PolyGraph: A Graph-Based Method for Floorplan Reconstruction From 3D Scans

Qian Sun, *Member, IEEE*, Chenrong Fang, Shuang Liu, Yidan Sun, Yu Shang, and Ying He

***Abstract*—The task of reconstructing indoor floorplans has become an increasingly popular subject, offering substantial benefits across various applications such as interior design, virtual reality, and robotics. Despite the growing interest, existing approaches frequently encounter challenges due to high computational costs and sensitivity to errors in primitive detection. In this article, we introduce PolyGraph, a new computational framework that combines a deep-learning based primitive detection network with an optimization-based reconstruction algorithm to facilitate high-quality reconstruction results. Specifically, we develop a novel guided wall point primitive estimation network capable of generating dense samples along wall boundaries. This network not only retains structural detail but also shows improved robustness in the detection phase. Then, PolyGraph utilizes wall points to establish a graph-based representation, formulating indoor floorplan reconstruction as a subgraph optimization problem. This approach significantly reduces the search space comparing to existing pixel-level optimization approaches. By utilizing "structural weight", we seamlessly integrate the structural information of walls and rooms into graph representations, ensuring high-quality reconstruction results. Experimental results demonstrate PolyGraph's effectiveness and its advantages compared to other optimization-based approaches, showcasing its computational efficiency, and its ability to preserve structural integrity and capture fine details, as quantified by the structure metrics.**

***Index Terms*—Floorplan reconstruction, structured reconstruction, deep learning, graph optimization.**

I. Introduction

HUMAN eyes have a remarkable capacity for holistic structural reasoning. We can effortlessly identify structural primitives (e.g., corners and edges of buildings) and their relationships. A fundamental challenge in computer vision research is to achieve this level of human perception and ultimately reconstruct global geometry from images. This would have profound implications for wider fields including visual effects, architecture, manufacturing, and robotics. Inferring the structure of indoor building floorplans is a representative task in structural reasoning, with the automatic reconstruction of floorplans from raw sensor data being one of the research hotspots. Floorplan reconstruction can improve the efficiency of virtual reality production [1], enabling visual preview and virtual evaluation of architectural design. Users can interact with the virtual reality environment, such as observing different perspectives of the building and entering the interior of the building. This enhanced interactivity can provide a richer virtual reality experience [2], allowing users to more deeply understand and experience various aspects of the building. Meanwhile, indoor plan reconstruction provides accurate spatial information, including the size, shape, direction, and layout of the rooms, which can provide a precise foundation for urban reconstruction tasks [3].

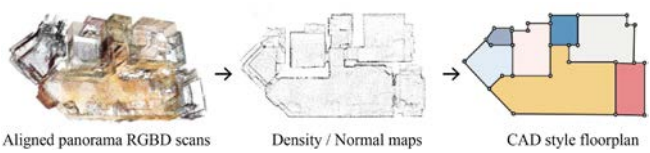

Fig. 1. Given an aligned panoramic RGBD scan, our task is to reconstruct it into a compact, closed and accurate CAD-like floorplan.

The central challenge in floorplan inference lies in the reasoning of wall structures, as their topology is unknown and differs from one example to another. The goal of floorplan reconstruction is to transform indoor scenes into bird's-eye view 2D vector graphs, i.e., extracting CAD-like interior floorplans from RGBD scans as shown in Fig. 1. A floorplan consists of corner points and edges, with the areas bounded by these edges representing rooms and the overall structure provides a concise depiction of the interior from a bird's-eye perspective. There are currently two mainstream methods for floorplan reconstruction: the first uses a bottom-up primitive detection method, extracting the corner position directly from the input, and predicting the edge relationship through the combination of corner points. Such methods often use an end-to-end network with fast inference speed, and the results are highly affected by the accuracy of primitive detection at the first stage. The second method which is top-down often adopts heuristic room optimization approaches. They get rough room masks based on image segmentation and

Received 28 February 2024; revised 6 February 2025; accepted 15 February 2025. Date of publication 24 February 2025; date of current version 5 September 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 61702363 and Grant 62472429, and in part by the Ministry of Education, Singapore, under its Academic Research Fund under Grant MOE-T2EP20220-0005 and Grant RT19/22. Recommended for acceptance by Haibin Ling. *(Corresponding author: Qian Sun.)*

Qian Sun is with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: sunqian@nuist.edu.cn).

Chenrong Fang and Yu Shang are with the College of Intelligence and Computing, Tianjin University, Tianjin 300000, China (e-mail: fangchenrong@tju.edu.cn; shangyu@tju.edu.cn).

Shuang Liu is with the School of Information, Renmin University of China, Beijing 100872, China (e-mail: shuang.liu@ruc.edu.cn).

Yidan Sun is with the Cyber Security Research Centre at NTU (CYSREN), Nanyang Technological University, Singapore 639798 (e-mail: yidan.sun@ntu.edu.sg).

Ying He is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: yhe@ntu.edu.sg).

The source code is publicly available at https://github.com/Fern327/PolyGraph.

Digital Object Identifier 10.1109/TVCG.2025.3544769

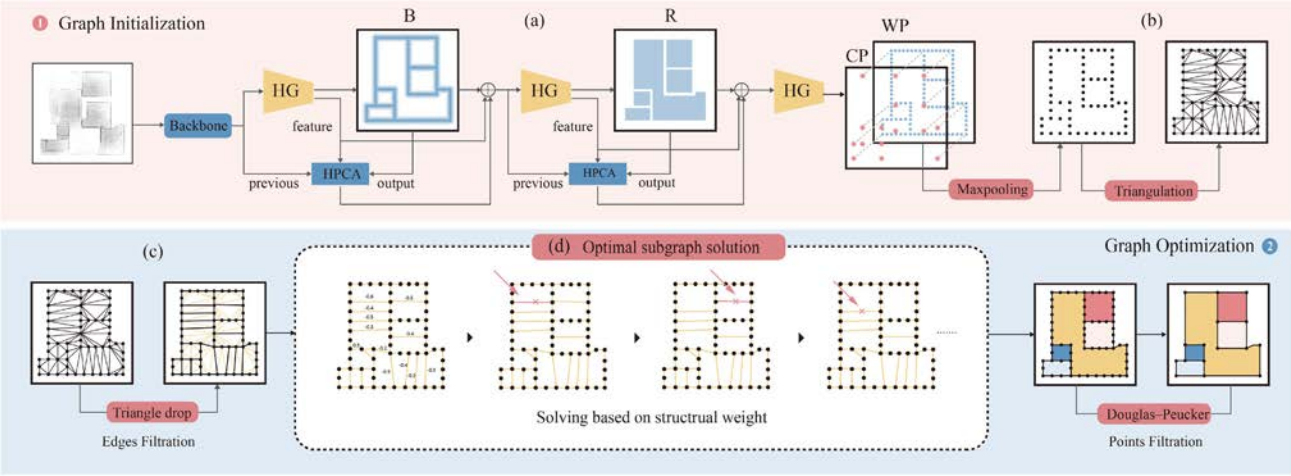

Fig. 2. The overall framework of PolyGraph consists of two processes: (1) Graph initialization and (2) Graph optimization. Given the 2D point density/normal map of the input point cloud, (a) the cross-guided neural network successively generates heatmaps for boundary (B), region (R), crucial points (CP), and wall points (WP). The wall points are obtained by max-pooling to determine the node positions and form the initial graph structure through (b) triangulation. After (c) triangle pruning, the (d) subgraph-solving process yields the optimal subgraph, and the point filtration produces the ultimate reconstruction result.

use heuristic functions to optimize room contours and combine them as an overall structure. However, this heuristic approach requires a high computing time cost. The recent study RoomFormer [4] implements an end-to-end approach based on the Transformer architecture, which regards the overall structure as a combination of contour loops of individual rooms. However, it has weak constraints on the shared edges between adjacent rooms, resulting in reconstruction results that lack overall structural compactness and consistency.

