HEAT: Holistic Edge Attention Transformer for Structured Reconstruction

Jiacheng Chen[1]   Yiming Qian[2]   Yasutaka Furukawa[1]

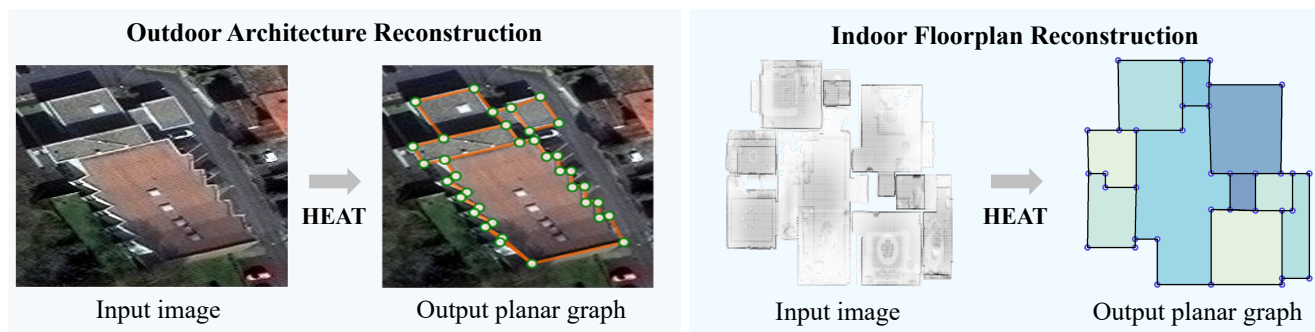[1]Simon Fraser University   [2]University of Manitoba

Figure 1. HEAT takes a 2D raster image as an input and reconstructs a planar graph by an end-to-end transformer based neural architecture, for example, from a satellite image to an outdoor building architecture, or from a point-density image to an indoor floorplan.

Abstract

This paper presents a novel attention-based neural network for structured reconstruction, which takes a 2D raster image as an input and reconstructs a planar graph depicting an underlying geometric structure. The approach detects corners and classifies edge candidates between corners in an end-to-end manner. Our contribution is a holistic edge classification architecture, which 1) initializes the feature of an edge candidate by a trigonometric positional encoding of its end-points; 2) fuses image feature to each edge candidate by deformable attention; 3) employs two weight-sharing Transformer decoders to learn holistic structural patterns over the graph edge candidates; and 4) is trained with a masked learning strategy. The corner detector is a variant of the edge classification architecture, adapted to operate on pixels as corner candidates. We conduct experiments on two structured reconstruction tasks: outdoor building architecture and indoor floorplan planar graph reconstruction. Extensive qualitative and quantitative evaluations demonstrate the superiority of our approach over the state of the art. Code and pre-trained models are available at https://heat-structured-reconstruction.github.io/

1. Introduction

Human vision has a remarkable capability in holistic structure reasoning. Looking at the building in Figure 1, we can effortlessly identify the structural primitives (*e.g.*, building corners and edges) and their relationships. A fundamental challenge in computer vision research is to acquire such human-level perceptual capability and ultimately reconstruct holistic geometric structures from an image, which would have tremendous impacts on broader domains such as visual effects, construction, manufacturing, and robotics.

Ever since the emergence of deep neural networks (DNNs), geometry reconstruction has seen a breakthrough in low-level primitive detection tasks (*e.g.*, corners and edges) by training DNNs on large annotated datasets [24, 25, 29, 31]. However, holistic structure reasoning (*e.g.*, graph inference from corner candidates) is still a challenge for end-to-end neural architectures. The performance is far below that of human-vision [26] and inferior to classical optimization or search methods [4, 17, 22].

This paper seeks to push the frontier of end-to-end neural architecture for structured reconstruction, in particular, we focus on inferring an outdoor building architecture or an indoor floorplan as a 2D planar graph from raster sensor data. Our approach resembles state-of-the-art Transformer based architectures DETR/LETR for object/edge detection [3, 24]. They extract image information by a ConvNet and pass to the "dummy query nodes" in a Transformer decoder by cross-attention. Our experiments reveal that they do not effectively learn structural regularities of target objects/edges.

Our idea is similar and simple, yet yields much more powerful holistic structural reasoning. For edge detection, instead of creating "dummy query nodes" as placeholders for regressing answers, we 1) create a node for every edge candidate and initialize the feature by a trigonometric positional

encoding of its end-points [23]; 2) fuse multi-scale image features from ConvNet backbone to each edge candidate by an adapted deformable attention mechanism [32] and filter out edge candidates; 3) learn structural patterns of edges by two weight-sharing Transformer decoders, one of which only sees the positional encoding (w/o image features) to enhance geometry learning; and 4) employ a masked learning strategy [6] for end-to-end training and iterative inference. A simple adaptation of the edge classification architecture is our corner detector, which further improves performance.

We have evaluated the proposed approach on two structured reconstruction benchmarks: outdoor building reconstruction from a satellite image and indoor floorplan reconstruction from a point-cloud density image. The qualitative and quantitative evaluations demonstrate that our approach outperforms all the competing methods for the outdoor reconstruction task. For the indoor reconstruction task, our approach outperforms all fully-neural methods and is on par with domain specific solutions [4, 22], which are more than 1000 times slower with the use of optimization and search.

2. Related Work

We review structured reconstruction algorithms in three groups: classical techniques, hybrid approaches incorporating deep learning, and end-to-end neural systems.

