2D-3D Interlaced Transformer for Point Cloud Segmentation with Scene-Level Supervision

Cheng-Kun Yang\textsuperscript{1,2}  Min-Hung Chen\textsuperscript{3}  Yung-Yu Chuang\textsuperscript{1}  Yen-Yu Lin\textsuperscript{4,5}
\textsuperscript{1}National Taiwan University  \textsuperscript{2}MediaTek  \textsuperscript{3}NVIDI
\textsuperscript{4}National Yang Ming Chiao Tung University  \textsuperscript{5}Academia Sinica

Abstract

We present a Multimodal Interlaced Transformer (MIT) that jointly considers 2D and 3D data for weakly supervised point cloud segmentation. Research studies have shown that 2D and 3D features are complementary for point cloud segmentation. However, existing methods require extra 2D annotations to achieve 2D-3D information fusion. Considering the high annotation cost of point clouds, effective 2D and 3D feature fusion based on weakly supervised learning is in great demand. To this end, we propose a transformer model with two encoders and one decoder for weakly supervised point cloud segmentation using only scene-level class tags. Specifically, the two encoders compute the self-attended features for 3D point clouds and 2D multi-view images, respectively. The decoder implements interlaced 2D-3D attention and carries out implicit 2D and 3D feature fusion. We alternately switch the roles of queries and key-value pairs in the decoder layers. It turns out that the 2D and 3D features are iteratively enriched by each other. Experiments show that it performs favorably against existing weakly supervised point cloud segmentation methods by a large margin on the S3DIS and ScanNet benchmarks. The project page will be available at https://jimmy15923.github.io/mit_web/.

1. Introduction

Point cloud segmentation offers rich geometric and semantic information of a 3D scene, thereby being essential to many 3D applications, such as scene understanding [5, 10, 16, 26, 36], augmented reality [2, 35], and autonomous driving [7, 8, 13]. However, developing reliable models is time-consuming and challenging due to the need for vast per-point annotations and the difficulty in capturing detailed semantic clues from textureless point clouds.

Research efforts have been made to address the aforementioned issues. Several methods are proposed to derive point cloud segmentation models using various weak supervisions, such as sparsely labeled points [24, 33, 58, 65], bounding box labels [9], subcloud-level annotations [54], and scene-level tags [40, 61]. These weak annotations are cost-efficient and can significantly reduce the annotation burden. On the other hand, recent studies [19, 20, 23, 32, 41, 52, 53, 62] witness the remarkable success of 2D vision, and utilize 2D image features to enhance the 3D recognition task. They show promising results because 2D detailed texture clues are well complementary to 3D geometry features.

Although 2D-3D fusion is effective, current methods require extra annotation costs for 2D images. To the best of our knowledge, no prior work has explored fusing 2D-3D features under extremely weak supervision, where only scene-level class tags of the 3D scene are given. It is challenging to derive a segmentation model that leverages both 2D and 3D data under scene-level supervision, as no per-point/pixel annotations or per-image class tags are avail-
able to guide the learning process. Furthermore, existing 2D-3D fusion methods require camera poses or depth maps to establish pixel-to-point correspondences, adding extra burdens on data collection and processing. In this work, we address these difficulties by proposing a Multimodal Interlaced Transformer (MIT) that works with scene-level supervision and can implicitly fuse 2D and 3D features without camera poses and depth maps.

Our MIT is a transformer model with two encoders and one decoder, and can carry out weakly supervised point cloud segmentation. As shown in Figure 1, the input to our method includes the 3D point cloud, multi-view images, and scene-level tags of a scene. The two encoders utilize the self-attention mechanism to extract the features of the 3D point cloud and the 2D multi-view images, respectively. The decoder computes the proposed interlaced 2D-3D attention and can implicitly fuse the 2D and 3D data.

Specifically, one encoder is derived for 3D feature extraction, where the voxels of the input point cloud yield the data tokens. The other encoder is for 2D multi-view images, where images serve as data tokens. Also, the multi-class tokens [57] are included to match the class-level annotations. The encoders capture long-range dependencies and aggregate class-specific features for their respective modalities. The decoder comprises 2D-3D interlaced layers, and is developed to fuse 2D and 3D features, where the correspondences between 3D voxels and 2D views are implicitly computed via cross-attention. In odd layers of the decoder, 3D voxels are enriched by 2D image features, while in even layers, 2D views are augmented by 3D geometric features. Specifically, in each odd layer, each 3D voxel serves as a query, while 2D views act as key-value pairs. Through cross-attention, a query is a weighted combination of the values. Together with residual learning, this query (3D voxel) is characterized by the fused 3D and 2D features. In each even layer, the roles of 3D voxels and 2D views switch: 3D voxels and 2D views become key-value pairs and queries, respectively. This way, 2D views are described by the augmented 2D and 3D features.