This study combines two mainstream approaches and proposes a new reconstruction pipeline, PolyGraph. It begins with a 2D density/normal map derived from raw 3D point data as the initial input, first detects structural wall points, and then proceeds to optimize the overall floorplan structure. Our main contributions are:

- For the first time, we model the problem of indoor holistic floorplan reconstruction as an optimal subgraph optimization task. To achieve this, we propose PolyGraph, to construct a graph-based representation of the overall floorplan, and then develop an efficient subsequent optimization procedure. The entire framework not only could achieve the highest computational efficiency among optimization-based approaches but also could generate high-quality reconstruction results.
- Unlike existing pixel-level optimization methods, our graph representations could largely reduce the search space, by building up upon a novel primitive (i.e., wall points). We propose a guided primitive estimation network to generate points that could capture more comprehensive structural information about the walls and possess higher error tolerance compared to sole corner points. After processing triangulation, the initial graph is generated with rich structural information.
- Then, we design a new structural weight, which plays a key role in solving the optimal subgraph problem. The weight considers both the confidence degree as a real wall and the impact of the length of the wall. By introducing it into our developed queue-based greedy algorithm, our method results in less deformation and successfully incorporate more details in the reconstructed results.
- Extensive experiments demonstrate that our method effectively captures fine details and maintains the overall integrity of the floor plan, resulting in a more visually pleasing and accurate representation of the floorplan structure. In addition, unlike metrics that solely measure the quality of individual primitive reconstructions, we introduce three metrics to evaluate the overall structural quality, for more comprehensive and overall comparison, in which we achieve the best performance.

II. Related Works

We classify structured reconstruction algorithms into three categories: classical techniques, Top-down methods and Bottom-up methods.

A. Classical Techniques

Structured reconstruction is a task of structure inference, same as human pose estimation [5] and image semantic relationships inference [6]. It focuses on extracting the vector structure from the input. Different reconstruction tasks require different transformed vectorized geometries, such as wireframes [7], planes [8], [9], [10], room layouts [11], [12], and polygonal loops [13]. Early methods relied on basic image processing methods such as histograms [14], Hough transform [15], [16], superpixel segmentation [17], and plane fitting [18]. For example, [14] reconstructs floorplan by detecting vertical planes in a 3D point cloud by building a histogram of the vertical positions of all measured points. [8], [9] perform planar structure reconstruction based on the graph-cut model. [19] inferences the floorplan by dynamic programming. And [20] use Bayesian networks for room layout estimation. However, these methods

rely heavily on heuristics, which are prone to failure on noisy data and have low robustness.

B. Bottom-Up Methods

Bottom-up reconstruction methods are usually divided into two stages, detecting low-level primitives such as corners, and then selecting these primitives to form high-level primitives such as edges or regions. In the wireframe parsing task, L-CNN [21] uses a likelihood prediction convolutional network (ConvNet) for node detection followed by an edge verification network. PPGNet [22] and HAWP [23] further refined the method for better performance. The recent success of the Transformer-based object detector DETR [24] is also extended to wireframe parsing in LETR [7]. DETR/LETR utilizes ”pseudo-nodes” as placeholders to store detection primitives. These techniques detect edge candidates independently, resulting in weak spatial correlation between rooms. While indoor structure reconstruction needs to pay more attention to the integrity and closure of the overall structure.

Floornet [25] proposes a benchmark for reconstructing floorplan from sensor data for the problem of indoor structural reconstruction. The corner primitives detected by the hybrid network are divided into thirteen categories according to the direction and the number of connections. The combined optimal solution of the wall is obtained based on integer optimization. However, Floornet has a limited number of primitive categories and cannot handle non-Manhattan scenes. HEAT [26] proposes an end-to-end model following a typical bottom-up pipeline. It detects the corners and then classifies the edge proposals between the corners. However, there are two drawbacks to this type of reconstruction pipeline. The absence of low-level primitives in the detection phase automatically leads to the loss of higher-level primitives (walls and rooms). Second, spurious candidate primitives may cause redundant walls and rooms to be reconstructed. Our method performs primitive detection based on heatmap regression instead of position regression for new types of primitives (wall points), reducing the impact of the above defects. Through overall structure optimization, our results achieve optimal structural integrity and consistency.

C. Top-Down Methods

Typical top-down methods usually view the floorplan as a collection of polygons representing individual rooms. The key lies in extracting the outline structures of each room.

For example, one approach is to extract the room outlines by tracing pixel chains in the point density image and then simplify them to form polygons. Contours can be extracted for instance by popular object saliency detection methods [27], [28] or interactive techniques such as Grabcut [29]. Chains of these pixels form the dense polygons then are simplified to concise polygons by the popular Douglas-Peucker algorithm [30] or edge contractions on Delaunay triangulation [31]. However, these vectorization pipelines cannot guarantee good topological accuracy because polygons are processed sequentially without global consistency.

More sophisticated solvers are then proposed. For example, [32] starts from room segmentation via instance semantic segmentation technique (Mask-RCNN [33]), transforming the extraction of room contours into a novel optimization problem, where room-wise coordinate descent sequentially solves shortest path problems to optimize the floorplan graph structure. After this, most top-down methods follow a similar detection pipeline. [34] uses the Monte Carlo tree search algorithm to reconstruct the floorplan and achieves better performance. The success of these techniques depends on the hand-optimized mastery of the domain of room shape and layout, however is a few orders of magnitude slower at test time. Recent work such as RoomFormer [4], based on the Transformer architecture, achieves state-of-the-art results by parallelly generating polygons for multiple rooms in a holistic manner. In contrast, our method converts the floorplan into a graph structure and optimizes the solution, leading to better performance in terms of structural compactness and consistency.

III. Method

We propose PolyGraph, a graph-based method for floorplan reconstruction. The overall architecture is shown in Fig. 2. It converts aligned panoramic RGBD images into a planar layout map in two stages: A) Graph initialization. In this step, the aligned panoramic RGBD scan is first transformed into a 2D point density/normal map, which serves as the input to a heatmap regression network. The network predicts dense wall points as the initial vertices of the graph, which are then triangulated to form the initial graph. B) Graph optimization. Subsequently, the floorplan reconstruction problem is formulated as a problem of finding the optimal subgraph, on the initial graph, we applied subtle pruning techniques and designed a novel algorithm with proposed structural weight for edges, which efficiently solves the subgraph problem.

A. Graph Initialization

1) Wall Point Detection: The first step of the graph initialization stage is to detect the nodes from the input raw image for constructing a graph. We propose a cross-attention sequence-guided heatmap regression network to obtain wall points in the 3D scan. These wall points are uniformly and discretely distributed along the boundaries of the walls. Unlike previous corner-based primitive structures for plane reconstruction (such as HEAT [26] and Floornet [32]), our proposed wall point primitive captures richer structural information. The reason is that corner primitives are sparser compared to wall point primitives, making methods based on corner primitives less tolerant of errors. Inspired by the beneficial effect of guided boundary maps on keypoint heatmap regression [35], we designed a sequence-guided network architecture that utilizes the structural correlation among multiple heatmaps to guide the generation process and improve the accuracy of keypoint heatmap detection. During the sequence-guided process, we utilize hyper-perceptual cross-attention (HPCA) for information feature extraction.

The cross-attention sequence-guided network consists of a backbone network and three layers of heatmap regression modules, as shown in the first-row block of Fig. 2. It takes 2D density/normal maps as input, extracts initial features through