2.1. Classical techniques

Structured geometry reconstruction has been an active area of research in computer vision, turning raster sensor data into vectorized geometries including wireframes, planes [8,21], room layouts [5,12], floorplans [2], and polygonal loops [9]. Traditional methods rely on low-level image processing techniques such as Hough transform [1, 16] or superpixel segmentation [19]. More sophisticated solvers are also proposed, for example, graphical model inference based on graph-cuts for planar reconstruction [8, 21], dynamic programming for floorplan recovery [2], and Bayesian network for room layout estimation [5]. These methods involve a plethora of heuristics or parameters to tweak by hand.

2.2. Hybrid approaches

With the development of neural networks, deep learning has become a *de facto* machinery for vector-geometry reconstruction. Many state-of-the-art systems adopt a two-stage pipeline, where neural networks first detect low-level primitives (*e.g*., corners, edges, region segments) then optimization techniques assemble them into the final models [15, 18, 33]. Nauata *et al*. [17] and FloorSP [4] rely on Mask R-CNN [10] for primitive detection and apply optimization techniques (*e.g*., integer programming) to reconstruct outdoor building and indoor floorplan as planar graphs, respectively. MonteFloor [22] takes similar detection framework while resorting to Monte Carlo Tree Search

for reconstructing the graph structure. Albeit effective, optimization/search requires domain-specific algorithm design by hand and is a few orders of magnitude slower at test time. A recent approach by Zhang *et al*. iterates exploration and classification steps to search for a better solution [27], while our approach is much faster and performs better.

2.3. End-to-end neural systems

End-to-end neural architectures require less hand-engineering and achieve fast inference. For the wireframe parsing task [13], L-CNN [30] adapts a likelihood prediction convolutional network (ConvNet) for junction detection, followed by an edge verification network to classify each line candidate. PPGNet [28] and HAWP [25] also use the two-stage framework as L-CNN while proposing more advanced model designs. Zhou *et al*. [31] extends the wireframe task to 3D by estimating depths and vanishing points together with the geometry primitives. These techniques handle edge candidates independently, while our approach jointly infers an overall structure by learning holistic structural patterns. ConvMPN is a special graph neural network designed for planar graph reconstruction [26], while our attention-based architecture yields much better results.

The recent success of Transformer-based object detector DETR [3] has also been extended to wireframe parsing by LETR [24]. DETR/LETR utilizes "dummy nodes" as placeholders for storing detection answers and refrain from heuristic-based steps like non-maximum suppression. Our approach designs decoders and training strategies over all edge candidates instead of dummy nodes, and demonstrates more effective holistic structural reasoning.

3. Preliminaries

The paper borrows two structured architecture reconstruction tasks in the literature to demonstrate our approach, which are to infer a planar graph (*i.e*., corners and edges) depicting an architectural structure given a 2D raster image.

Outdoor architecture reconstruction is borrowed from the work by Nauata *et al*. [17]. A cropped satellite image is the input, containing one building from either Paris, Las Vegas, or Atlanta. An output planar graph depicts a roof structure, where each flat roof component is bounded by edges (See Fig. 1). The images come from the SpaceNet Challenge [7] and are distributed under the CC BY-SA 4.0 license. The dataset contains 1601, 50, and 350 samples for training, validation, and test, respectively. The precision/recall/F-1 scores of the corner/edge/region primitives are the metrics as in their work [17]. This task is more challenging than the following floorplan task, because satellite images suffer from perspective distortions and the Manhattan assumption does not hold. Furthermore, a roof component cannot be reliably extracted by an instance segmentation technique. In the floor-
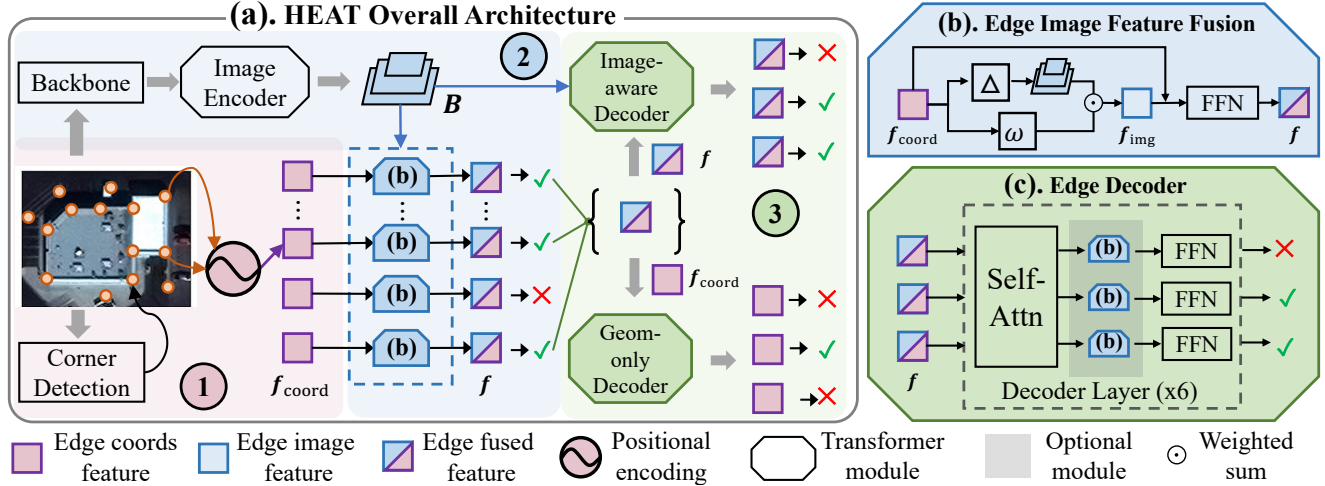
Figure 2. **(a)**. The overall architecture of HEAT, which consists of three steps: 1) edge node initialization; 2) edge image feature fusion and edge filtering; and 3) holistic structural reasoning with two weight-sharing Transformer decoders. **(b)**. The image feature fusion module for edge nodes. **(c)**. The edge Transformer decoder. For the geometry-only (geom-only) decoder, f is replaced by f_{coord} and the image feature fusion module (gray part) is discarded. The add-norm layers in (b) and (c) are omitted for simplicity.

plan task, rooms can be detected easily which are utilized in state-of-the-art floorplan reconstruction techniques [4, 22]. An input image crop is of resolution 256×256 or 512×512. The average/maximum numbers of corners and edges in the graphs are 12.6/93 and 14.2/101, respectively.