By leveraging multi-view information without extra annotation effort, our proposed MIT effectively fuses the 2D and 3D features and significantly improves 3D point cloud segmentation. The main contribution of this work is threefold. First, to the best of our knowledge, we make the first attempt to fuse 2D-3D information for point cloud segmentation under scene-level supervision. Second, we enable this new task by presenting a new model named Multimodal Interlaced Transformer (MIT) that implicitly fuses 2D-3D information via interlaced attention, which does not rely on camera pose information. Besides, a contrastive loss is developed to align the class tokens across modalities. Third, our method performs favorably against existing methods on the large-scale ScanNet [11] and S3DIS [3] benchmarks.

2. Related Work

Weakly supervised point cloud segmentation. This task aims at learning a point cloud segmentation model using weakly annotated data, such as sparsely labeled points [16, 18, 27, 28, 30, 33, 43, 45, 47, 49, 56, 58, 65, 66], box-level labels [9], subcloud-level labels [54, 61] and scene-level labels [24, 40]. Significant progress has been made in the setting of using sparsely labeled points: The state-of-the-art methods [18, 33, 63] show comparable performances with supervised ones. These methods usually utilize self-supervised pre-training [16, 33], graph propagation [33, 47], and contrastive learning [28, 66] to derive the models. Despite the effectiveness, they require at least one annotated point for each category in a scene. Hence, it is not straightforward to extend these methods to work with scene-level or subcloud-level annotations.

In this work, we aim to develop a segmentation method based on a more challenging setting of using scene-level annotations. The literature about point cloud segmentation with scene-level annotations is relatively rare. Yang et al. [61] derive a transformer by applying multiple instance learning to paired point clouds. However, their performance is much inferior to fully supervised methods. Kweon and Yoon [24] leverage 2D and 3D data for point cloud segmentation by introducing additional image-level class tags, which require extra annotation efforts. Our method compensates for the lack of point-level or pixel-level annotations by integrating additional 2D features while using scene-level annotation only.

2D and 3D fusion for point cloud applications. Given the accessible or synthetizable [23] 2D images in most 3D dataset, research studies [14, 17, 19, 20, 23, 24, 29, 32, 41, 46, 52, 53, 55, 59, 60, 62, 64] explore 2D data to enhance 3D applications. Hu et al. [19] and Robert et al. [41] construct a pixel-point mapping matrix to fuse 2D and 3D features for point cloud segmentation. Despite the effectiveness, existing methods rely on camera poses and/or depth maps to build the correspondences between the 2D and 3D domains. In contrast, our method learns a transformer with interlaced 2D-3D attention, enabling the implicit integration of 2D and 3D features without the need for camera poses or depth maps.

Query and key-value pair swapping. Cross-attention is widely used in the transformer decoder. It captures the dependency between queries and key-value pairs. Umam et al. [48] and Lim et al. [21] swap queries and key-value pairs for point cloud decomposition and generation, respectively. Different from their methods working with data in a domain, our method generalizes query and key-value pair swapping to cross-domain feature fusion. In addition, we develop a contrastive loss for 2D and 3D feature alignment.
3. Proposed Method

We present the proposed method in this section. We first give the problem statement in Section 3.1. Then, we specify the developed MIT with an encoder-decoder architecture in Section 3.2 and Section 3.3. Finally, the implementation details are provided in Section 3.4.

3.1. Problem Statement

We are given a set of $N$ point clouds as well as their corresponding RGB multi-view images and the class tag annotations, i.e., $\{P_n, V_n, y_n\}_{n=1}^N$, where $P_n$ is the $n$th point cloud, $V_n$ represents the multi-view images, and $y_n$ is the class-level labels. Note that $P_n$, $V_n$, and $y_n$ are acquired from the same scene. Without loss of generality, we assume that each point cloud consists of $M$ points, i.e., $P_n = \{p_{nm}\}_{m=1}^M$, where each point $p_{nm} \in \mathbb{R}^d$ is represented by its 3D coordinate and RGB color. The RGB multi-view images are grabbed in the same scene as $P_n$, and consist of a set of $T$ images, i.e., $V_n = \{v_{nt}\}_{t=1}^T$. Each image $v_{nt} \in \mathbb{R}^{H \times W \times S}$ is of resolution $H \times W$ with RGB channels. The class tags of $P_n$, i.e., $y_n \in \{0, 1\}^C$, are $C$-dimensional binary vectors storing which categories are present, where $C$ is the number of categories of interest.

