This is the Pre-Published Version.

The following publication L. Yao, Y. Wang, M. Liu and L. -P. Chau, "SGIFormer: Semantic-Guided and Geometric-Enhanced Interleaving Transformer for 3D Instance Segmentation," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 35, no. 3, pp. 2276-2288, March 2025 is available at https://doi.org/10.1109/tcsvt.2024.3498041.

SGIFormer: Semantic-guided and Geometric-enhanced Interleaving Transformer for 3D Instance Segmentation

Lei Yao, Yi Wang, *Member, IEEE,* Moyun Liu, and Lap-Pui Chau, *Fellow, IEEE*

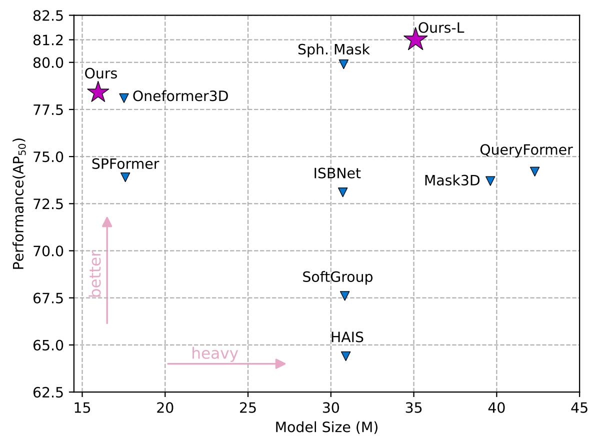
***Abstract*—In recent years, transformer-based models have exhibited considerable potential in point cloud instance segmentation. Despite the promising performance achieved by existing methods, they encounter challenges such as instance query initialization problems and excessive reliance on stacked layers, rendering them incompatible with large-scale 3D scenes. This paper introduces a novel method, named SGIFormer, for 3D instance segmentation, which is composed of the Semantic-guided Mix Query (SMQ) initialization and the Geometric-enhanced Interleaving Transformer (GIT) decoder. Specifically, the principle of our SMQ initialization scheme is to leverage the predicted voxel-wise semantic information to implicitly generate the scene-aware query, yielding adequate scene prior and compensating for the learnable query set. Subsequently, we feed the formed overall query into our GIT decoder to alternately refine instance query and global scene features for further capturing fine-grained information and reducing complex design intricacies simultaneously. To emphasize geometric property, we consider bias estimation as an auxiliary task and progressively integrate shifted point coordinates embedding to reinforce instance localization. SGIFormer attains state-of-the-art performance on ScanNet V2, ScanNet200, S3DIS datasets, and the challenging high-fidelity ScanNet++ benchmark, striking a balance between accuracy and efficiency. The code, weights, and demo videos are publicly available at https://rayyoh.github.io/SGIFormer/.**

***Index Terms*—Point Clouds, 3D Instance Segmentation, Transformer, Semantic Features**

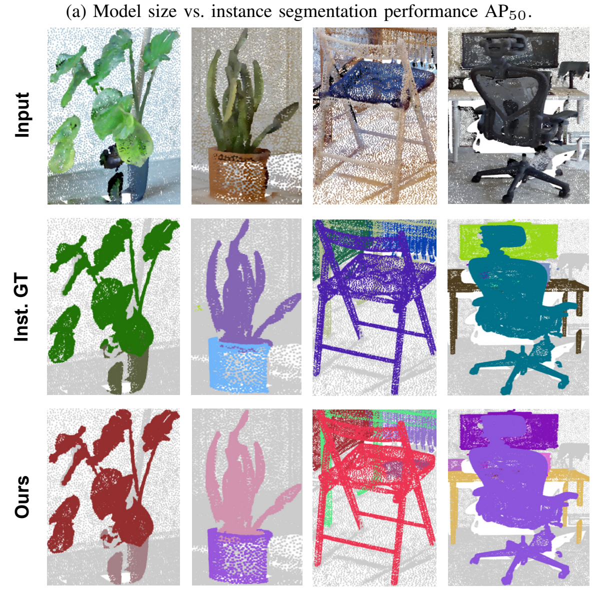
I. Introduction

POINT cloud instance segmentation serves as a fundamental task for 3D scene understanding across various applications such as embodied AI [1], [2], autonomous driving [3], [4], and metaverse [5]. The primary objective of this task is to identify each instance using binary masks and assign corresponding semantic categories within a scanned scene. However, due to the unordered nature of points and the sophisticated layout of scenes, accurately segmenting objects with proximity and varying sizes remains challenging in 3D point cloud instance segmentation.

Current approaches for point cloud instance segmentation can be categorized into three distinct pipelines: proposal-based [8], [9], [10], [11], [12], grouping-based [13], [14], [15], [16], [17], [18], [19], and transformer-based [20], [21], [22], [23], [24], [25], [26]. Proposal-based methods follow a top-down strategy by generating a series of preliminary proposals and refining them to obtain accurate instance masks. Grouping-based methods directly aggregate points into instance clusters according to point-wise semantic information and positional



(a) Model size vs. instance segmentation performance AP_{50}.



(b) Examples of fine-grained segmentation on the ScanNet++ validation set.

Fig. 1. **Performance evaluation of our proposed SGIFormer.** (a) We present the performance comparison of various methods based on AP_{50} and model size on ScanNet V2 [6] validation split. SGIFormer outperforms previous methods, and even the smaller version achieves competitive results. (b) We showcase the fine-grained segmentation results of SGIFormer on ScanNet++ [7] validation set, demonstrating its ability to segment small objects within large-scale scenes accurately.

Manuscript received xx, xx; revised xx, xx. *(Corresponding author: Lap-Pui Chau)*

Lei Yao, Yi Wang, and Lap-Pui Chau are with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong SAR (e-mail: rayyoh.yao@connect.polyu.hk; yi-eie.wang@polyu.edu.hk; lap-pui.chau@polyu.edu.hk).

Moyun Liu is with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: lmomoy@hust.edu.cn).

relationships in a bottom-up manner. However, both proposal-based and grouping-based architectures share a common limitation: reliance on accurate intermediate results. For example, incorrect object bounding boxes or false class labels [13] may lead to error accumulation during subsequent processes, resulting in suboptimal performance. In contrast, transformer-based methods operate in a fully end-to-end style, leveraging the attention mechanism [27] to capture global context information of the scene efficiently. This approach typically utilizes a 3D backbone to extract global scene features from the input point cloud and feed them into a stacked transformer decoder with a fixed number of instance queries to refine them iteratively. The method regards each query as a potential object instance and simultaneously predicts the corresponding mask and category.

While transformer-based methods have shown great promise in 3D instance segmentation and achieved superior performance with compact pipelines, existing algorithms still have inherent limitations. Considering that the quality of the query affects both the final performance and the convergence speed of the model, we suggest that *query initialization* is the first crucial component that requires improvement. As pioneers in this field, SPFormer [20] and Mask3D [21] have attempted different strategies for query initialization. The former employs randomly initialized parametric queries that are learnable, while the latter samples queries from the raw input, which are non-parametric, and it observes performance improvement similar to [28]. While existing sampling methods, such as Farthest Point Sampling (FPS) [29], allow for varying numbers of queries during inference, they cannot guarantee high-quality queries, as the sampled points may overlook small instances or be located in non-informative background regions [30]. Additionally, the uncertainty in FPS might result in multiple queries covering the same object, further decreasing the quality of segmentation tasks. Another limitation lies in the *deficiency of geometric information and fine-grained details* during the query refinement stage. The decoder was initially designed to update instance queries by aggregating the scene features. Nevertheless, due to the quadratic complexity of the attention mechanism [27], the features are often pooled from point-level embedding [20], [24]. Although this reduces the computational cost, it neglects the fine-grained details of the original scene, which has not been adequately addressed in previous works.

Motivated by the abovementioned analysis, this paper presents a **S**emantic-guided and **G**eometric-enhanced **I**nterleaving Trans**Former** (SGIFormer) for point cloud instance segmentation, which aims to overcome the limitations of existing transformer-based methods. Concretely, we propose a Semantic-guided Mix Query (SMQ) initialization scheme by incorporating voxel-wise semantic prediction as guidance to filter out weak semantic regions and generate scene-aware queries from remaining voxels, maintaining scene prior and local details. Additionally, we combine another set of learnable queries to form the overall query set to further compensate for the deficiency of the semantic-guided queries and enhance the diversity of the queries. To emphasize geometric information, we introduce a Geometric-enhanced Interleaving Transformer (GIT) decoder to gradually participate in point coordinates embedding into global scene features and update them as well as queries alternately to avoid the loss of details. With diverse queries and the interleaving mechanism, our SGIFormer can benefit from the local semantic information and efficiently reduce the reliance on heavy stacked layers, improving both efficiency and accuracy. The main contributions of this work can be summarized as follows:

- We propose a novel query initialization scheme to obtain an instance query set with semantic guidance. This scheme effectively integrates scene prior and preserves local information, improving the quality and diversity of queries.
- We introduce an interleaving transformer decoder progressively incorporating geometric information to refine instance queries and global scene features alternately, reducing the reliance on heavy stacked layers and enhancing the preservation of fine-grained details.
- Comprehensive experiments are conducted on various datasets, including ScanNet V2, ScanNet200, S3DIS, and ScanNet++, to evaluate the proposed SGIFormer. The experimental results demonstrate the superiority of our method over existing state-of-the-art methods, establishing its effectiveness in 3D instance segmentation.

II. Related work

A. 3D Instance Segmentation Methods

3D instance segmentation methods can be broadly categorized into proposal-based, grouping-based, and transformer-based methods.

Proposal-based methods [8], [9], [10], [11], [12] generate object proposals at the first phase following a refinement stage to obtain instance masks in a top-down manner. Taking RGB-D scans as input, 3D-SIS [8] used predicted 3D bounding boxes to acquire associated fine masks. 3D-BoNet [10] conducted mask prediction by concatenating intermediate results and per-point features after getting instance bounding boxes using the Hungarian matching [31] algorithm. Following a similar style, TD3D [11] proposed a data-driven, fully convolutional network without relying on prior knowledge or handcrafted parameters. Differently, GSPN [9] adopted an analysis-by-synthesis strategy by reconstructing shapes to get proposals. Due to the inherent dependence on object proposals, false negative error might accumulate from inaccurate bounding box prediction in the abovementioned methods. Therefore, Spherical Mask [12] represented a 3D polygon using spherical coordinates for coarse instance detection and incorporated a point migration module to alleviate error propagation. However, the complicated multi-stage nature and post-processing steps lead to significant latency overhead. Instead, our SGIFormer benefits from the advantages of end-to-end set prediction and avoids redundant computation.

