EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

Volume 45 (2026), Number 2

Raster2Seq: Polygon Sequence Generation for Floorplan Reconstruction

paper1137

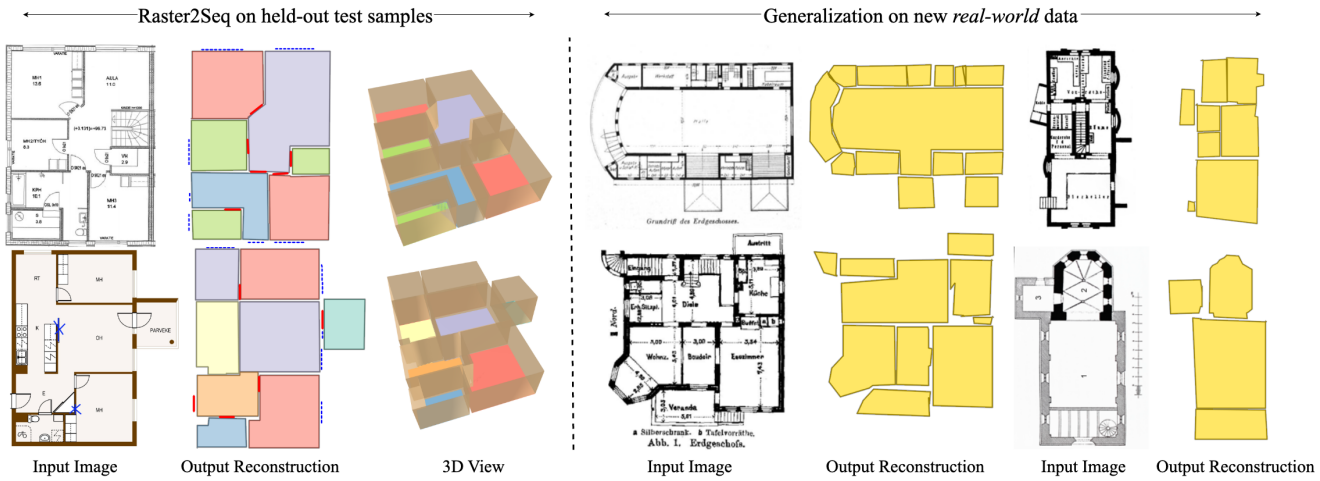

Figure 1: Our approach transforms rasterized floorplan images to vectorized format, reconstructing both its structure and semantics. We illustrate* results on held-out CubiCasa5K [KYH*19] test samples (left). The colors denote unique semantic categories (*e.g.*, Outdoor, Bedroom, bath, and entry). Additionally, we highlight our model's generalization capabilities over complicated real-world floorplan images from WAFFLE [GAMAE25] (right). *3D visualizations are constructed by extending the 2D boundaries vertically.

Abstract

Reconstructing a structured vector-graphics representation from a rasterized floorplan image is typically an important prerequisite for computational tasks involving floorplans such as automated understanding or CAD workflows. However, existing techniques struggle in faithfully generating the structure and semantics conveyed by complex floorplans that depict large indoor spaces with many rooms and a varying numbers of polygon corners. To this end, we propose Raster2Seq*, framing floorplan reconstruction as a sequence-to-sequence task, where each room is represented as a polygon sequence—labeled with the room's semantics. Our approach introduces an autoregressive decoder that learns to predict the next corner conditioned on image features and previously generated corners using guidance from learnable anchors. These anchors represent spatial coordinates in image space, hence allowing for effectively directing the attention mechanism to focus on informative image regions. By embracing the autoregressive mechanism, our method offers flexibility in the output format, enabling for efficiently handling complex floorplans with numerous rooms and diverse polygon structures. Our method achieves state-of-the-art performance on standard benchmarks such as Structure3D, CubiCasa5K, and Raster2Graph, while also demonstrating strong generalization to more challenging datasets like WAFFLE, which contain diverse room structures and complex geometric variations.*

CCS Concepts

• ***Computing methodologies*** → *Object detection;*

1. Introduction

Floorplans are a fundamental element of architectural design that define the structure and semantics of indoor spaces, from the tiny studio apartment in Manhattan to the historic Café Helms in Berlin (depicted in the top right corner of Figure 1). While floorplans are typically drawn in a vector-graphics representa-

tion using specialized softwares (e.g., AutoCAD), they are usually distributed in rasterized image formats. This rasterization process strips away the structured geometric and semantic information, severely limiting their utility for computational tasks such as automated editing [PKS*21, SPH*23, ZHY*24], floorplan understanding [WFU15, NWC*20], or 3D reconstruction [MBHRS14, LSK*15, NCV*24].

To unlock computational capabilities over rasterized floorplans, several works have explored the *raster-to-vector* conversion task [DLHAL*14, LWKF17, ZLYF19], which aims to transform an input floorplan image back to vectorized format. However, despite the significant advancements enabled by Transformer-based architectures [CQF22, YKSE23, HWS*24], existing methods face challenges in capturing the structure and semantics conveyed by complicated real-world floorplans, often depending on pretrained detectors and constructing sub-optimal multi-stage pipelines for performing the conversion.