With the weakly annotated dataset $\{P_n, V_n, y_n\}_{n=1}^N$, we aim to derive a model for point cloud segmentation that classifies each point of a testing cloud into one of the $C$ categories. Note that in this weakly supervised setting, neither points nor pixels are labeled, and camera poses are unavailable, making it challenging to enhance 3D point cloud segmentation by incorporating additional 2D features due to the absence of point/pixel supervision and explicit correspondences between 2D pixels and 3D points. Furthermore, as multi-view images share the same scene-level class tag, the lack of individual class tag annotation for each view image may lead to an inaccurate semantic understanding of each image.

Method overview. Figure 2 illustrates the network architecture of MIT, which comprises two transformer encoders, $f_{3D}$ and $f_{2D}$, and one decoder $f_d$. The two encoders extract features for 3D point clouds and 2D multi-view images, respectively. The decoder is developed for 2D-3D feature fusion, which utilizes cross-attention to connect 2D and 3D data implicitly. They are elaborated in the following.

3.2. Transformer Encoders

3D point cloud feature extraction. A 3D backbone $f_{3D}$, e.g., MinkowskiNet [10] or PointNet++ [38], is applied to extract the point embedding $s_{3D} \in \mathbb{R}^{M \times D}$ for all $M$ points of a point cloud $P$. Like WYPR [31], we perform supervoxel partition using an unsupervised off-the-shelf algorithm [25]. The 3D coordinates of $P$ are fed into a coordinate embedding module $f_{emb}$, which is composed of two $1 \times 1$ convolution layers with ReLU activation, to get the positional embedding $z_{3D} \in \mathbb{R}^{M \times D}$, where $D$ is the embedding dimension. We aggregate both the point features and point positional embedding through supervoxel average pooling [33], producing the supervoxel features $s_{3D} \in \mathbb{R}^{S \times D}$ and pooled positional embedding $z_{3D} \in \mathbb{R}^{S \times D}$, where $S$ is the number of the supervoxels in $P$. The supervoxel features are added to the positional embedding.

To learn the class-specific representation for fitting the scene-level supervision, we prepend $C$ learnable class tokens [57] $c_{3D} \in \mathbb{R}^{C \times D}$ with $S$ supervoxel tokens. Total $(C + S)$ tokens are fed into the transformer encoder $f_{3D}$.
through the self-attention mechanism, the dependencies of the class and supervoxel tokens are captured, producing the self-attended 3D features $F_{3D} \in \mathbb{R}^{(C+S) \times D}$.

2D multi-view images feature extraction. A 2D backbone network $f_{2D}$, e.g., ResNet [15], is employed to extract image features $s_{2D} \in \mathbb{R}^{T \times H' \times W' \times D}$, where $H' = H/32$ and $W' = W/32$. We apply global average pooling to image features $s_{2D}$ along the spatial dimensions. The pooled image features $\tilde{s}_{2D} \in \mathbb{R}^{T \times D}$ are added to the learnable positional embedding $\mathbf{e}_{2D} \in \mathbb{R}^{T \times D}$, producing $T$ view tokens. Analogous to 3D feature extraction, another transformer encoder $\tilde{f}_{2D}$ is applied to $C$ class tokens $c_{2D} \in \mathbb{R}^{C \times D}$ and $T$ view tokens, obtaining the self-attended 2D features $F_{2D} \in \mathbb{R}^{(C+T) \times D}$.

Encoder optimization. During training, we consider a point cloud $P$ and its associated $T$ multi-view images $\{v_t\}$ and scene-level label $y$. The 2D and 3D self-attended features $F_{2D}$ and $F_{3D}$ are compiled as specified above. We conduct the multi-label classification losses [40, 57] for optimization.