Grouping-based methods cluster points or superpoints [32], [33] into object instances based on corresponding semantic categories [13], [14], [15], [16], geometric offsets [13], [14], [15] or feature similarity [17], [18]. With predicted points-wise class labels, PointGroup [13] considered both shifted coordinates and original ones simultaneously to get groups followed by Non-Maximum Suppression (NMS) for final instance masks. To further improve clustering accuracy, HAIS [14] proposed hierarchical aggregation, which gathers points into a series of sets and obtains complete instances by set aggregation. SoftGroup [15] introduced a soft grouping mechanism that allows each point to be assigned multiple categories, reducing errors from semantic prediction. In [34], a binary clustering strategy is proposed as an alternative to traditional methods. SSTNet [16] and GraphCut [18] utilized superpoints as a mid-level representation and directly merged them into instances using semantic tree structure or graph neural networks. By eliminating the need for fancy offline clustering operation, our SGIFormer offers a more streamlined architecture

Building upon the flexibility and superiority of Transformer, DETR [35] introduced a compact set prediction pipeline for object detection, which was later extended to dense segmentation tasks [36], [37]. Inspired by DETR [35], a series of works have adapted the paradigm to 3D domain for object detection [38], [39] and instance segmentation [20], [21], [22], [23], [24], [25]. SPFormer [20] and Mask3D [21] are pioneers in achieving end-to-end 3D instance segmentation by iteratively refining a fixed number of instance queries. Additionally, 3IS-ESS [25] pointed out that the current pipeline lacks information exchange between the backbone and query refinement decoder. To moderate this, they proposed using voxel-wise semantic label prediction and raw coordinate regression as auxiliary tasks to enhance semantic and spatial understanding. Nevertheless, the intermediate results in their method are underutilized. In contrast, we propose using sufficient semantic cues to guide scene-aware query initialization.

B. Query Initialization and Refinement

Although the end-to-end transformer-based architecture is elegant, one crucial challenge is efficiently initializing queries to accelerate the training process further and boost the final performance. Similar to DETR [35] and Mask2Former [37], parametric learnable queries are used in SPFormer [20] and MAFT [23], while other approaches like Mask3D [21] demonstrated that sampling points from raw input scene according to their coordinates to initialize queries can lead to improved performance, as observed by [28] in 2D domain. Following this configuration, QueryFormer [22] proposed a tailored query aggregation module to reduce duplicate queries belonging to the same instance, aiming to increase coverage, and LCP-Former [40] utilized local context propagation strategy. In LeadNet [41], multiple dynamic prototypes are leveraged in the object query decoder. However, these modules introduced additional computational expenditure with limited benefits. OneFormer3D [24] randomly selected features from available over-segments of ScanNet V2 [6] as queries for training, but for better performance, all superpoints were used during inference. Even though this approach achieved one model for three different segmentation tasks on a specific dataset, it does not readily generalize to other datasets, as it showed a significant performance decline on S3DIS [42]. Differently, we introduce a novel mix query initialization strategy that combines scene-aware query and learnable query to achieve better generalization and performance.

For query refinement, existing methods [20], [23], [21], [24] typically adopt standard transformer decoders to progressively attend to features from superpoints or voxels, guided by masked attention [37], to enhance the stability of the training process. These approaches rely on heavily stacked transformer layers and do not consider the built-in advantages of geometric property for point cloud data in the refinement process. However, our SGIFormer participates in shifted point coordinates embedding to reinforce the global features for better instance localization progressively and adopts a novel interleaving update mechanism to alternately refine instance query and global scene features for more effective information exchange.

III. Methodology

In this section, we provide an overview of our pipeline and discuss the details of our methodology. We begin by describing the input data and backbone for feature extraction in Sec. III-A. We then elaborate on our novel semantic-guided mix query initialization scheme and geometric-enhanced interleaving transformer decoder in Sec. III-B and Sec. III-C, respectively. Finally, we discuss the loss function used in our method in Sec. III-D.

A. Backbone

Given a series of point cloud coordinates $\mathcal{C} \in \mathbb{R}^{n\times 3}$ and corresponding r, g, b colors $\mathcal{F} \in \mathbb{R}^{n\times 3}$, where n is the number of points. Our method takes a scene point set $\mathcal{P} = \{\mathcal{C}, \mathcal{F}\}$ as input as illustrated in Fig. 2. Before feeding them into the 3D backbone, we perform quantization by dividing the input points into m voxels $\mathcal{V}$ by a grid sampling pattern $\mathcal{G} : \mathcal{P} \mapsto \mathcal{V} = \{\hat{\mathcal{C}}, \hat{\mathcal{F}}\}$. Note that $\hat{\mathcal{C}} \in \mathbb{R}^{m\times 3}$ and $\hat{\mathcal{F}} \in \mathbb{R}^{m\times 3}$ are initialized by averaging the coordinates $\mathcal{C}$ and colors $\mathcal{F}$ of points within each voxel. Then the Submanifold Sparse Convolution [43] is adopted to implement our symmetrical U-Net backbone to extract voxel-wise global features $\mathbf{F} \in \mathbb{R}^{m\times d_o}$ similar to [20], [12], where d_o is the channel dimension of the features.

B. Semantic-guided Mix Query Initialization

As a pivotal component of transformer-based decoders, query initialization has been explored in various 3D tasks [39], [24], [44]. Generally, the query can be divided into parametric and non-parametric. The former initializes queries with learnable parameters, while the latter employs sampled point features as queries. FPS [29] is a typical selection mechanism in the point cloud domain. Parametric queries are flexible but suffer from slow convergence, while non-parametric queries are efficient but lack specific guidance and may include background points, leading to suboptimal results [22]. To tackle these problems, in this work, we propose a novel Semantic-guided Mix Query (SMQ) initialization scheme that implicitly obtains scene-aware queries, combining with learnable queries as the whole query set.

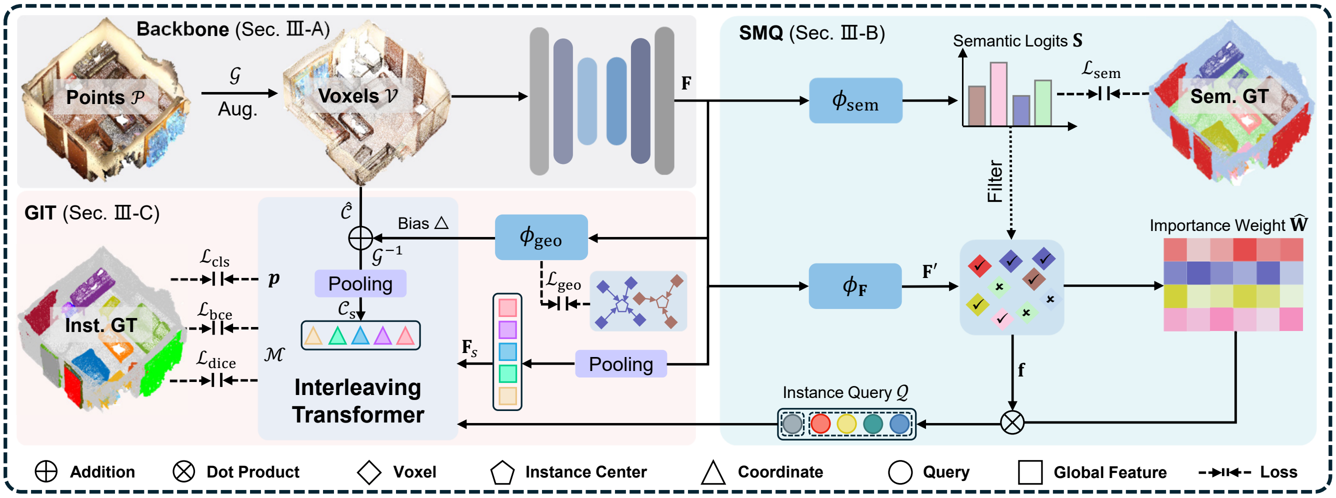


Fig. 2. **Overall architecture of SGIFormer.** The method comprises three main components: a symmetrical U-Net backbone, a Semantic-guided Mix Query (SMQ) initialization scheme, and a Geometric-enhanced Interleaving Transformer (GIT) decoder. Aug. in this figure denotes data augmentation. The 3D backbone extracts voxel-wise global features $\mathbf{F}$ from the input point cloud $\mathcal{P}$ (Sec. III-A). SMQ constructs instance queries $\mathcal{Q}$ with semantic guidance (Sec. III-B). GIT alternately refines the queries and scene features to enhance geometric information and capture fine-grained details. The final instance masks $\mathcal{M}$ and categories $\boldsymbol{p}$ are predicted by the decoder (Sec. III-C).

To achieve semantic guidance, we construct a branch that predicts voxel-wise class labels facilitating semantic prior formulated as $\mathbf{S} = \phi_{\texttt{sem}}(\mathbf{F}) \in \mathbb{R}^{m\times(c+1)}$, where $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^{m}$ and c represents the number of instance categories. Here, an extra label $\varnothing$ indicates the background. With available voxel-level semantic information, we train this branch using cross-entropy loss:

$$\mathcal{L}_{\texttt{sem}} = -\frac{1}{|\mathcal{V}|}\sum_{v_i\in\mathcal{V}}\sum_{j\in\mathcal{J}} \mathbf{1}_{\{y_{v_i}=j\}} \log(s_{i,j}), \tag{1}$$

where $s_{i,j}$ is the probability of voxel v_i belonging to class j, y_{v_i} is the ground truth label, $\mathcal{J} = \{1, \ldots, c+1\}$ is the set of class labels and $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function.