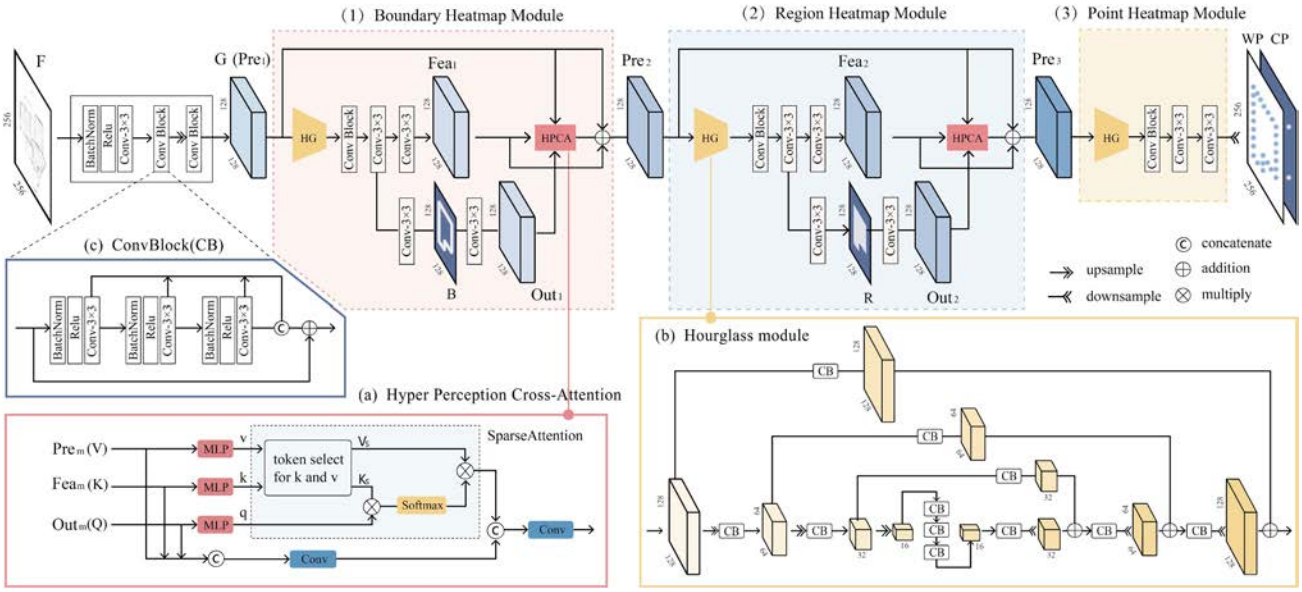

Fig. 3. Overall architecture of heatmap regression network, which consists of three core parts: (1) Boudary heatmap Module, (2) Region heatmap Module, (3) Point heatmap Module. (a) Hyper Perception Cross-Attention (HPCA) module. (b) Hourglass (HG) module. (c) ConvBlock module.

CNN, and then sequentially generates three types of heatmaps: wall (B), room (R), and point (P) masks. Each mask guides the generation of subsequent masks in a sequential manner. Our final point masks include not only general wall points but also crucial points (i.e., WP and CP in Fig. 2). HourGlass (HG) module [35] is utilized to capture feature information at different scales. Due to the diverse dimension of features and masks, the fusion of them should consider the loss of semantic information, therefore, we employ the Hyper-Perceptual Cross-Attention (HPCA) module to fuse their semantic features. HPCA employs a dual-branch neural network similar to HPB [36] for information fusion. We utilize cross-attention in its attention branch to perform weighted computation, enabling us to better capture the correlated features among different structures.

Specifically, as shown in Fig. 3, the input 2D density/normal map $F \in \mathbb{R}^{H\times W\times 3}$ goes through several convolution blocks (i.e., Fig. 3(c)) and convolution layers (e.g., Conv3x3) to obtain preliminary features $G \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times C}$. After undergoing three similar heatmap regression modules (i.e., Fig. 3(1), (2) and (3)), final wall point heatmaps which include rich wall boundary information are generated. In the following description, the variable m is used to identify the module index refers to dash blocks, where we highlight Pre_m, Fea_m, and Out_m since they are important intermediate features with diverse effects in the corresponding the $m-th$ heatmap regression module. For instance, when $m=1$, G serves as the preceding feature Pre_1, Pre_1 is fed into the hourglass (HG) module stack with a single convolution block (CB) as well as several convolution layers to generate two intermediate latent embeddings: Fea_1 and B, where Fea_1 are extracted as implicit representation from the input image, while B represent wall information explicitly. The detailed structure of HG module is shown in Fig. 3(b), with ConvBlocks, upsampling and downsampling steps, it has been proven to be capable to extract information at different scales [36]. Then, B is further processed by a conv3x3 layer to generate Out_1 for HPCA fusion.

The HPCA is utilized for fusing aforementioned three feature maps attentively. As shown in the left bottom block in Fig. 3, the Pre_m, Fea_m, and Out_m are treated as the inputs, i.e., value (V), key (K) and query (Q) of attention layer, then inputs are transferred into q, k and v, whose dimension are $\mathbb{R}^{N\times C}$, $N=\frac{H}{2}\times\frac{W}{2}$. The formula for high-contribution scores of tokens is as follows:

$$Score_r = \sum_{i=0}^{N-1}\sum_{j=0}^{\frac{W}{2}-1} q_i k'^{T}_{r,j,:} \quad r\in\left\{0\ldots\frac{H}{2}-1\right\}, \tag{1}$$

$$Score_c = \sum_{i=0}^{N-1}\sum_{j=0}^{\frac{H}{2}-1} q_i k'^{T}_{j,c,:} \quad c\in\left\{0\ldots\frac{W}{2}-1\right\}. \tag{2}$$

Sparse attention reduces computational costs [36] by retaining highly contributing rows and columns. In (1)–(2), $q_i \in \mathbb{R}^C$, is the i-th row of q, $k' \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times C}$, $k'_{r,j,:}$ is the slice based on the first and second dimension. Then, Ns rows and columns are selected based on the following:

$$Index_r = ArgMaxScore\left(Score_r\right)[:N_s], \tag{3}$$

$$Index_c = ArgMaxScore\left(Score_c\right)[:N_s]. \tag{4}$$

N_s is set to 16 in our experiments. Tokens K_s and V_s are selected from k' (reshaped k) and v' (reshaped v) based on $Index_r$ and $Index_c$, which are used for sparse attention weighted computation in HPCA:

$$SparseAttention(q,k,v) = Softmax\left(qK_s^T\right)V_s \tag{5}$$

Our network possesses more accurate reconstruction capabilities through cross-information fusion by using HPCA. This is validated in the ablation experiments discussed in Section IV-D.

The output of HPCA will be added with Pre_1 and Fea_1 to form the $Pre_2 \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times C}$, which is also the input of the next heatmap regression module (i.e., Fig. 3(2)). In this way, the previously learned features and marks could guide the sequential module to capture the wall message. After same process of generating wall mask B, the room mask matrix R, Fea_2 and Out_2 are also learned. Finally, Pre_3 will go through the final heatmap regression modules (i.e., Fig. 3(3)) for generating WP and CP including wall points and corner points, respectively. Benefiting from the guidance of the previous two modules, the third one could possess simpler architecture, i.e., one HG component stack with a CB and two convolution layers, which guarantee the performance while lightening the neural network complexity.

The loss function of the heatmap regression network consists of three components: the wall heatmap loss L_{wall}, the room heatmap loss L_{region}, and the point heatmap loss L_{points}. In the wall heatmap regression layer, to ensure the coherence of heat values in the wall heatmap, we utilize the Adaptive WingLoss [37] as shown in (6)–(8) for calculating the loss of B. y is the groundtruth of the wall heatmap and $\hat{y}$ is the predicted wall heatmap. The values of ω, ϵ, α, and θ are set to 14, 1, 2.1, and 0.5 respectively.

$$\mathcal{L}_{walls}(y, \hat{y}) = \begin{cases} \omega \ln\left(1 + \left|\frac{y-\hat{y}}{\epsilon}\right|^{\alpha-y}\right) & \text{if } |(y-\hat{y})| < \theta \\ A|y-\hat{y}| - C & \text{otherwise} \end{cases}, \tag{6}$$

$$A = \omega\left(\frac{1}{1+\left(\frac{\theta}{\epsilon}\right)^{(\alpha-y)}}\right)(\alpha-y)\left(\left(\frac{\theta}{\epsilon}\right)^{(\alpha-y-1)}\right)\left(\frac{1}{\epsilon}\right), \tag{7}$$

$$C = \left(\theta A - \omega ln\left(1+\left(\frac{\theta}{\epsilon}\right)^{(\alpha-y)}\right)\right), \tag{8}$$

In the room heatmap regression layer, as the number of positive and negative samples in the room area heatmap R differs only slightly, we use the mean squared error (MSE) loss as shown in (9). y_r is the groundtruth of the region heatmap and $\hat{y}_r$ is the predicted region heatmap.

$$\mathcal{L}_{region}(y_r, \hat{y}_r) = \frac{\sum_i^n \left(y_r^i - \hat{y}_r^i\right)^2}{n}, \tag{9}$$

Furthermore, for the final output point heatmap, where the number of negative samples is much larger than the positive samples, we utilize the L1 loss as shown in (10). y_p is the groundtruth of the points heatmap and $\hat{y}_p$ is the predicted points heatmap.

$$\mathcal{L}_{points}(y_p, \hat{y}_p) = \frac{\sum_i^n \left|y_p^i - \hat{y}_p^i\right|}{n}, \tag{10}$$

The total loss function for the final heatmap regression network is defined as (11):

$$\mathcal{L} = \lambda_w \mathcal{L}_{walls} + \lambda_r \mathcal{L}_{region} + \lambda_p \mathcal{L}_{points}. \tag{11}$$

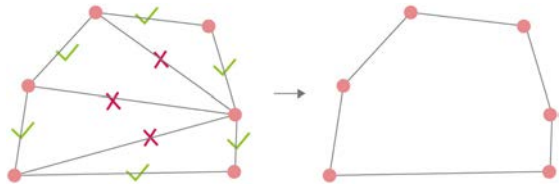

Fig. 4. Visualization of the triangle pruning process. Red cross marks indicate deleted edges.