For 3D attended features $F_{3D} \in \mathbb{R}^{(C+S) \times D}$, we divide it into $C$ class tokens $F_{3D}^{C \times D}$ and $S$ supervoxel tokens $F_{3D}^{S \times D}$. For the class tokens $F_{3D}^{C}$, the $C$ class scores are estimated by applying average pooling along the feature dimension. The multi-label classification loss $L_{3D}^{\text{enc}}$ is computed based on the estimated class scores and the scene-level ground-truth labels $y$. For the supervoxel tokens $F_{3D}^{S}$, we introduce a class-aware layer [44], i.e., a $1 \times 1$ convolution layer with $C$ filters, which maps the supervoxel tokens $F_{3D}^{S}$ into the class activation maps (CAM) $F_{3D}^{C} \in \mathbb{R}^{S \times C}$. The estimated class scores are obtained by applying global average pooling to $F_{3D}^{C}$ along the dimension of supervoxels. The multi-label classification loss $L_{3D}^{\text{enc}}$ is computed based on the class scores and label $y$. The loss for the 3D modality is defined by $L_{3D} = L_{3D}^{\text{enc}} + L_{3D}^{\text{sup}}$. For the self-attended 2D features $F_{2D} \in \mathbb{R}^{(C+T) \times D}$ of the $C$ class tokens and $T$ view tokens, the 2D loss is similarly defined by $L_{2D} = L_{2D}^{\text{enc}} + L_{2D}^{\text{sup}}$.

In sum, both encoders are derived in a weakly-supervised manner using the objective function

$$L_{\text{enc}} = L_{2D} + L_{3D}. \quad (1)$$

3.3. Transformer Decoder

The two encoders produce self-attended 3D features $F_{3D}$ of $C + S$ tokens and 2D features $F_{2D}$ of $C + T$ tokens, respectively. We propose a decoder that performs interlaced 2D-3D cross-attention for feature fusion. The decoder $f_d$ in Figure 2 is a stack of $R$ interlaced blocks. Each interlaced block is composed of two successive decoder layers, as shown in Figure 3. In the first layer of this block, 3D tokens are enriched by 2D features, while in the second layer, 2D tokens are enriched by 3D features.

In the odd/even layer (the blue-shaded region in Figure 3), the $C + S$ tokens in $F_{3D}$ serve as the queries, while the $C + T$ tokens in $F_{3D}$ act as the key-value pairs. Through scaled dot-product attention [50], the cross-modal attention matrix $A \in \mathbb{R}^{(C+S) \times (C+T)}$ (the yellow-shaded region) is computed to store the consensus between the 3D tokens and 2D tokens. As we focus on exploring the relationships between 3D tokens and merely 2D view tokens in this layer, we ignore the attention values related to the 2D class tokens in $A$. Specifically, only the query-to-view attention values $A_{q2v} \in \mathbb{R}^{(C+S) \times T}$ (green dots in Figure 3) are considered. This is implemented by applying submatrix extraction to the attention matrix $A$ and the value matrix $V \in \mathbb{R}^{(C+T) \times D}$, i.e., $A_{q2v} = A[1: C+S, C+1: C+T]$ and $V_d = V[C+1: C+T, :]$.

After applying the softmax operation to $A_{q2v}$, we perform matrix multiplication between the query-to-view attention matrix $A_{q2v}$ and the masked value matrix $V_d$. This way, each query (3D token) is a weighted sum of the values (2D view tokens). Together with a residual connection, the resultant 3D tokens $F_{3D}$ are enriched by 2D features. It turns out that implicit feature fusion from 3D features to 2D features is carried out without using annotated data.

In the even/second layer (the green-shaded region in Figure 3), the roles of $F_{3D}$ and $F_{2D}$ switch: The former serves as the key-value pairs while the latter yields the queries. After a similar procedure, the resultant 2D tokens $F_{2D} \in \mathbb{R}^{(C+T) \times D}$ are augmented with 3D information. $F_{2D}$ and $F_{3D}$ are the output of the interlaced block. By stacking $R$ interlaced blocks, the proposed decoder is built to fuse 2D and 3D features iteratively.

Decoder optimization. In the last interlaced block, the 2D class scores and 3D class scores can be estimated by applying average pooling to the corresponding class tokens. The multi-label classification losses for 2D $L_{2D}$ and 3D $L_{3D}$ data can be computed between the ground truth and the estimated class scores.
To mine additional supervisory signals, we employ contrastive learning on the class-to-class attention matrix 
\[ \alpha \in \mathbb{R}^{C \times C} \]. Though the 2D class tokens and 3D class tokens attend to respective modalities, they share the same class tags. Hence, the attention value between a pair of class tokens belonging to the same class should be larger than those between tokens of different classes, which can be enforced by the N-pair loss [39].