To yield the scene-aware queries $\mathbf{Q}^s = \{\boldsymbol{Q}_1^s, \ldots, \boldsymbol{Q}_{q_s}^s\} \in \mathbb{R}^{q_s\times d}$ from voxel features $\mathbf{F}$, we propose an implicit query initialization algorithm with semantic prediction results as guidance outlined in Alg. 1. Initially, the method calculates semantic score $\mathbf{S}' \in \mathbb{R}^{m\times c}$ for each voxel, excluding the background, and selects top-k favorable voxels with high semantics. Rather than straightforwardly applying a fixed threshold, we dynamically adjust the number of selected voxels αm using a ratio α based on the scale m of the scene. This adaptive approach enables us to efficiently filter out background noise and concentrate on foreground voxels $\mathbf{f} = \mathbf{F}'[idx] \in \mathbb{R}^{\alpha m\times d}$ with a higher probability of being relevant instances of interest. After the selection stage, the algorithm generates a sparse set of query weights $\mathbf{W} = [w_{u,v}]_{u=1,v=1}^{q_s,\alpha m}$ by:

$$\mathbf{W} = \psi(\mathbf{f})^{\top}, \tag{2}$$

where $\psi(\cdot) : \mathbb{R}^{\alpha m\times d} \mapsto \mathbb{R}^{\alpha m\times q_s}$ is a voxel-wise feature transformation implemented by MLP: $\psi(\mathbf{f}) = \texttt{ReLU}(\boldsymbol{w}_{\psi}^{\top}\mathbf{f} + \boldsymbol{b}_{\psi})$. We then normalize the weight by *softmax* function:

$$\hat{w}_{u,v} = \frac{\exp(w_{u,v})}{\sum_{j=1}^{\alpha m}\exp(w_{u,j})}, \tag{3}$$

$\hat{w}_{u,v}$ represents the importance weight of the v-th selected voxel $\mathbf{f}_v \subseteq \mathbf{f}$ for the u-th query. For simplicity, we denote the above operation as $\mathcal{I}$. Finally, the initialized scene-aware query $\mathbf{Q}^p$ is formulated by summing the chosen voxel features with the normalized weights:

$$\boldsymbol{Q}_u^s = \sum_{v=1}^{\alpha m} \hat{w}_{u,v}\cdot \mathbf{f}_v. \tag{4}$$

With the overall process, we can implicitly cluster the representative voxels into a set of scene-aware instance queries containing fine-grained foreground features. At the same time, our design can be more robust to noise in semantic prediction, avoiding potential error propagation indicated in [15].

Algorithm 1 Implicit Scene-aware Query Initialization

Require: voxel-wise features $\mathbf{F}$, semantic logits $\mathbf{S}$, ratio α, number of scene-aware query q_s
Ensure: initial scene-aware query $\mathbf{Q}^s$

```
/* get semantic score without background */
S' ∈ R^{m×c} ← softmax(S)[:, : −1]
j ← argmax(S', dim = −1)
/* filter out disruptive voxels */
score, idx ← topk(S'[:, j], αm)
/* project original voxel features */
F' ∈ R^{m×d} ← Linear(F)
f ∈ R^{αm×d} ← F'[idx]
/* get initialized scene-aware query */
Q^s ∈ R^{q_s×d} ← I(f) ⊙ f
Return: Q^s
```

The final query set $\mathcal{Q} = [\mathbf{Q}^s, \mathbf{Q}^l] \in \mathbb{R}^{q\times d}$ for transformer decoder is achieved by combining the scene-aware query $\mathbf{Q}^s$ with another set of randomly initialized learnable query $\mathbf{Q}^l \in \mathbb{R}^{q_l\times d}$, where $[\cdot,\cdot]$ denotes concatenation operation and $q = q_s + q_l$ means the total query number. The inclusion of parametric queries aims to capture missing local information and adapt the model to different scenes. We argue that our query initialization scheme benefits from both semantic cues and learnable queries. The guided queries can provide semantic prior through fine-grained voxel features, while additional parametric queries can mitigate the deficiency of the semantic misjudgment and improve the diversity of the query set. Furthermore, with the assistance of our implicitly generated queries, inter-query interaction during instance self-attention can extract more informative global contexts, reducing the dependence on heavily stacked layers. The mix query initialization module can achieve a balance between diverse queries and further promote subsequent transformer decoder.

C. Geometric-enhanced Interleaving Transformer Decoder

Generally, in DETR-based 3D instance segmentation methods [20], [24], [21], transformer decoder is utilized to refine instance queries with context information from the global scene features by stacking multiple layers. However, the vanilla decoder, whether for parametric query in [20], [23] or non-parametric query in [24], ignores the geometric property [45] of the input point clouds during query refinement. These methods directly update the query by attending to features of coarse-grained superpoints, which leads to a lack of inter-superpoint communication and the potential loss of fine-grained details in the input scene. Recognizing this limitation, we propose an interleaving transformer decoder capable of capturing instance geometric and detailed information more efficiently.

We consider the coordinate of voxels $\hat{\mathcal{C}}$ as the key component for geometry. Nevertheless, substantial variations in the scene range cause unstable training of raw voxel coordinates regression as in [25]. Instead, we estimate the bias vectors $\mathbf{\Delta} = \phi_{\text{geo}}(\mathbf{F}) \in \mathbb{R}^{m\times 3}$ of each voxel relative to the instance geometric center it belongs to. Since ground-truth instance centers are available, we apply ℓ_1 loss to supervise the geometric branch:

$$\mathcal{L}_{\text{geo}} = \frac{1}{\sum_{v_i\in\mathcal{V}} \mathbf{1}_{\{v_i\}}} \sum_{v_i\in\mathcal{V}} \mathbf{1}_{\{v_i\}} \|\mathbf{\Delta}_i - (\mathcal{O}^*_i - \hat{\mathcal{C}}_i)\|_1, \quad (5)$$

where $\mathcal{O}^*_i$ is the ground truth geometric center of the instance to which voxel v_i belongs, $\mathbf{1}_{\{v_i\}}$ is the indicator function to determine whether a voxel belongs to an instance. Then we shift the learned bias to raw coordinates $\hat{\mathcal{C}}$ to obtain refined coordinates, given by:

$$\hat{\mathcal{C}}_{\texttt{ref}} = \hat{\mathcal{C}} + \mathbf{\Delta}. \quad (6)$$

By refining the coordinates, we bring voxels belonging to the same instance closer, promoting the similarity between corresponding features. It is important to note that large-scale scene point clouds typically contain numerous voxels, although they carry rich information, directly leveraging them as scene features for the transformer decoder can be computationally demanding under the quadratic complexity of the attention mechanism. Therefore, we further cluster voxels into superpoints $\mathcal{S} = \{\mathcal{C}_{\text{s}}, \mathbf{F}_{\text{s}}\}$ aiming to reduce the complexity:

$$\mathcal{C}^i_{\text{s}} = \frac{1}{|\mathcal{N}_i|} \sum_{j\in\mathcal{N}_i} \mathcal{G}^{-1}(\hat{\mathcal{C}}^j_{\texttt{ref}}), \quad (7)$$

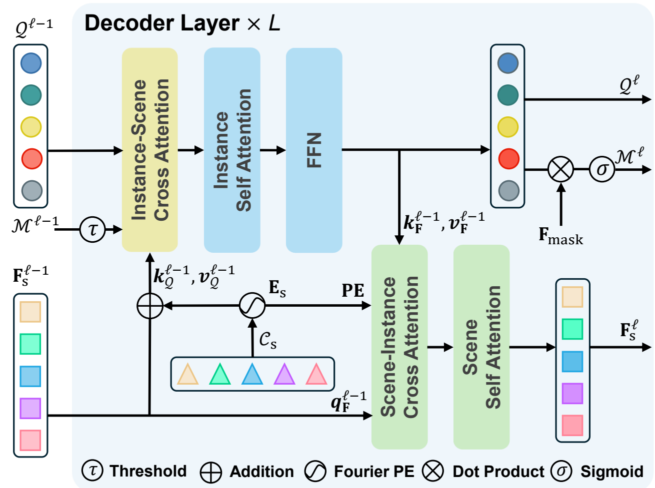


Fig. 3. **Geometric-enhanced Interleaving Transformer (GIT) decoder.** The diagram illustrates the detailed structure of our designed decoder. The decoder consists of L layers and employs an alternating update scheme to capture fine-grained features. In each layer, the instance queries $\mathcal{Q}$, and scene features $\mathbf{F}_{\text{s}}$ are iteratively refined by incorporating shifted coordinates embedding $\mathbf{E}_{\text{s}}$. The refined instance queries are then utilized to predict masks $\mathcal{M}$ and categories $\boldsymbol{p}$.

where $\mathcal{N}_i$ is the set of point indices belonging to the i-th superpoint, and $\mathcal{G}^{-1}(\cdot)$ is the inverse mapping that assigns the same coordinates to the points within the same voxel. We apply a similar operation following a non-linear transformation ϕ_{s} to features $\mathbf{F}$ to obtain $\mathbf{F}_{\text{s}}$. $\mathcal{C}_{\text{s}} \in \mathbb{R}^{n_{\text{s}}\times 3}$ and $\mathbf{F}_{\text{s}} \in \mathbb{R}^{n_{\text{s}}\times d}$ are superpoint coordinates and features, respectively. We consider $\mathbf{F}_{\text{s}}$ as scene features and send them with the overall instance query set $\mathcal{Q}$ into our tailored interleaving transformer decoder illustrated in Fig. 3.

Our decoder consists of L layers, each comprising a query refinement block (*cf*. Fig. 3 upper part) and a scene feature update block (*cf*. Fig. 3 lower part). These blocks are updated alternately to enhance geometric information and capture fine-grained details. The query refinement block is specifically designed to update the mix instance query $\mathcal{Q}$ by progressively attending to the scene features $\mathbf{F}_{\text{s}}$. Unlike previous transformer-based methods [20], [24] that overlook the position of superpoints, we incorporate the refined superpoint coordinates to provide geometric information, helping better instance localization. Practically, we exploit Fourier positional encoding [46] to get the position embedding $\mathbf{E}_{\text{s}} \in \mathbb{R}^{n_{\text{s}}\times d}$:

$$\mathbf{E}_{\text{s}} = \phi_{\text{E}}(\mathbf{Fourier}(\mathcal{C}_{\text{s}})). \quad (8)$$

We further add resulting embedding to original scene features as geometry reinforced features $\mathbf{F}_{\text{s}}^{\ell-1} + \mathbf{E}_{\text{s}}$. Here, compared to static positional information, our $\mathbf{E}_{\text{s}}$ is derived from the previously predicted bias vector in Eq. 6, which means the positional encoding is dynamically updated based on the estimated bias vectors, allowing for additional information exchange between the backbone and decoder. This technique makes the auxiliary tasks and instance segmentation complementary. By applying linear projection to the enhanced features, we obtain keys $\boldsymbol{k}_{\mathcal{Q}}^{\ell-1}$ and values $\boldsymbol{v}_{\mathcal{Q}}^{\ell-1}$ respectively for the queries $\mathcal{Q}^{\ell-1}$. We utilize masked cross-attention proposed in [37] to refine the queries constrained by attention mask $\mathcal{A}^{\ell-1}$ from the last predicted instance mask $\mathcal{M}^{\ell-1}$:

$$\mathcal{A}_{i,j}^{\ell-1} = -\infty \cdot [\mathcal{M}_{i,j}^{\ell-1} < \tau], \tag{9}$$

where $[\cdot]$ denotes Iverson Brackets and τ is a threshold. This operation can ensure each query only focuses on relevant context information, improving model robustness. Self-attention and feed-forward networks are also applied to facilitate the interaction between queries, avoiding duplicate instances and capturing discriminative representations.

