D priors, outdoor scenes, and more open-world 3D understanding remains an important direction for future work.

References

[1] R. Chen, Y. Liu, L. Kong, X. Zhu, Y. Ma, Y. Li, Y. Hou, Y. Qiao, and W. Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023.

[2] C. Choy, J. Gwak, and S. Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019.

[3] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[4] R. Dong, Z. Qi, L. Zhang, J. Zhang, J. Sun, Z. Ge, L. Yi, and K. Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv preprint arXiv:2212.08320*, 2022.

[5] W. Dou, X. Zhang, Y. Bin, J. Liu, B. Peng, G. Wang, Y. Yang, and H. T. Shen. Geopurify: A data-efficient geometric distillation framework for open-vocabulary 3d segmentation. *arXiv preprint arXiv:2510.02186*, 2025.

[6] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9031–9040, 2020.

[7] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020.

[8] T. He, C. Shen, and A. Van Den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 354–363, 2021.

[9] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[10] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020.

[11] S. Jung, J. Zheng, K. Zhang, N. Qiao, A. Y. Chen, L. Xia, C. Liu, Y. Sun, X. Zeng, H.-W. Huang, et al. Details matter for indoor open-vocabulary 3d instance segmentation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9627–9637. IEEE, 2025.

[12] J. Lahoud, B. Ghanem, M. Pollefeys, and M. R. Oswald. 3d instance segmentation via multi-task metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9256–9266, 2019.

[13] X. Lai, Y. Yuan, R. Chu, Y. Chen, H. Hu, and J. Jia. Mask-attention-free transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3693–3703, 2023.

[14] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018.

[15] R. Li, Z. Zhang, C. He, Z. Ma, V. M. Patel, and L. Zhang. Dense multimodal alignment for open-vocabulary 3d scene understanding. In *European Conference on Computer Vision*, pages 416–434. Springer, 2024.

[16] Z. Li, J. Lu, J. Deng, H. Chang, L. Wu, Y. Liang, and T. Zhang. Sas: Segment any 3d scene with integrated 2d priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8306–8318, 2025.

[17] Z. Liang, Z. Li, S. Xu, M. Tan, and K. Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2783–2792, 2021.

[18] J. Lu and J. Deng. Relation3d: enhancing relation modeling for point cloud instance segmentation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8889–8899. IEEE, 2025.

[19] J. Lu, J. Deng, C. Wang, J. He, and T. Zhang. Query refinement transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18516–18526, 2023.

[20] T. D. Ngo, B.-S. Hua, and K. Nguyen. Isbnet: a 3d point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13550–13559, 2023.

[21] P. Nguyen, M. Luu, A. Tran, C. Pham, and K. Nguyen. Any3dis: Class-agnostic 3d instance segmentation by 2d mask tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3636–3645, 2025.

[22] P. Nguyen, T. D. Ngo, E. Kalogerakis, C. Gan, A. Tran, C. Pham, and K. Nguyen. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4018–4028, 2024.

[23] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 604–621. Springer, 2022.

[24] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023.

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

[26] Z. Qi, R. Dong, G. Fan, Z. Ge, X. Zhang, K. Ma, and L. Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, pages 28223–28243. PMLR, 2023.

[27] J. Qu, H. Li, X. Chen, S. Liu, Y. Shi, T. Ren, R. Jing, and L. Zhang. Segdino3d: 3d instance segmentation empowered by both image-level and object-level 2d features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 8649–8658, 2026.

[28] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[29] T. Ren, Y. Chen, Q. Jiang, Z. Zeng, Y. Xiong, W. Liu, Z. Ma, J. Shen, Y. Gao, X. Jiang, et al. Dino-x: A unified vision model for open-world object detection and understanding. *arXiv preprint arXiv:2411.14347*, 2024.

[30] W. Roh, H. Jung, G. Nam, D. I. Lee, H. Park, S. H. Yoon, J. Joo, and S. Kim. Insightful instance features for 3d instance segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14057–14067, 2025.