We employ this regularization in all attention matrices in the decoder layers

\[
\mathcal{L}_{con} = \frac{1}{2R} \sum_{r=1}^{2R} \sum_{c=1}^{C} - \log \frac{A^r_{ss}}{\sum_{i=1}^{C} A^r_{sj}}, \quad (2)
\]

\[
+ \frac{1}{2R} \sum_{r=1}^{2R} \sum_{c=1}^{C} - \log \frac{A^r_{js}}{\sum_{i=1}^{C} A^r_{ij}},
\]

where \( A^r \) is the attention matrix in the \( r \)th decoder layer.

The objective function of learning the decoder is

\[
\mathcal{L}_{dec} = \mathcal{L}_{2D} + \mathcal{L}_{3D} + \alpha \mathcal{L}_{con}, \quad (3)
\]

where \( \alpha \) is a positive constant.

### 3.4. Implementation Details

The proposed method is implemented in PyTorch. ResNet-50 [15] pre-trained on ImageNet [42] serves as the 2D feature extractor while MinkowskiNet [10] works as the 3D feature extractor in the experiments. The numbers of heads, encoder layers, interlaced blocks, embedding dimension, and the width of FFN in the transformer are set to 4, 3, 2, 96, and 96, respectively. 16 multi-view images are randomly sampled for each scene. The model is trained on eight NVIDIA 3090 GPUs with 500 epochs. The batch size, learning rate, and weight decay are set to 32, \( 10^{-2} \), and \( 10^{-4} \), respectively. We use AdamW [22] as the optimizer. The weight \( \alpha \) for \( \mathcal{L}_{con} \) is set to 0.5.

**Inference.** Given a point cloud \( P \) for inference, we feed it into our 3D encoder for feature extraction. The 3D CAM \( F_{3D} \in \mathbb{R}^{S \times C} \), i.e., the segmentation result, is then obtained by passing the extracted features into the class-aware layer, as specified in Section 3.2. In MCTormer [57], 3D CAM can be further refined by the class-to-voxel attention maps \( A_{c2v} \in \mathbb{R}^{C \times 3} \) from the last \( K \) transformer encoder layers, where \( K = 3 \). The refined 3D CAM is obtained through element-wise multiplication between CAM and the attention maps: \( F = F_{3D} \odot A_{c2v} \), where \( \odot \) denotes Hadamard product. In addition, we consider the class-to-voxel attention maps in the interlaced decoder, if multi-view images are provided, which can be extracted from all the even layers, producing another refined 3D CAM \( \hat{F} \). Finally, the segmentation results can be obtained by applying the element-wise max operation to \( F \) and \( \hat{F} \).
methods. However, the reliance on camera poses or depth maps limits their applicability. In contrast, our MIT can benefit from 2D images without such requirements, enhancing its generalizability.

By using efficient scene-level annotation, our MIT with 3D data only (the blue-shaded region in Figure 2) shows comparable results to the state-of-the-art weakly supervised method [40], demonstrating the effectiveness of transformer encoder with the multi-class token [37]. The proposed interlaced decoder further enhances the performance of the MIT with 3D-only data by incorporating the 2D image information. Without introducing extra annotation costs, our method with 2D-3D fusion outperforms the existing methods by a large margin on both the ScanNet and S3DIS datasets. This result demonstrates once again that 2D and 3D data are complementary. More importantly, the proposed method is capable of utilizing their complementarity in a weakly supervised manner.

Kweon et al. [24] also confirms the effectiveness of combining 2D-3D data. However, their method requires non-negligible extra annotation costs on the images. According to [4, 61], their method incurs more than five times the annotation cost required for scene-level annotation and even more than the sparsely labeled points setting.

We summarize the advantages of the scene-level setting in three aspects: 1) Efficiency: Scene-level supervision is much more efficient to collect than other weak supervision types. According to [40, 61], the labeling cost of sparse points (1% of points in ScanNet) is more than ten times higher than our scene-level setting. 2) Generalization: Our method based on scene-level supervision can be extended to other forms of weak supervision. Section 4.3.2 evaluates our method trained with diverse weak supervision types. 3) Multiview aggregation: The view quality differs in different views of the same 3D point, such as occlusion, or no 2D-3D correspondence. Through the attention mechanism, the decoder learns how to effectively aggregates the multi-view information based on the semantic information. Global attention: The decoder can capture long-range dependencies.