Although using superpoint features as global information can reduce computational complexity, we argue that simply pooling voxels into superpoints (*cf*. Eq. 7) loses fine-grained features and lacks inter-superpoint communication. Therefore, we introduce a superpoint update block to refine $\mathbf{F}_{\mathrm{s}}^{\ell-1}$. Benefiting from our semantic-guided mix query initialization scheme, the refined query $\mathcal{Q}^{\ell}$ already contains fine-grained details from voxel features, which can mitigate the loss of information. By attending to the refined queries $\mathcal{Q}^{\ell}$ with superpoint position embedding $\mathbf{E}_{\mathrm{s}}$, we can obtain updated scene features $\mathbf{F}_{\mathrm{s}}^{\ell}$, which are then passed to the next layer. During the query refinement stage, our decoder iteratively participates in geometric information to emphasize the localization, while during the superpoint update stage, fine-grained features can be captured. Working interleavingly, our proposed decoder can thoroughly interact with instance queries and superpoints to improve performance.

Given refined queries $\mathcal{Q}^{\ell}$ after each decoder layer and mask-aware features $\mathbf{F}_{\mathtt{mask}}$, we can get final binary instance masks by:

$$\mathcal{M}^{\ell} = \left\{ b_{i,j} = \left[\sigma\left(\mathbf{F}_{\mathtt{mask}} \cdot \phi_{\mathtt{m}}(\mathcal{Q}^{\ell})^{\top}\right)_{i,j} > \tau \right] \right\} \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function, τ is the same as in Eq. 9, and $\phi_{\mathtt{m}}(\cdot)$ is a normalization layer. Besides, we predict corresponding instance categories $\boldsymbol{p}^{\ell} = \phi_{\mathtt{cls}}(\mathcal{Q}^{\ell})$ implemented by a shared shallow MLP.

D. Training and Inference

Training Objectives. Following exiting transformer-based methods [20], [21], [22], we use bipartite graph matching [47] for instance pairing during training. Note that we perform matching at every decoder layer. We omit the layer index ℓ for simplicity in the following description. The bipartite matching cost between the i-th predicted instance and the j-th ground truth is defined as:

$$\begin{aligned} \mathcal{U}_{i,j} = &- \lambda_{\mathtt{cls}} \cdot p_{i,j} + \lambda_{\mathtt{bce}} \mathbf{BCE}(\mathcal{M}_i, \mathcal{M}_j^{gt}) \\ &+ \lambda_{\mathtt{dice}} \mathbf{Dice}(\mathcal{M}_i, \mathcal{M}_j^{gt}), \end{aligned} \tag{11}$$

where $p_{i,j} \subseteq \boldsymbol{p}$ is the predicted probability, $\mathcal{M}_i$ and $\mathcal{M}_j^{gt}$ are the i-th predicted mask and the j-th ground truth mask, respectively. $\lambda_{\mathtt{cls}}$, $\lambda_{\mathtt{bce}}$, and $\lambda_{\mathtt{dice}}$ are hyper-parameters to balance classification, binary cross-entropy $\mathbf{BCE}(\cdot,\cdot)$, and dice $\mathbf{Dice}(\cdot,\cdot)$. We then use Hungarian algorithm [31] on the cost matrix $\mathcal{U}$ to find the optimal one-to-one matching between predicted instances and the ground truth instances, formulated as:

$$\begin{aligned} \hat{\boldsymbol{\kappa}} = \arg\min_{\kappa} \sum_{i=1}^{q} \mathcal{U}_{i,\kappa_i} \\ \text{s.t. } \forall i \neq j, \kappa_i \neq \kappa_j. \end{aligned} \tag{12}$$

where $\hat{\boldsymbol{\kappa}}$ is the optimal matching result. Finally, we calculate the classification cross-entropy loss $\mathcal{L}_{\mathtt{cls}}$, binary cross-entropy loss $\mathcal{L}_{\mathtt{mask}}$, and dice loss $\mathcal{L}_{\mathtt{dice}}$ for each matched pair. The overall loss function is given by:

$$\begin{aligned} \mathcal{L} = &\lambda_{\mathtt{aux}}(\mathcal{L}_{\mathtt{sem}} + \mathcal{L}_{\mathtt{geo}}) \\ &+ \sum_{\ell=1}^{L} \lambda_{\mathtt{cls}} \mathcal{L}_{\mathtt{cls}}^{\ell} + \lambda_{\mathtt{bce}} \mathcal{L}_{\mathtt{bce}}^{\ell} + \lambda_{\mathtt{dice}} \mathcal{L}_{\mathtt{dice}}^{\ell}, \end{aligned} \tag{13}$$

where $\lambda_{\mathtt{aux}}$ is used to balance the auxiliary loss and main loss.

Inferences. By taking a point cloud as input, SGIFormer directly predicts K instances with classification scores $\{\boldsymbol{p}_i\}_{i=1}^{K}$ and associated masks $\{\boldsymbol{m}_i\}_{i=1}^{K}$ during inference. Besides, we assign each mask a score $z_i \in [0,1]$ by averaging superpoint probability exceeding the threshold and obtain the final score $r_i = \boldsymbol{p}_i \cdot z_i$. Finally, we map the predicted masks back to the original point cloud by the inverse mapping $\tilde{\boldsymbol{m}}_i = \mathcal{G}^{-1}(\boldsymbol{m}_i)$.

IV. Experiment

A. Datasets

To verify the effectiveness of our proposed method, we conduct experiments on three indoor datasets: ScanNet V2 [6], ScanNet200 [49], and ScanNet++ [7]. *ScanNet V2* [6] is a widely used dataset containing 1201 fully labeled indoor scans for training and 312 scans for validation, respectively. Each scene is carefully annotated with 20 semantic categories, out of which 18 are instance classes (excluding *wall* and *floor*). ScanNet200 [49] is an extended version of ScanNet V2 with more fine-grained annotations and long-tail distribution covering 198 classes for 3D instance segmentation evaluation. We use the training and validation splits provided by the official toolkit for both datasets. *S3DIS* [42] dataset consists of 6 areas with 271 rooms, and the scans are annotated by 13 instance categories. We follow previous works to assess the segmentation performance on scenes from Area 5. *ScanNet++* [7] is a recently introduced indoor dataset providing sub-millimeter resolution 3D scan geometry along with 84 categories of dense annotations for evaluating instance segmentation quality. The dataset is divided into training, validation, and testing splits of 230, 50, and 50 scenes. Compared to previous datasets, ScanNet++ presents more challenges due to its large-scale and complicated layouts. It provides more fine-grained details and high-quality scans, making it a realistic and demanding dataset for evaluating 3D instance segmentation.

B. Evaluation Metrics

For evaluating the performance of our framework, we utilize standard average precision mAP, AP_{50} and AP_{25}, which are commonly used in point cloud instance segmentation tasks [6], [49], [7]. AP_{50} and AP_{25} represent the scores obtained with intersection over union (IoU) thresholds of 50% and 25%, respectively, while mAP is calculated as the average score

TABLE I

COMPARISON WITH STATE-OF-THE-ART METHODS ON SCANNET V2 [6] HIDDEN TEST SET. THE BEST RESULTS ARE SHOWN IN **BOLD**. RESULTS ARE ASSESSED ON JULY 10TH, 2024.

Methods	mAP	AP_{50}	bath	bed	bkshf	cabinet	chair	counter	curtain	desk	door	other	picture	fridge	s. cur.	sink	sofa	table	toilet	window
PointGroup [13]	40.7	63.6	63.9	49.6	41.5	24.3	64.5	2.1	57.0	11.4	21.1	35.9	21.7	42.8	66.6	25.6	56.2	34.1	86.0	29.1
HAIS [14]	45.7	69.9	70.4	56.1	45.7	36.4	67.3	4.6	54.7	19.4	30.8	42.6	28.8	45.4	71.1	26.2	56.3	43.4	88.9	34.4
OccuSeg [17]	48.6	67.2	80.2	53.6	42.8	36.9	70.2	20.5	33.1	30.1	37.9	47.4	32.7	43.7	**86.2**	48.5	60.1	39.4	84.6	27.3
TD3D [11]	48.9	75.1	85.2	51.1	43.4	32.2	73.5	10.1	51.2	35.5	34.9	46.8	28.3	51.4	67.6	26.8	67.1	51.0	90.8	32.9
SoftGroup [15]	50.4	76.1	66.7	57.9	37.2	38.1	69.4	7.2	**67.7**	30.3	38.7	53.1	31.9	58.2	75.4	31.8	64.3	49.2	90.7	38.8
SSTNet [16]	50.6	69.8	73.8	54.9	49.7	31.6	69.3	17.8	37.7	19.8	33.0	46.3	57.6	51.5	85.7	**49.4**	63.7	45.7	94.3	29.0
SPFormer [20]	54.9	77.0	74.5	64.0	48.4	39.5	73.9	31.1	56.6	33.5	46.8	49.2	55.5	47.8	74.7	43.6	71.2	54.0	89.3	34.3
ISBNet [48]	55.9	75.7	93.9	65.5	38.3	42.6	76.3	18.0	53.4	38.6	49.9	50.9	**62.1**	42.7	70.4	46.7	64.9	57.1	94.8	**40.1**
Mask3D [21]	56.6	78.0	92.6	59.7	40.8	42.0	73.7	23.9	59.8	38.6	45.8	**54.9**	56.8	**71.6**	60.1	48.0	64.6	57.5	92.2	36.4
OneFormer3D [24]	56.6	**80.1**	78.1	69.7	**56.2**	43.1	77.0	33.1	40.0	37.3	52.9	50.4	56.8	47.5	73.2	47.0	**76.2**	55.0	87.1	37.9
MAFT [23]	57.8	77.4	77.8	64.9	52.0	44.9	76.1	25.3	58.4	39.1	**53.0**	47.2	61.7	49.9	79.5	47.3	74.5	54.8	96.0	37.4
QueryFormer [22]	58.3	78.7	92.6	**70.2**	39.3	**50.4**	73.3	27.6	52.7	37.3	47.9	53.4	53.3	69.7	72.0	43.6	74.5	59.2	95.8	36.3
SGIFormer (Ours)	**58.6**	79.9	**100.0**	59.3	44.0	48.0	**77.1**	**34.5**	43.7	**44.4**	49.5	54.8	57.9	62.1	72.0	40.9	71.2	**59.3**	**96.0**	39.5

with a set of IoU thresholds from 50% to 95% with an increased step size of 5%. It is noted that higher values of these metrics indicate superior model performance. In addition to performance metrics, we also report the model size and average inference time on the real-world ScanNet V2 validation set to evaluate the model efficiency.

