Computational Visual Media
https://doi.org/10.26599/CVM.2026.9450531

Research Article

Bridging geometry and semantics for 3D point cloud instance segmentation

Huixiao Tian[1,*], Zhipeng Jiang[1,*], Shimin Song[1], Saishang Zhong[2], Zheng Liu[1] (✉), and Ying He[3]

Abstract 3D instance segmentation is a pivotal yet demanding problem in scene understanding and perception. Existing approaches still fall short of fully exploiting geometric and semantic cues, and the weak coupling between these modalities often leads to over- and under-segmentation artifacts. In this paper, we present a novel framework that couples structural perception with semantic learning to exploit their complementary strengths for 3D instance segmentation. Our framework centers on instance mask prediction and augments it with semantic classification and bounding-box regression as auxiliary objectives. First, a knowledge embedding module initializes instance queries alongside point-level structural features, stabilizing training and accelerating convergence. Second, a two-stage refinement module iteratively updates the instance queries and their associated point features, strengthening the network's ability to align each instance with its constituent points. Finally, a joint mask module fuses geometric and semantic cues, capitalizing on their synergy to improve instance mask accuracy. Extensive evaluation on ScanNetV2, ScanNet200, and S3DIS benchmarks demonstrate that our method achieves state-of-the-art performance for 3D instance segmentation. Furthermore, our underlying architecture generalizes naturally to 3D object detection, where it achieves competitive performance. Source code is available at `https://github.com/tianhuixiao12138/BGS`.

Keywords 3D instance segmentation; semantic segmentation; transformers; point clouds

1 Introduction

With rapid advances in scanning hardware and data acquisition, 3D scene understanding has become a core challenge in computer graphics and 3D vision [1–3]. Point clouds, as an inherent and easily accessible data format, provide a foundation for building and recreating the real world in virtual environments [4, 5]. Following recent advances in 3D perception tasks [6], including 3D object detection and semantic segmentation [7–12], instance segmentation [13] has emerged as a finer-grained scene understanding problem; it requires not only assigning semantic labels to points, but also delineating individual object instances. Fueled by the rapid progress in deep learning [14–16] and recent self-supervised methods [7], learning-based instance segmentation has attracted increasing attention and has been widely adopted in various applications, including digital twins [17–19], autonomous driving [20], robotics [21], and virtual reality [22, 23].

Existing 3D instance segmentation methods broadly fall into grouping-based and transformer-based paradigms. Grouping-based approaches [24–26] segment instances by analyzing gaps between clustered points. However, their performance is

* Huixiao Tian and Zhipeng Jiang contributed equally to this work.

1 School of Computer Science, China University of Geosciences (Wuhan), Wuhan 430074, China. E-mail: H. Tian, tianhuixiao@cug.edu.cn; S. Song, songshimin@cug.edu.cn; Z. Jiang, jiangzhipeng@cug.edu.cn; Z. Liu, liu.zheng.jojo@gmail.com (✉).

2 School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan 430200, China. E-mail: cugsaishang@foxmail.com.

3 College of Computing and Data Science, Nanyang Technological University, Singapore 639798, Singapore. E-mail: yhe@ntu.edu.sg.

Manuscript received: 2025-06-10; accepted: 2026-01-13

highly sensitive to hyper-parameters such as the clustering radius, undermining robustness, and accuracy. Transformer-based approaches [27–33] instead represent objects with instance queries and assign points by matching query and point features. While less dependent on hyper-parameter tuning, many of these approaches refresh instance queries using semantic cues alone. As a result, semantically heterogeneous regions of a single object may be split into multiple instances, while semantically similar but distinct objects may be erroneously merged. Such over- and under-segmentation arise from ignoring the critical role of instance geometric information in boundary differentiation and overall structure perception.

Recent work has attempted to incorporate geometric features into query formulation to address the above challenge. For example, MAFT [34] replaces traditional masked attention with a center regression objective to accelerate convergence. However, the geometric cue it introduces is too coarse to capture object structure, limiting the model's ability to perceive instances accurately. MSTA3D [35] mitigates over-segmentation by use of dual-attention and bounding-box queries, yet the boxes remain auxiliary and interact weakly with semantic features, leaving the model susceptible to under-segmentation when box estimates drift.

To address these challenges, we propose a collaborative framework that jointly models structural perception and semantic learning. Rather than relegating geometry to an auxiliary role, we treat structural perception and semantic learning as coupled yet distinct tasks, which provides more robust instance boundary localization through an interactive compensation mechanism. To do so, we first use a knowledge embedding module that filters non-instance points via semantic logits and uses learnable offsets to generate pointwise bounding boxes, yielding instance-focused queries that capture instance-specific structures. Next, a two-stage refinement module progressively optimizes the instance queries and their associated point features, narrowing the representational gap between them. Finally, we mitigate over- and under-segmentation by coupling geometric and semantic cues within a geometric–semantic mask module, allowing mutual constraints and compensation to improve segmentation performance.

Our main contributions are in summary:

- a collaborative 3D instance segmentation framework that exploits the complementary strengths of structural perception and semantic learning to suppress over- and under-segmentation artifacts,
- a dual-branch knowledge embedding module in which one branch produces high-quality initial queries and the other estimates instance-level structural features, accelerating convergence during training,
- a two-stage refinement module that iteratively updates instance queries and point features, allowing the network to align instances with their constituent points via mutual optimization, and
- a joint mask prediction module that leverages similarities in both semantic and geometric features to produce more accurate instance masks.

2 Related work

Current techniques for 3D instance segmentation can be roughly classified into four categories: proposal-based, grouping-based, dynamic convolution-based, and transformer-based methods.

Proposal-based methods typically follow a top-down paradigm. 3D-SIS [36] first localizes instances by regressing bounding boxes and then refines the points enclosed by each box to produce instance masks. 3D-BoNet [37] detects instances by aligning predicted boxes with ground truth via Hungarian matching, and then segments the foreground points inside each box to recover the instance masks. The effectiveness of proposal-based approaches is significantly influenced by the accuracy of the predicted bounding boxes. However, the inherent complexity and non-uniform distribution of raw point clouds make it challenging to predict stable bounding boxes for instances.

In contrast, grouping-based approaches follow a bottom–up paradigm, where scene points are first aggregated into clusters that are subsequently refined to produce instance masks. PointGroup [24] predicts per-point offsets from corresponding instance centers, facilitating clustering and improving mask accuracy. HAIS [25] introduces a hierarchical clustering mechanism that progressively merges smaller clusters belonging to the same instance, while a dedicated mask loss further refines the predicted

masks. SoftGroup [26] adopts a soft classification strategy that allows each point to be associated with multiple semantic categories during clustering, effectively reducing the adverse impact of semantic misclassifications on instance segmentation. Although grouping-based methods have shown strong ability to generate instance masks, their robustness is often limited by the high computational overhead of clustering and sensitivity to hyperparameters such as the clustering radius.

Dynamic convolution-based methods adopt a distinct strategy by generating instance-specific convolutional kernels and computing similarity maps between these kernels and point features to produce instance masks. DyCo3D [38] employs the clustering mechanism from PointGroup [24] to generate discriminative kernels, but its performance is limited by the inherent drawbacks of clustering-based instance grouping. ISBNet [39] replaces the clustering step with a foreground point sampling strategy, effectively improving kernel generation and segmentation accuracy. DKNet [40] further enhances instance representation via a mining algorithm that selects and progressively aggregates candidate points, embedding instance information into dynamic kernels. Despite their effectiveness, dynamic convolution-based methods typically rely on locally generated kernels, restricting their ability to capture long-range dependencies and accurately delineate large or spatially extended instances.

Recently, transformer-based methods have made remarkable progress in 3D instance segmentation. Mask3D [31] uses multiscale features and cross-attention to iteratively refine instance queries and generate masks through similarity maps between queries and point features. SPFormer [32] introduced structured superpoint features to reduce computational costs while maintaining segmentation accuracy. MAFT adds an auxiliary center regression task to address convergence challenges caused by low-quality initial instance queries. OneFormer3D [33] adopts disentangled bipartite matching to align predicted masks with ground truth, reducing instability associated with Hungarian matching. MSTA3D incorporates box queries as structural guidance alongside semantic queries to enhance query representations.

Despite their state-of-the-art performance, these transformer-based approaches often overlook the intrinsic complementarity between semantic cues and geometric structures, which may lead to over- or under-segmentation artifacts.

3 Method

This section begins with an overview of the proposed network architecture. Next, the core modules of the network are described in detail, followed by the formulation of the multitask loss function.

3.1 Architecture

Figure 1 illustrates the overall architecture, which comprises a feature backbone (■), a knowledge embedding module (■), and a query decoder (■) consisting of a feature and query refinement module (■) and a geometric-semantic mask module (■). Given an input point cloud $\mathrm{P} \in \mathbb{R}^{N\times 6}$ containing coordinates and color values, where N is the number of points, a sparse U-Net first computes point features $\mathrm{F}^0 \in \mathbb{R}^{N\times d}$, where d denotes the feature dimensionality and is set to 256. The F^0 are

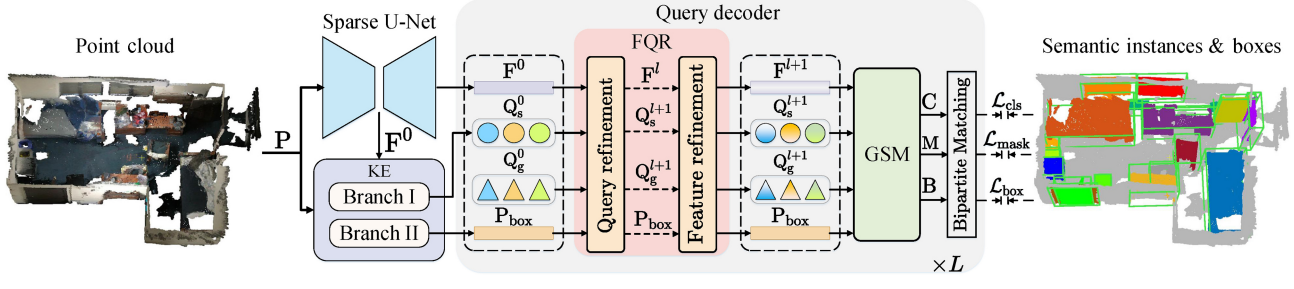

Fig. 1 Our 3D instance segmentation framework. Starting from a scene point cloud P, the network predicts instance masks M, semantic labels C, and bounding boxes B. A sparse U-Net first extracts point features F^0. Together with the coordinates of P, these features are processed by the knowledge embedding (KE) module, which consists of two branches, to produce initial semantic queries Q_s^0 and pointwise box predictions P_box. Subsequently, the decoder performs L iterative refinements, each using two components. The feature–query refinement (FQR) module updates the point features F^{l+1} and the semantic and geometric queries $\mathrm{Q}_\mathrm{s}^{l+1}$ and $\mathrm{Q}_\mathrm{g}^{l+1}$. The geometric–semantic mask (GSM) module then exploits the complementary nature of geometric and semantic cues to produce the corresponding outputs.