### Table 1: Quantitative results (mIoU) of several point-cloud segmentation methods with diverse supervisions and input data settings on the ScanNet and S3DIS datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sup.</th>
<th>Extra inputs</th>
<th>ScanNet</th>
<th>S3DIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepViewAgg [41]</td>
<td>✓✓</td>
<td>RGB, Pose</td>
<td>72.2</td>
<td>65.8</td>
</tr>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>-</td>
<td>71.0</td>
<td>67.2</td>
</tr>
<tr>
<td>WYPR [40]</td>
<td>✓✓</td>
<td>RGB, Pose</td>
<td>74.9</td>
<td>71.6</td>
</tr>
<tr>
<td>MIT (3D-only)</td>
<td>✓✓</td>
<td>RGB, Pose</td>
<td>59.4</td>
<td>50.1</td>
</tr>
<tr>
<td>MIT (Ours)</td>
<td>✓✓</td>
<td>RGB, Pose</td>
<td>63.9</td>
<td>-</td>
</tr>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>RGB</td>
<td>49.4</td>
<td>47.4</td>
</tr>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>RGB, Pose</td>
<td>26.4</td>
<td>10.3</td>
</tr>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>RGB, Pose</td>
<td>26.2</td>
<td>12.9</td>
</tr>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>RGB, Pose</td>
<td>29.6</td>
<td>24.0</td>
</tr>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>RGB, Pose</td>
<td>31.6</td>
<td>26.4</td>
</tr>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>RGB, Pose</td>
<td>35.8</td>
<td>31.7</td>
</tr>
</tbody>
</table>

**Potential:** Existing weakly supervised point cloud segmentation methods focus on the sparse-point supervision setting and achieve performances almost as good as fully supervised ones. Therefore, working with lower annotation costs, such as scene-level tags, shows potential and is worth exploring. Our method effectively carries it out by utilizing information from unlabeled images.

### 4.2.2 2D-3D Fusion Method Comparison

As far as we are aware, our MIT is the first attempt at exploring 2D-3D fusion without poses, and the model is derived through scene-level supervision. Hence, there is no existing fusion method for performance comparison. To evaluate our method, we explore two approaches for 2D-3D feature fusion. First, we design a baseline method using a simple multi-layer perceptron for 2D-3D fusion. For each 3D voxel, we locate the nearest 2D pixel and concatenate the 3D feature with the 2D feature, followed by a 1x1 convolution to perform 2D-3D feature fusion. Second, we employ the bidirectional projection module [19] for 2D-3D fusion, which utilizes the pixel-to-point link matrix to fuse the 2D-3D features.

We apply the 2D-3D fusion methods on a weakly supervised point cloud segmentation method, MIL-transformer [61], as well as our proposed method. Table 2 provides the mIoU results of the competing 2D-3D fusion methods. Our proposed interlaced decoder achieves superior results compared to the two competing 2D-3D fusion methods. More importantly, the interlaced decoder implements 2D-3D fusion without using poses or depths and performs even better when camera information is available (more details in Section 4.3.2 and supplementary materials).

### Table 2: Quantitative results (mIoU) of our method (interlaced decoder) and competing methods with different 2D-3D fusion strategies on the ScanNet validation set using scene-level annotations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Fusion</th>
<th>Pose</th>
<th>Depth</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>-</td>
<td>-</td>
<td>25.6</td>
</tr>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>✓</td>
<td>✓</td>
<td>28.9</td>
</tr>
<tr>
<td>MIL-Trans [61]</td>
<td>✓✓</td>
<td>✓</td>
<td>✓</td>
<td>32.6</td>
</tr>
<tr>
<td>MIT</td>
<td>✓✓</td>
<td>✓</td>
<td>✓</td>
<td>32.4</td>
</tr>
<tr>
<td>MIT</td>
<td>Interlaced</td>
<td>✓</td>
<td>✓</td>
<td>35.8</td>
</tr>
<tr>
<td>MIT</td>
<td>Interlaced</td>
<td>✓</td>
<td>✓</td>
<td>37.1</td>
</tr>
</tbody>
</table>
Figure 4: Qualitative results on the ScanNet dataset with scene-level supervision. The colored boxes highlight the differences between our MIT and MIT with 3D data only, and their corresponding views are shown on the right with outlines of the same color. For each view, the tags at the top indicate the results of the multi-label classification.

### 4.2.3 Qualitative Results

Figure 4 shows the qualitative results of our MIT with and without using the complementary 2D data. By utilizing both 3D and 2D data, our method achieves promising segmentation results without using any point-level supervision. With the help of detailed texture features in the 2D image, our method is able to classify objects with very similar geometric shapes, for example, door and wall. Take the second row of Figure 4 as an example: MIT successfully segments the points belonging to the door by cooperating with the correct prediction from our 2D view (marked in blue), while the 3D only MIT fails to locate the points of the door, by only considering geometric and color features.