C. Implementation and Training Details

We implement our SGIFormer based on the Pointcept [50] toolkit by PyTorch. Following previous work [24], [12], we train our model for 510 epochs using the AdamW optimizer with a weight decay of 0.05. We employ a polynomial learning rate scheduler with a base value of 0.9 for the initialized learning rate $3e^{-4}$, but for voxel-wise heads, we set the learning rate to $3e^{-3}$. The model is trained using 4 NVIDIA RTX 4090 GPUs with a batch size of 12.

For a fair comparison with prior methods, we provide two versions of our model: SGIFormer and SGIFormer-L, specifically designed for ScanNet V2 and ScanNet200 datasets. The only difference between them lies in the choice of feature extractor architecture. SGIFormer adopts a 5-layer Sparse Convolution U-Net [43] as the backbone for both datasets, following [20], [23], [24]. While SGIFormer-L uses a 7-layer Sparse Convolution U-Net and Res16UNet34C of MinkowskiEngine [51] for ScanNet V2 and ScanNet200, respectively, following [21], [24], [12], [15], [48]. But for ScanNet++, we just provide a smaller version. The input point clouds are voxelized with a voxel size of 0.02m for all experiments, and we utilize graph-based over-segmentation [32] to cluster points into superpoints. We apply standard augmentation techniques for point coordinates, including random dropout, horizontal flipping, random rotation around the z-axis, random translation, scaling, and elastic distortion. Our color augmentations include random jittering and auto-contrast following normalization. During the training process, we randomly crop $250k$ points from the original input for each scene in ScanNet V2 and ScanNet200 to reduce the memory consumption while sampling 0.8× points for ScanNet++. We set the number of scene-aware queries q_s and learnable queries q_l to 200 and 200, respectively, and use 3 stacked layers to yield the decoder. For the selection ratio α, we set it to 0.4. To balance the loss terms, we set $\lambda_{\texttt{cls}}, \lambda_{\texttt{bce}}, \lambda_{\texttt{dice}}$ and $\lambda_{\texttt{aux}}$ to 0.8, 1.0, 1.0 and 0.4, respectively. The same hyperparameters are used for all experiments unless otherwise stated.

TABLE II

COMPARISON WITH STATE-OF-THE-ART METHODS ON SCANNET V2 [6] VALIDATION SPLIT. P, G, AND T MEAN PROPOSAL-, GROUP-, AND TRANSFORMER-BASED METHODS, RESPECTIVELY. THE BEST RESULTS ARE SHOWN IN **BOLD**, AND THE SECOND BEST ARE <u>UNDERLINED</u>.

Methods	Year	Types	mAP	AP_{50}	AP_{25}
3D-SIS [8]	2019	P	-	18.7	35.7
GSPN [9]	2019	P	19.3	37.8	53.4
TD3D [11]	2024	P	47.3	71.2	81.9
Spherical Mask [12]	2024	P	**62.3**	<u>79.9</u>	<u>88.2</u>
JSNet++ [19]	2022	G	-	39.2	56.8
PointGroup [13]	2020	G	34.8	56.9	71.3
SSTNet [16]	2021	G	49.4	64.3	74.0
HAIS [14]	2021	G	43.5	64.4	74.6
SoftGroup [15]	2022	G	45.8	67.6	78.9
PBNet [34]	2023	G	54.3	70.5	78.9
Mask3D [21]	2023	T	55.2	73.7	82.9
SPFormer [20]	2023	T	56.3	73.9	82.9
QueryFormer [22]	2023	T	56.5	74.2	83.3
EipFormer [26]	2023	T	56.9	74.6	-
3IS-ESSS [25]	2023	T	56.1	75.0	83.7
EASE [52]	2024	T	60.2	77.2	-
OneFormer3D [24]	2024	T	59.3	78.1	86.4
SGIFormer (Ours)	2024	T	58.9	78.4	86.2
SGIFormer-L(Ours)	2024	T	<u>61.0</u>	**81.2**	**88.9**

D. Comparisons with State-of-the-art Methods

We present a comprehensive comparison of our method with state-of-the-art 3D instance segmentation models, including proposal-based (P), group-based (G), and transformer-based (T) methods on ScanNet V2 validation and hidden test splits as illustrated in Tab. I and Tab. II. We report average mAP, AP_{50} for the overall categories and class-wise mAP scores on the test set in Tab. I. SGIFormer obtains the highest average mAP of 58.6% and achieves the best scores for 6 out of 18 classes. The detailed quantitative results in Tab. II demonstrate

TABLE III
INFERENCE TIME AND MEMORY COMPARISON. WE RECORD THE AVERAGE INFERENCE TIME PER SCENE AND PEAK GPU MEMORY ON SCANNET V2 [6] VALIDATION SET USING A SINGLE NVIDIA RTX 4090 GPU. SP. EXT. REFERS TO SUPERPOINT EXTRACTION.

Methods	Parts	Device	Time (ms)	Total (ms)↓	Memory (MB)↓	AP_{50} ↑
SPFormer [20]	Sp. Ext. Model	CPU GPU	151.85 41.30	**193.15**	**1238**	73.9
OneFormer3D [24]	Sp. Ext. Model	CPU GPU	151.85 53.13	204.98	1556	78.1
Spherical Mask [12]	Sp. Ext. Model	CPU GPU	151.85 80.10	231.95	1567	<u>79.9</u>
SGIFormer (Ours)	Sp. Ext. Model	CPU GPU	151.85 42.14	<u>193.99</u>	<u>1418</u>	78.4
SGIFormer-L(Ours)	Sp. Ext. Model	CPU GPU	151.85 49.00	200.85	1492	**81.2**

TABLE IV
COMPARISON WITH PREVIOUS METHODS ON SCANNET200 [49] VALIDATION SET AND S3DIS [42] AREA5. THE BEST RESULTS ARE SHOWN IN **BOLD**, AND THE SECOND BEST ARE <u>UNDERLINED</u>. * MEANS THE RESULTS OF PRETRAINED ON SCANNET V2.

Methods	ScanNet200 *Val*		
	mAP	AP_{50}	AP_{25}
PointGroup [13]	-	24.5	-
SPFormer [20]	25.2	33.8	39.6
TD3D [11]	23.1	34.8	40.4
Mask3D [21]	27.4	37.0	42.3
QueryFormer [22]	28.1	37.1	43.4
SGIFormer (Ours)	<u>28.9</u>	<u>38.6</u>	<u>43.6</u>
SGIFormer-L(Ours)	**29.2**	**39.4**	**44.2**

Methods	S3DIS	
	mAP	AP_{50}
PointGroup [13]	-	57.8
SoftGroup [15]	51.6	66.1
PBNet [34]	<u>53.5</u>	66.4
SPFormer [20]	-	66.8
OneFormer3D* [24]	-	68.5
MAFT [23]	-	<u>69.1</u>
SGIFormer (Ours)	**56.2**	**69.2**

that SGIFormer-L achieves the best performance in terms of AP_{50} and AP_{25}, surpassing other methods by a margin of 1.3% and 0.7%, respectively. Although our model is slightly inferior to Spherical Mask [12] in terms of mAP, it is worth noting that our method significantly improves inference speed by 31ms per scene illustrated in Tab. III, making it well-suited for latency-sensitive scenarios. And SGIFormer and SGIFormer-L achieve lower memory consumption compared to OneFormer3D and Spherical Mask, which is crucial for real-world applications. The speed and memory advantages are attributed to our model's end-to-end design, whereas Spherical Mask relies on a coarse-to-fine strategy involving complex point migration and mask assembly operations. Although compared with SPFormer, our method consumes slightly higher memory, the performance is significantly improved. Furthermore, compared with other transformer-based counterparts, our variant SGIFormer achieves equivalent performance as the state-of-the-art OneFormer3D [24], but with fewer parameters and lower latency. This advantage is primarily due to our proposed geometric enhanced interleaving decoder, which reduces the dependence on heavy transformer layers.

We then evaluate our method on ScanNet200 dataset to verify its robustness and generalization, and the results are showcased in Tab. IV. Our SGIFormer-L, engaged with SGIFormer, consistently outperforms other methods across all metrics, demonstrating its superiority in handling sophisticated semantics and long-tailed distributions. Concretely, SGIFormer-L achieves 1.1% and 2.3% improvements in mAP and AP_{50}, respectively. Our method also achieves comparable performance on S3DIS Area5, as shown in Tab. IV, which further validates the superiority of our method. In Tab. V, we benchmark SGIFormer against leading segmentation algorithms on ScanNet++ dataset for the effectiveness in processing large-scale and high-fidelity scenes. Our method achieves state-of-the-art performance on both the validation and hidden test sets, achieving an AP_{50} of 37.5% and 41.1%, respectively.

TABLE V
COMPARISON WITH STATE-OF-THE-ART METHODS ON SCANNET++ [7] BENCHMARK. THE BEST RESULTS ARE SHOWN IN **BOLD**, AND THE SECOND BEST RESULTS ARE <u>UNDERLINED</u>. RESULTS ON THE HIDDEN TEST SET ARE ASSESSED ON JUNE 24TH, 2024. † DENOTES METRICS ARE REPORTED BY [7].

Instance Seg.	ScanNet++ *Val*†			ScanNet++ *Test*		
Methods	mAP	AP_{50}	AP_{25}	mAP	AP_{50}	AP_{25}
PointGroup [13]	-	14.8	-	8.9	14.6	21.0
HAIS [14]	-	16.7	-	12.1	19.9	29.5
SoftGroup [15]	-	<u>23.7</u>	-	16.7	29.7	38.9
BFL [53]	-	-	-	<u>22.2</u>	<u>32.8</u>	<u>42.5</u>
SGIFormer (Ours)	**23.9**	**37.5**	**46.6**	**27.3**	**41.0**	**48.4**

TABLE VI
VARIANTS OF QUERY INITIALIZATION AND DECODER DESIGNS. LEARN., FPS, R.S. AND SMQ INDICATE LEARNABLE QUERY [20], FPS-BASED QUERY [21], RANDOM SELECTION MECHANISM [54], AND SEMANTIC-GUIDED MIX QUERY, RESPECTIVELY. THE BEST RESULTS ARE SHOWN IN **BOLD**, AND THE SECOND BEST RESULTS ARE <u>UNDERLINED</u>.