Floorplan reconstruction recovers the floorplan of an indoor scene as a planar graph from a point-cloud density image in the top view [4, 22]. In particular, we take the Structured3D dataset [29], convert the registered multi-view RGBD panorama images of each scene to a point cloud, project the 3D points to an XY image plane, and generate a 256×256 density image where each pixel is the number of projected points after normalization (See Fig. 1). The benchmark 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. Following the recent work MonteFloor [22], we use the same precision/recall of room/corner/angle as the evaluation metrics. Note that the metrics require a set of closed polygons for evaluation. Therefore, edges that are not part of a closed polygon are discarded before computing the metric.

4. Holistic Edge Attention Transformer

Holistic edge attention transformer (HEAT) consists of three steps: 1) edge node initialization; 2) image feature fusion and edge filtering; and 3) holistic edge self-attention for structural reasoning (See Fig. 2). The section explains the three steps, followed by the description of 4) the training/inference schemes, and 5) the corner detector, which is

an adaptation of HEAT. Input images are either 256×256 or 512×512, where we explain the architecture parameters in the former case. In the latter case, the spatial resolutions of the ConvNet features simply become double, while the rest of the architecture remains the same. We refer the full network specification to the supplementary.

4.1. Edge node initialization

We start with detecting a set of corner candidates from the input image by a corner detector (See Sect. 4.5). Each pair of corners is an edge candidate and becomes a Transformer node, whose feature f_{coord} is initialized by the 256-dimensional trigonometric positional encoding [23]:

$$
\begin{aligned}
\boldsymbol{f}_{\text{coord}} &= \boldsymbol{M}_{\text{coord}} \left[\boldsymbol{\gamma}(e_1^x),\ \boldsymbol{\gamma}(e_1^y),\ \boldsymbol{\gamma}(e_2^x),\ \boldsymbol{\gamma}(e_2^y) \right], \\
\boldsymbol{\gamma}(t) &= \left[\sin(w_0 t), \cos(w_0 t), \cdots \sin(w_{31} t), \cos(w_{31} t) \right], \\
w_i &= (1/10000)^{2i/32} \quad (i = 0, 1, \cdots 31).
\end{aligned}
$$

e_1 and e_2 are the two corners. e_1^x (resp. e_1^y) denotes the x (resp. y) coordinate of e_1. $\boldsymbol{M}_{\text{coord}}$ is a 256×256 learnable matrix for linear mapping. The function γ encodes ordinal priors including relative distance between coordinates.

4.2. Image feature fusion and edge filtering

We inject image features into each edge node by adapting a deformable attention technique, originally developed for object detection in deformable-DETR [32]. The technique enables adaptive attention over the image features, while existing edge-feature initialization techniques sample pixels uniformly along an edge [25, 30].

We use a ResNet [11] backbone and a Transformer encoder borrowed from deformable-DETR [32] to build a 3-

level image feature pyramid, whose shapes are 64x64x256, 32x32x256, and 16x16x256, respectively. For each level $l(= 1, \cdots 3)$ of the feature pyramid, we use $\boldsymbol{f}_{\text{coord}}$ to generate sampling locations around an edge, as well as their attention-weights for aggregation:

$$\boldsymbol{\Delta}^l = \boldsymbol{M}_{loc}^l \boldsymbol{f}_{\text{coord}}, \tag{1}$$

$$\boldsymbol{w}^l = \text{softmax}\left(\boldsymbol{M}_{agg}^l \boldsymbol{f}_{\text{coord}}\right). \tag{2}$$

$\boldsymbol{M}_{loc}^l$ and $\boldsymbol{M}_{agg}^l$ are learnable weights for the feature level l. $\boldsymbol{\Delta}^l$ contains four 2D sampling offsets with respect to the edge center for the feature level l, and $\boldsymbol{w}^l$ is the corresponding attention weights after the softmax over all the levels and samples. The image feature $\boldsymbol{f}_{\text{img}}$ at each edge is given by

$$\boldsymbol{f}_{\text{img}} = \sum_{l=1}^{3} \sum_{i=1}^{4} \boldsymbol{w}^l(i) \left[\boldsymbol{M}_{\text{img}}^l \, \boldsymbol{B}^l \left(\frac{e_1 + e_2}{2^{l+2}} + \frac{\boldsymbol{\Delta}^l(i)}{2^{l+1}} \right) \right] \tag{3}$$

$\boldsymbol{f}_{\text{img}}$ is a 256-dimensional vector. $\boldsymbol{M}_{\text{img}}^l$ is a learnable weight matrix for level l. $\boldsymbol{B}^l$ denotes the feature map at level l of its feature pyramid. Note that we also employ 8-way multi-head attention strategy. Finally, we obtain a fused feature $\boldsymbol{f}$ by a standard add-norm layer and a feed forward network (FFN) as in the original Transformer [23]:

$$\boldsymbol{f} = \text{FFN}\left(\text{Add\&Norm}\left(\boldsymbol{f}_{\text{img}}, \boldsymbol{f}_{\text{coord}}\right)\right). \tag{4}$$

We illustrate the image feature fusion module in Figure 2b.

