HiT: Building Mapping with Hierarchical Transformers

Mingming Zhang, Qingjie Liu, *Member, IEEE,* and Yunhong Wang, *Fellow, IEEE*

Abstract—Deep learning-based methods have been extensively explored for automatic building mapping from high-resolution remote sensing images over recent years. While most building mapping models produce vector polygons of buildings for geographic and mapping systems, dominant methods typically decompose polygonal building extraction in some sub-problems, including segmentation, polygonization, and regularization, leading to complex inference procedures, low accuracy, and poor generalization. In this paper, we propose a simple and novel building mapping method with Hierarchical Transformers, called HiT, improving polygonal building mapping quality from high-resolution remote sensing images. HiT builds on a two-stage detection architecture by adding a polygon head parallel to classification and bounding box regression heads. HiT simultaneously outputs building bounding boxes and vector polygons, which is fully end-to-end trainable. The polygon head formulates a building polygon as serialized vertices with the bidirectional characteristic, a simple and elegant polygon representation avoiding the start or end vertex hypothesis. Under this new perspective, the polygon head adopts a transformer encoder-decoder architecture to predict serialized vertices supervised by the designed bidirectional polygon loss. Furthermore, a hierarchical attention mechanism combined with convolution operation is introduced in the encoder of the polygon head, providing more geometric structures of building polygons at vertex and edge levels. Comprehensive experiments on two benchmarks (the CrowdAI and Inria datasets) demonstrate that our method achieves a new state-of-the-art in terms of instance segmentation and polygonal metrics compared with state-of-the-art methods. Moreover, qualitative results verify the superiority and effectiveness of our model under complex scenes.

Index Terms—Building mapping, transformer, bidirectional polygon loss, hierarchical attention mechanism.

I. INTRODUCTION

BUILDING mapping from remote sensing images is an essential task for geographic and mapping applications, including disaster management and assessment, city planning, human activity monitoring, and demographics. Deep learning methods have emerged over recent years due to their powerful representation learning and success in many tasks (*e.g.*, classification, detection, segmentation). Meanwhile, the quick development of satellite and sensor techniques makes large-scale high-resolution remote sensing images easy to access, and some building segmentation benchmarks have been built for automatic building extraction. Therefore, deep learning-based building mapping from high-resolution remote sensing images has attracted more and more attention in the remote sensing community.

Early deep learning-based methods apply semantic segmentation models, such as fully convolutional network (FCN) [1], U-Net [2], and DeepLabs [3], [4], to classify each pixel as building or background [5]–[9]. However, semantic segmentation-based methods can not distinguish individual buildings. Therefore, some building extraction methods based on instance segmentation models [10]–[13] have been studied for building instance segmentation. All these pixel-wise segmentation-based methods fail to obtain accurate building boundaries due to dense buildings and similar backgrounds in remote sensing images. To refine blurred boundaries, some studies [14]–[17] introduce boundary-preserved modules to regularize building boundaries. Although recent pixel-wise segmentation methods produce accurate buildings with precise boundaries, they usually output raster building segmentation masks, requiring a delicate post-vectorization pipeline to meet real-world geographic applications.

To vectorize building masks, researchers have formulated building extraction as a multi-stage task and produced vectorized buildings by post-processing or multi-task learning. Early multi-stage methods [18], [19] usually decompose this task into different sub-tasks, including binary building segmentation, polygon generation (or initialization), and boundary regularization. Since these methods are not end-to-end trainable due to separate sub-tasks, building segmentation errors will accumulate throughout the pipeline, resulting in irregular building boundaries. Another line of multi-stage methods [20]–[23] has integrated building segmentation, polygonization, and refinement into a framework by multi-task learning. These methods usually design complex pipelines with different threshold constraints for each sub-task, resulting in complex pipelines and hard-to-train.

Recently, dominant building mapping approaches [24]–[26] have represented building extraction as polygonal building vertex prediction and directly predicted building vertices to produce vector polygons of buildings. These methods can be categorized as follows: (1) Predict serialized vertices clockwise or counterclockwise from the building feature map of the candidate building region aligned from remote sensing image features. A CNN-RNN architecture is adopted to extract feature maps using convolution neural networks (CNNs) and predict serialized vertices iteratively using recurrent neural networks (RNNs, *e.g.*, ConvLSTM [27]). Since they output a vector polygon vertex by vertex, this type of method is sensitive to buildings with complex structures or a large number of vertices due to the long dependency problem.

Mingming Zhang, Qingjie Liu, and Yunhong Wang are with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, and also with the Hangzhou Innovation Institute, Beihang University, Hangzhou 310051, China (e-mail: sara_@buaa.edu.cn; qingjie.liu@buaa.edu.cn; yhwang@buaa.edu.cn). This work was supported by Science and Technology Innovation 2030-Key Project of "New Generation Artificial Intelligence" under Grant 2020AAA0108205 and NSFC No. 62176017.

(2) Predict a vertex set from the extracted feature map of remote sensing images. The primary concern of these methods is determining the vertices of one building and the vertex sequence of each building from the unordered vertex set, requiring complex human-crafted polygonal constraints and generating irregular building polygons.

In this paper, we present an end-to-end building mapping method with a hierarchical transformer (HiT), which is built on a two-stage detection architecture by adding a polygon head to produce vector polygons of buildings. In particular, the polygon head casts polygonal building as a bidirectional vertex sequence without start or end vertices hypothesis, making serialized vertices prediction order-independent (i.e., the order can be clockwise or equivalently counterclockwise). Through this new perspective, the polygon head pays more attention to the prediction of vertex position and the relationship between two vertices, regardless of whether the order is clockwise or counterclockwise. Unlike RNNs that first define the start vertex and then predict serialized vertices one by one, HiT adopts learnable and order invariant vertex queries to automatically predict serialized vertices of a building at one time. Moreover, we introduce a hierarchical attention mechanism of vertex and edge levels to encode the building feature of the candidate building region aligned from remote sensing image features, providing more geometric information of building boundaries and corners to embed into the building feature.

We evaluate our proposed HiT on two building benchmarks, including the CrowdAI [28] and Inria Polygonized [29] datasets. Since the building mapping task can be seen as an instance segmentation task and a vector polygon extraction task, we compare the proposed HiT with classical instance segmentation methods and state-of-the-art polygonal building extraction methods to evaluate pixel-level and geometric-level performance. Finally, experimental results demonstrate that HiT improves performance to a new state-of-the-art by considerable margins on the two benchmarks.

The contributions of our work include:

1) We propose HiT, a two-stage model with three parallel heads to simultaneously detect buildings and extract vector polygons of buildings from remote sensing images, which includes classification, bounding box regression, and polygon heads. HiT is end-to-end trainable and simple yet powerful to achieve building mapping.

2) HiT represents a polygonal building as a bidirectional vertex sequence, making serialized vertices of a building order-agnostic. In particular, the polygon head applies a transformer-based architecture to directly produce serialized vertices of a building at one time rather than one by one. Besides, we introduce a novel bidirectional polygon loss to supervise the polygon head, avoiding complex polygonal constraints and improving the generalization ability.

3) We introduce a hierarchical attention mechanism of vertex and edge levels in the encoder of the polygon head, embedding more geometric information (*e.g.*, building boundaries and corners) into aligned building features. Finally, comprehensive experimental results on two building benchmarks demonstrate that our method

achieves new state-of-the-art performance on instance segmentation and polygonal building extraction.

The remainder of this paper is organized as follows: Section II reviews related studies. Section III introduces our proposed HiT in detail. Section IV describes experiment settings, including comparison datasets, methods, and evaluation metrics, then reports and discusses the experimental results quantitatively and qualitatively. Section V concludes this work.