2) Triangulation: Based on the results of wall point detection, we construct the initial graph structure using Delaunay triangulation [38]. The graph node positions are extracted by maximum pooling, and nodes that are within 6 pixels of each other are merged to reduce redundancy. We utilize the Bowyer-Waston [39] algorithm for triangulation to ensure the connectivity of the graph structure and prevent edge intersections. The initial graph structure formed by dense wall points and corner points effectively maintains the corresponding relationships with the real walls.

B. Graph Optimization

In this step, we propose pruning techniques and structural weight terms. Also, we design an optimization pipeline for subgraph generation in the initial graph of wall points, as shown in the second row of Fig. 2. Thanks to vector representation, pruning techniques, and structural weight terms, the search space of the optimal subgraph-solving problem is reduced effectively. The computational speed of PolyGraph is improved by three orders of magnitude compared to other optimization methods for floorplan structure reconstruction, such as MonteFl [34] and FloorSP [32].

1) Edge Pruning: Our pruning strategy is designed based on the fact that "In most buildings, primary functional spaces are designed to have high occupancy rates and to avoid wasting space, resulting in fewer instances of acute angle spaces being utilized" [41]. Therefore, the longest edge of a triangle generated through triangulation is guaranteed not to lie on a wall. As shown in Fig. 4, by removing the cross-marked edges and retaining the check-marked edges, we can obtain a simplified graph structure. Specifically, as shown in (12)–(14), for the set of triangles T generated from the wall point set V, we remove the longest edge of each triangle to obtain the set of edges E, resulting in the graph G. Also, we remove the vertices with a degree less than 2.

$$T = DelaunayTriangulation(V), \tag{12}$$

$$E = \{e \mid e \in \{e_t\} - \{e_t^{\max}\},\ t \in T\}, \tag{13}$$

$$G = \{E, V\}, \tag{14}$$

where e_t and $e_t^{\max}$ represent the edges and the longest edge of triangle t, respectively.

2) Structral Weight: The structural weight of an edge is the key component in solving the optimal subgraph problem. It consists of two terms: the confidence term and the length term.

Confidence term(E_{conf}): This weight term is designed to reflect the probability that an edge corresponds to a real wall

and is calculated using a combination of the wall (B) and room (R) masks. The higher the value $B(e)$ in the wall mask for an edge, the higher the probability that it corresponds to a wall. Conversely, the lower the value $R(e)$ in the room mask, the higher the probability that the edge corresponds to a wall. The confidence term is calculated using (15)–(17), where v_1 and v_2 are the two endpoints of the edge e, T_c is a threshold:

$$e(u) = u\mathbf{v}_1 + (1-u)\mathbf{v}_2, \tag{15}$$

$$H(e) = \int_0^1 1 - \min\left(1, 1 + R\left(e(u)\right) - B\left(e(u)\right)\right) d_u, u \in [0,1], \tag{16}$$

$$E_{conf}(e) = \frac{H(e)}{|e|} - T_c. \tag{17}$$

Length term(E_{len}): It is used to refine the confidence of short edges in the structural weight. Due to the slight misalignment between wall points and edge heatmap, short edges are more sensitive to this misalignment, resulting in low confidence values and potential abandonment. Therefore, we use E_{len} to refine the weights of short edges in E_{conf}. Specifically, when dealing with edges that have lower confidence, we set a threshold T_d, as shown in (18). For edges with E_{conf} less than zero, if the length is smaller than T_d, E_{len} is set to 1, ensuring that the final edge weight is greater than 0. For edges with higher confidence, no adjustment is made, resulting in $E_{len} = 0$. The reason is that edges with higher confidence values do not need to be refined.

$$E_{len}(e) = \begin{cases} \frac{T_d - |e|}{|T_d - |e||} & \text{if } E_{conf}(e) < 0 \\ 0 & \text{otherwise} \end{cases}, \tag{18}$$

The overall representation of the structural weight is as follows:

$$w_{struct}(e) = \begin{cases} E_{conf}(e) + E_{len}(e) & DT_{e_{v_1}} > 1 \text{ and } DT_{e_{v_2}} > 1 \\ -100 & \text{otherwise} \end{cases}. \tag{19}$$

To ensure the connectivity in the reconstructed graph structure and include all edges in the graph cycle, we assign a weight of -100 to edges with node degrees (DT) less than 2.

3) Optimal Subgraph Solution: We further optimize the graph using optimal subgraph optimization, based on the structural weights we proposed and using a greedy algorithm for the solution, which improves the accuracy and efficiency of structural reconstruction. Based on the definition of structural weights, the objective of this step is to find a subgraph with the maximum summation of weights over it, and our greedy strategy could guarantee maximization in each step. It further ensures the sum of all structural weights in the final subgraph is maximized, indicating a state of local optimality. It is worth noting that we traversed all negative edges in turn, and deleted all of the edges that can be deleted step by step according to the optimization strategy. While for the negative edges that are retained in the final subgraph, any attempt to remove them would inevitably lead to an increase in the sum of the structural weights. Although it cannot be guaranteed that the resulting subgraph is globally optimal, adopting a local optimal strategy can significantly enhance the computational efficiency, while the use of structural weight ensures the reliability of the algorithm. Furthermore, to improve the computational efficiency of the greedy algorithm, our algorithm is carried out on an increasing sequence queue Q that only contains edges with structural weights less than 0. Specifically, as shown in Fig. 5, on the queue Q, we traverse the edges in the queue according to the following steps:

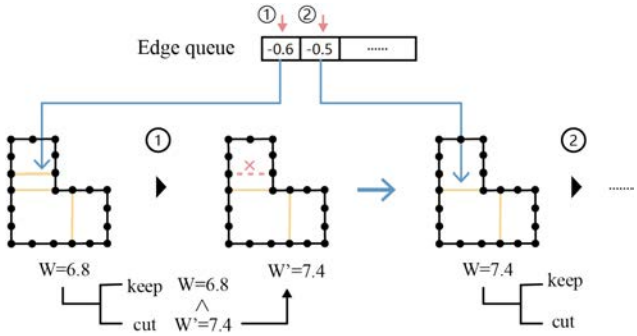

Fig. 5. Visualization of subgraph solving process.

a. Compute the sum of structural weight values, denoted as W, for all edges in the current graph.
b. Compute the sum of structural weight values, denoted as W', for all edges in the current graph after removing the edge with the minimum structural weight of the queue.
c. Compare the values of W and W'. If the structural weight increases ($W' > W$), remove the edge with the minimum structural weight. Otherwise, keep the edge. Pop the first element from the queue.
d. Repeat steps a-c until the queue is empty.

To make the final reconstructed result more compact, we select a set of connected edges composed of nodes with a degree of 2 for Douglas-Peucker line simplification and then delete nodes with a degree of 2 that form an angle greater than 150 degrees to obtain the final result.

IV. Experiment

A. Experiment Settings

1) Sample Processing: Given a set of panoramic RGBD scans, we first convert them into a point cloud. We then project the 3D points onto the XY image plane to generate a 256×256 density image and a 256×256 normal image. In the density image, the density at any point is positively correlated with the number of 3D points at that coordinate. It is linearly scaled to the range [0.0, 1.0]. The pixel values in the normal image represent the average normal vector of the 3D points at that coordinate. The heatmaps for wall points and corner points are generated by applying Gaussian filtering ($\sigma = 2$) to each labeled point. The wall points are uniformly generated at intervals of 10pixels along the labeled edges. The enclosed area formed by the edges has a value of 1, while other areas have a value of 0, resulting in the room heatmap.