then fed, along with coordinates of P, into the knowledge embedding module to generate initial semantic queries $\mathrm{Q}_\mathrm{s}^0 \in \mathbb{R}^{K\times d}$ and pointwise bounding boxes $\mathrm{P}_\mathrm{box} \in \mathbb{R}^{N\times 9}$, where K denotes the number of instances. With these features in hand, the query decoding process can be performed. First, the semantic queries Q_s^0, geometric queries $\mathrm{Q}_\mathrm{g}^0 \in \mathbb{R}^{K\times 9}$ and point features F^0 are iteratively refined within the feature and query refinement module. Here, Q_g^0 is randomly initialized through learnable parameters. Then, the refined features, queries, along with P_box, are fed into the geometric-semantic mask module to generate the instance masks $\mathrm{M} \in \mathbb{R}^{K\times N}$. This query decoding process is executed for L iterations to obtain more accurate results.

3.2 Knowledge embedding module

3.2.1 Overview

Our knowledge embedding (KE) module uses scene-specific priors to stabilize training and accelerate convergence. As Fig. 2 shows, the KE module comprises two branches—semantic filtering and structural perception—that operate on the point features F^0 and coordinates $\mathcal{P} = \{p_i\}_{i=1}^N \in \mathbb{R}^{N\times 3}$. These branches yield high-quality initial semantic queries Q_s^0 and per-point bounding boxes P_box, computed as

$$\{\mathrm{Q}_\mathrm{s}^0, \mathrm{P}_\mathrm{box}\} \leftarrow \mathrm{KE}\left(\mathrm{F}^0, \mathcal{P}\right) \tag{1}$$

3.2.2 Branch I: Semantic filtering

The point features F^0 are fed into a semantic head to predict semantic logits $\mathcal{S} \in \mathbb{R}^{N\times C}$, where C represents the number of semantic categories.

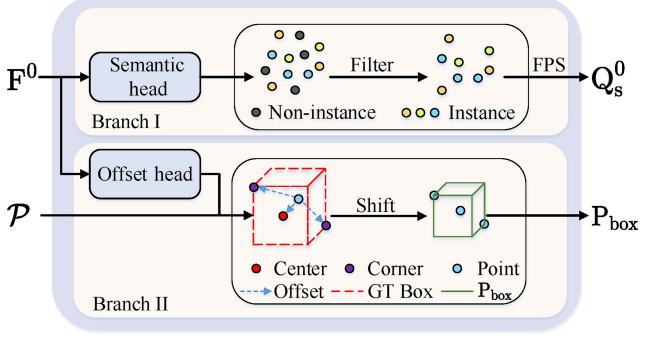

Fig. 2 The knowledge embedding (KE) framework consists of two branches: one for semantic filtering and one for structural perception. The initial point features F^0 are fed into the semantic head to filter out non-instance points, followed by farthest point sampling (FPS) to generate initial semantic queries Q_s^0. Simultaneously, the F^0 are fed into the offset head to predict offsets, which are used to shift original coordinates $\mathcal{P}$ to produce pointwise bounding boxes P_box.

Farthest point sampling (FPS) then selects K representative points from the filtered set and their features serve as the initial semantic queries Q_s^0, as shown in Fig. 2(above). The semantic branch is trained with a cross-entropy loss,

$$\mathcal{L}_\mathrm{sem} = \mathrm{CE}(\mathcal{S}, \hat{\mathcal{S}}) \tag{2}$$

where $\hat{\mathcal{S}}$ are the ground-truth pointwise labels. As a result, semantic queries Q_s^0, obtained through semantic filtering, focus more on instances rather than non-instance points.

3.2.3 Branch II: Structure perception

In this branch, we predict the offsets $\mathrm{O} = \{o_i \in \mathbb{R}^{1\times 9}\}_{i=1}^N \in \mathbb{R}^{N\times 9}$ for each point with respect to the center, top-left corner, and bottom-right corner of its instance bounding box, as shown in Fig. 2(below). Adding these offsets to the original coordinates $\mathcal{P}$ yields the pointwise boxes P_box as Eq. (3):

$$\mathrm{P}_\mathrm{box} = \mathrm{Concat}(\mathrm{P}_\mathrm{center}, \mathrm{P}_{\mathrm{corner}_1}, \mathrm{P}_{\mathrm{corner}_2}) \tag{3}$$

where $\mathrm{P}_\mathrm{center} = \mathcal{P} + \mathrm{O}[:, 0:3]$, $\mathrm{P}_{\mathrm{corner}_1} = \mathcal{P} + \mathrm{O}[:, 3:6]$, $\mathrm{P}_{\mathrm{corner}_2} = \mathcal{P} + \mathrm{O}[:, 6:9]$, and $\mathrm{Concat}(\cdot)$ denotes the concatenation operation. Then, offset prediction is supervised with an ℓ_1 loss

$$\mathcal{L}_\mathrm{offset} = \sum_{i=1}^{N} \mathbb{1}_{\{p_i\}} \|o_i - \hat{o}_i\|_1 / \sum_{i=1}^{N} \mathbb{1}_{\{p_i\}} \tag{4}$$

where $\hat{\mathrm{O}} = \{\hat{o}_i\}_{i=1}^N$ are the ground-truth offsets to the corresponding instance centers and corners, and $\mathbb{1}_{\{p_i\}}$ is an indicator function that specifies whether point p_i belongs to an instance.

Note that the pointwise boxes P_box encode each point's offsets to the center and corners of its corresponding instance box, thereby capturing its relative position within the instance. Analogous to how pointwise semantic features aggregate local semantic cues, these offsets provide each point with structural context. As a result, points belonging to the same instance share consistent geometric properties, which in turn promotes robust instance segmentation.

3.3 Feature and query refinement module

3.3.1 Overview

Our refinement module iteratively updates instance queries and point features to strengthen their mutual dependency. In addition, geometric queries are introduced to encode localization cues for each instance, enriching the geometric context and enabling queries to more accurately attend to points belonging to the same object. As Fig. 3 shows,

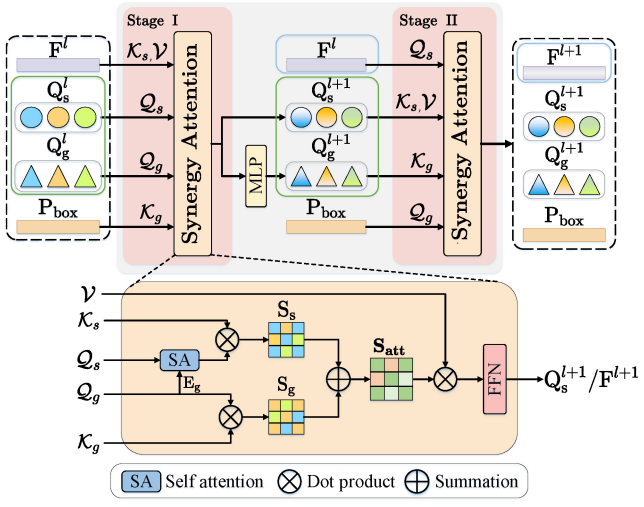

Fig. 3 Feature and query refinement (FQR). Semantic and geometric queries Q_s^l, Q_g^l, point features F^l, and bounding boxes P_box are alternately utilized as sources for $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ to refine the input features through our synergic attention (below). In stage I, $\mathrm{Q}_\mathrm{s}^{l+1}$ and $\mathrm{Q}_\mathrm{g}^{l+1}$ are obtained using F^l and P_box; in stage II, F^{l+1} are updated using refined queries $\mathrm{Q}_\mathrm{s}^{l+1}$, $\mathrm{Q}_\mathrm{g}^{l+1}$, and P_box.

our refinement module operates in two stages: first, queries Q_s^l and Q_g^l are updated; later, point features F^l are refined. Throughout this process, the pointwise bounding boxes P_box serve as auxiliary geometric features. The process is in outline:

$$\{\mathrm{F}^{l+1}, \mathrm{Q}_\mathrm{s}^{l+1}, \mathrm{Q}_\mathrm{g}^{l+1}, \mathrm{P}_\mathrm{box}\} \leftarrow \mathrm{FQR}\left(\mathrm{F}^l, \mathrm{Q}_\mathrm{s}^l, \mathrm{Q}_\mathrm{g}^l, \mathrm{P}_\mathrm{box}\right) \tag{5}$$

3.3.2 Stage I: Query refinement

To establish mappings between instance queries and point features, we use a synergic attention mechanism. This mechanism adaptively computes similarity scores between instance queries and point features, allowing the network to emphasize features relevant to each query while suppressing unrelated ones. The definitions of $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ are as Eq. (6):

$$\begin{cases} \{\mathcal{Q}_\mathrm{s}, \mathcal{K}_\mathrm{s}, \mathcal{V}\} = \left\{\psi(\mathrm{Q}_\mathrm{s}^l), \psi(\mathrm{F}^l), \psi(\mathrm{F}^l)\right\} \\ \{\mathcal{Q}_\mathrm{g}, \mathcal{K}_\mathrm{g}, \mathcal{V}\} = \left\{\psi(\mathrm{Q}_\mathrm{g}^l), \psi(\mathrm{P}_\mathrm{box}), \psi(\mathrm{F}^l)\right\} \end{cases} \tag{6}$$

where $\mathrm{Q}_\mathrm{g}^l \in \mathbb{R}^{K\times 9}$ are initialized randomly as learnable queries, and represent the queries for instance bounding boxes. The second dimension corresponds to the centers of the instances and the two corner points, and $\psi(\cdot)$ denotes a linear mapping.