II. RELATED WORK

Recently, deep learning has been a prevalent technology for remote sensing mapping [30], [31]. Since building mapping has been a hot research topic in the remote sensing community, many attempts based on deep learning have been widely explored. Early work treats building extraction as a semantic segmentation task [5]–[9], [32], [33] or an instance segmentation task [10]–[13], [34], [35], as shown in Figure 1(a) and (b). However, they typically output raster building segmentation masks and are not suitable for real-world applications. Recently, polygonal building extraction has directly outputted vector polygons of buildings, which is more suitable for real-world geographic and mapping applications. Therefore, we review literature closely related to our research in this section.

A. Multi-stage polygonal building mapping

Multi-stage polygonal building mapping first extracts binary building masks and then obtains polygonal buildings using post-processing or multi-task learning, as exemplified in Figure 1(c) and (d). Some multi-stage methods [18], [19], [36] based on post-processing decomposes polygonal building mapping into sequential sub-tasks: (1) Extract individual building masks by segmentation models (*e.g.*, Mask RCNN [37]); (2) Generate or initialize polygons by heuristic post-processing (*e.g.*, Marching Cubes algorithm [38]); (3) Regularize or simplify polygons using Douglas–Peucker algorithm [39] to refine boundaries or vertices. Since they are not end-to-end trainable, building segmentation errors of the first stage will be accumulated throughout the pipeline, resulting in sub-optimal performance and irregular buildings.

To tackle these problems, some works adopt multi-task learning for polygonal building mapping. [20]–[23], [40] have emerged by integrating building segmentation, polygonization, and refinement into a unified framework to obtain polygonal buildings. FrameField [29] generates a frame field to provide structural information and then aligns the frame field to raster building segmentation for building polygonization. FrameField leverages multi-task learning to achieve a polygonization algorithm utilizing the frame field along with the raster segmentation. [41]–[44] first sample serialized vertices from building masks and then refine vertex positions using the designed refinement module. BuildMapper [25] is an end-to-end learnable building contour extraction framework with a learnable contour initialization module and a contour evolution module, which can directly extract building polygons. These models include complex modules with different threshold constraints to achieve each sub-task, making training challenging and computationally intensive. When dealing with inconsistent

Fig. 1. Different building mapping categorized into rasterized and polygonal mapping based on the output format. Rasterized mapping employs semantic or instance segmentation frameworks to obtain pixel-wise buildings shown in (a) and (b). Polygonal mapping is subdivided into multi-stage and single-stage pipelines based on whether to segment buildings explicitly. Multi-stage mapping typically adopts post-processing or multi-task learning for transforming pixel-wise masks to polygonal buildings shown in (c) and (d). Single-stage mapping designs serialized vertices or vertex connection prediction modules to obtain building serialized vertices directly shown in (e) and (f).

remote sensing images caused by complex imaging conditions, these methods remain the challenge of performance degradation. Additionally, these methods generally predict a fixed vertex number, resulting in vertex redundancy and insufficiency for different buildings.

B. Single-stage polygonal building mapping

Single-stage polygonal building mapping casts polygonal building mapping as extracting serialized vertices of a building. As shown in Figure 1(e), [24], [45], motivated by PolyRNNs [46], [47], first extracts building features and then iteratively predicts building serialized vertices by RNNs. TransBuilding [48] predicts polygonal buildings with a vertex transformer module and designs three self-attention modules in row-wise, column-wise, and vertex-wise to enhance geometric information of building features. These methods typically employ a two-stage detection framework (*i.e.*, Faster RCNN [49]) and add a serialized vertices prediction head parallel with building classification and bounding box regression heads. Figure 1(f) shows another line of works, which first detects all the vertices and then predicts the vertex connection matrix for assembling serialized vertices. PolyWorld [26] directly predicts a connection matrix to find the vertices of one building and the order of building vertices. Since PolyWorld [26] produces serialized vertices in a bottom-up pathway, missing or error vertices will influence the connection matrix learning, leading to self-intersection or non-closed polygons.

C. Transformers in CV

Since Transformers [50] are successful in natural language processing (NLP) with their powerful feature encoding abil-

ity, some researchers have extended them to computer vision (CV). Self-attention mechanism enables Transformers to model long dependencies, and multiple attention heads learn appropriate inductive bias, avoiding spatial constraints and inductive bias in convolutional operations. Hence, Transformers have achieved promising performance over CNN-based approaches [51] in CV. ViT [52] splits an image into non-overlapping patches and employs the standard transformer-based structure to process sequences of image patches, which has become a milestone work in vision transformers. Since ViT is a plain and non-hierarchical network, Swin Transformer [53] introduces a feature pyramid to extract multi-scale feature maps, which serves as a hierarchical backbone and facilitates Transformers in other tasks. Follow-up works [54]–[57] adopt hierarchical stages with spatial reduction layers and hybrid architectures with convolutional operations to efficiently extract local and global information. With notably advanced vision transformers have emerged in image classification, transformers have been successfully applied in various fields, such as detection [58], segmentation [59], and video [60] in CV. DETR [58] is the first transformer-based detection framework, representing object detection as a set prediction and matching problem and removing additional operations such as anchor generation and non-maximum suppression (NMS). SETR [59] reformulates semantic segmentation as a sequence-to-sequence prediction task and uses a pure Transformer to model the global context in transformer layers, which can provide a powerful segmentation model. This line of work formulates detection or segmentation problems as set prediction tasks and introduces learnable queries to extract targets in an auto-regressive manner, which can be applied in vertex prediction. Alfieri et al. [61] explore transformer-based architecture in

Fig. 2. Overview of HiT. HiT is a two-stage building mapping framework, which includes classification, bounding box regression, and polygon heads. The polygon head predicts serialized vertices of a building, together with building detection. We introduce a novel bidirectional polygon loss to train the polygon head without complex constraints.

polygon prediction, but it still needs a multi-layer Elman RNN [62] to generate serialized vertices iteratively. However, no prior work has exploited Transformer for serialized vertex prediction. This work leverages a pure Transformer to predict serialized vertices for polygonal building mapping, aiming to mitigate the research gap.

III. METHOD

In this section, We introduce the proposed HiT, a single-stage polygonal building mapping approach. In the following, we will first describe the overall pipeline of the proposed HiT and then describe each component in detail.

A. Overall pipeline

Polygonal building mapping has recently focused on iteratively predicted vertex on the condition of the predicted first vertex and the previous predicted vertices. Due to the long dependency problem, it remains very challenging to handle buildings with complex structures or occlusions and shadows caused by the imaging conditions. In response to these challenges, HiT models the vertex sequence on the condition of all the vertices by the designed polygon head. Specifically, we build the polygon head based on the insight that building polygons should be effectively delineated using a bidirectional vertex sequence. This innovative perspective makes serialized vertex prediction order-independent, enabling the model to concentrate on predicting vertex positions and relationships of any two adjacent vertices. Consequently, this alleviates the reliance on assumptions about the starting or ending vertices within the single-stage pipeline.

As shown in Figure 2, HiT employs a transformer-based architecture to simultaneously predict serialized vertices. Notably, the polygon head incorporates learnable and order-invariant vertex queries to dynamically predict serialized vertices for buildings with varying numbers at one time. Furthermore, HiT introduces a hierarchical attention mechanism at the vertex and edge levels, integrating convolution operations to amplify the encoding of geometric information for building features. A novel bidirectional polygon loss is further introduced to supervise the polygon head, thereby improving

the learning of sequence relationships within the query-based transformer module. In addition, the designed loss disregards the clockwise or counterclockwise orientation of the sequence, enhancing greater flexibility when predicting polygon vertex sequences. Finally, HiT directly extracts building polygons with appropriate vertices, achieving high performance compared to multi-stage and single-stage polygonal building mapping methods detailed in subsection IV-C.