In this work, we propose *Raster2Seq*, an approach that transforms rasterized floorplan images to vectorized format using a labeled polygon sequence representation. Unlike prior work that simultaneously predict all structural floorplan elements [SRFL21, YKSE23, CQF22], our framework autoregressively outputs a polygon sequence, directly modeling both spatial structure and semantic attributes. Our key observation, motivating our framework design, is that floorplan elements can be effectively modeled as a sequence, leveraging the left-to-right generation bias of masked attention models [VSP*17]. We represent each polygon as a sequence of labeled corners, *i.e.*, spatial coordinates labeled with semantic information, and sort the floorplan's polygons using a left-to-right ordering. Specifically, we consider rooms, windows and doors, but this representation could easily accommodate additional labeled entities. At its core, our framework introduces an anchor-based autoregressive decoder that effectively fuses information from image features and the previously generated corners to predict the next labeled corner. In particular, our autoregressive module is guided by learnable anchors that direct the attention mechanism to focus on informative regions, enabling for efficiently handling complex floorplan images.

We show the effectiveness of our framework on multiple benchmarks, conducting experiments in different floorplan reconstruction settings—considering both rasterized RGB images and 2D density maps as input. Our approach consistently surpasses existing methods over a wide range of geometric and semantic metrics. Notably, our results show that more complicated floorplans—containing higher quantities of corners and rooms—yield larger performance gaps. We also show strong generalization capabilities over challenging real-world Internet datasets, demonstrated both qualitatively and quantitatively. Code and models will be available.

2. Related Work

2.1. Floorplan Reconstruction

Raster-to-vector floorplan conversion aims to reconstruct structured, vectorized representations directly from rasterized floorplan images. Prior to the Deep Learning era, complex multi-step systems [MLVT10, ALWD11, DLHAL*14] were proposed for addressing this task. These systems typically involved the extraction and processing of various handcrafted features for detecting different floorplan components (such as walls of varying thickness).

Liu *et al.* [LWKF17] first integrated neural networks for solving this task, predicting intermediate corner representations followed by integer programming to recover geometric primitives such as edges. Zeng *et al.* [ZLYF19] proposed to utilize pixel-wise segmentations for floorplan conversion, linking elements through their hierarchical relationships. Similarly, Sun *et al.* [SWL*22] employed a graph neural network to model the hierarchy of floorplan elements based on boundary constraints. Raster2Graph [HWS*24] employs a transformer network [ZSL*21] and uses image-space augmentation to highlight visible corners, enabling sequential corner prediction. By contrast, our method frames floorplan conversion as a sequence-to-sequence task, generating polygon coordinates token-by-token in an autoregressive manner. Our recursive formulation enables generating variable-length polygons and effectively adapting to dense layouts, avoiding the need for additional steps, such as the image augmentation mechanism and corner sampling strategy used in Raster2Graph.

Several works aim at related floorplan reconstruction tasks, utilizing different modalities such as point-cloud density maps [SRFL21, CQF22, YKSE23], RGB panoramas [CF14, LWF18], rather than rasterized floorplan images. These methods typically reconstruct the entire floorplan in a single forward pass and can be broadly categorized into instance segmentation-based approaches and object detection-based approaches.

Floor-SP [CLWF19] and MonteFloor [SRFL21], for instance, frame the task as instance segmentation, followed by additional optimization steps to group room regions and refine geometric boundaries. While effective in certain settings, these multi-stage pipelines typically do not generalize well to diverse and irregular floorplan layouts. More recent work [CQF22, YKSE23, XXH*24, LZM*24] has demonstrated that end-to-end floorplan reconstruction is possible without the need of post-optimization stage. Both HEAT [CQF22] and FRI-Net [XXH*24] follow bottom-up strategies, with HEAT first detecting room corners and then classifying edge candidates between them, while FRI-Net predicts line primitives first and then groups them into rooms. Later works [YKSE23, LZM*24] formulate floorplan reconstruction as an object detection task by predicting room coordinates and relying on the Hungarian matching algorithm during training. In contrast, our work requires no post-processing steps to produce final outputs and can be efficiently trained with a token-wise supervision loss.

While these methods were originally designed for 3D-scan-based inputs—typically represented as 2D density maps—we demonstrate in our experiments that they can be adapted for the raster-to-vector conversion task. Nonetheless, these methods assume a fixed number of corners and rooms per image, which are all simultaneously predicted. By contrast, our method is designed to predict corners and rooms recursively, emulating the iterative process of a typical CAD workflow. As demonstrated in our experiments, our approach is better suited for handling dense layouts with varying numbers of corners. Additionally, unlike most prior work that focus solely on predicting structural information, our method incorporates semantic room labels into the floorplan reconstruction task. This is achieved using a simple yet effective token-level classification loss, which

eliminates the complexity of indirect label prediction methods used in prior semantic-augmented works [YKSE23, HWS*24].