Filtering edge candidates: Transformer is memory intensive and we filter out bad candidates by passing $\boldsymbol{f}$ to a 2-layer MLP followed by a sigmoid function and computing a confidence score. We keep the top-K candidates, where K is three times the number of corner candidates. This filtering often takes a few thousand candidates and cuts down to a few hundred. This module is trained with the rest of the network with a binary cross entropy (BCE) loss. We have four BCE losses in total and refer to Sect. 5 for the balancing weights.

4.3. Holistic edge decoders

HEAT adopts two weight-sharing Transformer [23] decoders to classify each edge candidate to be correct or not.

Image-aware decoder: The first decoder takes the fused feature $\boldsymbol{f}$ for each edge candidate as a node. The network contains six layers of (self-attention, edge image feature fusion module, and feed forward network), with 8-way multi-head attention (See Fig. 2c). The BCE is the loss function.

Geometry-only decoder: The second decoder has exactly the same architecture and shares the weights, while we disable the use of image information. First, we pass the coordinate feature $\boldsymbol{f}_{\text{coord}}$ as the initialization, which does not have image information. Second, this decoder does not have the
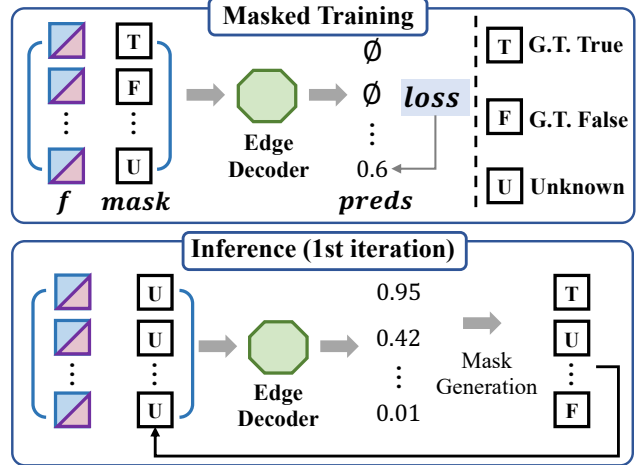

Figure 3. (Top) The masked edge learning strategy. (Bottom) The iterative edge inference at the 1st iteration.

image feature fusion module (gray parts in Fig. 2c). The decoder is forced to solve the task solely with geometry information, enhancing holistic geometric reasoning and the performance of the image-aware decoder through weight-sharing. This decoder is trained with BCE loss, while only the image-aware decoder is used for test time results.

4.4. Masked training and iterative inference

While the proposed architecture works well with vanilla supervised learning, a masked learning strategy inspired by the masked language modeling in BERT [6] can further encourage the learning of structural patterns and boost performance. The idea is to randomly provide ground-truth (GT) labels for some edge candidates and only ask to infer missing labels (See Fig. 3). Each edge candidate has one of the three states: (T) A GT label is given as true; (F) A GT label is given as false; or (U) A GT label is unknown and the network needs to infer. We modify the architecture slightly to inject the state information. Concretely, we represent the state as an one-hot encoding vector, convert to 256-dimension by a linear layer, concatenate with $\boldsymbol{f}_{\text{coord}}$, and downscale to 256-dimension by another linear layer.

During training, we pick a ratio uniformly from [0, 0.5] and provide GT labels for randomly sampled edge candidates based on the ratio. At test time, we perform iterative label inference with the image-aware decoder. At the first iteration, the state mask contains all U labels (Fig. 3). For the second iteration, edges with confidence predictions less than 0.01 (resp. more than 0.9) will have a F (resp. a T) label, while the remaining will have a U label. At the last iteration, a threshold of 0.5 is used to produce the final predictions. We set the number of iterations to be 3 for all experiments.

4.5. Corner detector

HEAT works with any corner detector. We found that an

adaptation of HEAT architecture achieves superior corner detection performance and further improves the edge classification. Concretely, we take the HEAT architecture till the edge-filtering module as a corner detector, as we found that self-attention did not improve corner detection. Every pixel is a corner candidate and hence becomes a Transformer node. The node feature is initialized by the positional encoding of the pixel coordinate, which is half the dimension of the edge feature. Thus we modify M_{coord} to be a 128×256 matrix so that the rest of the architecture remains the same. Non-maximum suppression is applied and we use a threshold of 0.01 to select corner candidates, such that important corners are unlikely to be missed. The corner model is also trained with a BCE loss. Please see the supplementary for details.

5. Experiments

We implemented our approach in Python3.7 and Pytorch1.5.1, and used a workstation with a 3.4GHz Xeon and dual NVIDIA RTX 2080 GPUs. Our image encoder contains only one Transformer layer while the edge decoders have six. We use ResNet-50 as the backbone to be consistent with the Transformer-based competing method LETR [24]. The loss balancing weights of the three edge BCE losses are all 1.0, while the weight for corner prediction BCE is 0.05 and 0.1 for the outdoor and indoor tasks, respectively. We train our model with Adam optimizer [14] with an initial learning rate 2e-4 and a weight decay factor 1e-5. The learning rate decays by a factor of 10 for the last 25% epochs. With a reference to LETR, our training schedule has 800 epochs for outdoor reconstruction and 400 epochs for floorplan reconstruction, based on the number of training images. Note that the datasets are both small so we set a large epoch number without hyper-parameter search, and the same setting is used for running competing methods. HEAT does not apply any post-processing for generating the final planar graph.