B. Building detection

In building detection, HiT first extracts multi-scale features from the input image by the feature extraction module and then detects buildings by classification and regression heads. The feature extraction module consists of multi-scale feature extraction and candidate building region generation. Following Faster RCNN [49], HiT takes the ResNet50 [63] as the multi-scale backbone network in the feature extraction module. In this paper, multi-scale features C_i ($i \in [1, 2, 3, 4, 5]$) are firstly extracted from a remote sensing image $X \in \mathbb{R}^{3 \times H_0 \times W_0}$ by ResNet-50, of which the feature channels are {64, 256, 512, 1024, 2048} and the resolutions are {1/2, 1/4, 1/8, 1/16, 1/32} of the input image X. In order to accurately detect buildings of different sizes, an FPN fuses multi-scale features {C_2, C_3, C_4, C_5} in a top-down pathway with the lateral connection, which obtains new multi-scale features P_i ($i \in [2, 3, 4, 5, 6]$) with 256 channels and {1/4, 1/8, 1/16, 1/32, 1/64} resolutions of the input image X. Finally, an RPN is adopted to generate candidate building regions with three anchor aspect ratios {0.5, 1.0, 2.0} from fusion features {P_2, P_3, P_4, P_5, P_6}.

The building detection module outputs building classification scores and bounding boxes through the building classification head and the bounding box regression head. Given candidate building regions from the RPN, the building detection module first extracts the building feature map $B \in \mathbb{R}^{256 \times 7 \times 7}$ from multi-scale features P_i of the corresponding scale through a ROIAlign operation [37]. Subsequently, the building feature map B is flattened along the spatial and channel dimensions and goes through two connected linear layers to reduce channel dimension, which obtains the

Fig. 3. Illustration of the polygon head. The encoder with a hierarchical attention mechanism embeds more geometric information into the building feature. The decoder learns vertex queries to predict serialized vertices.

instance-level feature representation. Finally, a classification linear layer takes as input the building representation and predicts the building score (*i.e.*, building or background); on the other hand, a bounding box regression linear layer inputs the building representation and produces the building bounding box including center, width, and length.

C. Polygon prediction

HiT represents the vector polygon of a building as a bidirectional vertex sequence, of which the vertex order can be clockwise or equivalently counterclockwise. Therefore, the polygonal building mapping can be formulated as a sequence prediction task. As shown in Figure 3, HiT introduces a transformer-based polygon head to directly predict serialized vertices of a building at one time. Firstly, the polygon head also uses a ROIAlign operation [37] to extract the corresponding building feature map $B \in \mathbb{R}^{256 \times 20 \times 20}$, which is a large resolution compared to the building detection. Then, the encoder of the polygon head adopts a hierarchical attention mechanism to embed geometric information into the building feature map B. Finally, the decoder of the polygon head learns dynamic vertex queries from the building embedding for automatically serialized vertices prediction.

Encoder. The encoder of the standard transformer architecture exploits the self-attention operation to calculate the similarity between every two tokens of the sequence. However, serialized vertices are sparsely present in the building feature map B,

leading to the self-attention operation intensive computation, memory cost, and low efficiency. To deal with the sparsity of attention weights, the encoder of the polygon head replaces the original self-attention mechanism with the hierarchical attention mechanism to encode the building feature map B efficiently, which avoids the complexity and speeds up the convergence speed by introducing the geometric information in terms of vertex and edge levels.

As depicted in Figure 3, the encoder consists of vertex-level and edge-level blocks. Similar to the original encoder in the transformer [50], the two blocks have an attention operation and a feed-forward network (FFN), after which the short-cut connection and the layer normalization are also added. Since HiT adopts the hierarchical attention mechanism, the encoder directly takes the building instance feature B as input and outputs the building embedding, avoiding feature patchy and positional encoding. As shown in Figure 4, the encoder first uses four 3×3 convolution layers with the batch normalization and a rectified linear unit (ReLU [64]) to process the building feature map B. Then, a 1×1 convolution layer and a sigmoid activation function are used to obtain the vertex attention weight, and another 1×1 convolution layer, followed by a sigmoid operation, is used to obtain the edge attention weight. Finally, the vertex-level attention is calculated by multiplying the building feature map B and the vertex attention weight. Similarly, the edge-level attention is calculated by multiplying the building feature map B and the edge attention weight. The

Fig. 4. Illustration of vertex-level and edge-level attention operations. Vertex-level and edge-level attention replace the original self-attention mechanism to encode the building feature map, avoiding the complexity and speeding up the convergence speed by introducing the geometric information in terms of vertex and edge levels.

outputs of the vertex-level and edge-level attention are defined as:

$$
\begin{aligned}
B' &= 4 * [\text{ReLU}(\text{BN}(\text{conv}(B)))] \\
Attn_v &= B \otimes \sigma(\text{conv}(B')) \\
Attn_e &= B \otimes \sigma(\text{conv}(B'))
\end{aligned}
\tag{1}
$$

Afterward, the FFN is added to generate the final building embedding B_{emd}, which can increase the expressive ability of the model. In this paper, we integrate the information from vertex to edge level, enhancing building features. Besides, we will discuss the combination manner of vertex-level and edge-level attention in ablation studies.

Decoder. Unlike RNNs, which iteratively predict serialized vertices of a building, the decoder uses the vertex query $Q \in \mathbb{R}^{M \times 256}$ to predict the vertex sequence at one time. Like the original decoder of the standard transformer architecture, the decoder consists of N identical decoder blocks to perform multi-head self-attention and cross-attention. Each block includes a multi-head self-attention sub-layer, a cross-attention sub-layer, and an FFN sub-layer. Besides, the short-cut connection and the layer normalization are added after attention operations and FFN.

As shown in Figure 3, the vertex query Q added with the sinusoidal positional encoding is transformed by the multi-head self-attention operation, which outputs the vertex embedding followed by the short-cut connection and the layer normalization operations. Subsequently, the cross-attention sub-layer (CA_v) inputs the building embedding from the encoder as key and value and uses the output vertex embedding to automatically integrate the information of serialized vertices, after which the short-cut connection and the layer normalization operations are also used to enhance the output vertex embedding. Finally, the FFN sub-layer is applied with the short-cut connection and the layer normalization operations

to generate the final vertex embedding. The output of each block is defined as:

$$
\begin{aligned}
V' &= Q \oplus \text{PE}(Q) \\
V_{emd} &= \text{LN}(\text{SA}_v(V') \oplus V') \\
V_{emd} &= \text{LN}(\text{CA}_v(B_{emd}, B_{emd}, V_{emd}) \oplus V_{emd}) \\
V_{emd} &= \text{LN}(\text{FFN}(V_{emd}) \oplus V_{emd})
\end{aligned}
\tag{2}
$$

where PE is the positional encoding. SA_v and CA_v are the multi-head self-attention and the multi-head cross-attention operations. LN is the layer normalization operation. $\oplus$ means the element-wise addition. In the prediction, the final vertex embedding V_{emd} is fed to a linear layer and is transformed to M one-hot vectors by the softmax operation, indicating whether the vertex is a building vertex or not.

D. Training objective

Building detection. HiT consists of building classification and bounding box regression. Building classification loss L_{cls} is calculated by the binary cross-entropy loss:

$$
L_{cls} = -\frac{1}{N} \sum_{i=1}^{N} (y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)) \tag{3}
$$

where y_i represents the class, which is 1 for the building class or 0 for the background, and p_i is the predicted classification score. For building bounding box regression loss L_{bbox}, building detection is trained using a L1 loss:

$$
L_{bbox} = \frac{1}{N} \sum_{i=1}^{N} |t_i - t_i^*| \tag{4}
$$

where $t_i = (cx_i, cy_i, w_i, h_i)$ and $t_i^* = (cx_i^*, cy_i^*, w_i^*, h_i^*)$ represent the ground truth and predicted bounding boxes, respectively.