#	*Query Initialization*				*VT*			*GIT*		
	Learn.	FPS	R.S.	SMQ	mAP	AP_{50}	AP_{25}	mAP	AP_{50}	AP_{25}
1	✓				56.6	76.6	85.2	57.8	77.3	85.7
2		✓			57.2	77.1	85.7	56.5	77.8	85.8
3	✓	✓			57.1	77.7	85.6	<u>57.9</u>	<u>77.9</u>	<u>86.1</u>
4			✓		57.1	76.8	84.9	57.1	77.3	85.3
Ours				✓	<u>57.9</u>	77.6	85.6	**58.9**	**78.4**	**86.2**

E. Ablation Studies

Impact of core designs. We conduct comprehensive ablation studies of SGIFormer on ScanNet V2 validation set, focusing on evaluating the impact of core designs within our framework. In Tab. VI, we explore various combinations of query initialization and decoder designs. Considering alternative parametric learnable query [20], FPS-based non-parametric query [21], random query selection mechanism [24], vanilla transformer decoder, as well as our proposed semantic-guided mix query (SMQ) initialization and geometric-enhanced interleaving transformer (GIT) decoder, we assess 9 different variants besides our method. When combining with the vanilla transformer decoder, we observe that our proposed SMQ (#Ours-VT) achieves at least a 0.5% improvement in terms of AP_{50} compared to learnable query (#1-VT), FPS-based query (#2-VT), and randomly selected query (#4-VT). Since our SMQ is designed in a hybrid form, we assemble the FPS-based query with the learnable query while maintaining the same number of queries to investigate SMQ's impact. Results of #3-VT and #Ours-VT show that our SMQ slightly improves mAP, indicating that our

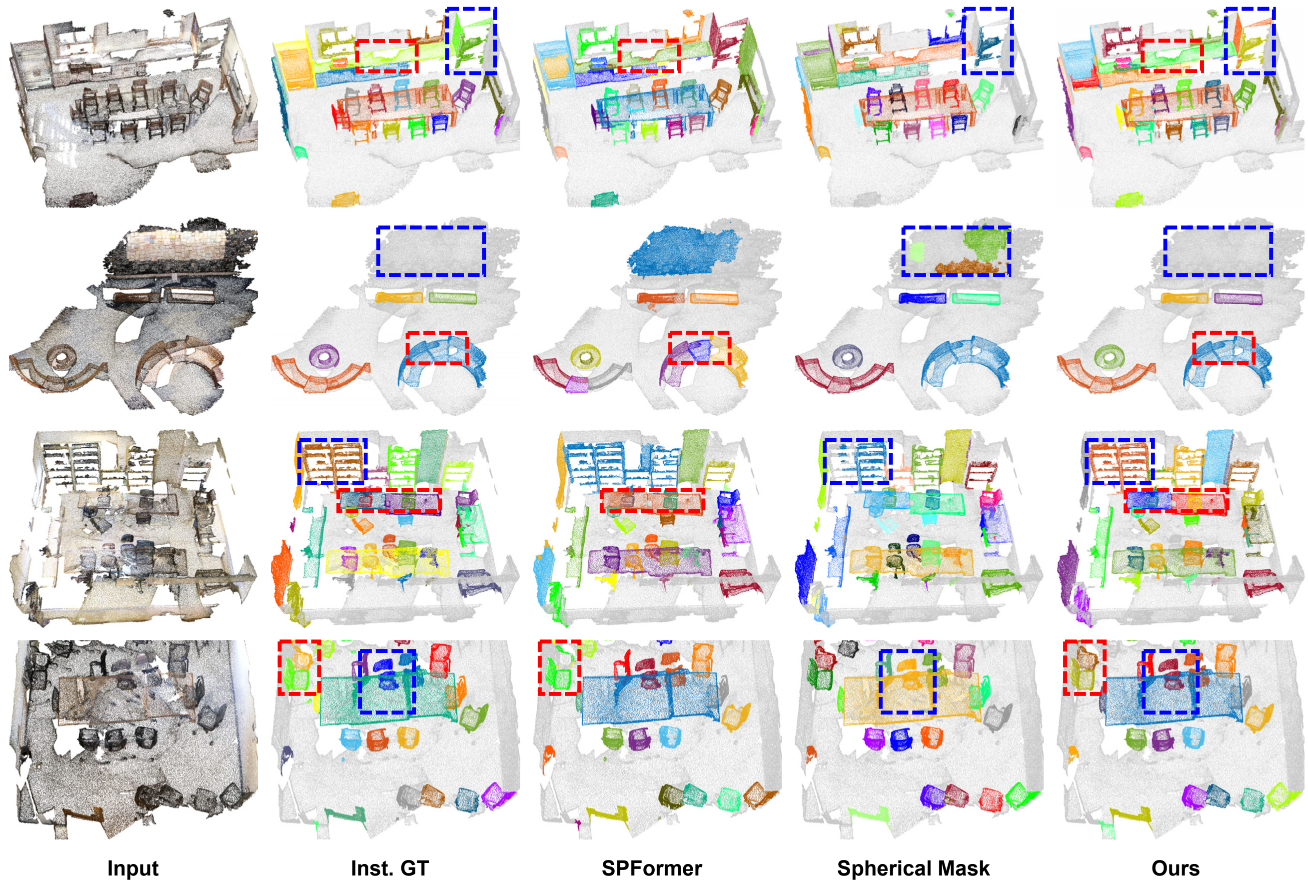


Fig. 4. **Visualization comparison on ScanNet V2 validation split.** We visualize the instance segmentation results of SGIFormer (ours), SPFormer [20], and Spherical Mask [12]. Inst. GT means instance ground truth, and different colors indicate different instance IDs. The comparison with SPFormer [20] is highlighted in red, while the comparison with Spherical Mask [12] is highlighted in blue. We use different colors in each subfigure to distinguish instances.

TABLE VII
EFFECT OF REMOVING EACH COMPONENT IN SGIFORMER. GEO. INDICATES PROGRESSIVELY GEOMETRIC ENHANCEMENT. THE BEST RESULTS ARE SHOWN IN **BOLD**.

$\mathbf{Q}^s$	$\mathbf{Q}^l$	w/ Geo.	mAP	AP_{50}	AP_{25}
w/ interleaving update					
✓	✓	✓	**58.9**	**78.4**	**86.2**
✓	✓	✗	57.6	78.2	86.1
✓	✗	✓	57.8	77.9	86.1
✗	✓	✓	56.6	77.8	85.6
w/o interleaving update					
✓	✓	✓	56.8	78.1	85.7

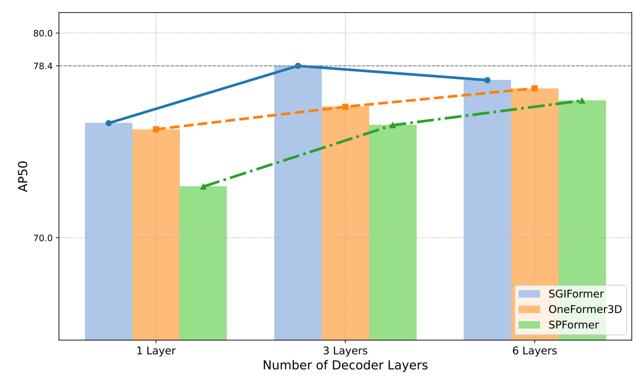
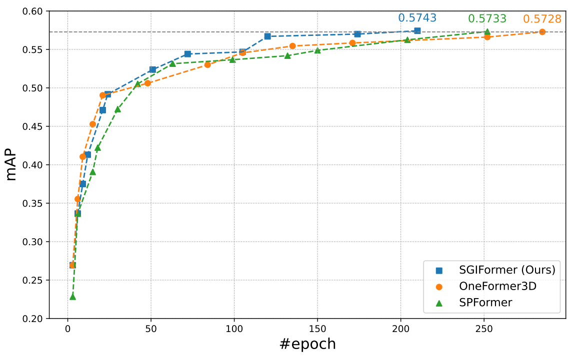


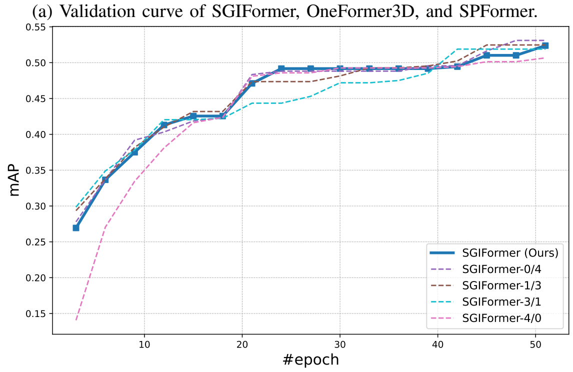
Fig. 5. Comparison of AP_{50} scores with different stacked decoder layers.

novel paradigm assists the ability to capture more informative context. Moreover, we validate the effectiveness of GIT by incorporating it with different initialization methods. The results clearly show that our interleaving mechanism boosts most of the metrics when combined with different query initialization strategies. In particular, our SGIFormer (#Ours-GIT) outperforms all other variants by at least 1.0% and 0.5% in mAP and AP_{50}. These results confirm the superior capability of our model in enhancing instance localization and demonstrate the plug-and-play ability of our proposed components.

Effect of removing each component. As depicted in Tab. VII, we perform a series of experiments by subsequently removing different components, including geometric enhancement (w/ Geo.), scene-aware queries $\mathbf{Q}^s$, and learnable queries $\mathbf{Q}^l$. This step-by-step procedure allows us to assess the individual contributions of each component to the overall results. Upon discarding geometric enhancement, we observe a noticeable performance drop across all metrics, particularly in terms of mAP, indicating the necessity of our proposed progressive geometric refinement mechanism. In addition, the table shows that our implicitly initialized scene-aware queries $\mathbf{Q}^s$ play a crucial role in improving the accuracy of instance segmentation, while the removal of learnable queries $\mathbf{Q}^l$ has a relatively minor impact. This suggests that our scene-aware queries are more effective in capturing instance-level information, and a combination of both queries can further enhance the capability of our interleaving decoder for better aggregating instance features from the global context. In addition, we investigate the impact of the interleaving update mechanism by removing the scene feature update block. The results in Tab. VII show that the model without the interleaving update achieves a dramatic performance drop in terms of mAP, indicating the importance of our proposed mechanism in capturing fine-grained details.