5.1. Competing methods

We evaluate the proposed approach on two tasks (i.e., outdoor architecture and indoor floorplan reconstructions). For the outdoor task, we compare with five methods: ConvMPN [26], IP [17], Exp-cls [27], HAWP [25] and LETR [24]. The first three methods were demonstrated for the same outdoor task, while the latter two methods were for the wire-frame parsing.

• **ConvMPN** is an improved graph neural network for edge classification and requires a pre-trained corner detector.

• **IP** and **Exp-cls** rely on heavy optimization or search process to reconstruct a planar graph based on geometric primitives detected by neural networks.

• **HAWP** [25] is a state-of-the-art wireframe parsing method, which is an improvement over LCNN [30] and performs independent edge classification.

• **LETR** is a Transformer-based line detection framework adapted from the object detection framework DETR [3].

For the indoor task, we compare with four methods: HAWP, LETR, Floor-SP [4], and MonteFloor [22]. HAWP and LETR have flexible designs with good performance for the outdoor task, and hence are chosen again.

• **Floor-SP** is one of the state-of-the-art floorplan reconstruction techniques with domain specific system design. The method first uses Mask-RCNN [10] to obtain room segmentation, and conducts complex optimization procedure.

• **MonteFloor** is an improvement over Floor-SP with a similar algorithm specifically designed for floorplan reconstruction, where a Monte Carlo tree search algorithm is employed.

For both outdoor and indoor tasks, we borrow the standard metrics in the literature for the evaluation (See Sect. 3). We use the numbers reported in the original papers for domain-specific baselines (i.e., ConvMPN, IP, Exp-cls, Floor-SP, and MonteFloor). The publicly available official implementations are used for HAWP and LETR.

5.2. Quantitative evaluation

Outdoor architecture reconstruction: Table 1 presents the main quantitative evaluation. HEAT outperforms all the competing methods on all the F-1 scores, including IP and Exp-Cls, which employ expensive optimization/search methods and are a few orders of magnitude slower than our method. Note that the outdoor reconstruction task [17] uses 256×256 images, while HAWP and LETR were demonstrated on larger images in their original papers. Therefore, we also conduct experiments in the 512×512 resolution, where we resize the training/testing images with the same data split. In the higher resolution, LETR exhibits higher region metrics but poor edge precision compared to HAWP. Our hypothesis is that LETR relies mostly on the image features and does not learn holistic geometric reasoning over the "dummy query nodes", resulting in many false edges and building reconstructions which do not look like buildings. HEAT is still a clear winner.

Floorplan reconstruction: Table 2 presents the main quantitative evaluation. The overall performance of HEAT surpasses HAWP, LETR, and Floor-SP, and is on par with MonteFloor. It is surprising that our HEAT is even comparable to MonteFloor, which utilizes both deep neural networks and Monte Carlo tree search algorithm. The average inference time of MonteFloor is more than a minute, while our method runs in a dozen milliseconds.

5.3. Qualitative evaluation

Outdoor architecture reconstruction: Figure 4 provides qualitative comparisons. The reconstruction quality of HEAT is clearly better than the competing methods and close to the ground-truth even on large and complex architectures.

Table 1. **Quantitative evaluations on outdoor architecture reconstruction.** Size: the size of input image. Fully-neural: not using hand-crafted optimization or search techniques. Joint: the edge prediction is trained end-to-end with corner detection. The colors cyan and orange mark the top-two results with different image sizes.

Evaluation Type →				Corner			Edge			Region		
Method	Size	Fully-neural	Joint	Prec	Recall	F-1	Prec	Recall	F-1	Prec	Recall	F-1
IP [17]	256	-	-	-	-	74.5	-	-	53.1	-	-	55.7
Exp-Cls [27]	256	-	-	92.2	75.9	83.2	75.4	60.4	67.1	74.9	54.7	63.5
ConvMPN [26]	256	✓	-	78.0	79.7	78.8	57.0	59.7	58.1	52.4	56.5	54.4
HAWP [25]	256	✓	✓	90.9	81.2	85.7	76.6	68.1	72.1	74.1	55.4	63.4
LETR [24]	256	✓	✓	87.8	74.8	80.8	59.7	58.6	59.1	68.3	48.7	56.8
HEAT(Ours)	256	✓	✓	91.7	83.0	87.1	80.6	72.3	76.2	76.4	65.6	70.6
HAWP [25]	512	✓	✓	90.6	83.7	87.0	78.8	72.0	75.2	77.5	57.8	66.2
LETR [24]	512	✓	✓	90.3	79.7	84.7	64.0	71.6	67.6	77.1	62.6	69.1
HEAT(Ours)	512	✓	✓	90.7	86.7	88.7	82.2	77.4	79.7	79.6	69.0	73.9

Table 2. **Quantitative evaluations on floorplan reconstruction.** Results for MonteFloor and Floor-SP are borrowed from MonteFloor paper. The colors cyan and orange mark the top-two results.

Eval Type →		Room		Corner		Angle	
Method	t(s)	Prec	Recall	Prec	Recall	Prec	Recall
HAWP [25]	0.02	0.78	0.88	0.66	0.77	0.60	0.70
LETR [24]	0.04	0.94	0.90	0.80	0.78	0.72	0.71
HEAT(Ours)	0.11	0.97	0.94	0.82	0.83	0.78	0.79
Floor-SP [4]	785	0.89	0.88	0.81	0.73	0.80	0.72
MonteFl. [22]	71	0.96	0.94	0.89	0.77	0.86	0.75

Table 3. **Ablation study for different corner/edge prediction methods**, evaluated with corner/edge/region F-1 scores on outdoor reconstruction. The image size is 256×256. Joint: the corner and edge prediction models are trained end-to-end. FRCNN: pre-trained Faster-RCNN provided by ConvMPN. HG: HourGlass netowork used by HAWP.