To facilitate the interaction of semantic queries between different instances, we employ self-attention, denoted $\mathrm{SA}(\cdot)$, and introduce positional embedding $\mathrm{E}_\mathrm{g} \in \mathbb{R}^{K\times d}$ to encode instance location information as Eq. (7):

$$\mathcal{Q}_\mathrm{s}' = \mathrm{SA}(\mathcal{Q}_\mathrm{s} + \mathrm{E}_\mathrm{g}) \tag{7}$$

where E_g is generated from $\mathcal{Q}_\mathrm{g}$ by Fourier transform and linear mapping.

Subsequently, we use $\mathcal{Q}$ and $\mathcal{K}$ in Eqs. (6) and (7) to compute the semantic and geometric attention scores, S_s and S_g, respectively. These scores are then aggregated using a weighted sum to derive the attention score S_att, as Eqs. (8)–(10):

$$\mathrm{S}_\mathrm{s} = \mathrm{Softmax}(\mathcal{Q}_\mathrm{s}'\mathcal{K}_\mathrm{s}^\mathrm{T}/\sqrt{d}) \tag{8}$$

$$\mathrm{S}_\mathrm{g} = \mathrm{Softmax}(\mathcal{Q}_\mathrm{g}\mathcal{K}_\mathrm{g}^\mathrm{T}/\sqrt{d}) \tag{9}$$

$$\mathrm{S}_\mathrm{att} = w_\mathrm{s}\mathrm{S}_\mathrm{s} + w_\mathrm{g}\mathrm{S}_\mathrm{g} \tag{10}$$

where w_s and w_g represent learnable weights.

Next, we use S_att, derived from the interaction between point features and instance queries, and $\mathcal{V}$ to update the queries, as Eqs. (11) and (12):

$$\mathrm{Q}_\mathrm{s}^{l+1} = \mathrm{FFN}((\mathrm{S}_\mathrm{att})\mathcal{V}) \tag{11}$$

$$\mathrm{Q}_\mathrm{g}^{l+1} = \mathrm{MLP}(\mathrm{Q}_\mathrm{s}^{l+1}) \tag{12}$$

where $\mathrm{Q}_\mathrm{s}^{l+1}$ and $\mathrm{Q}_\mathrm{g}^{l+1}$ represent the refined queries, FFN denotes a feed-forward network, and MLP refers to a multilayer perceptron.

3.3.3 Stage II: Feature refinement

To ensure that the point features adapt to the changes introduced by the updated instance queries $\mathrm{Q}_\mathrm{s}^{l+1}$ and $\mathrm{Q}_\mathrm{g}^{l+1}$, we propose a feature refinement stage to further enhance the point features based on the updated queries. The process of this stage closely resembles that of query refinement (Stage I), with the primary distinction lying in the sources of $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$, as Eqs. (13) and (14):

$$\{\mathcal{Q}_\mathrm{s}, \mathcal{K}_\mathrm{s}, \mathcal{V}\} = \left\{\psi(\mathrm{F}^\mathrm{l}), \psi(\mathrm{Q}_\mathrm{s}^{l+1}), \psi(\mathrm{Q}_\mathrm{s}^{l+1})\right\} \tag{13}$$

$$\{\mathcal{Q}_\mathrm{g}, \mathcal{K}_\mathrm{g}, \mathcal{V}\} = \left\{\psi(\mathrm{P}_\mathrm{box}), \psi(\mathrm{Q}_\mathrm{g}^{l+1}), \psi(\mathrm{Q}_\mathrm{s}^{l+1})\right\} \tag{14}$$

Similarly to Eq. (11), the updated point features F^{l+1} are computed as

$$\mathrm{F}^{l+1} = \mathrm{FFN}((w_\mathrm{s}\mathrm{S}_\mathrm{s} + w_\mathrm{g}\mathrm{S}_\mathrm{g})\mathcal{V}) \tag{15}$$

3.4 Geometric-semantic mask module

3.4.1 Overview

As Fig. 4 shows, given updated point features F^{l+1} and refined instance queries $\mathrm{Q}_\mathrm{s}^{l+1}$ and $\mathrm{Q}_\mathrm{g}^{l+1}$, we introduce two auxiliary tasks to support the mask prediction task: one for predicting instance semantics $\mathrm{C} \in \mathbb{R}^{K\times N}$ and another for inferring bounding boxes $\mathrm{B} \in \mathbb{R}^{K\times 9}$. This process is formulated as

$$\{\mathrm{M}, \mathrm{C}, \mathrm{B}\} \leftarrow \mathrm{GSM}\left(\mathrm{F}^{l+1}, \mathrm{Q}_\mathrm{s}^{l+1}, \mathrm{Q}_\mathrm{g}^{l+1}, \mathrm{P}_\mathrm{box}\right) \tag{16}$$

To begin, we compute the geometric similarity

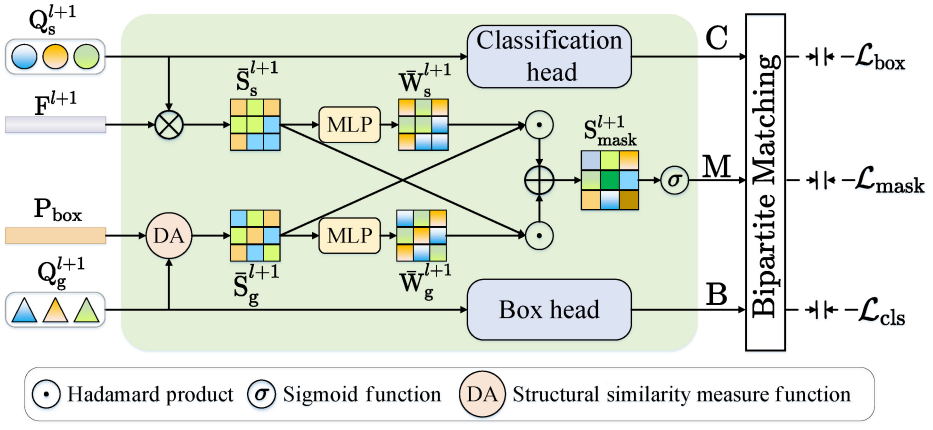

Fig. 4 Geometric-semantic mask (GSM) module. Refined queries $\mathrm{Q}_{\mathrm{s}}^{l+1}$, $\mathrm{Q}_{\mathrm{g}}^{l+1}$ are used to generate semantic and geometric similarity scores $\bar{\mathrm{S}}_{\mathrm{s}}^{l+1}$, $\bar{\mathrm{S}}_{\mathrm{g}}^{l+1}$, respectively. These scores are then fused to produce the final mask scores $\mathrm{S}_{\mathrm{mask}}^{l+1}$ for instance mask prediction. Simultaneously, the semantic label and bounding box for each instance are inferred.

scores between geometric queries and individual bounding boxes as

$$\bar{\mathrm{S}}_{\mathrm{g}}^{l+1} = \mathrm{DA}(\mathrm{Q}_{\mathrm{g}}^{l+1}, \mathrm{P}_{\mathrm{box}}) \tag{17}$$

where $\mathrm{DA}(\cdot)$ denotes the structural similarity function, which is described later. Next, the semantic similarity scores between the semantic instances and point features are obtained using the dot product:

$$\bar{\mathrm{S}}_{\mathrm{s}}^{l+1} = \mathrm{Q}_{\mathrm{s}}^{l+1} \cdot \mathrm{F}^{l+1} \tag{18}$$

With the geometric and semantic scores obtained from Eqs. (17) and (18), we can calculate the instance mask score leveraging the mutual compensation between the geometry and semantics as Eq. (19):

$$\mathrm{S}_{\mathrm{mask}}^{l+1} = \bar{\mathrm{W}}_{\mathrm{s}}^{l+1} \odot \bar{\mathrm{S}}_{\mathrm{g}}^{l+1} + \bar{\mathrm{W}}_{\mathrm{g}}^{l+1} \odot \bar{\mathrm{S}}_{\mathrm{s}}^{l+1} \tag{19}$$

where $\bar{\mathrm{W}}_{\mathrm{g}}^{l+1} \in \mathbb{R}^{K\times N}$ and $\bar{\mathrm{W}}_{\mathrm{s}}^{l+1} \in \mathbb{R}^{K\times N}$ are learnable weights, and $\odot$ is the Hadamard product. After that, the final masks are obtained as Eq. (20):

$$\mathrm{M} = [\sigma(\mathrm{S}_{\mathrm{mask}}^{l+1})_{k,i} > 0.5] \in \{0,1\}^{K\times N} \tag{20}$$

where $\sigma(\cdot)$ denotes the sigmoid function.

3.4.2 DA function

We propose a structural similarity metric, denoted DA, to ensure that points belonging to the same instance exhibit consistent structural characteristics. For any given point within an instance, the centers of its pointwise bounding box and the corresponding instance query are spatially close. Moreover, a point located near the instance center tends to form larger angles with the two corner points of the bounding box, whereas points outside the instance yield smaller angles. Based on these observations, we define the distance factor $\mathrm{D} = \{d_{k,i}\} \in \mathbb{R}^{K\times N}$ and angle factor $\mathrm{A} = \{a_{k,i}\} \in \mathbb{R}^{K\times N}$as Eq. (21):

$$d_{k,i} = \|\tilde{q}_k - \tilde{p}_i\|, \quad a_{k,i} = \frac{v_{ki}^1 \cdot v_{ki}^2}{\|v_{ki}^1\|_2 \, \|v_{ki}^2\|_2} \tag{21}$$

Here, $\tilde{p}_i$ is the center of the pointwise bounding box in Eq. (3), and $\tilde{q}_k$ is the center of the instance bounding box, as shown in Fig. 5. v_{ki}^1 and v_{ki}^2 denote the vectors from $\tilde{p}_i$ to the upper left and bottom right corners of the instance bounding box. To adaptively balance the contributions of D and A, the DA metric is formulated as

$$\mathrm{DA}(\mathrm{Q}_{\mathrm{g}}^{l+1}, \mathrm{P}_{\mathrm{box}}) = \bar{w}_{\mathrm{D}}\mathrm{D} + \bar{w}_{\mathrm{A}}\mathrm{A} \tag{22}$$

where $\bar{w}_{\mathrm{D}}$ and $\bar{w}_{\mathrm{A}}$ are learnable weights that adjust the scaling between the two factors.