Polygon prediction. Since polygon prediction introduces geometric constraints in terms of vertex-level and edge-level, the vertex and edge prediction is trained by the focal loss [65]:

$$
\begin{aligned}
L_{ver} &= \begin{cases} -\alpha(1 - p^v)^\gamma \log(p^v) & \text{vertex} \\ -\alpha(p^v)^\gamma \log(1 - p^v) & \text{otherwise} \end{cases} \\
L_{edge} &= \begin{cases} -\alpha(1 - p^e)^\gamma \log(p^e) & \text{edge} \\ -\alpha(p^e)^\gamma \log(1 - p^e) & \text{otherwise} \end{cases}
\end{aligned}
\tag{5}
$$

where α and γ are the hyper-parameters, which is 2.0 and 4.0 in this paper. p^v and p^e denotes the predicted probability of vertex and edge, respectively.

For serialized vertices prediction loss L_{sv}, we use the binary cross-entropy loss to automatically predict a polygon with orientation invariant. In specific, we define $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ as the ground truth and predicted serialized vertices, respectively. For ease of understanding, we describe the calculation of L_{sv} by using the first vertex y_0 of $\mathbf{Y}$. As illustrated in Figure 5, we first search the reference vertex $\hat{y}_t$ from $\hat{\mathbf{Y}}$, which is closest to y_0. Subsequently, we move $\hat{y}_t$ to the first vertex $\hat{y}_0^{new}$ by sequentially shift the predicted sequence. Then, we fix $\hat{y}_0^{new}$ and inverse the other vertices in counter clockwise to get $\hat{\mathbf{Y}}_{inv}$. Finally, we calculate the minimum of the binary cross-entropy

Fig. 5. Illustration of the serialized vertices prediction loss L_{sv}. (a)Search the corresponding vertex. (b)Shift the predicted serialized vertices. (c)Inverse the predicted serialized vertices.

loss between ground truth $\mathbf{Y}$ and $\hat{\mathbf{Y}}$, as well as $\hat{\mathbf{Y}}_{inv}$. The L_{sv} is defined as follows:

$$
\begin{aligned}
\hat{y}_t &= \text{Search}(\hat{\mathbf{Y}}, y_0) \\
\hat{\mathbf{Y}} &= \text{Shift}(\hat{\mathbf{Y}}, \hat{y}_t) \\
\hat{\mathbf{Y}}_{inv} &= \text{Inverse}(\hat{\mathbf{Y}}) \\
L_{sv} &= \min(\mathbf{L}_{ce}(\mathbf{Y}, \hat{\mathbf{Y}}), \mathbf{L}_{ce}(\mathbf{Y}, \hat{\mathbf{Y}}_{inv}))
\end{aligned}
\tag{6}
$$

The polygon prediction loss L_{poly} is the the sum of L_{ver}, L_{edge}, and L_{poly}. Finally, the total loss of HiT is defined as:

$$
L = L_{cls} + L_{bbox} + L_{poly}
\tag{7}
$$

E. Implementation Details

HiT is implemented based on Faster RCNN [49] with a ResNet50 backbone, which is trained end-to-endly by using the PyTorch framework. For data augmentations during training and inference, input images are generally resized to 512×512, randomly flipped with a probability of 0.5, and normalized. During training, HiT is optimized by AdamW [66] optimizer. The initial learning rates of the backbone and the other are set to 1e-5 and 1e-4, respectively. The weight decay is set to 1e-4. The model is trained for 150 epochs with the learning rate dropped by 10 at the 90 and 130 epochs. For the model hyper-parameters and the joint training strategy, we have conducted extensive ablation studies, as illustrated in the section IV-D.

IV. EXPERIMENTS

A. Dataset

The proposed HiT is evaluated on two public building segmentation benchmark datasets, namely the CrowdAI Mapping Challenge dataset (CrowdAI dataset) [28] and the Inria Aerial Image Labeling dataset (Inria dataset) [67], to assess its performance and generalization. Buildings in the two large-scale building datasets cover many regions with different complex scenes and significantly vary in size, shape, structure, and appearance.

(1) CrowdAI dataset: The CrowdAI dataset is a large-scale satellite imagery of about 30 cm resolution with RGB channels, in which images have a size of 300×300 pixels and are annotated with polygonal building instances in MS-COCO [68] format. The training set consists of 280,741 images with

around 2,400,000 polygonal building instances. The test set has 60,317 images with 515,364 polygonal building instances. In addition, a small version that only includes 8,366 images for the training set and 1,820 images for the test set is also provided for comparison experiments. In our paper, the small version is used to conduct the following ablation studies to consider time-consuming and resource constraints.

(2) Inria dataset: The Inria dataset contains 180 aerial images of $5,000 \times 5,000$ pixels, covering different geographic locations (*i.e.*, United States and Austria) ranging from highly dense metropolitan financial districts to alpine resorts. Buildings of the Inria dataset have different urban settlement appearances and are annotated in binary masks, indicating pixels into building and not building classes. The Inria dataset has a spatial resolution of 30 cm and is split by the cities for the training set and the test set for assessing the model's generalization.

Since the source annotations in the Inria dataset are pixel-wise semantic masks, the Inria dataset is not suitable for supervising the model to extract polygonal building instances. Following the FrameField [29], we use the Inria Polygonized dataset to train our model, which converts the source annotations to polygonal MS-COCO [68] format. The images of the Inria Polygonized dataset are cropped into 512×512 patches with an overlap of 128. In the cropping stage, we remove buildings with an area smaller than 50% compared to the original building instance. Finally, we split the cropped images with the 75% for the training set and the 25% for the test set.

B. Evaluation Metrics

Building mapping from remote sensing images is a building instance segmentation task and a polygonal building extraction task. Therefore, we adopt two evaluation criteria to compare the proposed method with other methods.

Instance metric. For the instance segmentation, we adopt the average precision (AP) and the average recall (AR) under different intersection over union (IoU) thresholds provided by the standard MS-COCO metrics [68]. In order to evaluate the overall performance, we use AP and AR metrics, which present average precision and average recall under IoU thresholds ranging from 0.50 to 0.95 with a step of 0.05. Moreover, AP_{50}, AP_{75}, AR_{50} and AR_{75} are also calculated under IoU thresholds of 0.5 and 0.75 to measure the model's basic and higher performance. Besides, we report $F1$ to comprehensively assess the model's precision and recall, which is calculated as shown in Eq. 8.

$$
\begin{aligned}
AP &= \frac{AP_{0.50} + AP_{0.55} + \cdots + AP_{0.95}}{10} \\
AR &= \frac{AR_{0.50} + AR_{0.55} + \cdots + AR_{0.95}}{10} \\
F1 &= \frac{2 \times AP \times AR}{AP + AR}
\end{aligned}
\tag{8}
$$

where AP_i and AR_i measure average precision and average recall under IoU threshold i that is calculated by $IoU = (Pre \cap GT)/(Pre \cup GT)$.

TABLE I
RESULTS ON THE CROWDAI DATASET UNDER THE INSTANCE SEGMENTATION METRIC. THE BEST RESULTS ARE MARKED IN BOLD.