2) Datasets: We conducted comparative experiments with other state-of-the-art methods on both the medium-scale dataset Structure3D and the small-scale dataset Lianjia-s.

Structure3D [40]: It contains a total of 3500 scenes (3000/250/250 for training/validation/test) with a diverse set of house floorplans covering both Manhattan and non-Manhattan layouts. The average/maximum numbers of corners and edges across all floorplan graphs are 22.0/52 and 27.6/74, respectively.

Lianjia-s [32]: It contains a total of 100 scenes (70/30 for training/test) released by Lianjia. The average/maximum numbers of corners and edges across all floorplan graphs are 27.8/60 and 34.1/71, respectively.

3) Baselines: We compared our method against five approaches: HEAT [26], Floor-SP [32], MonteFloor [34], SLIBO-Net [43], and RoomFormer [4]. HEAT [26] first detects corners and then uses a neural network to determine the connectivity between them. Both Floor-SP [32] and MonteFloor [34] initially employ Mask-RCNN [33] for room segmentation, followed by non-learning optimization techniques for room vectorization. SLIBO-Net [43] uses a transformer architecture to generate slicing boxes and room centroids, which are then aggregated and refined through post-processing to reconstruct the floorplan. RoomFormer [4] leverages a transformer architecture to simultaneously output multiple corner sequences for floorplan reconstruction.

4) Evaluation Metrics: To evaluate the performance of PolyGraph, quantitative experiments were conducted using precision/recall/F1 metrics for room/corner/angle (same as Roomformer [4]). When calculating the metrics for corner points and angles of a room, T-junction points on room edges resulting from adjacent rooms are not considered. For calculating the accuracy of wall points, wall points within 10 pixels of the GT points and edges are considered successful reconstructions. Additionally, due to the limitations of existing evaluation metrics in fully capturing the quality issues of generated floorplans and rooms (i.g., room overlap and angular deviation), we propose three new metrics (S_{cons}, $MAnE$ and S_{comp}) to evaluate the reconstruction results.

The S_{cons} (structural consistency) metric measures the quality of reconstruction from two aspects: the consistency of adjacent room structures (no overlap with neighboring rooms is expected) and the consistency between points and room structures (existing points belong to the existing rooms is expected). The formula is as follows:

$$S_{cons} = \lambda \times \frac{R_{overlap} + V_{hc} + 1}{TP_R + TP_V + 1}, \tag{20}$$

where $R_{overlap}$ represents the number of rooms that have overlaps with other rooms, V_{hc} denotes the number of hanging points and cut points in the structure, TP_R and TP_V represent the number of correctly reconstructed rooms and corner points respectively, λ is a scaling coefficient. In practice, we use $\lambda = 10$.

We propose MAnE (Mean Angle Error) metric to measure the overall deformation of the structure. Unlike angle recall/precision/F1, which only calculate the quantity of correct angles, MAnE also takes into account the deviation between the reconstructed angles and the ground truth values. It is defined as in (21):

$$MAnE = \frac{\sum_{i=0}^{n} A_i + \theta \times FN_{angle}}{TP_{angle} + FN_{angle}}, \tag{21}$$

where A_i represents the i-th angle deviation among all angles deviations within 5 degrees, FN_{angle} is the number of false negative angles, TP_{angle} is the number of correctly reconstructed angles. For angles that have not been correctly reconstructed, we assign an angle deviation $\theta = 10^{\circ}$.

We use the S_{comp} metric to measure the compactness of reconstructed structures. A higher S_{comp} indicates that fewer points are used to correctly reconstruct more rooms. It is defined as follows:

$$S_{comp} = \frac{TP_R}{N_v}. \tag{22}$$

where TP_R represents the number of correctly reconstructed rooms and N_v is the total number of corner points in the reconstruction results.

5) Implementations: We have implemented our method in Python 3.8 and PyTorch 1.9.0, using a workstation with a 2.6 GHz CPU and NVIDIA RTX 3090 GPU. We trained the model using the Adam optimizer, with an initial learning rate of 1e-4 and a weight decay factor of 1e-5. In the last 20% of the epochs, the learning rate was decayed by a factor of 0.1. We set the coefficients for losses in (11) to $\lambda_w = 1$, $\lambda_r = 1$, $\lambda_p = 1$ and hyperparameters in (17) and (18) as $T_c = 0.6$, $T_d = 12$, which are recommended for optimal performance, based on experimental results. Depending on the size of the dataset, we trained the model for 700 epochs on the Structured3D dataset and 500 epochs on the Lianjia-s dataset. The specific hyperparameter settings are shown in the Table III.

B. Quantitative Evaluation

Tables I and II report the quantitative results of PolyGraph and SOTA methods for plane structure reconstruction on the Structure3D and Lianjia-s datasets. It is worth noting that we focus on proposing an alternative method that combines the stability of optimization-based approaches and the efficiency of end-to-end methods. Specifically, PolyGraph outperforms other methods in terms of structural consistency (S_{cons}), mean angle error ($MAnE$), and structural compactness (S_{comp}), indicating that our reconstruction method achieves better integrity, compactness, and lower deformation. We also achieve the highest computational efficiency among optimization-based methods, with a computational time difference of 0.033s compared to end-to-end SOTA methods. The time cost of the graph initialization and optimization steps is shown in parentheses. Our method outperforms optimization-based methods (Floor-SP and MonteFl) across all metrics. On the Structure3D dataset, our corner recall is lower than the end-to-end method RoomFormer but higher than SLIBO-Net. We observe that RoomFormer performs poorly in structural compactness, and we speculate that its higher corner recall is due to predicting more corner points, which is also reflected in the visualization results in Section IV-C. Our angle recall is lower than SLIBO-Net but higher than other methods. We observe that the Structure3D dataset contains a high frequency of right angles, and SLIBO-Net's approach, which uses box-based sliding and stitching, is highly effective at detecting 90- and 180-degree angles. In addition, we noticed that PolyGraph has the highest room recall rate

TABLE I
QUANTITATIVE EVALUATION OF METRICS ON STRUCTURE3D DATASET

Method	Time	$S_{cons}\downarrow$	$MAnE\downarrow$	$S_{comp}\uparrow$	Room			Angle			Corner		
	t(s)				Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1
Floor-SP	785	0.625	4.35°	0.207	88.0	89.0	88.5	72.0	80.0	75.8	73.0	81.0	76.8
MonteFl	71	-	-	-	94.4	95.6	95.0	75.4	86.3	80.5	77.2	88.5	82.5
HEAT	0.11	0.234	2.75°	0.273	94.3	96.6	95.4	79.1	88.9	83.7	81.8	91.8	86.5
RoomFormer	**0.01**	0.034	2.74°	0.178	96.3	97.8	97.0	77.9	82.1	79.9	**84.2**	88.8	86.4
SLIBO-Net	0.17	-	-	-	97.8	**99.1**	**98.4**	**81.2**	87.8	84.4	82.1	88.9	85.4
Ours	0.043 (0.004+0.039)	**0.019**	$\mathbf{2.422^\circ}$	**0.285**	**97.9**	95.7	96.7	79.4	**89.2**	**85.4**	82.2	**92.4**	**88.3**

Time: time required to reconstruct a floorplan. The results of MonteFl and SLIBO-Net are from their respective papers. Bold numbers indicate the top results. Underlined numbers indicate the second-best results.

TABLE II
QUANTITATIVE EVALUATION OF METRICS ON LIANJIA-S DATASET

Method	Time	$S_{cons}\downarrow$	$MAnE\downarrow$	$S_{comp}\uparrow$	Room			Angle			Corner		
	t(s)				Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1
Floor-SP	785	0.921	8.31°	0.114	39.7	36.3	37.9	18.5	18.9	18.7	46.1	49.5	47.7
HEAT	0.11	2.334	7.53°	0.156	56.1	61.5	58.7	30.1	43.4	35.5	84.2	**88.1**	**86.1**
Ours	**0.043 (0.006+0.037)**	**0.012**	$\mathbf{5.76^\circ}$	**0.209**	**82.2**	**74.9**	**78.4**	**49.2**	**45.4**	**47.2**	**84.8**	80.7	82.7

Bold numbers indicate the top results. Underlined numbers indicate the second-best results.