The DA function measures the distance between each point and its corresponding instance query center through a distance factor, which captures the relative geometric position of the point within the instance. Unlike the Euclidean distance, which measures only straight-line proximity, this metric merely indicates how close two points are in

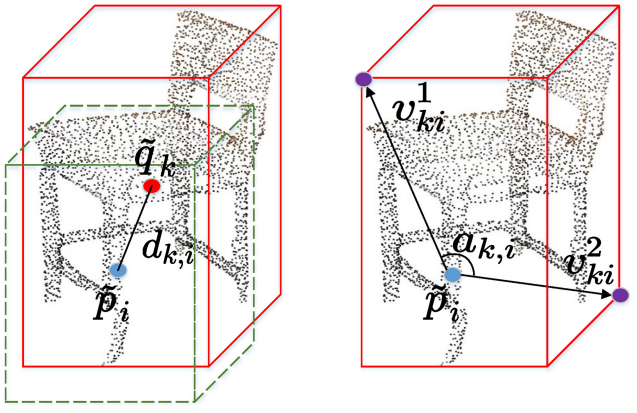

Fig. 5 DA metric is based on a distance factor (left) and an angle factor (right).

space and cannot distinguish whether a point lies inside, on the boundary, or outside an instance. Moreover, it fails to capture directional relationships or structural characteristics, making it ineffective for accurately identifying boundary points. To address these limitations, we introduce an angular factor that determines point membership by analyzing the geometric angles formed between each point and the vertices of the corresponding instance bounding box. This combined metric is inherently insensitive to variations in instance size and shape, while effectively distinguishing boundary points. In summary, by jointly modeling relative position and geometric angle, the proposed similarity metric provides a comprehensive representation of the spatial relationship between points and instances, offering robust geometric support for accurate instance segmentation.

3.4.3 Bipartite matching

Since there is no inherent order between the ground-truth instances and the predicted instances, we follow prior work [31, 32, 39] and employ bipartite matching to establish correspondences between the two sets. To achieve this, we construct a cost matrix $\mathcal{C} \in \mathbb{R}^{K \times \hat{K}}$, where $\hat{K}$ is the number of ground-truth instances. The matching cost in $\mathcal{C}$ for a predicted instance with index k and a ground-truth instance with index $\hat{k}$ is calculated as Eq. (23):

$$\begin{aligned}\mathcal{C}(k,\hat{k}) =& \lambda_{\text{cls}}\mathcal{C}_{\text{cls}}(k,\hat{k}) + \lambda_{\text{dice}}\mathcal{C}_{\text{dice}}(k,\hat{k}) \\ &+ \lambda_{\text{bce}}\mathcal{C}_{\text{bce}}(k,\hat{k}) + \lambda_{\text{center}}\mathcal{C}_{\text{center}}(k,\hat{k})\end{aligned} \tag{23}$$

With the constructed matching cost matrix, we apply the Hungarian algorithm [41] to efficiently determine the optimal correspondences between predicted results and ground truth. Once the correspondences are established, the multiple tasks can be optimized as Eqs. (24)–(26):

$$\mathcal{L}_{\text{cls}} = \lambda_{\text{cls}}\text{CE}(\text{C},\hat{\text{C}}) \tag{24}$$

$$\mathcal{L}_{\text{mask}} = \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}(\text{M},\hat{\text{M}}) + \lambda_{\text{bce}}\mathcal{L}_{\text{bce}}(\text{M},\hat{\text{M}}) \tag{25}$$

$$\mathcal{L}_{\text{box}} = \lambda_{\text{center}}\mathcal{L}_1(\text{B}_{\text{center}},\hat{\text{B}}_{\text{center}}) + \lambda_{\text{IoU}}\mathcal{L}_{\text{IoU}}(\text{B},\hat{\text{B}}) \tag{26}$$

where $\mathcal{L}_{\text{bce}}$, $\mathcal{L}_1$, $\mathcal{L}_{\text{dice}}$, and $\mathcal{L}_{\text{IoU}}$ denote the binary cross-entropy loss, L_1 norm loss, Dice loss [42], and gIoU loss [43], respectively. B_{center} denotes the centers of the predicted bounding boxes B. $\hat{\text{C}}$, $\hat{\text{B}}_{\text{center}}$, $\hat{\text{B}}$, and $\hat{\text{M}}$ represent the corresponding ground truth, respectively. The matching cost and loss weights are denoted by λ_{cls}, λ_{dice}, λ_{bce}, λ_{center}, and λ_{IoU}, respectively.

3.5 Multi-task learning

The entire network is trained with a multi-task loss:

$$\mathcal{L} = \sum_{l=1}^{L}\left(\mathcal{L}_{\text{cls}}^{l} + \mathcal{L}_{\text{box}}^{l} + \mathcal{L}_{\text{mask}}^{l}\right) + \mathcal{L}_{\text{sem}} + \mathcal{L}_{\text{offset}} \tag{27}$$

where L denotes the number of decoder layers.

4 Experiments

In this section, we describe comprehensive experiments to evaluate our method using diverse benchmarks. Section 4.1 details the experimental setup; Sections 4.2 and 4.3 present quantitative and qualitative comparisons; Section 4.4 reports ablation studies; and Section 4.5 discusses generalization performance.

4.1 Experimental setting

4.1.1 Datasets

We evaluated our approach using the standard data splits of several challenging benchmarks, including ScanNetV2 [44], ScanNet200 [45], and S3DIS [46]. ScanNetV2 contains 1613 annotated indoor scans covering hundreds of reconstructed scenes, with both semantic and instance-level annotations in 20 semantic and 18 instance categories. The dataset is split into 1201 scans for training, 312 for validation, and 100 for testing. ScanNet200 extends ScanNetV2 with finer-grained annotations, introducing 200 semantic categories and 198 instance classes to enable more detailed segmentation of complex indoor environments. S3DIS consists of 271 indoor scenes in six areas, annotated with 13 semantic classes, including furniture and structural elements. Area 5 is used for testing, while the remaining areas serve as the training set.

4.1.2 Evaluation metrics

We adopt average precision (AP) as the primary metric to evaluate segmentation accuracy. For instance segmentation, we employ mean average precision (mAP) along with AP_{50} and AP_{25} as evaluation criteria. mAP represents the average precision score calculated with a step size of 5% for intersection over union (IoU) thresholds ranging from 50% to 95%. AP_{50} and AP_{25} correspond to precision scores at fixed IoU thresholds of 50% and

25%, respectively. For object detection, we use box AP as the metric, which measures the average precision of 3D axis-aligned bounding box predictions.

4.1.3 Implementation details

We conduct training on ScanNetV2 and ScanNet200 using an RTX 4090 GPU, and on S3DIS using an A100 GPU. During training, data augmentation is applied by randomly cropping each scene, with the maximum point count capped at 250,000. To enforce this limit, scenes are iteratively cropped using cubic windows. For optimization, we adopt the AdamW optimizer with a learning rate of 0.0001 and a weight decay of 0.05. The batch size is set to 4 in all datasets. Training on the ScanNet datasets requires approximately 38 h on an RTX 4090 GPU, while training on S3DIS takes about 54 h on an NVIDIA A100 GPU for one complete training cycle. The model backbone is similar to SPFormer and generates per-point features with a dimension of 32. The matching cost and loss weights (λ_{cls}, λ_{dice}, λ_{bce}, λ_{center}, λ_{IoU}) are configured as (0.5, 1.0, 1.0, 0.5, 0.5) for ScanNet and ScanNet200, respectively; and for S3DIS, the matching cost and loss weights are configured as (2.0, 5.0, 1.0, 0.5, 0.5). The voxel size is set at 0.02 m for ScanNet and ScanNet200, and at 0.05 m for S3DIS.

4.2 Quantitative results

We have performed numerical comparisons on the benchmark datasets to verify the superiority of our method over state-of-the-art approaches.

4.2.1 ScanNetV2

We record results of the quantitative comparison on the ScanNetV2 validation set in Table 1. As we can see, our method achieves the highest performance in terms of mAP, AP_{50}, and AP_{25}, demonstrating its proficiency in capturing fine details and structures, as well as producing more precise instance masks. Specifically, our results improve the scores by +1.3 mAP and +2.2 AP_{50} compared to MAFT, and by +1.3 mAP, +1.1 AP_{50}, and +1.1 AP_{25} over MSTA3D. In addition, as shown in Table 2, our approach outperforms the competing methods in 12 out of the 18 categories, further validating its superior capability in fine-grained category segmentation.