Method	AP ↑	AP_{50} ↑	AP_{75} ↑	AR ↑	AR_{50} ↑	AR_{75} ↑	F1 ↑
Mask RCNN [37]	41.9	67.5	48.8	47.6	70.8	55.5	44.6
PANet [69]	50.7	73.9	62.6	54.4	74.5	65.2	52.5
PolyMapper [24]	55.7	86.0	65.1	62.1	88.6	71.4	58.7
FrameField [29]	61.3	87.5	70.6	65.0	89.4	73.9	63.1
PolyWorld [26]	63.3	88.6	70.5	75.4	93.5	83.1	68.8
TransBuilding [48]	54.4	88.6	64.1	62.1	91.6	72.7	56.0
BuildMapper [25]	63.9	90.1	75.0	-	-	-	-
HiT (ours)	**64.6**	**91.9**	**78.7**	**75.5**	**93.8**	**83.5**	**69.6**

TABLE II
RESULTS ON THE CROWDAI DATASET UNDER THE POLYGONAL METRIC. "N RATIO (=1)" DENOTES THAT THE PERFORMANCE IS BETTER WHEN THE N RATIO IS CLOSER TO 1. THE BEST RESULTS ARE MARKED IN BOLD.

Method	C-IoU ↑	MTA ↓	N ratio (=1)
PolyMapper [24]	65.3	32.8	1.29
FrameField [29]	73.7	33.5	1.13
PolyWorld [26]	88.2	32.9	0.93
HiT (ours)	**88.6**	**31.7**	**1.00**

Polygonal metric. The polygonal metric considers the geometric properties of the extracted buildings, so we adopt three indicators to evaluate the extracted polygonal buildings, which consist of the N ratio [26], the complexity aware IoU (C-IoU) [26], and the Max Tangent Angle Error (MTA) [29]. The N ratio measures the simplicity of polygonal buildings by calculating the ratio between the predicted vertex number and the ground truth, which is defined as:

$$N\ ratio = \frac{\hat{V}_N}{V_N} \tag{9}$$

where V_N and $\hat{V}_N$ denote the vertex number of the prediction and the ground truth. When the model predicts redundant or insufficient vertices, the N ratio would be greater or less than 1. So, the N ratio is closer to 1, illustrating the better performance of the model. C-IoU is used to jointly assess the complexity and the segmentation of the extracted polygonal buildings, which is defined as:

$$RD = \frac{|V_N - \hat{V}_N|}{V_N + \hat{V}_N} \tag{10}$$

$$C\text{-}IoU = IoU(M, \hat{M}) \cdot (1 - RD)$$

where M and $\hat{M}$ denote masks of the predicted polygonal building and the ground truth. The C-IoU is higher when the model extracts polygonal buildings with accurate segmentation and precise polygonal complexity. MTA measures geometric shape by calculating the tangent angles between the predicted polygonal building and the ground truth, which is defined as:

$$T(V_i) = (V_{i+1} - V_i)/\|V_{i+1} - V_i\|$$
$$T(\hat{V}_i) = (\hat{V}_{i+1} - \hat{V}_i)/\|\hat{V}_{i+1} - \hat{V}_i\| \tag{11}$$
$$MAT = \max_{1 \le x \le N} \cos^{-1}(\langle T(V_i), T(\hat{V}_i) \rangle)$$

C. Results and Discussion

In our experiments, we select PolyMapper [24], Framefield [29], and PolyWorld [26]) for comparison, which are recently proposed state-of-the-art (SOTA) methods for polygonal building extraction. Besides, we compare HiT with more polygonal building mapping methods (TransBuilding [48] and BuildMapper [25]) on CrowdAI dataset. Moreover, we compare our method with classical instance segmentation methods, including Mask RCNN [37] and PANet [69] following the recent SOTA methods. We adopt the Douglas-Peucker algorithm [39] to polygonize pixel-wise segmentation masks to obtain polygonal results from binary building instance masks predicted by instance segmentation methods (Mask RCNN and PANet), which can be compared with the other polygonal building extraction for fair comparison. Our proposed HiT is similar to Mask RCNN, which all add branches for task-specific prediction in a standard two-stage detection framework (a mask prediction head in the Mask RCNN and a polygon prediction head in the proposed HiT). Therefore, we select the Mask RCNN as the baseline.

(1) Results on CrowdAI dataset.

For the CrowdAI dataset, the proposed HiT is compared with Mask RCNN [37], PANet [69], PolyMapper [24], FrameField [29], PolyWorld [26], TransBuilding [48] and BuildMapper [25] under the instance segmentation metric and the polygonal metric, respectively. Since TransBuilding [48] and BuildMapper [25] only evaluates performance on the instance segmentation metric, we have excluded them from Table II.

Quantitative Evaluation. Table I and II report the quantitative comparison results under the two metrics. From Table I, we can see that the proposed method outperforms all the comparison methods under the instance segmentation metric. Compared with the baseline instance segmentation method, the AP, AR, and F1 scores have been improved by +22.5%, +27.9%, and +24.9%, respectively. These significant improvements show that the designed polygon head can more efficiently extract building instances than the mask head in Mask RCNN. For polygonal segmentation methods, the proposed HiT has achieved AP gains of +8.7%, +2.7%, and +1.1% compared with polygonal building extraction methods (*i.e.*, PolyMapper, FrameField, and PolyWorld). In addition, the AR scores are +13.4%, +10.1%, and +0.1% higher than the three polygonal building extraction methods, respectively. Specifically, the F1 score of the proposed HiT has outperformed the three polygonal building methods by +10.8%, +6.0%, and +0.7%. All these results consistently show that the proposed HiT generates building instances with high precision and recall.

For the polygonal evaluation, our method compares with the three polygonal building extraction methods. As reported

Fig. 6. Qualitative results on CrowdAI. The proposed HiT can generate high-quality polygonal buildings of different sizes and shapes.

Fig. 7. Additional qualitative results on CrowdAI. HiT can accurately extract polygonal buildings.

in Table II, the performance of the proposed HiT significantly increases compared with other methods. Especially, the N ratio of HiT is closer to 1, which has redundant vertices for FrameField (N ration=1.13 > 1.0) and insufficient vertices for PolyWorld (N ratio=0.93 < 1.0). The results illustrate that our method generates more accurate vertices than polygonal building extraction methods. Moreover, our method has achieved the highest C-IoU score (88.6%), demonstrating that our method has better balanced building segmentation and geometric complexity among comparison methods. To measure the performance in polygonal building shape and

structure, HiT obtains the lowest MTA value, which means the lower the MTA value, the better the performance. Specifically, HiT gets 31.7% of the MTA indicator, which is 1.4%, 1.8%, and 1.2% lower than PolyMapper, FrameField, and PolyWorld, respectively.

The results in terms of instance segmentation and polygonal metrics have proved that our model has a high ability for accurately building segmentation with a more precise polygonal structure. Besides, a comprehensive comparison has verified our proposed HiT's superiority and effectiveness.

Qualitative Comparison. Figure 6 and 7 shows some visual-

TABLE III
RESULTS ON THE INRIA POLYGONIZED DATASET UNDER THE INSTANCE SEGMENTATION METRIC. THE BEST RESULTS ARE MARKED IN BOLD.

Method	AP $\uparrow$	AP_{50} $\uparrow$	AP_{75} $\uparrow$	AR $\uparrow$	AR_{50} $\uparrow$	AR_{75} $\uparrow$	$F1$ $\uparrow$
Mask RCNN [37]	40.0	79.2	35.3	51.5	87.3	54.4	45.0
PANet [69]	39.6	79.0	35.0	51.5	87.3	54.2	44.8
PolyMapper [24]	44.9	82.5	45.4	55.4	90.8	61.7	49.6
FrameField [29]	38.3	67.3	39.8	49.0	78.1	53.4	43.0
HiT (ours)	**50.5**	**86.1**	**56.6**	**60.6**	**91.2**	**71.0**	**55.1**

TABLE IV
RESULTS ON THE INRIA POLYGONIZED DATASET UNDER THE POLYGONAL METRIC. "N RATIO (=1)" DENOTES THAT THE PERFORMANCE IS BETTER WHEN THE N RATIO IS CLOSER TO 1. THE BEST RESULTS ARE MARKED IN BOLD.