Corner Model	Edge Model	Joint	Corner	Edge	Region
FRCNN	ConvMPN [26]	-	78.8	58.1	54.4
FRCNN	HEAT(Ours)	-	78.8	68.2	62.5
HG	HAWP [25]	✓	85.7	72.1	63.4
HG	HEAT(Ours)	✓	85.9	74.3	69.3
HEAT(Ours)	HEAT(Ours)	✓	87.1	76.2	70.6

Table 4. **Ablation study for the technical components of HEAT edge prediction**, evaluated with edge/region F-1 scores of outdoor reconstruction. The image size is 256×256. The pre-trained FRCNN from ConvMPN is used as the corner model.

Coord	D-attn	Dec^{img}	Mask	Dec^{geom}	Edge	Region
-	✓	-	-	-	59.2	21.0
✓	-	-	-	-	62.8	43.5
✓	✓	-	-	-	67.3	48.6
✓	✓	✓	-	-	67.3	60.2
✓	✓	✓	✓	-	68.5	60.7
✓	✓	✓	✓	✓	68.2	62.5

Looking carefully at the structures, HEAT is especially good at capturing fine details and keep the overall prediction consistent and geometrically valid (*e.g.*, less dangling edges, not disturbed by background buildings).

Floorplan reconstruction: Figure 5 shows the qualitative comparisons. Only valid polygons of the predicted planar graph are colored. LETR and HEAT are better at recovering complete room structures compared to HAWP, as the use of Transformer enables better structured reasoning. HEAT still outperforms LETR in terms of producing accurate angles, which is consistent with the numbers in Table 2.

5.4. Ablation studies

We conduct extensive ablation studies on the outdoor reconstruction task.

Separating corner and edge modules: Most methods are combinations of corner detection and edge selection modules. Table 3 assesses individual contributions. In particular, we compare three corner models: 1) a pre-trained Faster-RCNN [20] (FRCNN) used and provided by ConvMPN; 2) an end-to-end trained HourGlass (HG) network used in LCNN and HAWP; and 3) HEAT (ours). The second, the fourth, and the fifth rows share the same HEAT edge model

and demonstrate that HEAT corner model makes consistent improvements in all the F1-scores. Next, we compare three edge selection models: 1) ConvMPN; 2) HAWP; and 3) HEAT (ours). The first two rows compare ConvMPN and HEAT, while both using FRCNN as the corner model to be fair. Similarly, the next two rows compare HAWP and HEAT, while using HG as the corner model that was used by HAWP. The table again shows that HEAT makes clear improvements over all the competing methods.
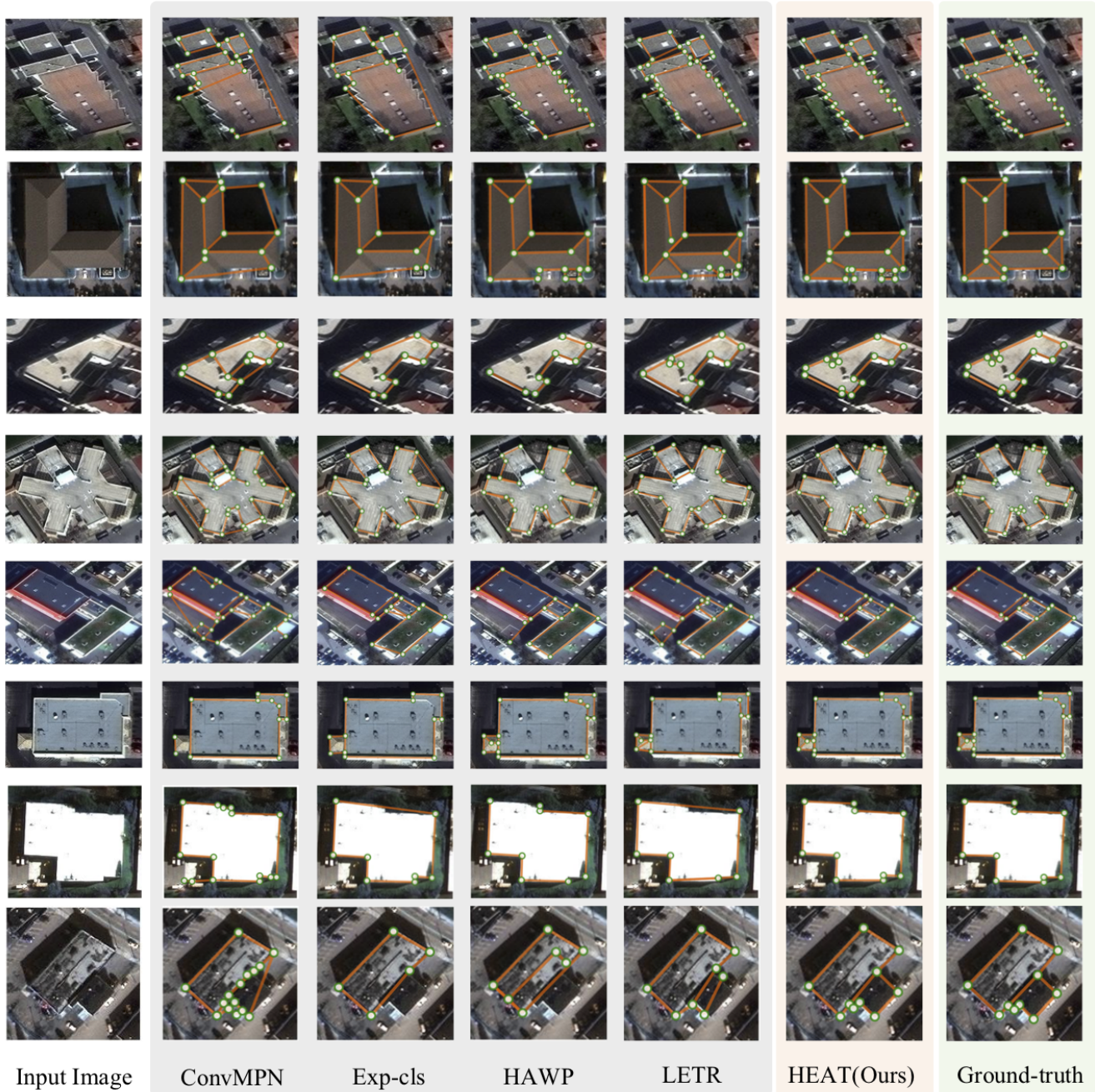
Figure 4. **Qualitative evaluations on outdoor architecture reconstruction** with image size 256×256. Large and complex samples are selected to demonstrate the challenges of the task. More results with higher resolution are available in the supplementary material.