In addition, the category co-occurrence issue could hinder the optimization of the model with 3D data only. Since optimization is based on scene-level labels, it is difficult to learn discriminative features for those co-occurring categories. As demonstrated in the second and third rows of Figure 4, MIT (3D-only) often fails to classify the chairs and tables since these categories often co-occur in a scene. In contrast, our method leverages multi-view information during training. As each view only captures a small part of the scene, the issue of category co-occurrence could be alle-
Table 3: The mIoU performance of different combinations of proposed components on the validation set of the ScanNet dataset. “Query” and “Key-Value” denote the input to the decoder. “MCT” and “Interlaced” are the multi-class tokens encoder and decoder architectures respectively. \( L_{\text{con}} \) denotes the contrastive loss on the class tokens.

<table>
<thead>
<tr>
<th>Query</th>
<th>Key-Value</th>
<th>MCT</th>
<th>Interlaced</th>
<th>( L_{\text{con}} )</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D</td>
<td>2D</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>33.7</td>
</tr>
<tr>
<td>2D</td>
<td>3D</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>35.2</td>
</tr>
<tr>
<td>3D</td>
<td>2D</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>35.8</td>
</tr>
</tbody>
</table>

Table 4: Performance with different numbers of views on the mIoU of pseudo labels on the ScanNet.

<table>
<thead>
<tr>
<th>Number of views</th>
<th>4</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>29.7</td>
<td>32.7</td>
<td>30.9</td>
<td>31.2</td>
</tr>
</tbody>
</table>

Table 5: Performance with different numbers of interlaced blocks on the mIoU of pseudo labels on the ScanNet.

<table>
<thead>
<tr>
<th>( R ) interlaced blocks</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>31.4</td>
<td>32.7</td>
<td>32.1</td>
<td>32.4</td>
</tr>
</tbody>
</table>

4.3. Ablation Studies and Performance Analysis

To evaluate the effectiveness of the proposed components, we perform ablation studies and analyze their performance. We present ablation studies to evaluate the impacts of the proposed components and provide performance analysis.

4.3.1 Contributions of Components

To evaluate the effectiveness of each proposed component, we first construct the baseline by considering only 3D data and utilizing class activation maps \([54,58]\). Then, we assess the contributions of each component, including the multi-class token transformer encoder (MCT), the interlaced decoder (Interlaced), and the N-pair loss (\( L_{\text{con}} \)), by successively adding each one to the baseline. In addition, we evaluate the roles for 2D and 3D, as query and key-value pairs, by switching them. The result of the standard transformer decoder is also reported (the third row of Table 3) by taking 3D as query and 2D as key-value. Table 3 illustrates the performance when using different combinations of the proposed modules and loss. The results validate that each component contributes to the performance of our method.

4.3.2 Performance Analysis

We discuss the extension of the proposed method and evaluate our method with different parameters and synthesized images in the following.

Extension with known poses and depths. When camera poses, and depth maps are available, the correspondence between 3D world coordinates and 2D pixels can be established. Therefore, we can explicitly construct the position correlation between 2D views and 3D voxels. To this end, we first generate the 3D world coordinate maps for each view by following Yu et al. \([63]\). All the 3D coordinate maps are fed into the coordinate embedding module \( f_{\text{emb}} \) to obtain positional embedding, which is then added to the 2D image features. Through explicit positional information between the 2D view and 3D voxel, we can further boost performance, as shown in the last row of Table 2.

Analysis of parameters. We explore the influence of the number of 2D views and interlaced blocks by evaluating the quality of pseudo labels on the training set. Table 4 shows the performance of our MIT with different numbers of views used in the transformer. We found that performance is stable when given a sufficient number of views, as also reported in \([19]\). Table 5 presents the performance by altering the number of the proposed interlaced blocks. The results indicate that stacking two interlaced blocks performs the best while being saturated by adding more blocks.

Experiments with virtual image rendering. Among the limitations of our method is the need for multi-view 2D images within the 3D dataset. A potential solution would be the virtual view rendering of the 3D data. Several studies \([23, 34]\) suggest that synthesized images can further improve 3D segmentation performance. With the help of virtual view rendering \([23]\), our method still achieves competitive results (34.3% mIoU on ScanNet validation set) using the synthesized RGB images.