Method	C-IoU $\uparrow$	MTA $\downarrow$	N ratio (=1)
PolyMapper [24]	41.5	34.4	1.6
FrameField [29]	49.4	**32.4**	2.1
HiT (ours)	**64.5**	33.2	**0.8**

TABLE V
MODEL COMPUTATIONAL COMPLEXITY. M AND G DENOTE MILLION AND GILLION, RESPECTIVELY.

Method	#Params (M) $\downarrow$	FLOPs (G) $\downarrow$
Mask RCNN [37]	43.8	114.7
PANet [69]	47.7	123.1
PolyMapper [24]	53.8	717.6
FrameField [29]	76.7	204.3
PolyWorld [26]	39.4	448.3
HiT (ours)	47.4	145.3

ization results generated by our approach and the comparison methods for qualitative comparison. HiT can successfully extract all polygonal buildings with high quality, including buildings of different sizes, appearances, and shapes.

Compared with FrameField, HiT predicts more precise vertices regarding number and position. While FrameField can predict all buildings, it predicts many redundant vertices, which is not suitable for real-world applications. In addition, PolyWorld extracts polygonal buildings in a down-top pathway, resulting in error vertex detection or insufficient vertices. Although both HiT and PolyMapper represent building mapping as a vertex sequence prediction task, HiT can better handle occlusions due to predicting serialized vertices simultaneously by the polygon prediction head. Moreover, the designed hierarchical attention mechanism embeds geometric information into the building feature map so that HiT can deal with buildings under complex scenes. Besides, the introduced polygon head exploits the supervisions from vertex, edge, and polygon, leading to more robustness and generalization. The qualitative results further demonstrate the superiority of HiT.

(2) Results on Inria Polygonized dataset.

In this subsection, we compare HiT on the Inria Polygonized dataset with Mask RCNN [37], PANet [69], PolyMapper [24], and FrameField [29] under the instance segmentation and polygonal metrics.

Quantitative Evaluation. Quantitative results are reported in Table III and IV from different methods under two metrics. For instance segmentation, our HiT has improved AP and AR on all the indicators shown in Table III. Compared with Mask RCNN, HiT achieves 50.5% AP (+10.0%), 60.6% AR (+9.1%), and 55.1% F1 (+10.1%), comprehensively indicating that HiT can generate highly accurate buildings. Since HiT is similar to Mask RCNN in the pipeline, the high performance of HiT demonstrates the effectiveness of the designed polygon prediction head. On the other hand, our HiT has improved the AP and AR scores by +5.6% and +5.2% compared with PolyMapper, respectively. In addition, the F1 score is +5.5% higher than the SOTA polygonal building extraction method, as shown in Table III. Specifically, the AP_{75} and AR_{75}

scores of the proposed HiT have outperformed the polygonal segmentation methods by +11.2% and +9.3%, respectively. These high improvements indicate that HiT generates building instances with high precision and recall.

Table IV reports the polygonal evaluation from different polygonal building extraction methods. We can see that HiT significantly increases performances than comparison methods on all indicators. HiT detects buildings with more accurate vertices than other methods on the N ratio, but FrameField generates many more vertices with a large N ratio. Moreover, our method has significantly improved the C-IoU score by +15.1%, illustrating that our method has better balanced building segmentation and geometric complexity.

The quantitative results on the Inria Polygonized dataset under different metrics have further proved that our model generates polygonal buildings with more accurate segmentation masks and precise polygonal structures.

Qualitative Comparison. We show some qualitative results from our approach and comparison methods in Figure 8. Compared with other methods, HiT extracts more accurate polygonal buildings of different sizes, appearances, and shapes. Moreover, HiT is more robust in dealing with images with complex scenes by using a transformer-based structure to predict the vertex sequence simultaneously. The visualization results consistently demonstrate the superiority of HiT.

(3) Discussion.

In this section, we evaluate model complexity among comparison methods and then discuss performance in terms of model structure, complexity, accuracy and robustness. For the model complexity, we calculate model parameters (#Params (M)) and floating point operations (FLOPs (G)) by testing an image with a resolution of 512×512 on 1 GPU for all comparison methods. As reported in Table V, our approach has lower #Params and higher FLOPs than instance segmentation methods since HiT directly generates polygonal buildings rather raster building masks. Compared with polygonal building mapping, our model has much lower parameters, except PolyWorld [26], and lower FLOPs than comparison methods, demonstrating our method can effectively extract polygonal

| GT | PolyMapper | FFL | HiT |

Fig. 8. Qualitative results on the Inria Polygonized dataset. The proposed HiT can better extract buildings in dense areas. Zoom in for a cleaner view.

Fig. 9. Model complexity and F1 score comparison among different methods. The floating point operations (FLOPs (G)) is denoted by the radius of the circle.

buildings. Figure 9 shows comprehensive comparison between complexity and accuracy (F1 score). We use the radius of the circle to denote the floating point operations (FLOPs (G)). We can see that HiT can accurately extract polygonal buildings with lower #Params and FLOPs. In addition, we discuss the model robustness using the CrowdAI dataset, which encompasses large-scale buildings across diverse regions, including urban, suburban, and rural landscapes. Quantitative results in Tables I and II reveal that HiT consistently achieves high performance and robustness compared to alternative methods. As depicted in Figures 6 and 7, HiT accurately delineates building polygons for sparse and dense buildings. HiT exhibits acceptable proficiency in handling occlusions or shadows, demonstrating its robustness.

HiT is built on a insightful perspective that a building polygon can be effectively formulated as a bidirectional vertex sequence. Hence, a simple polygon head is designed for serialized vertex prediction. The polygon head combine attention with convolution operations to encoding building features with rich geometric and semantic information in a hierarchical manner. Moreover, a bidirectional polygon loss

TABLE VI
RESULTS FOR DIFFERENT ENCODING MECHANISMS IN THE POLYGON HEAD ON CROWDAI-S DATASET UNDER THE INSTANCE SEGMENTATION METRIC.
THE BEST RESULTS ARE MARKED IN BOLD.

Method	AP ↑	AP_{50} ↑	AP_{75} ↑	AR ↑	AR_{50} ↑	AR_{75} ↑	F1 ↑
Baseline	31.1	64.1	27.5	44.3	78.4	45.9	36.5
(a) Original	36.6	72.3	34.4	48.5	82.4	51.8	41.7
(b) Vertex-enhanced	34.0	68.9	30.9	46.4	80.4	48.8	39.2
(c) Edge-enhanced	31.7	65.5	27.9	44.7	78.9	46.3	37.1
(d) Vertex-edge-enhanced	37.5	74.1	35.7	48.7	82.9	52.1	42.4
(e) Vertex-wise	32.0	66.4	28.3	45.1	79.2	47.0	37.4
(f) Edge-wise	33.7	68.3	30.6	46.4	80.2	48.9	39.0
(g) Vertex-edge-wise	36.6	72.6	34.7	48.2	82.4	51.6	41.6
(h) Hierarchical (ours)	**38.5**	**75.3**	**37.6**	**49.3**	**83.5**	**53.2**	**43.2**

guide the model to pay more attention on vertex positions and relationships, rather than the clockwise or counterclockwise orientation of the vertex sequence. Consequently, HiT has greater flexibility in polygonal building mapping.