Table 1 Quantitative comparison on ScanNetV2 with mAP, AP_{50}, and AP_{25} metrics

Method	mAP	AP_{50}	AP_{25}
MTML [47]	20.3	40.2	55.4
3D-MPA [48]	35.5	59.1	72.4
PointGroup [24]	34.8	56.7	71.3
OccuSeg [49]	44.2	60.7	71.9
DyCo3D [38]	35.4	57.6	72.9
HAIS [25]	43.5	64.1	75.6
SSTNet [50]	49.4	64.3	74.0
SoftGroup [26]	46.0	67.6	78.9
DKNet [40]	50.8	66.7	76.9
ISBNet [39]	54.5	73.1	83.5
Mask3D [31]	55.2	73.7	83.5
ISES [51]	56.1	75.0	83.7
SPFormer [32]	56.3	73.9	82.9
QueryFormer [52]	56.5	74.2	83.3
MAFT [34]	58.4	75.9	–
MSTA3D [35]	58.4	77.0	85.4
OneFormer3D[33]	59.3	**78.1**	–
Sonata [7]	42.4	63.9	79.2
Ours	**59.7**	**78.1**	**86.5**

4.2.2 ScanNet200

We compare results of our method to those of competing approaches, on the ScanNet200 validation

Table 2 Results on ScanNetV2 with AP_{50} metric. Per-category and average scores are reported. The best and second-best performances are highlighted in bold and underlined, respectively

Method	Bathtub	Bed	Bkshelf	Cabinet	Chair	Counter	Curtain	Desk	Door	Picture	Fridge	S. curtain	Sink	Sofa	Table	Toilet	Window	Other	AP_{50}
MTML [47]	70.8	54.0	21.9	14.5	79.2	0.8	39.9	14.2	32.4	10.9	42.1	64.3	36.4	48.8	42.7	96.5	32.7	21.5	40.2
PointGroup [24]	80.5	69.6	54.9	48.1	87.7	22.4	44.9	41.6	42.0	37.7	37.2	64.4	61.1	71.5	62.9	98.3	46.2	53.0	56.9
DyCo3D [38]	77.4	70.4	48.4	52.3	90.2	34.9	47.5	52.3	40.5	44.7	51.5	70.3	54.3	69.6	**94.8**	47.2	46.2	56.4	61.0
HAIS [25]	87.1	70.2	49.4	55.4	82.5	47.8	55.7	58.5	48.1	48.7	53.0	76.1	69.2	67.7	75.3	**100.0**	51.5	56.3	64.0
SoftGroup [26]	86.7	71.8	64.3	61.7	85.6	38.6	55.6	57.5	53.1	53.9	76.7	75.6	70.9	67.5	77.6	98.3	56.3	60.3	67.6
Mask3D [31]	87.0	79.1	66.7	65.5	94.4	63.1	**73.6**	63.5	74.4	65.8	**77.1**	71.4	77.5	**78.0**	82.8	**100.0**	**65.1**	73.2	73.7
ISES [51]	90.1	76.7	54.5	67.6	95.0	63.7	71.0	66.3	73.5	65.8	**77.1**	71.4	77.5	**78.0**	82.8	**100.0**	**65.1**	73.9	75.0
MAFT [34]	90.1	80.1	65.6	65.6	94.7	66.2	73.3	**68.7**	74.4	73.8	72.6	76.2	78.5	67.2	83.5	96.6	63.8	73.8	75.9
Ours	**92.4**	**83.9**	**75.3**	**68.0**	**95.3**	**66.7**	71.1	**68.7**	**78.6**	**75.7**	75.4	**78.8**	**79.6**	77.4	83.7	96.6	63.3	**75.0**	**78.1**

set, in Table 3. Compared to MAFT, our method achieves performance improvements of +0.5 mAP, +0.6 AP_{50}, and +0.8 AP_{25}. Against MSTA3D, our method shows a higher margin of +3.5 on mAP, +3.6 on AP_{50}, and +4.0 on AP_{25}. These results shows that our method handles more categories yet has competitive performance, demonstrating its robustness and effectiveness.

4.2.3 S3DIS

We compare our method to other competing approaches on S3DIS (Area 5) in Table 4. It can be observed that our method achieves the highest scores on both AP_{50} and AP_{25}. Compared to MAFT, our method improves AP_{50} by +1.5 and AP_{25} by +0.9, respectively. Furthermore, it also outperforms MSTA3D by +2.4 on AP_{50}. These results demonstrate that our method still has superior performance even for indoor scenes with complex layouts, highlighting the generalizability of our network.

4.2.4 Runtime analysis

Table 5 reports the average inference time per scan for different methods, on the ScanNetV2 dataset. For fair comparison, all methods were evaluated on the same RTX 4090 GPU. Our method achieves an average inference speed of 198 ms/scan, a mere 35 ms slower than the fastest model, while outperforming it by +3.4 mAP, +4.2 AP_{50}, and +3.6 AP_{25}. Component analysis shows that our backbone, knowledge embedding module, and query decoder take 18 ms, 25 ms, and 34 ms, respectively. We note that superpoint extraction can be precomputed offline during the training phase, thus significantly reducing the computational load during training. In summary, our method is highly accurate yet efficient. It provides efficient, fine-grained 3D instance segmentation for complex scenes.

Table 3 Quantitative comparison on ScanNet200 with mAP, AP_{50}, and AP_{25} metrics

Method	mAP	AP_{50}	AP_{25}
SPFormer [32]	23.3	38.5	**48.6**
Mask3D [31]	27.4	37.0	42.3
MAFT [34]	29.2	38.2	43.3
MSTA3D [35]	26.2	35.2	40.1
Sonata [7]	25.4	35.6	42.1
Ours	**29.7**	**38.8**	44.1

Table 4 Quantitative comparison on S3DIS with AP_{50} and AP_{25} metrics

Method	AP_{50}	AP_{25}
PointGroup [24]	57.8	—
SSTNet [50]	59.3	—
SoftGroup [26]	66.1	—
ISBNet [39]	67.5	—
SPFormer [32]	66.8	—
MAFT [34]	69.1	75.7
Mask3D [31]	71.9	77.2
MSTA3D [35]	70.0	—
Sonata [7]	57.4	63.8
Ours	**72.4**	**77.9**

Table 5 Inference time per scan on the ScanNetv2 dataset. S.p. means superpoint

Method	Component time (ms)	Total (ms)
PointGroup [24]	Backbone (GPU): 26 Grouping (GPU+CPU): 165 ScoreNet (GPU): 54	245
SSTNet [50]	S.p. extraction (CPU): 132 Backbone (GPU): 31 Tree network (GPU+CPU): 114 ScoreNet (GPU): 43	320
SoftGroup [26]	Pointwise prediction (GPU): 77 Soft grouping (GPU+CPU): 108 Top-down refinement (GPU): 54	239
SPFormer [32]	S.p. extraction (CPU): 132 Backbone (GPU): 18 S.p. pooling (GPU): 10 Query decoder (GPU): 14	**174**
MAFT [34]	S.p. extraction (CPU): 132 Backbone (GPU): 18 S.p. pooling (GPU): 10 Query decoder (GPU): 23	183
OneFormer3D [33]	S.p. extraction (CPU): 132 Backbone (GPU): 18 S.p. pooling (GPU): 10 Query decoder (GPU): 21	181
Ours	S.p. extraction (CPU): 132 Backbone (GPU): 18 KE (GPU): 25 Query decoder (GPU): 34	209

4.3 Qualitative results

Figure 6 provides a visual comparison of results of MAFT, OneFormer3D, and our method on ScanNetV2. The dashed elliptical regions in Fig. 6, show how both MAFT and OneFormer3D can produce noticeable segmentation artifacts, including over-segmentation, where a single object instance

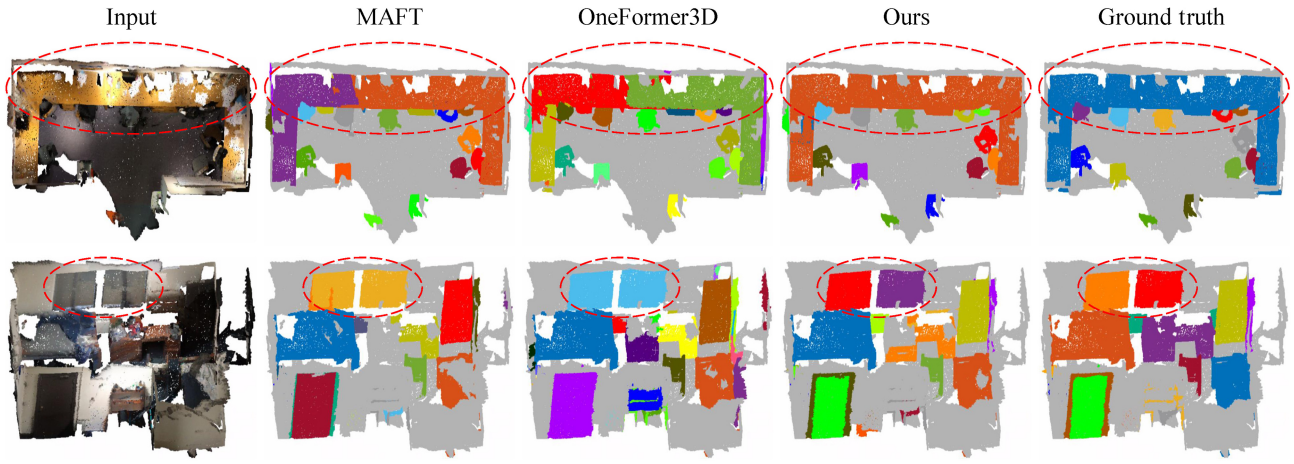

Fig. 6 Qualitative comparison of instance segmentation results on ScanNetV2. Left to right: input point clouds, results produced by MAFT, OneFormer3D, our method, and ground truth. Dashed ellipses highlight how our method can produce more accurate instance masks, effectively mitigating over- and under-segmentation artifacts.

is incorrectly split into multiple instances, and under-segmentation, where multiple distinct instances are erroneously merged as a single instance. In contrast, our method avoids these over- and under-segmentation artifacts. This improvement stems from the introduction of a collaborative mask generation approach, which enables our method to exploit complementary information from different feature spaces.

To further assess the robustness and generalization ability of our approach, we evaluated it on the Matterport3D dataset [53], which includes noisy real-world RGB-D scans. Some qualitative results are shown in Fig. 7: our method consistently avoids both over- and under-segmentation errors compared to the baseline and yields substantially finer delineation of object instances. For example, in the dashed ellipse in Fig. 7(above), our approach more accurately isolates the pillow from the sofa, demonstrating its ability to handle fine-grained structures. These observations suggest that our method is robust and produces more precise instance masks for such challenging data.

4.4 Architectural study

We conducted a series of experiments using the ScanNetV2 validation set to evaluate the core components and configurations of our method. Using MAFT as the baseline, we focused on three issues: analysis of query initialization strategies, evaluation of individual modules to assess their contributions, and exploration of decoder layer depths and numbers of queries.