HEAT system components: Table 4 evaluates the five HEAT system components in the edge selection module. The pre-trained FRCNN model by ConvMPN is used for corner detection to be simple. The five columns indicate:

• [Coord] Node initialization with edge coordinates information or not (*i.e.*, a zero vector);

• [D-attn] The edge deformable attention for image feature extraction or the LoI-Pooling from LCNN [30];

• [Dec^{img}] The image-aware decoder or without it (*i.e.*, using the confidence scores from edge filtering for prediction);

• [Mask] The masked learning and iterative inference or single-shot training and inference; and

• [Dec^{geom}] The weight-sharing geometry-only decoder or the image-aware decoder alone.

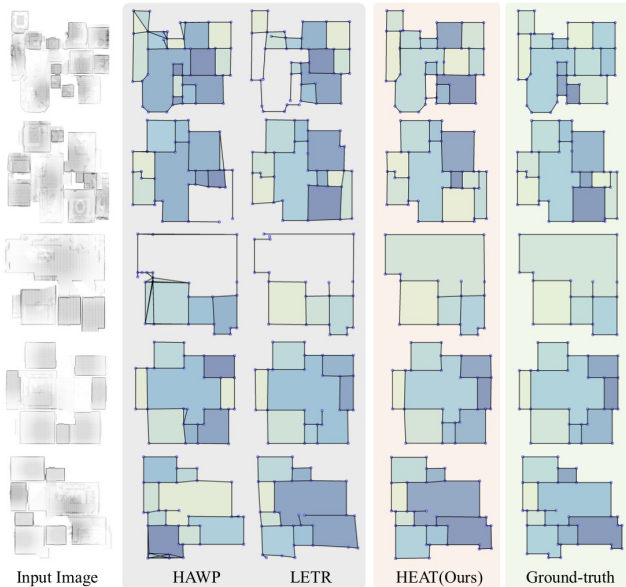The first row shows that the edge deformable attention cannot

Figure 5. **Qualitative evaluations on floorplan reconstruction**. Closed polygons in the predicted planar graph are colored while broken ones are not.

Table 5. **Edge prediction with the geometry-only decoder**, with different types of corners. "edge filter" means all remained edge candidates after the edge filtering. "HEAT (geom)" means using the geometry-only decoder for inference. "GT" uses ground-truth information to select corner pairs as the answers, representing the performance upper-bound given the corners.

Eval Type →		Edge			Region		
Corners	Method	Prec	Recall	F-1	Prec	Recall	F-1
FRCNN	edge filter	25.8	68.5	37.5	3.4	22.4	5.9
	HEAT (geom)	63.3	56.5	59.7	49.7	44.3	46.8
	HEAT	77.5	60.9	68.2	74.7	53.8	62.5
	GT	83.6	66.8	74.2	80.4	63.4	70.9
GT	edge filter	38.5	99.9	55.6	5.8	35.9	10.0
	HEAT (geom)	91.6	92.6	91.9	71.9	76.2	74.0
	HEAT	96.6	93.8	95.2	91.8	84.5	88.0
	GT	100	100	100	100	100	100

work well without proper coordinate features. Other rows in the table validate the contributions of the components, which consistently improve the metrics.

Geometry-only decoder: Table 5 shows the power of the geometry-only decoder that conducts only geometric reasoning without any image information. Concretely, we compare performance of the four edge-selection results: the edge-filtering, the geometry-only decoder, the image-aware decoder (the full HEAT), and the ground-truth (GT), while using either FRCNN or GT corners for corner detection.

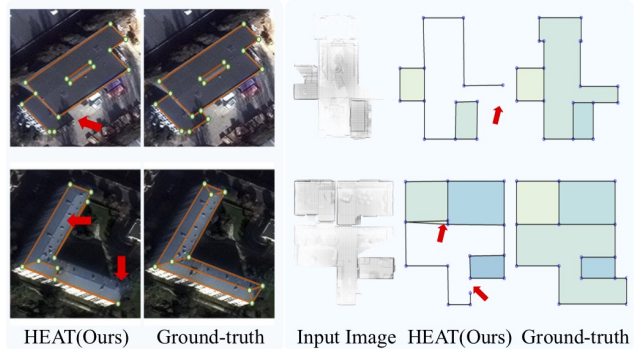HEAT(Ours)   Ground-truth   Input Image   HEAT(Ours)   Ground-truth

Figure 6. **Typical failure modes** on the two reconstruction tasks.

The geometry-only decoder makes significant improvements over the edge-filtering, which is the input of the decoder. The most striking result is that the geometry-only decoder with the GT corners is much better than the HEAT full system with FRCNN corners. The result demonstrates that HEAT learns to conduct powerful holistic geometric reasoning.