D. Ablation Study

Our method designs a transformer-based polygon prediction head to extract building serialized vertices parallel with building classification and building bounding box regression by a two-stage detection framework. In the polygon prediction head, the encoder with the hierarchical attention operation is proposed to encode building feature maps with geometric and semantic information. The designed polygon prediction head is optimized using the serialized vertices prediction loss joint with the vertex and edge prediction loss. In this subsection, we perform ablation studies on the CrowdAI dataset to further analyze the effectiveness of the details of our approach, including the encoding mechanism, the decoding setting, and the training strategy. In the ablation experiments, we remove the encoder from the polygon prediction head and train the modified model on the small version of the CrowdAI dataset (CrowdAI-S), which is used as the baseline.

(1) Encoding mechanism. The encoder of the polygon head plays a significant role in the serialized vertex prediction. In the encoding ablation experiments, we conduct eight different encoding mechanisms and remove the encoder from the polygon prediction head as a baseline. Figure 10 shows different encoding methods discussed in the following.

(a) The original encoding mechanism. The original encoding manner uses an element-wise self-attention operation to obtain the relationship from the input sequence. As shown in Figure 10(a), the flattened building feature map is processed through N identical layers with self-attention and FFN operations.

(b) The vertex-enhanced mechanism. We use convolutional operations to get a vertex feature map shown in Figure 10(b). Then, the building and vertex feature maps are concatenated to enhance vertex information.

(c) The edge-enhanced mechanism. In Figure 10(c), the edge-enhanced encoding fashion is similar to the vertex-enhanced encoding, which can introduce edge information for serialized vertex prediction.

(d) The vertex-edge-enhanced mechanism. This encoding method obtains the vertex and edge features separately and then concatenates with the building feature, evaluating a joint

Fig. 10. Illustration of the vertex-level and the edge-level attention operations. The vertex-level and the edge-level attention replace the original self-attention mechanism to encode the building feature map, avoiding the complexity and speeding up the convergence speed by introducing the geometric information in terms of vertex and edge levels.

enhanced encoding in vertex and edge levels shown in Figure 10(d).

(e) The vertex-wise attention mechanism. We formulate the building vertex probabilities as attention weights and multiply them with the building feature, enhancing vertex information in the building feature. Besides, the short-cut connection and layer normalization are used to get the building embedding, as described in Figure 10(e).

(f) The edge-wise attention mechanism. Like vertex-wise attention, the edge-wise attention mechanism multiplies edge prediction probabilities with the building feature, followed by short-cut connection and layer normalization operations.

(g) The vertex-edge-wise attention. We simultaneously exploit vertex-wise and edge-wise attention products to enhance the building feature map, as shown in Figure 10(g).

(h) The hierarchical attention mechanism. Motivated by the original encoding mode, the hierarchical attention mechanism replaces the self-attention operation with vertex-wise and edge-wise attentions due to the sparsity of serialized vertices in the building feature map B.

As shown in Table VI, the hierarchical attention mechanism significantly improves all the evaluation metrics compared

with other encoding methods. We can see that encoding methods in concatenation and multiplication manners improve performance in all the indicators and show comparable performance to the original encoding method, proving that geometric information and the original encoding pipeline are effective in feature encoding. Motivated by this observation, we replace self-attention with hierarchical attention to introduce geometric information in building embeddings.

(2) Decoding setting. The polygon head uses N identical decoder blocks to predict serialized vertices simultaneously. In each decoder block, the multi-head self-attention operation encodes the relationship among all the vertex queries. In this ablation experiment, we test head number H and block number N to select the optimal hyper-parameters. Finally, we can observe from Table VII that optimal hyper-parameters are set as H=4 and N=8.

TABLE VII
RESULTS FOR DIFFERENT HYPER-PARAMETERS IN THE DECODER OF THE POLYGON HEAD. H AND N REPRESENT HEAD NUMBER AND BLOCK NUMBER. THE BEST RESULT IS MARKED IN BOLD.

	H=1		H=2		H=4		H=8	
	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}
N=1	31.1	65.5	33.4	68.6	35.3	72.3	35.8	72.5
N=2	33.1	67.8	34.9	70.6	32.8	67.0	36.9	73.3
N=4	31.4	65.9	31.9	67.6	34.0	68.3	33.4	67.8
N=6	33.0	67.4	34.3	69.3	35.8	71.6	33.4	67.9
N=8	34.4	70.4	37.1	73.8	**38.5**	**75.3**	37.3	73.2

(3) Training strategy. HiT is jointly trained by building classification, bounding box regression, and polygon prediction losses. In this ablation study, we select the optimal weighting coefficients λ_{cls}, λ_{bbox}, and λ_{poly} to balance different modules. In Table VIII, the training objective is optimal when weighting coefficients are all set to 1.0.

TABLE VIII
RESULTS FOR WEIGHTING COEFFICIENTS λ_{cls}, λ_{bbox}, AND λ_{poly} SELECTION IN THE JOINT TRAINING. THE BEST RESULT IS MARKED IN BOLD.

λ_{cls}	λ_{bbox}	λ_{poly}	AP	AP_{50}	AR	AR_{50}
1.0	1.0	0.01	31.9	64.6	44.6	76.0
1.0	1.0	0.1	34.3	69.1	46.6	79.4
10.0	10.0	0.1	32.1	65.3	44.7	76.0
10.0	10.0	1.0	34.3	69.1	46.8	79.3
1.0	1.0	1.0	**38.5**	**75.3**	**49.3**	**83.5**

V. CONCLUSION

We have presented HiT for automatically building mapping from remote sensing images. In this paper, we represent a building with serialized vertices, which can be formulated as a bidirectional vertex sequence. Based on this new observation, we apply a hierarchical transformer-based structure to predict serialized vertices. In the hierarchical transformer, we combine the CNN operation and transformer structure to embed semantic and geometry information, obtaining more effective building representations and capturing better building boundaries and corners. Moreover, we introduce a novel bidirectional polygon loss with bidirectional properties to train HiT end-to-endly. Finally, our extensive experiments illustrate

that HiT significantly outperforms state-of-the-art methods, demonstrating its superiority.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[4] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[5] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Transactions on geoscience and remote sensing*, vol. 57, no. 1, pp. 574–586, 2018.

[6] Z. Zhang, W. Guo, M. Li, and W. Yu, "Gis-supervised building extraction with label noise-adaptive fully convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 12, pp. 2135–2139, 2020.

[7] M. Guo, H. Liu, Y. Xu, and Y. Huang, "Building extraction based on u-net with an attention block and multiple losses," *Remote Sensing*, vol. 12, no. 9, p. 1400, 2020.

[8] Q. Zhu, Z. Li, Y. Zhang, and Q. Guan, "Building extraction from high spatial resolution remote sensing images via multiscale-aware and segmentation-prior conditional random fields," *Remote Sensing*, vol. 12, no. 23, p. 3983, 2020.

[9] C. Wang and L. Li, "Multi-scale residual deep network for semantic segmentation of buildings with regularizer of shape representation," *Remote Sensing*, vol. 12, no. 18, p. 2932, 2020.

[10] Q. Wen, K. Jiang, W. Wang, Q. Liu, Q. Guo, L. Li, and P. Wang, "Automatic building extraction from google earth images under complex backgrounds based on deep instance segmentation network," *Sensors*, vol. 19, no. 2, p. 333, 2019.

[11] T. Wu, Y. Hu, L. Peng, and R. Chen, "Improved anchor-free instance segmentation for building extraction from high-resolution remote sensing images," *Remote Sensing*, vol. 12, no. 18, p. 2910, 2020.