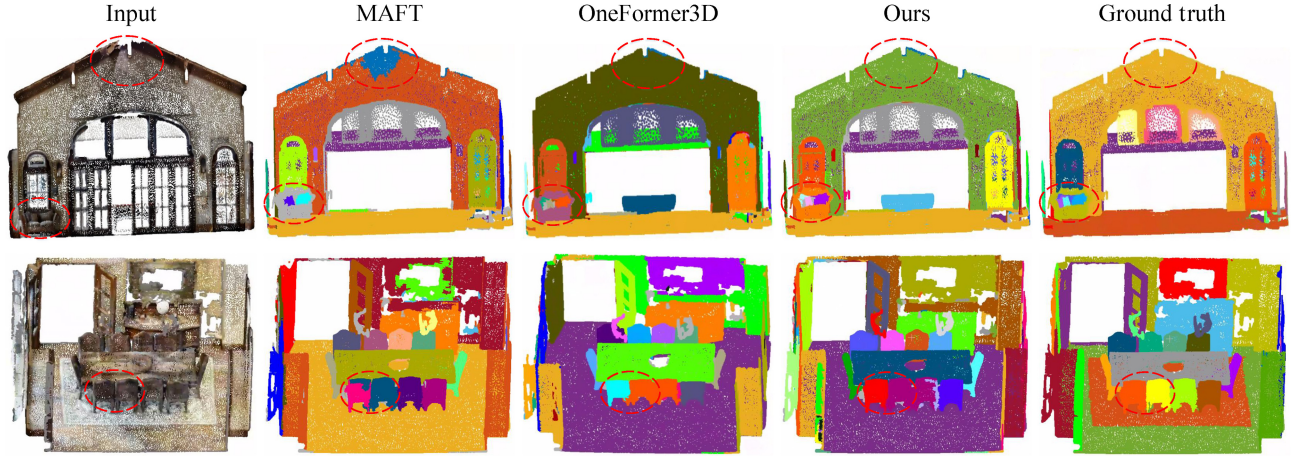

Fig. 7 Comparison of sample instance segmentation results on Matterport3D. Left to right: input point clouds, segmentation results from MAFT, OneFormer3D, our method, and ground truth. Dashed ellipses highlight how our method is robust, and generates more precise instance masks for these challenging examples.

4.4.1 *Initialization of queries*

We evaluated the performance of our framework with different query initialization strategies, with results summarized in Table 6. Existing methods commonly use either the learned query generation technique or farthest point sampling (FPS). In contrast, our proposed semFPS strategy first filters out non-instance points using semantic predictions, then applies FPS to the remaining instance points to generate queries. The results demonstrate that semFPS produces superior initial semantic queries. Additionally, the learnable technique outperforms FPS for initializing geometric queries, as it better captures structural information and adapts to complex scenes.

4.4.2 *Effects of core modules*

We validated the contribution of our individual modules by assessing the following three variants on the ScanNetV2 validation set.

- V1: baseline + KE.
- V2: baseline + KE + FQR.
- V3: baseline + KE + GSM.

The quantitative comparison of these variants shown in Table 7 highlights the contribution of each module and their synergistic effects. As we can see, our full model outperforms all variants, highlighting the strong synergy and cooperation between the modules. In addition, integrating any single module improves the segmentation accuracy compared to the baseline, demonstrating the effectiveness of each module. In particular, removing GSM from the full model results in a significant accuracy drop, as observed in the third and fifth rows of Table 7. This decline arises from a failure to leverage the complementary nature of semantic and geometric features, underscoring the importance of integrating both for instance segmentation.

The visual comparison provided in Fig. 8 intuitively demonstrates the impact of each module. From left to right, V1 reduces mis-segmentation artifacts compared to the baseline, as evident in the zoomed-in view where the number of errors decreases. This highlights the effectiveness of KE in leveraging geometric structure and filtering out non-instance points. V2, with the addition of FQR that allows mutual optimization between queries and features, further enhances the segmentation accuracy compared to V1. V3, integrating geometric–semantic mutual guidance, shows significant improvements over V1. As we can see, both the left and right bookshelves are almost correctly segmented, with only minor under-segmentation remaining in the right bookshelf. Finally, our full model achieves near-perfect segmentation of all instances, demonstrating its superior ability to address over- and under-segmentation challenges.

4.4.3 *Numbers of decoder layers and queries*

We analyzed the impact on model performance of

Table 6 Impact of initialization strategies on semantic and geometric query generation

Q_s	Q_g	mAP	AP_{50}	AP_{25}
Learnable	FPS	58.3	75.5	84.6
Learnable	Learnable	59.5	77.4	85.9
SemFPS	FPS	58.6	76.0	85.2
SemFPS	Learnable	**59.7**	**78.1**	**86.5**

Table 7 Impact of individual modules on our method

Variant	KE	FQR	GSM	mAP	AP_{50}	AP_{25}
Baseline				58.4	75.9	84.6
V1	✓			58.7	76.4	85.4
V2	✓	✓		58.8	76.7	84.9
V3	✓		✓	59.4	77.5	86.0
Ours	✓	✓	✓	**59.7**	**78.1**	**86.5**

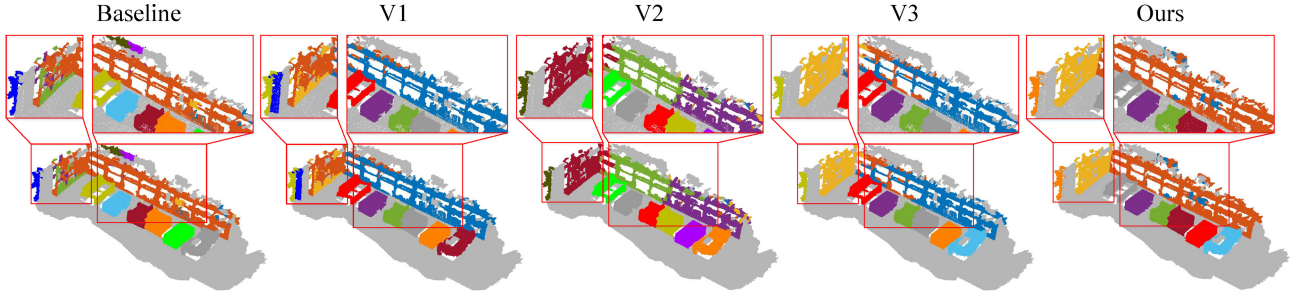

Fig. 8 Qualitative results from the study considering the impact of individual modules. Left to right: baseline, variants V1 (KE), V2 (KE + FQR), V3 (KE + GSM), and our full model. The close-up views highlight the effectiveness of each module and the full model in addressing over- and under-segmentation issues commonly encountered in 3D instance segmentation.

varying the number of layers and queries, with results shown in Table 8. Optimal performance is achieved with a network depth of six layers. Fewer layers fail to capture sufficient semantic and geometric information, whereas deeper networks introduce unnecessary computational load and produce negative results. Too few queries hinders the learning of sufficient instance features, while too many can cause instance loss during the matching process, reducing performance.

4.4.4 Dimensionality of point feature F^0

We evaluated the impact of different dimensionalities for the point feature F^0, with results reported in Table 9: a feature dimensionality of 256 yields the best instance segmentation performance. Lower dimensionality limits the capacity to encode semantic and geometric information, while overly large dimensionality introduces redundancy and increases the risk of overfitting, degrading performance. Although a 512-dimensional setting also achieves competitive results, we adopted $d = 256$ to match mainstream methods (e.g., Mask3D and MAFT) and to ensure a fair comparison.

4.4.5 Hyperparameter settings

Most parameters in our framework are learnable and are adaptively optimized during training, including those in Eqs. (10) and (19). To assess the influence of manually specified hyperparameters, we conducted a sensitivity study on the weighting factors λ_{center} and λ_{IoU}, which govern the balance between different loss terms.

Table 8 Performance comparison for different decoder layer depths and numbers of queries

Layers	Queries	mAP	AP_{50}	AP_{25}
1	400	53.3	69.2	78.9
3	400	56.9	74.9	84.7
6	400	**59.7**	**78.1**	**86.5**
12	400	59.4	77.4	85.9
6	100	56.8	74.3	82.6
6	200	57.6	75.6	84.6
6	600	59.2	77.1	86.1

Table 9 Comparison of instance segmentation performance for different dimensionalities of point feature F^0

d	mAP	AP_{50}	AP_{25}
128	58.5	76.2	84.2
256	**59.7**	78.1	**86.5**
512	59.6	**78.4**	86.0

For the loss weights related to mask generation and category prediction ($\lambda_{\text{cls}}, \lambda_{\text{dice}}, \lambda_{\text{bce}}$), we adopted the standard settings used in prior work (e.g., MAFT and Mask3D) to ensure fair comparison. As Table 10 shows, adjusting these factors results in minor performance variations, confirming that the proposed framework is robust to reasonable hyperparameter changes and is not unduly sensitive to these settings.