Limitations Figure 6 presents failure modes of HEAT. The model still misses important corners even with a small threshold, leading to further mistakes in edge prediction. Rare structures (*e.g.*, the bottom-left L-shape building) are also challenges. We do not touch 3D structured reconstruction (*e.g.*, the 3D wireframe parsing task [31]) with HEAT, which can be a potential future work.

Please see the supplementary for more qualitative visualizations as well as quantitative ablation studies (*e.g.*, details of iterative inference, choice of positional encoding, etc.).

6. Conclusion

This paper presents HEAT, a novel attention-based neural network that takes 2D raster image and reconstructs a planar graph depicting the underlying geometric structures. HEAT detects corners and learns to exploit both image information and geometric patterns between edge candidates in reconstructing a planar graph. Two weight-sharing Transformer decoders are the technical core of the approach. The entire system is trained end-to-end with a masked learning strategy and performs iterative inference at test time. Qualitative and quantitative evaluations demonstrate that our method pushes the frontier of end-to-end neural architecture for structured reconstruction. Extensive ablation studies further justify the design choices. We will share our code and models.

Potential negative societal impact: Outdoor reconstruction could potentially facilitate satellite imaging for military tasks. Indoor floorplan estimation could raise privacy concerns for house scanning without user permission.

References

[1] Antonio Adan and Daniel Huber. 3d reconstruction of interior wall surfaces under occlusion and clutter. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 275–281. IEEE, 2011. 2

[2] Ricardo Cabral and Yasutaka Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 628–635. IEEE, 2014. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020. 1, 2, 5

[4] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2661–2670, 2019. 1, 2, 3, 5, 6

[5] Erick Delage, Honglak Lee, and Andrew Y Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR)*, volume 2, pages 2418–2428. IEEE, 2006. 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2, 4

[7] Adam Van Etten, David Lindenbaum, and Todd M. Bacastow. Spacenet: A remote sensing dataset and challenge series. *ArXiv*, abs/1807.01232, 2018. 2

[8] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Manhattan-world stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1422–1429. IEEE, 2009. 2

[9] Lucile Gimenez, Jean-Laurent Hippolyte, Sylvain Robert, Frédéric Suard, and Khaldoun Zreik. reconstruction of 3d building information models from 2d scanned plans. *Journal of Building Engineering*, 2:24–35, 2015. 2

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 2020. 2, 5

[11] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[12] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th international conference on computer vision (ICCV)*, pages 1849–1856. IEEE, 2009. 2

[13] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 626–635, 2018. 2

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 5

[15] Chen Liu, Jiajun Wu, Pushmeet Kohli, and Yasutaka Furukawa. Raster-to-vector: Revisiting floorplan transformation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2195–2203, 2017. 2

[16] Josep Lladós, Jaime López-Krahe, and Enric Martí. A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform. *Machine Vision and Applications*, 10(3):150–158, 1997. 2

[17] Nelson Nauata and Yasutaka Furukawa. Vectorizing world buildings: Planar graph reconstruction by primitive detection and relationship inference. In *ECCV*, 2020. 1, 2, 5, 6

[18] Yiming Qian and Yasutaka Furukawa. Learning pairwise inter-plane relations for piecewise planar reconstruction. In *European Conference on Computer Vision (ECCV)*, pages 330–345. Springer, 2020. 2

[19] Xuebin Qin, Shida He, Xiucheng Yang, Masood Dehghan, Qiming Qin, and Jagersand Martin. Accurate outline extraction of individual building from very high-resolution optical images. *IEEE Geoscience and Remote Sensing Letters*, 15(11):1775–1779, 2018. 2

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems (NeurIPS)*, 28:91–99, 2015. 6

[21] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision (ECCV)*, pages 746–760. Springer, 2012. 2

[22] Sinisa Stekovic, Mahdi Rad, Friedrich Fraundorfer, and Vincent Lepetit. Montefloor: Extending mcts for reconstructing accurate large-scale floor plans. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 3, 5, 6

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3, 4

[24] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line segment detection using transformers without edges. In *CVPR*, 2021. 1, 2, 5, 6

[25] Nan Xue, Tianfu Wu, Song Bai, Fudong Wang, Guisong Xia, Liangpei Zhang, and Philip H. S. Torr. Holistically-attracted wireframe parsing. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2785–2794, 2020. 1, 2, 3, 5, 6

[26] Fuyang Zhang, Nelson Nauata, and Yasutaka Furukawa. Conv-mpn: Convolutional message passing neural network for structured outdoor architecture reconstruction. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2795–2804, 2020. 1, 2, 5, 6

[27] Fuyang Zhang, Xiang Xu, Nelson Nauata, and Yasutaka Furukawa. Structured outdoor architecture reconstruction by exploration and classification. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 5, 6

[28] Ziheng Zhang, Zhengxin Li, Ning Bi, Jia Zheng, Jinlei Wang, Kun Huang, Weixin Luo, Yanyu Xu, and Shenghua Gao. Ppgnet: Learning point-pair graph for line segment detection.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7105–7114, 2019. 2

[29] Jia Zheng, Junfei Zhang, J. Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *ECCV*, 2020. 1, 3

[30] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 962–971, 2019. 2, 3, 5, 7

[31] Yichao Zhou, Haozhi Qi, Yuexiang Zhai, Qi Sun, Zhili Chen, Li yi Wei, and Yi Ma. Learning to reconstruct 3d manhattan wireframes from a single image. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7697–7706, 2019. 1, 2, 8

[32] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ArXiv*, abs/2010.04159, 2021. 2, 3

[33] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2051–2059, 2018. 2