[12] L. Xu, Y. Li, J. Xu, and L. Guo, "Gated spatial memory and centroid-aware network for building instance extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[13] F. Shi and T. Zhang, "A multi-task network with distance–mask–boundary consistency constraints for building extraction from aerial images," *Remote Sensing*, vol. 13, no. 14, p. 2656, 2021.

[14] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1480–1484.

[15] Q. Zhu, C. Liao, H. Hu, X. Mei, and H. Li, "Map-net: Multiple attending path neural network for building footprint extraction from remote sensed imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 6169–6181, 2020.

[16] Q. Li, L. Mou, Y. Hua, Y. Shi, and X. X. Zhu, "Building footprint generation through convolutional neural networks with attraction field representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2021.

[17] S. Chen, W. Shi, M. Zhou, M. Zhang, and Z. Xuan, "Cgsanet: A contour-guided and local structure-aware encoder–decoder network for accurate building extraction from very high-resolution remote sensing imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 1526–1542, 2021.

[18] K. Zhao, J. Kang, J. Jung, and G. Sohn, "Building extraction from satellite images using mask r-cnn with building boundary regularization," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 247–251.

[19] S. Wei, S. Ji, and M. Lu, "Toward automatic building footprint delineation from aerial images using cnn and regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 3, pp. 2178–2189, 2019.

[20] S. Zorzi, K. Bittner, and F. Fraundorfer, "Machine-learned regularization and polygonization of building segmentation masks," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 3098–3105.

[21] S. Wei and S. Ji, "Graph convolutional networks for the automated production of building vector maps from aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2021.

[22] S. Wei, T. Zhang, and S. Ji, "A concentric loop convolutional neural network for manual delineation-level building boundary segmentation from remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2021.

[23] Z. Xu, C. Xu, Z. Cui, X. Zheng, and J. Yang, "Cvnet: Contour vibration network for building extraction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1383–1391.

[24] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1715–1724.

[25] S. Wei, T. Zhang, S. Ji, M. Luo, and J. Gong, "Buildmapper: A fully learnable framework for vectorized building contour extraction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 87–104, 2023.

[26] S. Zorzi, S. Bazrafkan, S. Habenschuss, and F. Fraundorfer, "Polyworld: Polygonal building extraction with graph neural networks in satellite images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1848–1857.

[27] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[28] S. P. Mohanty, J. Czakon, K. A. Kaczmarek, A. Pyskir, P. Tarasiewicz, S. Kunwar, J. Rohrbach, D. Luo, M. Prasad, S. Fleer *et al.*, "Deep learning for understanding satellite imagery: An experimental survey," *Frontiers in Artificial Intelligence*, vol. 3, p. 534696, 2020.

[29] N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka, "Polygonal building extraction by frame field learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5891–5900.

[30] Y. Su, L. Gao, M. Jiang, A. Plaza, X. Sun, and B. Zhang, "Nsckl: Normalized spectral clustering with kernel-based learning for semisupervised hyperspectral image classification," *IEEE Transactions on Cybernetics*, 2022.

[31] T. Guo, R. Wang, F. Luo, X. Gong, L. Zhang, and X. Gao, "Dual-view spectral and global spatial feature fusion network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[32] S. Liu, H. Ye, K. Jin, and H. Cheng, "Ct-unet: Context-transfer-unet for building segmentation in remote sensing images," *Neural Processing Letters*, vol. 53, pp. 4257–4277, 2021.

[33] L. Mou and X. X. Zhu, "Rifcn: Recurrent network in fully convolutional network for semantic segmentation of high resolution remote sensing images," *arXiv preprint arXiv:1805.02091*, 2018.

[34] J. Li, Y. Zhuang, S. Dong, P. Gao, H. Chen, L. Chen, and L. Li, "Hierarchical disentangling network for building extraction from very high resolution optical remote sensing imagery," *Remote Sensing*, vol. 14, no. 7, p. 1767, 2022.

[35] S. Li, T. Bao, H. Liu, R. Deng, and H. Zhang, "Multilevel feature aggregated network with instance contrastive learning constraint for building extraction," *Remote Sensing*, vol. 15, no. 10, p. 2585, 2023.

[36] Y. Xie, J. Zhu, Y. Cao, D. Feng, M. Hu, W. Li, Y. Zhang, and L. Fu, "Refined extraction of building outlines from high-resolution remote sensing imagery based on a multifeature convolutional neural network and morphological filtering," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1842–1855, 2020.

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[38] L. We, "Marching cubes: A high resolution 3d surface construction algorithm," *Comput Graph*, vol. 21, pp. 163–169, 1987.

[39] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[40] A. Hu, L. Wu, S. Chen, Y. Xu, H. Wang, and Z. Xie, "Boundary shape-preserving model for building mapping from high-resolution remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[41] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler, "Fast interactive object annotation with curve-gcn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5257–5266.

[42] B. Xu, J. Xu, N. Xue, and G.-S. Xia, "Accurate polygonal mapping of buildings in satellite imagery," *arXiv preprint arXiv:2208.00609*, 2022.

[43] W. Li, W. Zhao, J. Yu, J. Zheng, C. He, H. Fu, and D. Lin, "Joint semantic–geometric learning for polygonal building segmentation from high-resolution remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 201, pp. 26–37, 2023.

[44] M. Khomiakov, M. R. Andersen, and J. Frellsen, "Polygonizer: An autoregressive building delineator," *arXiv preprint arXiv:2304.04048*, 2023.

[45] W. Zhao, C. Persello, and A. Stein, "Building outline delineation: From aerial images to polygons with an improved end-to-end learning framework," *ISPRS journal of photogrammetry and remote sensing*, vol. 175, pp. 119–131, 2021.

[46] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5230–5238.

[47] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 859–868.

[48] M. Zhang, Q. Liu, W. Wang, and Y. Wang, "Transbuilding: An end-to-end polygonal building extraction with transformers," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 460–464.

[49] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[50] 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.

[51] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[52] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[53] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[54] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 22–31.

[55] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pvt v2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.

[56] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 009–12 019.

[57] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4794–4803.

[58] 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.

[59] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[60] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8741–8750.

[61] A. Alfieri, Y. Lin, and J. C. van Gemert, "Investigating transformers in the decomposition of polygonal shapes as point collections," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2076–2085.

[62] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[64] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[65] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[66] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," in *International Conference on Learning Representations*, 2018.

[67] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017, pp. 3226–3229.

[68] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[69] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

Mingming Zhang received the BS degree in information and computer science from Liaoning University, Shenyang, China and the MS degree in software engineering from Beihang University, Beijing, China. She is currently pursuing the Ph.D. degree in computer science with the Laboratory of Intelligent Recognition and Image Processing, School of Computer Science and Engineering, Beihang University. Her research interests include remote sensing image analysis and computer vision.

Qingjie Liu received the BS degree in computer science from Hunan University, Changsha, China, in 2007, and the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2014. He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University. He is also a Distinguished Research Fellow with the Hangzhou Institute of Innovation, Beihang University, Hangzhou. His current research interests include image fusion, object detection, image segmentation, and change detection. He is a member of the IEEE.

Yunhong Wang received the BS degree in electronic engineering from Northwestern Polytechnical University, Xi'an, China, in 1989, and the MS and Ph.D. degrees in electronic engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1995 and 1998, respectively.

She was with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, from 1998 to 2004. Since 2004, she has been a Professor with the School of Computer Science and Engineering, Beihang University, Beijing, where she is also the Director of the Laboratory of Intelligent Recognition and Image Processing. Her research interests include biometrics, pattern recognition, computer vision, data fusion, and image processing. She is a Fellow of IEEE, IAPR, and CCF.