[31] W. Roh, H. Jung, G. Nam, J. Yeom, H. Park, S. H. Yoon, and S. Kim. Edge-aware 3d instance segmentation network with intelligent semantic prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20644–20653, 2024.

[32] D. Rozenberszki, O. Litany, and A. Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European conference on computer vision*, pages 125–141. Springer, 2022.

[33] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023.

[34] S. Shin, K. Zhou, M. Vankadari, A. Markham, and N. Trigoni. Spherical mask: Coarse-to-fine 3d point cloud instance segmentation with spherical representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4060–4069, 2024.

[35] M. Soori, B. Arezoo, and R. Dastres. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive robotics*, 3:54–70, 2023.

[36] Y. Su, S. Zhang, X. Jia, and Q. Fan. Pointalign: Feature-level alignment regularization for 3d vision-language models. *arXiv preprint arXiv:2603.00412*, 2026.

[37] J. Sun, C. Qing, J. Tan, and X. Xu. Superpoint transformer for 3d scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2393–2401, 2023.

[38] A. Takmaz, E. Fedele, R. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 68367–68390. Curran Associates, Inc., 2023.

[39] T. Vu, K. Kim, T. M. Luu, T. Nguyen, and C. D. Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2708–2717, 2022.

[40] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024.

[41] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. 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*, pages 19313–19322, 2022.

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

[43] R. Zhang, L. Wang, Y. Qiao, P. Gao, and H. 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*, pages 21769–21780, 2023.

[44] S. Zhang, M. Liu, J. Yan, H. Zhang, L. Huang, X. Yang, and P. Lu. M-mix: Generating hard negatives via multi-sample mixing for contrastive learning. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2461–2470, 2022.

[45] S. Zhang, F. Zhu, J. Yan, R. Zhao, and X. Yang. Zero-cl: Instance and feature decorrelation for negative-free symmetric contrastive learning. In *International Conference on Learning Representations*, 2022.

[46] X. Zhang, S. Zhang, and J. Yan. Pcp-mae: Learning to predict centers for point masked autoencoders. *Advances in Neural Information Processing Systems*, 37:80303–80327, 2024.

[47] M. Zhong, X. Chen, X. Chen, G. Zeng, and Y. Wang. Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022.

[48] M. Zhou, C. He, R. Wang, and X. Chen. Ov3d-cg: Open-vocabulary 3d instance segmentation with contextual guidance. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5305–5314, 2025.

A Additional Experimental Results

A.1 Generalization to ScanNet++

We evaluate Relation3D and Relation3D(Ours) on ScanNet++ using a 100-scene training subset and the complete official 50-scene validation set. Relation3D and Relation3D(Ours) use identical 3D preprocessing, initialization, and training schedules. Table 8 shows that mAP increases from 15.4% for Relation3D to 21.5% for Relation3D(Ours). AP@50 increases from 24.3% to 30.2%, and AP@25 increases from 34.0% to 39.3%. The improvements on ScanNet++ support the effectiveness of our method in scenes with denser geometry.

Table 8: Results on ScanNet++.

Method	mAP	AP@50	AP@25
Relation3D	15.38	24.29	34.03
Relation3D(Ours)	21.46	30.20	39.32

A.2 ScanNet200 Comparisons

Table 9 compares SegDINO3D and SegDINO3D(Ours) on the Head, Common, and Tail categories of ScanNet200. On Head categories, mAP increases from 39.2% to 41.4%, and on Common categories it increases from 29.4% to 30.9%. Tail mAP increases from 28.7% to 33.1%, giving the largest absolute increase among the three groups in this experiment. The larger gain on Tail categories suggests that our method is particularly beneficial for rare categories.

Table 9: Results on the Head, Common, and Tail categories of ScanNet200.

Framework	Head mAP	Common mAP	Tail mAP
SegDINO3D	39.19	29.37	28.70
SegDINO3D(Ours)	41.41	30.92	33.09

Table 10 reports ScanNet200 results for the supervised closed-set model Relation3D(Ours) and the listed open-vocabulary methods. Relation3D(Ours) obtains 36.6% mAP, 46.1% AP@50, and 52.3% AP@25. Details Matter [11] obtains 32.7%, 41.4%, and 45.3%, respectively, the highest scores among the listed open-vocabulary methods.