(a) Validation curve of SGIFormer, OneFormer3D, and SPFormer.



(b) Validation curve of the first 51 epochs for different mix ratios.

Fig. 6. **Convergence analysis of SGIFormer.** We compare the convergence speed of SGIFormer with OneFormer3D and SPFormer on ScanNet V2 validation set and investigate the impact of different mix ratios.

Bias Estimation. As discussed in Sec. III-C (*cf*. Eq. 6), we argue that refining the coordinates by shifting learned bias can intuitively lead to more discriminative voxel representations, ultimately resulting in improved 3D instance segmentation. To validate this claim, we conduct an ablation study by replacing the bias estimation with raw coordinate regression, and the quantitative results are presented in Tab. VIII. It is evident that our design can significantly improve the performance by 1.5% for mAP. We also compare the performance of our method with the utilization of original coordinates in Tab. VIII. Although our geometric enhancement introduces a slight increase in the computational cost, the performance gain justifies the additional overhead.

Selection ration. We then analyze the influence of selection ratio α defined in Alg. 1. We vary α from 0.2 to 1.0 with a step size of 0.2 to explore its impact, where a value of 1.0 means the utilization of all voxels for scene-aware queries. The results presented in Tab. IX left demonstrate that the model achieves the best performance when α is set to 0.4, which indicates that a moderate selection ratio can effectively filter out disruptive and redundant voxels to balance the trade-off between the global context and local details.

TABLE VIII
COMPARISON OF USING DIFFERENT GEOMETRIC ESTIMATION METHODS AND ORIGINAL COORDINATES.

Methods	mAP	AP_{50}	AP_{25}	Memory(MB)↓
Original Coordinates	56.4	76.9	85.5	**1414**
Coordinate Regression	57.4	77.4	85.0	1418
Bias Estimation	**58.9**	**78.4**	**86.2**	1418
Improvement	*+1.5*	*+1.0*	*+0.7*	*+4*

TABLE IX
IMPACT ANALYSIS OF SELECTION RATIO α AND MIX RATIO ON THE INSTANCE SEGMENTATION PERFORMANCE.

Ratio	α					mix ratio				
	0.2	0.4	0.6	0.8	1.0	0/4	1/3	2/2	3/1	4/0
mAP	57.4	**58.9**$^{\uparrow 1.2}$	<u>57.7</u>	57.0	57.0	56.6	57.6	**58.9**$^{\uparrow 1.2}$	57.0	<u>57.7</u>
AP_{50}	78.0	**78.4**$^{\uparrow 0.3}$	<u>78.1</u>	77.9	77.9	77.3	77.5	**78.4**$^{\uparrow 0.4}$	77.1	<u>78.0</u>
AP_{25}	86.0	<u>86.2</u>$^{\downarrow 0.7}$	85.8	<u>86.2</u>	**86.9**	<u>86.0</u>	85.8	**86.2**$^{\uparrow 0.2}$	85.1	85.9

Layer and query numbers. In Tab. X, we investigate the effect of query numbers q and stacked layers L in the decoder. During these experiments, we maintain the same number of scene-aware and learnable queries. We can observe from the results that the model achieves the worst performance with only one decoder layer, and increasing the query number will not bring significant improvement. While slightly increasing the number of stacked layers can boost the performance, the model with 3 layers and 400 queries achieves the best results. However, further increasing the layers to 6 even degrades the performance. This decline can be attributed to the excessive number of layers that will devastate our interleaving mechanism and introduce more noise to the instance features. Moreover, we compare the AP_{50} scores of our method with SPFormer [20] and OneFormer3D [24] by varying the stacked decoder layers, as shown in Fig. 5. The results present that our method consistently outperforms the baselines across different numbers of layers and shows the advantage of reducing the dependence on heavy decoder layers.

Convergence analysis and mix ratio. We further analyze the convergence speed of SGIFormer with SPFormer [20] and OneFormer3D [24] in Fig. 6 (a). The results demonstrate that, in the initial training stage, SGIFormer converges slower than OneFormer3D due to the integrated coarse learnable queries. However, our method can catch up and surpass the baselines quickly and achieve the best performance. We also explore the impact of different mix ratios in Tab. IX right. We set the mix ratio to $0/4, 1/3, 2/2, 3/1, 4/0$, which means the ratio of scene-aware queries and learnable queries in order to investigate the influence, and we find that the model achieves the best performance when the mix ratio is set to $2/2$. We then visualize the validation curve of the first 51 epochs for different mix ratios in Fig. 6 (b), which shows that the convergence speed is not significantly affected by mix ratio.

F. Qualitative Results

Fig. 4 illustrates several representative qualitative examples of our method on ScanNet V2 validation set, comparing with

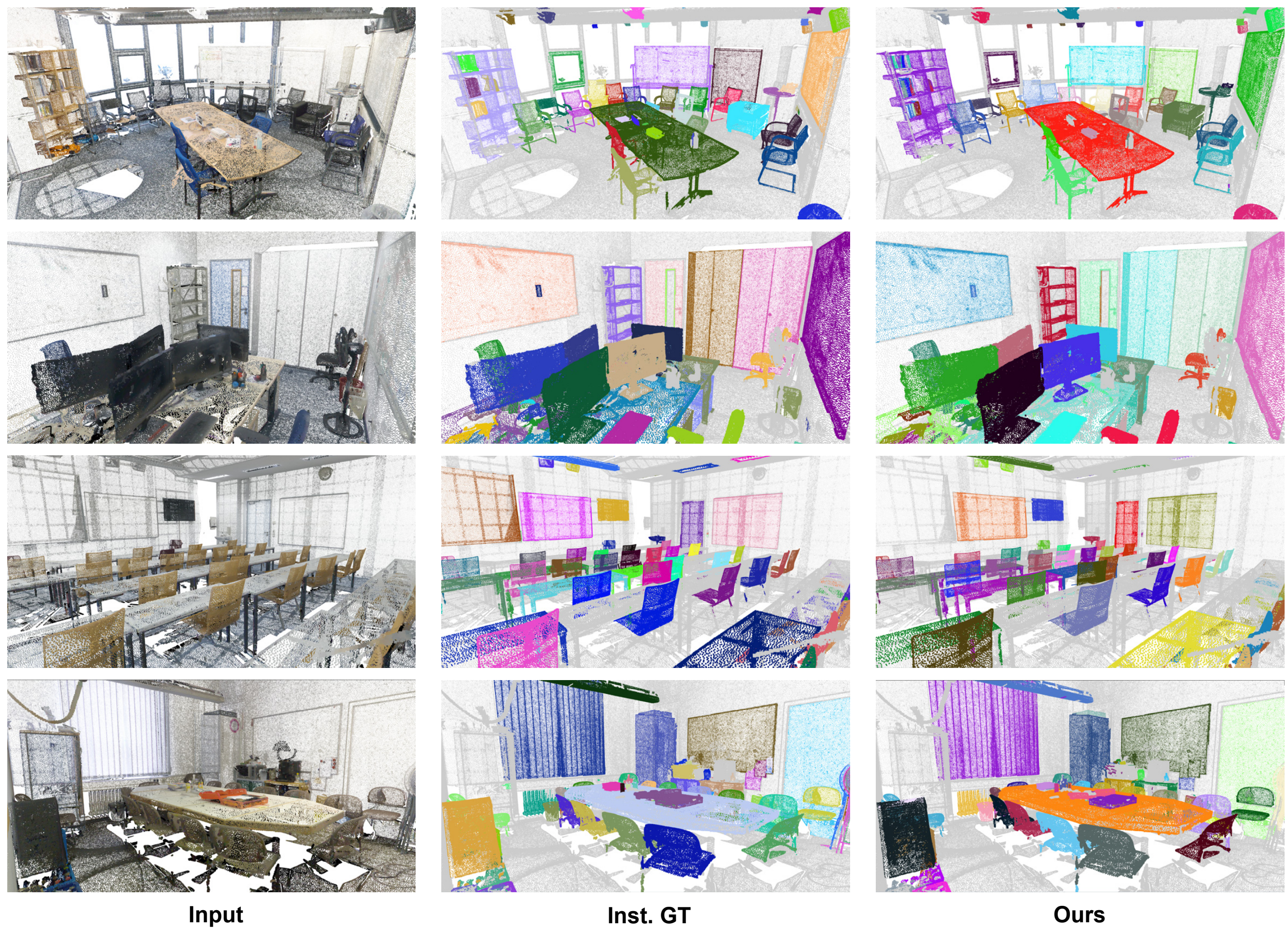


Fig. 7. **Qualitative results of ScanNet++ validation set.** We present 4 representative examples selected from ScanNet++ validation set to showcase the input point clouds, instance ground truth, and the segmentation results of SGIFormer. The visualization comprehensively illustrates our method's capability in handling large-scale and high-fidelity scenes. We use different colors to distinguish instances.

TABLE X
EFFECT OF QUERY NUMBERS AND STACKED LAYERS IN THE DECODER.

Layers	1 Layer			3 Layers			6 Layers		
#num	200	400	600	200	400	600	200	400	600
mAP	54.7	55.8	55.9	<u>57.6</u>	**58.9**	57.1	56.5	57.6	57.4
AP_{50}	75.4	75.6	75.6	77.4	**78.4**	77.8	77.4	<u>77.7</u>	77.5
AP_{25}	84.4	84.4	84.2	**86.3**	<u>86.2</u>	85.7	85.8	85.5	85.2
Params(M)	12.75	12.80	12.85	15.91	15.96	16.01	20.66	20.71	20.76

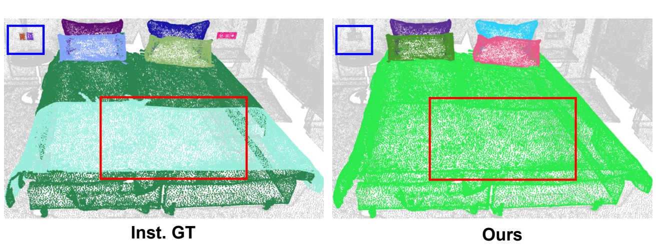


Fig. 8. **Failure cases on ScanNet++ validation set.** The visualization demonstrates that our method remains challenging when handling imbalanced categories and semantically analogous instances.