TABLE III
THE IMPLEMENTATION DETAILS OF OUR POLYGRAPH

Hyperparameter	Value
N of Res-AI Modules	3
G of attention groups	8
r of information bottleneck	16
hidden dimension d	64
kernel size	3
batch size	8
optimizer	Adam
learning rate	10^{-4}
learning scheduler	1000
weight decay	10^{-5}
momentum	0.99
T_c	0.6
T_d	12
S3D_epoch	700
Lianjia-s_epoch	500
The Threshold of the Douglas-Peucker Algorithmpx	3
λ_w	1
λ_r	1
λ_r	1

and angel precision, but a lower room precision. This will be discussed in Section IV-E. Notably, because of the small scale and complexity of the Lianjia-s dataset, all methods experience a significant performance decline across all metrics, while our method demonstrates relatively better performance under these challenging conditions. Moreover, due to the Transformer-based architecture, RoomFormer performs poorly on the small-scale Lianjia-s dataset, and therefore, we have not included it in the comparison for this dataset. Given the constraints on paper length, additional comparative experiments on other dataset (e.g., SceneCAD) have been made publicly available on GitHub.

C. Qualitative Evaluations

Fig. 6 presents the visual results of our method compared to SOTA methods on the Structure3D test dataset. We demonstrate superior structural details, completeness, and compactness compared to other methods. Wall point information in PolyGraph enables more accurate reconstruction of wall details, as demonstrated in the highlighted examples within the black rectangles. The graph optimization module we designed improves the integrity and compactness of the reconstructed structure. Compared to RoomFormer, our reconstructed structure is more concise, while RoomFormer exhibits noticeable redundant edges and corner points, as shown with the balck dashed rectangles in the fourth row of Fig. 6. HEAT exhibits incomplete structural representations, as shown with the balck dashed rectangles in the second row.

Fig. 7 shows the visualization results on the Lianjia-s test dataset. PolyGraph maintains good structural integrity on this small-scale dataset, with no obvious room missing issues, as shown with the balck dashed rectangles in the second and fourth columns. From the results in the first column, as shown with the balck rectangles, PolyGraph and HEAT outperform FloorSP in capturing slanted wall structures. This is because FloorSP employs discrete angle constraints to reduce the search space. Similar observations can be made on the S3D dataset, as shown in the fifth column of Fig. 6. We observe that all methods experience a decrease in accuracy on Lianjia-s dataset, indicating the challenges of reconstructing small-scale datasets.

The qualitative results demonstrate that PolyGraph exhibits more complete and compact structural representations compared to the state-of-the-art methods, reducing the gap between automated and manual approaches in floorplan reconstruction. To visually evaluate the reconstruction effect, we assigned a certain height to the walls in the reconstructed plan for 3D visualization, as shown in Fig. 8. Compared to the initial RGBD scans, our reconstruction result accurately reproduces the overall building space.

D. Ablation Studies

In order to examine the performance of different modules in PolyGraph, we conducted experiments with six combinations

Fig. 6. Qualitative comparison with SOTA methods on Structure3D dataset.

TABLE IV
ABLATION STUDY FOR THE TECHNICAL COMPONENTS OF POLYGRAPH

setting name	SubGraph	$Point_{wall}$	$Mask_b$	$Mask_r$	HPCA	Room			Angle			Corner		
						Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1
setting 1	/	✓	/	/	/	54.8	18.3	24.4	33.2	15.3	20.9	35.3	16.3	22.3
setting 2	✓	/	/	/	/	57.5	77.3	65.9	39.0	58.0	46.6	43.0	64.0	51.4
setting 3	✓	✓	/	/	/	89.0	87.9	88.4	68.6	73.5	71.1	73.1	78.3	75.6
setting 4	✓	✓	✓	/	/	94.6	91.1	92.8	76.8	82.6	79.6	80.0	86.0	82.9
setting 5	✓	✓	✓	✓	/	94.9	91.5	93.2	77.8	83.3	80.5	80.8	86.6	83.6
setting 6	✓	✓	✓	✓	✓	**97.9**	**95.7**	**96.7**	**79.4**	**89.2**	**85.4**	**82.2**	**92.4**	**88.3**

For each row of settings, '✓' means the column component is set, and '/' means not. Bold numbers indicate the top results.

of modules and evaluated the results using visualization results and recall/precision/F1 metrics for rooms/angles/corner points on Structure3D datasets. Columns 2-6 of Table IV represent the following configurations:

- *SubGraph:* The subgraph solution module in graph optimization. Without this module, structure-weight-based graph optimization is not performed.
- *Point*$_{\mathbf{wall}}$*:* The primitives for constructing the graph structure. Without this item, only crucial points are used as nodes.
- *Mask*$_{\mathbf{r}}$*:* One of the guided heatmaps in the graph initialization network. Without this component, the room mask heatmap is not generated, and the corresponding weight value $R(e)$ in the E_{conf} weight term is set to 0.
- *Mask*$_{\mathbf{b}}$*:* One of the guided heatmaps in the graph initialization network. Without this component, the edge mask heatmap is not generated, and the corresponding weight value $B(e)$ in the E_{conf} weight term is set to 1.
- *HPCA:* It is the feature fusion module in the heatmap generation network. Without this component, the multiple preceding inputs are directly added together.

Fig. 9 and Table IV present the comparison results of different settings. Compared to Setting 1 and Setting 2, Setting 3 shows significant improvement in recall, precision and F1,

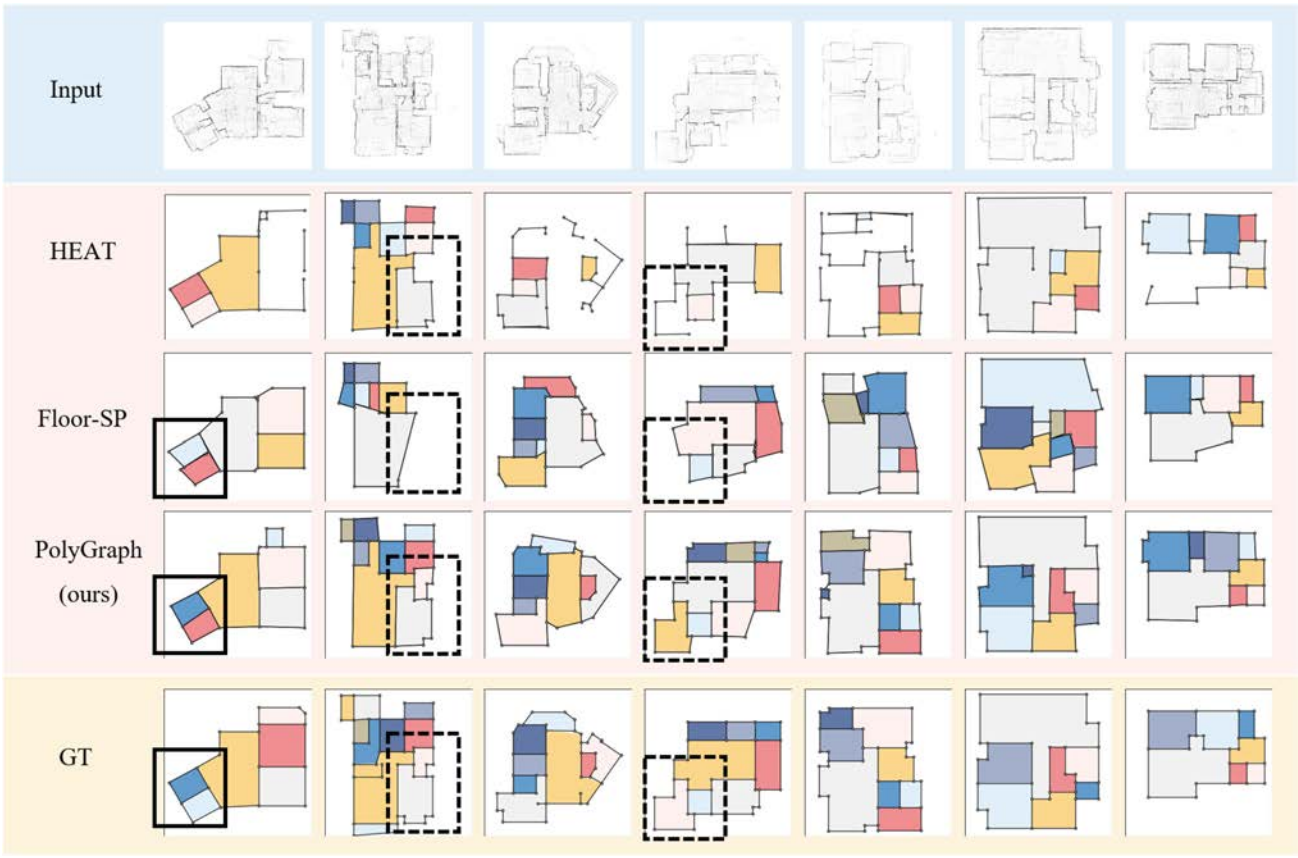

Fig. 7. Qualitative comparison with SOTA methods on Lianjia-s dataset.