Extensions to other weak supervisions. Thanks to the flexibility of the transformer, our method can be easily adapted to other weakly supervised settings, such as additional image-level labels, subcloud-level annotation, or sparsely labeled points.
Several methods [1, 44] are proposed to derive a 2D tag indicating the existing object category within each view ScanNet. annotation time per scene of different weak supervisions on Table 6: The mIoU performance of our MIT and its average efficient [61], its performance has room for improvement. While scene-level annotation is the most different types of weak supervision and the corresponding an-
notation cost. More importantly, our MIT shows promising results by us-
The extra image-label annotation, it provides the class tag indicating the existing object category within each view image. Several methods [1, 44] are proposed to derive a 2D segmentation model based on this supervision and achieve promising results. Our method can easily train on image-level supervision by computing the multi-label classification loss on each image token. Regarding the subcloud-level annotation, it sequentially crops a sphere point cloud from the scene and labels the existing objects within the sphere. This type of supervision alleviates the severe class imbalance issue in scene-level supervision. Our approach can be directly trained on subcloud-level supervision by considering the corresponding multi-view images in the subcloud. For the setting with sparsely labeled points [33, 61], we can calculate the cross entropy loss on the self-attended voxel features $F_{2D}$ and the labeled points. Furthermore, we note that the sparsely labeled 3D points can be projected onto the 2D image pixels, generating 2D pixel annotation. In spite of this, we do not explore this operation in our experiments and leave it for future research. Table 6 shows the performance of our method under different types of weak supervision and the corresponding annotation cost. While scene-level annotation is the most efficient [61], its performance has room for improvement. The extra image-level labels can improve the performance of scene-level supervision but introduce additional burdens due to the large number of view images in each scene. According to [4], the image-level labels cost 20 seconds per image. In line with [24], which utilized 17 multi-view images in their setting, we used 16 images per scene, resulting in an additional five-minute annotation time. Even though both image-level and subcloud-level supervision types do not require point-level annotation, they could require more annotation efforts due to the large number of views and subclouds that need to be annotated. Sparsely labeled points, on the other hand, may perform better with less annotation effort. Our approach can work effectively with diverse weak supervision, allowing for flexible savings in annotation costs. More importantly, our MIT shows promising results by using the most efficient scene-level supervision, while other weakly supervised methods cannot be straightforwardly applied in this scenario.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Scene+Image</th>
<th>Subcloud</th>
<th>20pts</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>35.8</td>
<td>45.4</td>
<td>46.8</td>
</tr>
<tr>
<td>label effort</td>
<td>&lt; 1 min</td>
<td>5 min</td>
<td>3 min</td>
</tr>
</tbody>
</table>

Table 6: The mIoU performance of our MIT and its average annotation time per scene of different weak supervisions on ScanNet.

Experiments with different backbones. Table 7 provides the performance (pseudo-label quality in mIoU) of our method on ScanNet with different 2D and 3D backbones, including different versions of 2D ResNet and 3D ResUNet. Our method’s performance is consistent across different backbones.

Table 7: Performance with different backbones on the mIoU of pseudo labels on the ScanNet.

5. Conclusion

This paper presents a multimodal interlaced transformer, MIT, for weakly supervised point cloud segmentation. Our method represents the first attempt at 2D and 3D information fusion with scene-level annotation. Through the use of the proposed interlaced decoder, which performs implicit 2D-3D feature fusion via cross-attention, we are able to effectively fuse 2D-3D features without using camera poses or depth maps. Our MIT achieves promising performance without using any point-level or pixel-level annotations. Furthermore, we develop class token consistency to align the multimodal features. MIT is end-to-end trainable. It has been extensively evaluated on two challenging real-world large-scale datasets. Experiments show that our method performs favorably against existing weakly supervised methods. We believe MIT has the potential to enhance other recognition tasks that involve both 2D and 3D observations, in an efficient manner.

Discussion and future work. Our current method has not utilized the spatial information conveyed in the images since global average pooling is applied to the image features. We attempted to flatten image features instead of using global average pooling to obtain patch tokens, similar to [12, 32], but achieved inferior results. One possible reason is that a large number of patch tokens introduces noise under scene-level supervision. A solution to this issue can achieve joint 2D-3D segmentation with weak supervision, which could be an interesting area for future research.

Acknowledgements. This work was supported in part by the National Science and Technology Council (NSTC) under grants 111-2628-E-A49-025-MY3, 112-2221-E-A49-090-MY3, 111-2634-F-002-023, 111-2634-F-006-012, 110-2221-E-002-124-MY3, and 111-2634-F-002-022. This work was funded in part by MediaTek, Qualcomm, NVIDIA, and NTU-112L900902.

985
References