SPFormer [20] and Spherical Mask [12]. Empowered by semantic guided and geometric enhanced properties, our method can accurately recognize large convex and complex shapes (row 1 and row 2), such as counter, sofa, and background. For challenging scenes with cluttered and nearby instances, SGIFormer can effectively separate instances with fine-grained details like tables, bookshelves (row 3), and chairs (row 4). In contrast, SPFormer [20] and Spherical Mask [12] tend to miss small parts or merge different instances into a single object. Furthermore, to demonstrate the ability of our method in handling large-scale and high-fidelity scenes, we visualize the results on ScanNet++ validation set in Fig. 7. Notably, these examples show that our method can accurately distinguish instances with similar shapes and textures under complex layouts, which is crucial for real-world applications. However, our method relies on semantic prior, making it challenging to handle imbalanced categories (*light switch*) and semantically analogous instances (*bed v.s. bedsheet*) as shown in Fig. 8. These failure cases suggest potential directions for future research.

V. Conclusion

This paper presents a novel transformer-based 3D point cloud instance segmentation method namely SGIFormer. The proposed semantic-guided mix query initialization scheme combines implicitly generated scene-aware query from the original input with the learnable query, which overcomes the query initialization dilemma in large-scale 3D scenes. This hybrid strategy can not only filter out irrelevant information but also empower the model to handle semantically sophisticated scenes. Considering the vanilla transformer decoder struggles to capture fine-grained instance details and relies on heavily stacked layers, we introduce a geometric-enhanced interleaving transformer decoder to update instance queries and global features alternately, progressively incorporating coordinate information for better instance localization. SGIFormer achieves state-of-the-art performance on ScanNet V2 and ScanNet200 datasets, and the latest high-quality ScanNet++ instance segmentation benchmark. Comprehensive ablation studies demonstrate the effectiveness of each component in our architecture.

VI. Limitations and Future Work

Although our method achieves state-of-the-art performance on several indoor benchmarks and successfully alleviates dependence on heavily stacked decoder layers, it remains challenging in handling imbalanced categories and semantically analogous instances. In the meanwhile, the proposed method relies on the superpoint extraction stage, which may restrict the inference speed of our model. In the future, we plan to explore leveraging the 2D foundation model to guide the superpoint generation process, which will further improve the efficiency of the model. In addition, we will investigate the potential of adapting the proposed method to outdoor scenes and facilitate a unified 3D instance segmentation model.

Acknowledgment

The research work was conducted in the JC STEM Lab of Machine Learning and Computer Vision funded by The Hong Kong Jockey Club Charities Trust.

References

[1] A. Delitzas, A. Takmaz, F. Tombari, R. Sumner, M. Pollefeys, and F. Engelmann, "SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[2] C. Qi, J. Yin, J. Xu, and P. Ding, "Instance-incremental scene graph generation from real-world point clouds via normalizing flows," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 2, pp. 1057–1069, 2023.

[3] D. Zhou, J. Fang, X. Song, L. Liu, J. Yin, Y. Dai, H. Li, and R. Yang, "Joint 3d instance segmentation and object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1839–1849.

[4] M. Liu, Y. Chen, J. Xie, Y. Zhu, Y. Zhang, L. Yao, Z. Bing, G. Zhuang, K. Huang, and J. T. Zhou, "Menet: Multi-modal mapping enhancement network for 3d object detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[5] F. Wirth, J. Quehl, J. Ota, and C. Stiller, "Pointatme: efficient 3d point cloud labeling in virtual reality," in *IEEE Intelligent Vehicles Symposium*, 2019, pp. 1693–1698.

[6] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[7] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "Scannet++: A high-fidelity dataset of 3d indoor scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12–22.

[8] J. Hou, A. Dai, and M. Nießner, "3d-sis: 3d semantic instance segmentation of rgb-d scans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4421–4430.

[9] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "Gspn: Generative shape proposal network for 3d instance segmentation in point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3947–3956.

[10] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni, "Learning object bounding boxes for 3d instance segmentation on point clouds," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[11] M. Kolodiazhnyi, A. Vorontsova, A. Konushin, and D. Rukhovich, "Top-down beats bottom-up in 3d instance segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 3566–3574.

[12] 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*, 2024.

[13] 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*, 2020, pp. 4867–4876.

[14] S. Chen, J. Fang, Q. Zhang, W. Liu, and X. Wang, "Hierarchical aggregation for 3d instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 467–15 476.

[15] 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*, 2022, pp. 2708–2717.

[16] 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*, 2021, pp. 2783–2792.

[17] L. Han, T. Zheng, L. Xu, and L. Fang, "Occuseg: Occupancy-aware 3d instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2940–2949.

[18] L. Hui, L. Tang, Y. Shen, J. Xie, and J. Yang, "Learning superpoint graph cut for 3d instance segmentation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 804–36 817, 2022.

[19] L. Zhao and W. Tao, "Jsnet++: Dynamic filters and pointwise correlation for 3d point cloud instance and semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 4, pp. 1854–1867, 2022.

[20] 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*, vol. 37, no. 2, 2023, pp. 2393–2401.

[21] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3d: Mask transformer for 3d semantic instance segmentation," in *IEEE International Conference on Robotics and Automation*, 2023, pp. 8216–8223.

[22] 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*, 2023, pp. 18 516–18 526.

[23] 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*, 2023, pp. 3693–3703.

[24] M. Kolodiazhnyi, A. Vorontsova, A. Konushin, and D. Rukhovich, "Oneformer3d: One transformer for unified point cloud segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 943–20 953.

[25] S. Al Khatib, M. El Amine Boudjoghra, J. Lahoud, and F. S. Khan, "3d instance segmentation via enhanced spatial and semantic supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 541–550.

[26] M. Zhao, L. Zhang, Y. Kong, and B. Yin, "Eipformer: Emphasizing instance positions in 3d instance segmentation," *arXiv preprint arXiv:2312.05602*, 2023.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[28] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2906–2917.

[29] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[30] F. Yin, Z. Huang, T. Chen, G. Luo, G. Yu, and B. Fu, "Dcnet: Large-scale point cloud semantic segmentation with discriminative and efficient feature aggregation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 4083–4095, 2023.

[31] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[32] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, pp. 167–181, 2004.

[33] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4558–4567.

[34] W. Zhao, Y. Yan, C. Yang, J. Ye, X. Yang, and K. Huang, "Divide and conquer: 3d point cloud instance segmentation with point-wise binarization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 562–571.

[35] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[36] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 864–17 875, 2021.

[37] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1290–1299.

[38] H. Wang, S. Dong, S. Shi, A. Li, J. Li, Z. Li, L. Wang *et al.*, "Cagroup3d: Class-aware grouping for 3d object detection on point clouds," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 975–29 988, 2022.

[39] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-free 3d object detection via transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2949–2958.

[40] Z. Huang, Z. Zhao, B. Li, and J. Han, "Lcpformer: Towards effective 3d point cloud analysis via local context propagation in transformers," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 9, pp. 4985–4996, 2023.

[41] C. Wang, J. Deng, J. He, T. Zhang, Z. Zhang, and Y. Zhang, "Long-short range adaptive transformer with dynamic sampling for 3d object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7616–7629, 2023.

[42] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1534–1543.

[43] B. Graham and L. Van der Maaten, "Submanifold sparse convolutional networks," *arXiv preprint arXiv:1706.01307*, 2017.

[44] Z. Wang, Y.-L. Li, X. Chen, H. Zhao, and S. Wang, "Uni3detr: Unified 3d detection transformer," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[45] Z. Lin, Z. He, X. Wang, B. Zhang, C. Liu, W. Su, J. Tan, and S. Xie, "Dbganet: dual-branch geometric attention network for accurate 3d tooth segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[46] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[47] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, "An optimal algorithm for on-line bipartite matching," in *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, 1990, pp. 352–358.

[48] 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*, 2023, pp. 13 550–13 559.

[49] D. Rozenberszki, O. Litany, and A. Dai, "Language-grounded indoor 3d semantic segmentation in the wild," in *European Conference on Computer Vision*. Springer, 2022, pp. 125–141.

[50] P. Contributors, "Pointcept: A codebase for point cloud perception research," https://github.com/Pointcept/Pointcept, 2023.

[51] 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*, 2019, pp. 3075–3084.

[52] 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*, 2024, pp. 20 644–20 653.

[53] J. Lu, J. Deng, and T. Zhang, "Beyond the final layer: Hierarchical query fusion transformer with agent-interpolation initialization for 3d instance segmentation," *Under Review*, 2024.

[54] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi, "Oneformer: One transformer to rule universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2989–2998.

Lei Yao (Graduate Student Member, IEEE) received the B.Eng. degree in measurement and control technology and instruments from Huazhong University of Science and Technology, Wuhan, China, in 2020, and the M.Eng. degree in mechanical engineering in 2023. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. His current research interests include 3D scene understanding and representation for Embodied AI.

Yi Wang (Member, IEEE) received BEng degree in electronic information engineering and MEng degree in information and signal processing from the School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China, in 2013 and 2016, respectively. He earned PhD in the School of Electrical and Electronic Engineering from Nanyang Technological University, Singapore, in 2021. He is currently a Research Assistant Professor at the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong. His research interest includes Image/Video Processing, Computer Vision, Intelligent Transport Systems, and Digital Forensics.

Moyun Liu (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from the Hubei University of Technology, Wuhan, China, in 2019, and the M.S. degree in the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. He is currently working toward the Ph.D. degree with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. He is also a visiting Ph.D. student in Agency for Science Technology and Research (A*STAR) Centre for Frontier AI Research (CFAR), Singapore. His current research interests mainly include visual perception in autonomous driving, especially with multi-modal fusion method.

Lap-Pui Chau (Fellow, IEEE) received a Ph.D. degree from The Hong Kong Polytechnic University in 1997. He was with the School of Electrical and Electronic Engineering, Nanyang Technological University from 1997 to 2022. He is currently a Professor in the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. His current research interests include image and video analytics and autonomous driving. He is an IEEE Fellow. He was the chair of Technical Committee on Circuits & Systems for Communications of IEEE Circuits and Systems Society from 2010 to 2012. He was general chairs and program chairs for some international conferences. Besides, he served as associate editors for several IEEE journals and Distinguished Lecturer for IEEE BTS.