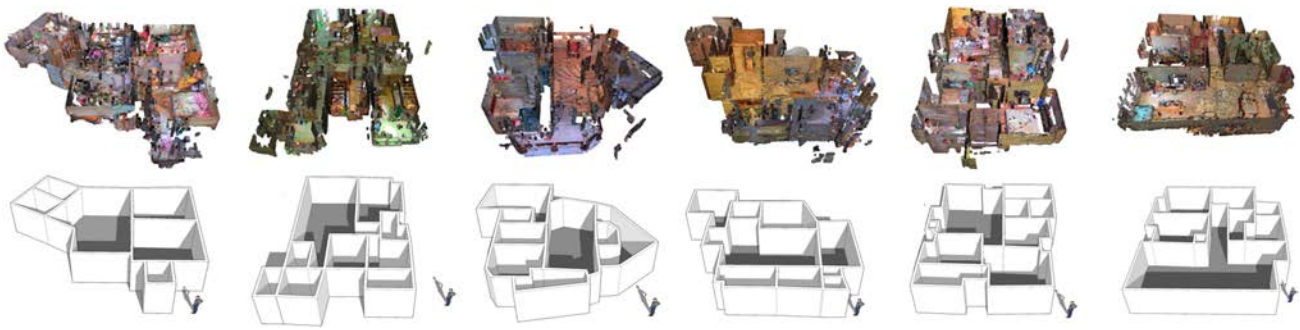

Fig. 8. 3D visualization of the reconstructed floorplan.

demonstrating the importance of subgraph optimization based on structural weight terms and the inclusion of wall point primitives. From rows 2-4 in Fig. 9, we observe that SubGraph effectively reduces redundant edges in the structure, and the use of WallPoints helps improve the completeness of rooms as shown with the black rectangels in rows 3-4. Settings 3, 4, and 5 exhibit steady improvement in all metrics, indicating the effectiveness of $Mask_r$ and $Mask_b$. Rows 4-6in Fig. 9 also show a gradual refinement of structural details, as marked by the black dashed rectangles. Setting 6 demonstrates a slight improvement of 1-2 percentage points in the metrics. Comparing Setting 5 and Setting 6in the figure, such as examples marked by black rectangles,it can be seen that adding the HPCA module leads to more accurate structures, validating the effect of using cross-attention fusion.

TABLE V
ABLATION STUDY FOR EDGE WEIGHTS ITEMS IN GRAPH OPTIMIZATION

Edge Weight	Room			Angle			Corner		
	Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1
w_{mask}	88.8	90.9	89.8	76.0	82.5	79.1	72.8	79.0	75.8
w/o E_{conf} & E_{len}	54.8	18.3	27.4	33.2	15.3	20.9	35.3	16.3	22.3
w/o E_{len}	97.0	92.9	94.9	**79.7**	84.6	82.1	**82.7**	87.8	85.2
w/o E_{conf}	95.1	94.0	94.5	78.0	84.4	81.8	81.1	87.8	84.3
w_{struct}	**97.9**	**95.7**	**96.7**	79.4	**89.2**	**85.4**	82.2	**92.4**	**88.3**

"Mask weight" is the common method for edge assignment. The second to fourth lines are the comparison of different combinations of the weight items in PolyGraph. Bold numbers indicate the top results.

Fig. 9. Ablation study on Structure3D dataset which demonstrats the role of each component of PolyGraph. Each setting evaluates the isolated effect of one or more terms. Please refer to the text for details on each setting.

As shown in Table V, we examined the effectiveness of the structural weight (as proposed in Section III-B2) by comparing it with different weights. Compared to the first row using the edge weight in [42] which we denoted as w_{mask}, PolyGraph achieved a 3-8 percentage point improvement in all metrics when replacing it with the structural weight w_{struct} in the fifth row. The weight w_{mask} used in the experiments is defined as follows:

$$w_{mask}(e) = \int_0^1 1/2 \times (1 - R(e(u)) + B(e(u)))\, d_u - T_c. \tag{23}$$

As shown in rows 2 to 5 of Table V, we conducted ablation experiments on each component of the structural weight term. Compared to row 4 where E_{conf} is not used, all metrics show improvements in row 5. On average, there is a 1.8 percentage point increase in recall and a 3.7 percentage point increase in precision, indicating the effectiveness of E_{conf}. In comparison to row 3 where E_{len} is not used, recall shows almost no increase (an average increase of 0.03) in row 5. This could be attributed to the removal of a small number of correctly reconstructed points that belong to the low-confidence edges when considering the length term. However, there is an improvement in precision by an average of 4 percentage points, indicating the contribution of E_{len} to the accuracy of detecting structural elements. Taking all factors into consideration, using the proposed calculation of the structural weight term yields better results.

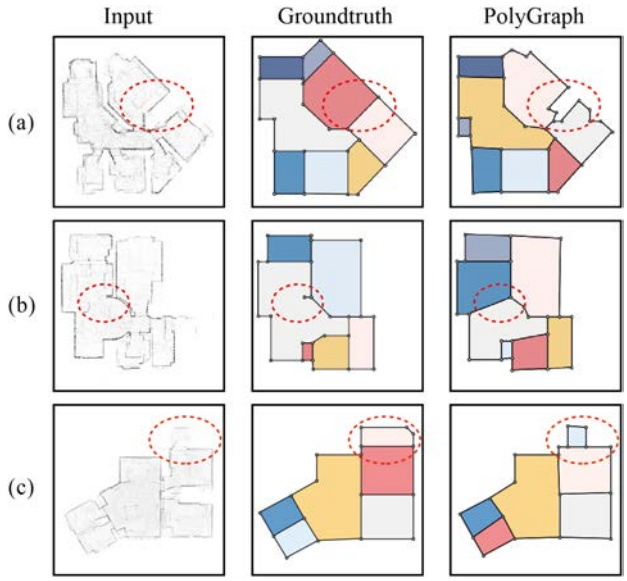

Fig. 10. Failure cases.

E. Discussions

We observed that furniture with a considerable height in the input can be mistakenly scanned as walls, resulting in wide gaps in the outer walls on the input image. This leads to the reconstruction of blank external areas instead of wall structures, as indicated by the red dashed line in Fig. 10(a). Additionally, PolyGraph cannot handle cases where the ground truth includes hanging points, as indicated by the red dashed line in Fig. 10(b), which can result in false positive edges and a decrease in room precision. This is because we model the overall indoor reconstruction as a subgraph optimization problem, which prioritizes maintaining the integrity of rooms. It is worth noting that not all redundant wall points will affect the final result, as our graph optimization process does not generate false positive edges when they are located outside the plane of the floorplan. This is different from existing top-down methods where false positives or false negatives in primitives will inevitably have an impact on the reconstruction result. Furthermore, for the room with strongly missing 3D data (e.g., loss majority of boundaries, Fig. 10(c)), we are unable to reconstruct those missing structures. Addressing this limitation could be an important direction for future work.

V. Conclusion

In this paper, we have presented PolyGraph, a novel reconstruction pipeline to obtain high-quality results, which combines a deep learning-based primitive detection network with an efficient optimization-based reconstruction algorithm. PolyGraph perform wall point primitive detection to preserve more structural details and strengthen the error tolerance of the primitive detection stage. We formulate the problem of holistic indoor floorplan reconstruction as a subgraph optimization problem by utilizing wall points to create a graph-based representation. Our method stands out among optimization-based approaches in terms of computational efficiency and performs exceptionally well on the newly proposed overall structure metrics. Experimental results validate the superior performance of PolyGraph in preserving structural integrity and capturing fine details. Extensive ablation studies further justify our design choices. The PolyGraph pipeline is also universal. For other reconstruction problems, only the corresponding weight items need to be set. Our ideas play a certain role in promoting the frontier field of structure reconstruction.