4.5 Generalization to object detection

To further assess the scalability and generalization ability of the proposed method, we evaluated its 3D object detection performance on the ScanNetV2 and S3DIS benchmarks. For each instance, an axis-aligned bounding box was obtained by computing the minimum and maximum coordinates of its corresponding mask. Following standard practice in 3D object detection, we report performance in terms of box AP_{50} and AP_{25}: see Table 11. On

Table 10 Comparison of instance segmentation performance using different weights λ_{center} and λ_{IoU}

λ_{center}	λ_{IoU}	mAP	AP_{50}	AP_{25}
2	2	56.9	75.7	85.1
1	1	59.3	77.1	85.8
1	0.5	58.7	77.0	85.8
0.5	1	59.1	77.6	86.1
0.5	0.5	**59.7**	**78.1**	**86.5**
0.25	0.25	58.5	76.6	85.6

Table 11 Numerical evaluation of object detection results on ScanNetV2 with box AP_{50} and box AP_{25} metrics

Method	Box AP_{50}	Box AP_{25}
VoteNet [54]	33.5	58.6
3D-MPA [48]	49.2	64.2
GSDN [55]	34.8	62.5
H3DNet [56]	48.1	67.2
3DETR [57]	47.0	65.0
Group-free [58]	52.8	70.1
RBGNet [59]	55.2	69.9
HyperDet3D [56]	57.2	70.9
FCAF3D [60]	57.3	71.5
CAGroup3D [61]	61.3	75.1
Mask3D [31]	56.2	70.2
MAFT [34]	63.9	73.5
MSTA3D[35]	64.3	**78.6**
V-DETR [62]	65.0	77.4
OneFormer3D [33]	65.3	76.4
Swin3D [9]	63.2	76.4
UniDet3 [8]	**66.1**	77.5
Ours	65.5	77.0

ScanNetV2 our method surpasses most existing object detection approaches in terms of AP_{50}, trailing only slightly behind UniDet3D, and achieves competitive performance on AP_{25}. Results on S3DIS are summarized in Table 12. Consistent with the observations on ScanNetV2, our method attains performance comparable to UniDet3D on AP_{50}, while establishing a new state of the art on AP_{25}. The qualitative results demonstrate that our method can produce more accurate and tightly fitting bounding boxes than those produced by the baseline, as also shown in Fig. 9.

4.6 Limitations in object detection

The proposed framework achieves state-of-the-art performance on instance segmentation benchmarks. However, its performance on 3D object detection is slightly inferior to methods specifically optimized for detection, such as UniDet3D. In addition, the framework exhibits limitations when detecting and segmenting fine-grained objects. These behaviors primarily arise from the following factors.

First, the core objective of this work was to generate high-quality instance masks rather than to perform precise regression of 3D object detection bounding boxes. Consequently, detection-oriented metrics (e.g., box AP) are not explicitly optimized during training.

Second, the proposed geometry–semantics co-modeling strategy alleviates over- and under-segmentation, but its tightly coupled design may propagate noise or classification biases from the semantic branch to the geometric branch. This interaction can lead to bounding box inflation, localization offsets, or shape distortions. Such effects are particularly pronounced for small, elongated, or non-rigid objects (e.g., cushions, curtains, and pillows), where minor inaccuracies or discontinuities in mask boundaries are amplified during bounding box derivation.

Importantly, these limitations reflect intentional, task-oriented design trade-offs. By prioritizing instance mask accuracy as the primary optimization target and treating detection bounding boxes as a secondary output, the geometric modeling module is deliberately embedded within a semantic–geometric collaborative framework rather than implemented as an independent, detection-focused regression head. While this design substantially enhances segmentation performance, it inevitably results in compromising detection accuracy somewhat.

Table 12 Numerical evaluation of object detection results on S3DIS with box AP_{50} and box AP_{25} metrics

Method	Box AP_{50}	Box AP_{25}
SPGroup3D [63]	47.2	69.2
FCAF3D [60]	45.9	66.7
Swin3D [9]	58.6	75.4
UniDet3D [8]	**60.8**	<u>75.2</u>
Ours	<u>58.8</u>	**75.6**

5 Conclusions

We have proposed a unified 3D instance framework that jointly predicts instance masks, semantic labels, and bounding boxes in an end-to-end

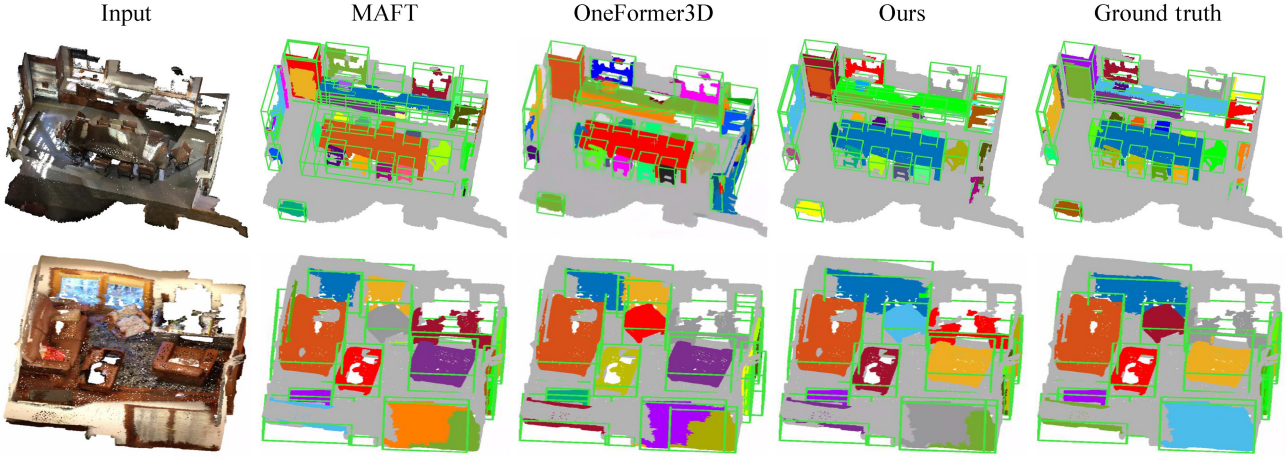

Fig. 9 Qualitative comparison of object detection results on ScanNetV2. Left to right: input point clouds, results produced by MAFT, OneFormer3D, our method, and the ground truth. Our method predicts more accurate and tightly fitting bounding boxes for object instances.

manner. Our key insight is that geometric and semantic information provide complementary and mutually reinforcing cues. By optimizing them jointly, our framework learns more discriminative and representative features, thereby enhancing the precision and robustness of instance mask prediction. Our method achieves state-of-the-art performance on multiple benchmarks, delivering significant improvements in both quantitative metrics and visual quality. Furthermore, although not explicitly designed for object detection, it surpasses most existing approaches in detection accuracy as well.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback. We gratefully acknowledge the support from the National Key R&D Program of China (2024YFA1016300) and the Ministry of Education, Singapore, under an Academic Research Fund Grant (RT19/22).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Ying He is the Associate Editor of this journal.

References

[1] Liu, Z.; Li, Y.; Wang, W.; Liu, L.; Chen, R. Mesh total generalized variation for denoising. *IEEE Transactions on Visualization and Computer Graphics* Vol. 28, No. 12, 4418–4433, 2022.

[2] Yu, Q.; Li, X.; Tang, Y.; Xu, J.; Hu, L.; Hao, Y.; Chen, M. JIMR: Joint semantic and geometry learning for point scene instance mesh reconstruction. *IEEE Transactions on Visualization and Computer Graphics* Vol. 31, No. 8, 4270–4282, 2025.

[3] Wang, J.; Fei, B.; de Silva, E. D.; Liu, Z.; He, Y.; Lu, X. A survey of deep learning-based point cloud denoising. *arXiv preprint* arXiv:2508.17011, 2025.

[4] Liu, Z.; Huang, Z.; Pan, M.; He, Y. Deterministic point cloud diffusion for denoising. *IEEE Transactions on Visualization and Computer Graphics*, doi: 10.1109/TVCG.2025.3621633, 2025.

[5] Liu, Z.; Zhang, J.; Zhang, M.; Ke, R.; Yu, C.; Liu, L. Unsupervised point cloud reconstruction via recurrent multi-step moving strategy. *IEEE Transactions on Multimedia* Vol. 28, 972–984, 2026.

[6] Xu, X.; Xia, C.; Wang, Z.; Zhao, L.; Duan, Y.; Zhou, J.; Lu, J. Memory-based adapters for online 3D scene perception. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 21604–21613, 2024.

[7] Wu, X.; DeTone, D.; Frost, D.; Shen, T.; Xie, C.; Yang, N.; Engel, J.; Newcombe, R.; Zhao, H.; Straub, J. Sonata: Self-supervised learning of reliable point representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 22193–22204, 2025.

[8] Kolodiazhnyi, M.; Vorontsova, A.; Skripkin, M.; Rukhovich, D.; Konushin, A. UniDet3D: Multi-dataset indoor 3D object detection. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 39, No. 4, 4365–4373, 2025.

[9] Yang, Y. Q.; Guo, Y. X.; Xiong, J. Y.; Liu, Y.; Pan, H.; Wang, P. S.; Tong, X.; Guo, B. Swin3D: A pretrained transformer backbone for 3D indoor scene understanding. *Computational Visual Media* Vol. 11, No. 1, 83–101, 2025.

[10] Huang, S. S.; Ma, Z. Y.; Mu, T. J.; Fu, H.; Hu, S. M. Supervoxel convolution for online 3D semantic segmentation. *ACM Transactions on Graphics* Vol. 40, No. 3, Article No. 34, 2021.

[11] Li, X.; Tan, X.; Zhang, Z.; Xie, Y.; Ma, L. Point mask transformer for outdoor point cloud semantic segmentation. *Computational Visual Media* Vol. 11, No. 3, 497–511, 2025.

[12] Liu, G.; van Kaick, O.; Huang, H.; Hu, R. Active self-training for weakly supervised 3D scene semantic segmentation. *Computational Visual Media* Vol. 10, No. 3, 425–438, 2024.

[13] Sun, C. Y.; Tong, X.; Liu, Y. Semantic segmentation-assisted instance feature fusion for multi-level 3D part instance segmentation. *Computational Visual Media* Vol. 9, No. 4, 699–715, 2023.

[14] Charles, R. Q.; Hao, S.; Mo, K.; Guibas, L. J. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 77–85, 2017.

[15] Liu, Z.; Zhao, Y.; Zhan, S.; Liu, Y.; Chen, R.; He, Y. PCDNF: Revisiting learning-based point cloud denoising via joint normal filtering. *IEEE Transactions on Visualization and Computer Graphics* Vol. 30, No. 8, 5419–5436, 2024.

[16] Liu, Z.; Zhou, W.; Guo, C.; Qiu, Q.; Xie, Z. PyramidPCD: A novel pyramid network for point cloud denoising. *Pattern Recognition*, Vol. 161, 111228, 2025.

[17] Guo, J.; Qin, H.; Zhou, Y.; Chen, X.; Nan, L.; Huang,

H. Fast building instance proxy reconstruction for large urban scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 46, No. 11, 7267–7282, 2024.