Table 10: Comparison with open-vocabulary methods on ScanNet200.

Method	Setting	mAP	AP@50	AP@25
OpenMask3D [38]	Open-vocabulary	15.4	19.9	23.1
Open3DIS [22]	Open-vocabulary	23.7	29.4	32.8
OV3D-CG [48]	Open-vocabulary	25.4	32.9	37.0
Details Matter [11]	Open-vocabulary	32.7	41.4	45.3
Relation3D(Ours)	Closed-set	36.6	46.1	52.3

A.3 Training Cost

Table 11 reports average iteration time and total optimization time for both frameworks. Relation3D(Ours) takes 17.75 GPU-hours, compared with 16.01 GPU-hours for Relation3D, under the same 102,912-iteration schedule. SegDINO3D(Ours) takes 28.04 GPU-hours, compared with 29.08 GPU-hours for SegDINO3D, with both trained for 150,000 iterations.

Table 11: Training cost on one NVIDIA RTX 5090.

Framework	Iterations	Time/iter. (s)	GPU-hours
Relation3D	102,912	0.560	16.01
Relation3D(Ours)	102,912	0.621	17.75
SegDINO3D	150,000	0.698	29.08
SegDINO3D(Ours)	150,000	0.673	28.04

A.4 Fusion Operator

Table 12 compares concatenation, addition, and FiLM without reconstruction using Relation3D on ScanNetV2. Addition first linearly projects the 3D features before combining them with the 2D priors. FiLM achieves 64.1% mAP, compared with 63.7% for concatenation and 63.6% for addition. The higher mAP suggests that FiLM provides a more effective fusion strategy than concatenation or addition.

Table 12: Fusion operator comparison without reconstruction on ScanNetV2 using the Relation3D framework.

Fusion	mAP	AP@50	AP@25
Concatenation	63.7	82.1	89.3
Addition	63.6	81.1	87.9
FiLM	64.1	81.3	88.5

A.5 Prior Robustness

We evaluate the robustness of our method to degraded lifted 2D priors, including noisy features and inaccurate 2D-to-3D correspondences. Table 13 reports test-time perturbation results without retraining. With 2D feature noise at $\sigma = 0.10$, mAP remains at 64.9%, compared with 65.0% on clean inputs. To simulate correspondence errors caused by imperfect camera calibration, we locally reassign 10% of lifted features, yielding 64.5% mAP. These results show that our method remains close to its clean-input performance under both perturbations, indicating robustness to feature noise and local correspondence errors.

Table 13: Test-time sensitivity to lifted 2D priors on ScanNetV2. Feature noise uses $\sigma = 0.10$. Simulated camera-calibration error locally reassigns 10% of lifted features. Results are mean $\pm$ standard deviation.

Perturbation	mAP	AP@50	AP@25
2D feature noise	64.93 ± 0.06	82.70 ± 0.10	88.37 ± 0.06
Simulated calibration error	64.47 ± 0.21	82.50 ± 0.20	87.93 ± 0.21

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are stated in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the Conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental details are provided in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The paper uses publicly available benchmarks but does not provide anonymized code at submission time.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test details are specified in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report results from standard benchmarks without error bars from multiple independent runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We report the GPU type (NVIDIA RTX 5090) and efficiency metrics, but do not provide detailed per-experiment training time or total compute hours. This information can be provided upon request.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work uses publicly available benchmarks and does not involve human subjects, proprietary data, or unethical practices. All research follows standard academic integrity and transparency principles.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This is foundational research on 3D scene understanding. While the method could enable beneficial applications in robotics and autonomous systems, the paper does not explicitly discuss broader societal implications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new models or datasets with high misuse risk. We use existing public benchmarks and pre-trained features without introducing novel high-risk assets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original datasets, baseline methods, and pre-trained models used in this work. These existing assets are credited in the paper through their corresponding references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce or release new datasets, models, or code assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects or crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not employed in developing the proposed method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.