References

[1] Y. He, Y.-T. Liu, Y.-H. Jin, S.-H. Zhang, Y.-K. Lai, and S.-M. Hu, "Context-consistent generation of indoor virtual environments based on geometry constraints," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 12, pp. 3986–3999, Dec. 2022.

[2] A. C. Haley, D. Thorpe, A. Pelletier, S. Yarosh, and D. F. Keefe, "Inward VR: Toward a qualitative method for investigating interoceptive awareness in VR," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 5, pp. 2557–2566, May 2023.

[3] W. Sui, L. Wang, B. Fan, H. Xiao, H. Wu, and C. Pan, "Layer-wise floorplan extraction for automatic urban building reconstruction," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 3, pp. 1261–1277, Mar. 2016.

[4] Y. Yue, T. Kontogianni, K. Schindler, and F. Engelmann, "Connecting the dots: Floorplan reconstruction using two-level queries," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 845–854.

[5] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1653–1660.

[6] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2017, pp. 3097–3106.

[7] Y. Xu, W. Xu, D. Cheung, and Z. Tu, "Line segment detection using transformers without edges," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4255–4264.

[8] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1422–1429.

[9] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.

[10] T. Nguyen, G. Reitmayr, and D. Schmalstieg, "Structural modeling from depth images," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 11, pp. 1230–1240, Nov. 2015.

[11] C. Zou, A. Colburn, Q. Shan, and D. Hoiem, "LayoutNet: Reconstructing the 3D room layout from a single RGB image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2051–2059.

[12] S. T. Yang, F. E. Wang, C. H. Peng, P. Wonka, M. Sun, and H. K. Chu, "DuLa-Net: A dual-projection network for estimating room layouts from a single RGB panorama," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3358–3367.

[13] L. Gimenez, J. Hippolyte, S. Robert, F. Suard, and K. Zreik, "Reconstruction of 3D building information models from 2D scanned plans," *J. Building Eng.*, vol. 2, pp. 24–35, 2015.

[14] B. Okorn, X. Xiong, B. Akinci, and D. Huber, "Toward automated modeling of floor plans," in *Proc. Symp. 3D Data Process. Vis. Transmiss.*, 2010, pp. 1–8.

[15] A. Adan and D. Huber, "3D reconstruction of interior wall surfaces under occlusion and clutter," in *Proc. Int. Conf. 3D Imag. Model. Process. Vis. Transmiss.*, 2011, pp. 275–281.

[16] J. Lladós, J. López-Krahe, and E. Martí, "A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform," *Mach. Vis. Appl.*, vol. 10, pp. 150–158, 1997.

[17] X. Qin, S. He, X. Yang, M. Dehghan, Q. Qin, and J. Martin, "Accurate outline extraction of individual building from very high-resolution optical images," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 11, pp. 1775–1779, Nov. 2018.

[18] A. Budroni and J. Boehm, "Automated 3D reconstruction of interiors from point clouds," *Int. J. Architect. Comput.*, vol. 8, pp. 55–73, 2010.

[19] R. Cabral and Y. Furukawa, "Piecewise planar and compact floorplan reconstruction from images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 628–635.

[20] E. Delage, H. Lee, and A. Y. Ng, "A dynamic Bayesian network model for autonomous 3D reconstruction from a single indoor image," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2418–2428.

[21] Y. Zhou, H. Qi, and Y. Ma, "End-to-end wireframe parsing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 962–971.
[22] Z. Zhang et al., "PPGNET: Learning point-pair graph for line segment detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7105–7114.
[23] N. Xue et al., "Holistically-attracted wireframe parsing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2785–2794.
[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.
[25] C. Liu, J. Wu, and Y. Furukawa, "FloorNet: A unified framework for floorplan reconstruction from 3D scans," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 201–207.
[26] J. Chen, Y. Qian, and Y. Furukawa, "HEAT: Holistic edge attention transformer for structured reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3856–3865.
[27] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.
[28] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 478–487.
[29] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, 2004.
[30] S. T. Wu and M. R. G. Marquez, "A non-self-intersection Douglas-Peucker algorithm," in *Proc. IEEE Symp. Comput. Graph. Image Process.*, 2003, pp. 60–66.
[31] C. Dyken, M. Dæhlen, and T. Sevaldrud, "Simultaneous curve simplification," *J. Geographical Syst.*, vol. 11, pp. 273–289, 2009.
[32] J. Chen, C. Liu, J. Wu, and Y. Furukawa, "Floor-SP: Inverse CAD for floorplans by sequential room-wise shortest path," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2661–2670.
[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[34] S. Stekovic, M. Rad, F. Fraundorfer, and V. Lepetit, "MonteFloor: Extending MCTS for reconstructing accurate large-scale floor plans," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16014–16023.
[35] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: A boundary-aware face alignment algorithm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2129–2138.
[36] W. Zheng, Q. Li, G. Zhang, P. Wan, and Z. Wang, "ITTR: Unpaired image-to-image translation with transformers," 2022, *arXiv:2203.16015*.
[37] X. Wang, L. Bo, and L. Fuxin, "Adaptive wing loss for robust face alignment via heatmap regression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6970–6980.
[38] O. R. Musin, "Properties of the Delaunay triangulation," in *Proc. 13th Annu. Symp. Comput. Geometry*, 1997, pp. 424–426.
[39] D. F. Watson, "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes," *Comput. J.*, vol. 24, no. 2, pp. 167–172, 1981.
[40] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou, "Structured3D: A large photo-realistic dataset for structured 3D modeling," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 519–535.
[41] Y. Mei and Y. Yang, "Application and practice of acute angle in architectural space," *Urban Archit.*, vol. 29, pp. 209–211, 2016.
[42] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1715–1724.
[43] J. W. Su, K. Y. Tung, C. H. Peng, P. Wonka, and H. K. Chu, "SLIBO-net: Floorplan reconstruction via slicing box representation with local geometry regularization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2023, pp. 48781–48792.

Qian Sun (Member, IEEE) received the BS degree from Tianjin University, China, and the PhD degree in computer science from Nanyang Technological University, Singapore. She was a research fellow with Fraunhofer Singapore and an associate professor with Tianjin University. She is now a professor with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology. Her research interests include intelligent graphics and image processing, and intelligent remote sensing.

Chenrong Fang received the bachelor's degree from the South China University of Technology, in 2021. She is currently working toward the master's degree with the College of Intelligence and Computing, Tianjin University. Her research interests include computer graphics and computer vision.

Shuang Liu received the bachelor's degree from the Renmin University of China, in 2010, and the PhD degree from the National University of Singapore, in 2015. She worked as a research fellow with SUTD, lecturer with SiT, and associate professor with Tianjin University, China. She is now with Renmin University of China. Her research interests include AI for SE and SE for AI.

Yidan Sun received the BEng degree from the School of Computer Science and Technology, Tianjin University, China, and the doctoral degree from the School of Computer Science and Engineering, Nanyang Technological University, Singapore. Currently, she works as a research fellow with the School of Computer Science and Engineering, NTU. Her research interests include AI security, energy efficient MPSoC design, Big Data analytics in transport, and urban computing.

Yu Shang received the bachelor's degree from Tianjin University, in 2019, and the master's degree from the College of Intelligence and Computing, Tianjin University, in 2022. His research interests include computer vision and image processing.

Ying He is an associate professor with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests are primarily in geometric computing and analysis. He actively participates in the technical program committees of major conferences in geometric modeling and is serving/has served on the editorial boards of *IEEE Transactions on Visualization and Computer Graphics*, *Computer Graphics Forum*, and *Computational Visual Media*. He has also served as general/program co-chair for the Shape Modeling International conference in 2022, the Symposium on Solid and Physical Modeling in 2022 and 2023, the Geometric Modeling and Processing conference in 2014 and 2021, and the Conference on Computational Visual Media in 2020.