[18] Yang, G.; Xue, F.; Zhang, Q.; Xie, K.; Fu, C. W.; Huang, H. UrbanBIS: A large-scale benchmark for fine-grained urban building instance segmentation. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference, Article No. 16, 2023.

[19] Yang, M.; Guo, J.; Chen, Y.; Chen, L. InstanceTex: Instance-level controllable texture synthesis for 3D scenes via diffusion priors. In: Proceedings of the SIGGRAPH Asia Conference Papers, Article No. 59, 2024.

[20] Zhou, D.; Fang, J.; Song, X.; Liu, L.; Yin, J.; Dai, Y.; Li, H.; Yang, R. Joint 3D instance segmentation and object detection for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1836–1846, 2020.

[21] Cao, H.; Xia, X.; Wu, G.; Hu, R.; Liu, L. ScanBot: Autonomous reconstruction via deep reinforcement learning. *ACM Transactions on Graphics* Vol. 42, No. 4, Article No. 157, 2023.

[22] Huang, J.; Artemov, A.; Chen, Y.; Zhi, S.; Xu, K.; Nießner, M. SSR-2D: Semantic 3D scene reconstruction from 2D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 46, No. 12, 8486–8501, 2024.

[23] Dong, Z. C.; Wu, W.; Xu, Z.; Sun, Q.; Yuan, G.; Liu, L.; Fu, X. M. Tailored reality: Perception-aware scene restructuring for adaptive VR navigation. *ACM Transactions on Graphics* Vol. 40, No. 5, Article No. 159, 2021.

[24] Jiang, L.; Zhao, H.; Shi, S.; Liu, S.; Fu, C. W.; Jia, J. PointGroup: Dual-set point grouping for 3D instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4866–4875, 2020.

[25] Chen, S.; Fang, J.; Zhang, Q.; Liu, W.; Wang, X. Hierarchical aggregation for 3D instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 15447–15456, 2021.

[26] Vu, T.; Kim, K.; Luu, T. M.; Nguyen, T.; Yoo, C. D. SoftGroup for 3D instance segmentation on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2698–2707, 2022.

[27] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 6000–6010, 2017.

[28] Wang, P. S. OctFormer: Octree-based transformers for 3D point clouds. *ACM Transactions on Graphics* Vol. 42, No. 4, Article No. 155, 2023.

[29] Wu, X.; Jiang, L.; Wang, P. S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; Zhao, H. Point transformer V3: Simpler, faster, stronger. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4840–4851, 2024.

[30] Guo, M. H.; Cai, J. X.; Liu, Z. N.; Mu, T. J.; Martin, R. R.; Hu, S. M. PCT: Point cloud transformer. *Computational Visual Media* Vol. 7, No. 2, 187–199, 2021.

[31] Schult, J.; Engelmann, F.; Hermans, A.; Litany, O.; Tang, S.; Leibe, B. Mask3D: Mask transformer for 3D semantic instance segmentation. In: Proceedings of the IEEE International Conference on Robotics and Automation, 8216–8223, 2023.

[32] Sun, J.; Qing, C.; Tan, J.; Xu, X. Superpoint transformer for 3D scene instance segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 37, No. 2, 2393–2401, 2023.

[33] Kolodiazhnyi, M.; Vorontsova, A.; Konushin, A.; Rukhovich, D. OneFormer3D: One transformer for unified point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 20943–20953, 2024.

[34] Lai, X.; Yuan, Y.; Chu, R.; Chen, Y.; Hu, H.; Jia, J. Mask-attention-free transformer for 3D instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 3670–3680, 2023.

[35] Tran, D. D. T.; Kang, B.; Lee, Y. MSTA3D: Multi-scale twin-attention for 3D instance segmentation. In: Proceedings of the 32nd ACM International Conference on Multimedia, 1467–1475, 2024.

[36] Hou, J.; Dai, A.; NieBner, M. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4416–4425, 2019.

[37] Yang, B.; Wang, J.; Clark, R.; Hu, Q.; Wang, S.; Markham, A.; Trigoni, N. Learning object bounding boxes for 3D instance segmentation on point clouds. In: Proceedings of the Advances in Neural Information Processing Systems, 6737–6746, 2019.

[38] He, T.; Shen, C.; van den Hengel, A. DyCo3D: Robust instance segmentation of 3D point clouds through dynamic convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 354–363, 2021.

[39] Ngo, T. D.; Hua, B. S.; Nguyen, K. 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, 13550–13559, 2023.

[40] Wu, Y.; Shi, M.; Du, S.; Lu, H.; Cao, Z.; Zhong, W. 3D Instances as 1D kernels. *arXiv preprint* arXiv:2207.07372, 2022.

[41] Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* Vol. 2, Nos. 1–2, 83–97, 1955.

[42] Deng, R.; Shen, C.; Liu, S.; Wang, H.; Liu, X. Learning to predict crisp boundaries. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11210.* Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 570–586, 2018.

[43] Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 658–666, 2019.

[44] Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2432–2443, 2017.

[45] Rozenberszki, D.; Litany, O.; Dai, A. Language-grounded indoor 3D semantic segmentation in the wild. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13693.* Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 125–141, 2022.

[46] Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1534–1543, 2016.

[47] Lahoud, J.; Ghanem, B.; Oswald, M. R.; Pollefeys, M. 3D instance segmentation via multi-task metric learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9255–9265, 2019.

[48] Engelmann, F.; Bokeloh, M.; Fathi, A.; Leibe, B.; NieBner, M. 3D-MPA: Multi-proposal aggregation for 3D semantic instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9028–9037, 2020.

[49] Han, L.; Zheng, T.; Xu, L.; Fang, L. OccuSeg: Occupancy-aware 3D instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2937–2946, 2020.

[50] Liang, Z.; Li, Z.; Xu, S.; Tan, M.; Jia, K. Instance segmentation in 3D scenes using semantic superpoint tree networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2763–2772, 2021.

[51] Al Khatib, S.; El Amine Boudjoghra, M.; Lahoud, J.; Khan, F. S. 3D instance segmentation via enhanced spatial and semantic supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 541–550, 2023.

[52] Lu, J.; Deng, J.; Wang, C.; He, J.; Zhang, T. Query refinement transformer for 3D instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 18470–18480, 2023.

[53] Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niebner, M.; Savva, M.; Song, S.; Zeng, A.; Zhang, Y. Matterport3D: Learning from RGB-D data in indoor environments. In: Proceedings of the International Conference on 3D Vision, 667–676, 2017.

[54] Qi, C. R.; Litany, O.; He, K.; Guibas, L. Deep Hough voting for 3D object detection in point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9276–9285, 2019.

[55] Gwak, J.; Choy, C.; Savarese, S. Generative sparse detection networks for 3D single-shot object detection. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12349.* Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J.-M. Eds. Springer Cham, 297–313, 2020.

[56] Zhang, Z.; Sun, B.; Yang, H.; Huang, Q. H3DNet: 3D object detection using hybrid geometric primitives. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12357.* Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J.-M. Eds. Springer Cham, 311–329, 2020.

[57] Misra, I.; Girdhar, R.; Joulin, A. An end-to-end transformer model for 3D object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2886–2897, 2021.

[58] Liu, Z.; Zhang, Z.; Cao, Y.; Hu, H.; Tong, X. Group-free 3D object detection via transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2929–2938, 2021.

[59] Wang, H.; Shi, S.; Yang, Z.; Fang, R.; Qian, Q.; Li, H.; Schiele, B.; Wang, L. RBGNet: Ray-based grouping for 3D object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1100–1109, 2022.

[60] Rukhovich, D.; Vorontsova, A.; Konushin, A. FCAF3D: Fully convolutional anchor-free 3D object detection. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13670.* Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 477–493, 2022.

[61] Wang, H.; Ding, L.; Dong, S.; Shi, S.; Li, A.; Li, J.; Li, Z.; Wang, L. CAGroup3D: Class-aware grouping for 3D object detection on point clouds. In: Proceedings of the 36th International Conference on Neural Information Processing System, Article No. 2173, 2022.

[62] Shen, Y.; Geng, Z.; Yuan, Y.; Lin, Y.; Liu, Z.; Wang, C.; Hu, H.; Zheng, N.; Guo, B. V-DETR: DETR with vertex relative position encoding for 3D object detection. *arXiv preprint* arXiv:2308.04409, 2024.

[63] Zhu, Y.; Hui, L.; Shen, Y.; Xie, J. SPGroup3D: Superpoint grouping network for indoor 3D object detection. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 38, No. 7, 7811–7819, 2024.

Huixiao Tian is currently an M.S. candidate in China University of Geosciences, Wuhan. He received his B.S. degree from Yangtze University in 2023. His research interests include 3D vision and deep learning.

Zhipeng Jiang is currently an M.S. candidate in China University of Geosciences, Wuhan. He received his B.S. degree from Beijing Institute of Petrochemical Technology in 2022. His research interests include 3D vision and deep learning.

Shimin Song is currently an M.S. candidate in China University of Geosciences, Wuhan. She received her B.S. degree from Liaocheng University in 2024. Her research interests include geometric modeling and 3D deep learning.

Saishang Zhong is an associate professor at Wuhan Textile University. He received his Ph.D. degree from China University of Geosciences, Wuhan. His research interests include geometry processing, 3D vision, and AIGC.

Zheng Liu is an associate professor at School of Computer Science, China University of Geosciences, Wuhan. He received his Ph.D. degree from Central China Normal University, and held a post-doctoral position in School of Mathematical Sciences, University of Science and Technology of China. His research interests include geometry processing, 3D deep learning, computer graphics, and 3D vision.

Ying He is an associate professor at the College of Computing and Data Science, Nanyang Technological University, Singapore, where he also serves as the Director of the Centre for Augmented and Virtual Reality. His research focuses on geometric computing and analysis, with applications in computer graphics, 3D vision, and computer-aided design. He serves or has served on the editorial boards of *IEEE TVCG, Computer Graphics Forum*, and *Computational Visual Media*. He has also held leadership roles in several conferences.