2.2. Sequence-to-Sequence Modeling for Visual Tasks

Sequence-to-sequence (seq2seq) modeling [SVL14] was originally proposed for machine translation, with the goal of learning a mapping from a source sequence to a target sequence. This framework was later adapted to a plethora of computer vision tasks by providing image features as input to a decoder (typically an RNN or Transformer) that generates a target sequence. Notable applications include image captioning [VTBE15, XBK*15, CSBC20], object detection [CSL*21], instance segmentation [ALKF18, LDC*23, CSL*22], and image generation [RPG*21, YXK*22]. The seq2seq paradigm enables end-to-end training and naturally accommodates inputs and outputs of variable lengths, eliminating the need for complex post-processing. This paradigm was adopted by Liu *et al.* [LDC*23] for representing object segmentations as polygon sequences, which can be utilized for the task of prompt-based segmentation. While our method is conceptually similar, our framework introduces several representation and architectural differences for performing floorplan reconstruction. For example, beyond predicting spatial coordinates, we introduce semantic labels into the representation and incorporate a novel semantic training objective for semantic-aware floorplan recognition. This semantic integration improves the utility of vectorized floorplans by producing both structural information and semantic labels.

Prior work has explored the effectiveness of recursive frameworks in modeling complex and structured visual data. For instance, GRASS [LXC*17] GRAINS [LPX*19], READ [PBEPAE20], SceneScript [AXHJ*24] demonstrated the utility of recursive prediction for 3D shapes, 3D indoor scene synthesis, 2D document layout generation, and 3D scene reconstruction, respectively. More closely related to our work, SceneScript formulates 3D scenes as text representations and learns to generate house layouts from input point clouds using predefined text commands for drawing objects (e.g. wall and object box). In our work, we adopt the sequence-to-sequence framework for floorplan transformation, predicting semantic polygon coordinates sequentially based on corner-based representation instead.

3. Method

An overview of our proposed method is presented in Figure 2. Our goal is to transform a rasterized floorplan image into vectorized format, reconstructing both its structure and semantics. Specifically, we assume that we are provided with an RGB image of a rasterized floorplan $I \in \mathbb{R}^{H\times W\times 3}$, where H and W denote the height and width of the image. The input image I is encoded via a *Feature Extractor* module to produce a feature vector $f_{img} \in \mathbb{R}^{L_I\times D}$ where L_I is the length of the image features and D is the number of channels.

Unlike existing floorplan reconstruction techniques [ZLYF19, SRFL21, SWL*22, CQF22] that extract vectorized floorplans via intermediate geometric elements such as edges, corners, or room segments, we propose to represent vectorized floorplans directly using a sequence of labeled polygons. We introduce this representation in Section 3.1. We then describe our *Anchor-based Autoregressive Decoder* module, the main architectural component in our framework, in Section 3.2. Finally, training and inference details are discussed in Section 3.3.

3.1. Labeled Polygon Sequence Floorplan Representation

We propose to represent vectorized floorplans using labeled polygon sequences. By labeled, we refer to the polygon's *semantics*. For instance, a room can be labeled as a *kitchen*, *bedroom*, etc. We parameterize a polygon as a sequence of labeled corner tokens c, where $c_i = (x_i, y_i, p_i)$ denotes the i-th corner in the polygon, $v_i = (x_i, y_i)$ denotes its spatial position, and $p_i \in [0,1]^C$ denotes its semantic probability vector (assuming C unique semantic categories). As we elaborate later in Section 3.3, room-level semantic predictions are obtained by aggregating semantic information at the token-level. We also consider windows and doors, in addition to rooms. These are simply represented as two additional semantic categories (on top of the room types).

To represent a floorplan that contains multiple rooms (or floorplan *entities*, such as windows)—each represented as a labeled polygon, as detailed above—we concatenate their sequences using a separator <SEP> token. We also use <BOS> and <EOS> tokens to indicate the beginning and the end of the sequence. Put together, the labeled polygon sequence is structured as follows:

$$[\texttt{<BOS>}, c_1^1, c_2^1, \cdots, \texttt{<SEP>}, c_1^n, c_2^n, \cdots, \texttt{<EOS>}]$$

As Raster2Seq is trained to regress continuous values without relying on a discrete tokenizer, each token is augmented with a token type probability vector $q \in [0,1]^3$, where the three token type categories are <CORNER>, <SEP> or <EOS>; a similar augmentation strategy was recently utilized in [LTL*24]. During training, the <CORNER> type is used as a supervision label for each corner token c_i but is not explicitly included in the sequence. <BOS> is omitted from the token type modeling. The training objective is to predict the next corner token in the sequence, where the output sequence contains the target tokens to be predicted; see Figure 2.

3.2. Anchor-based Autoregressive Decoder

Next, we present our *Anchor-based Autoregressive Decoder* module which predicts labeled polygon sequences; see Figure 3 for an illustration. Our proposed module is provided with three different inputs: (i) image features extracted with the *Feature Extractor* module, (ii) a sequence of coordinate tokens, and (iii) learnable anchors.

The sequence of coordinate tokens are provided after quantization of the continuous 2D coordinates into a discrete 1D embedding space using a learnable codebook $C \in \mathbb{R}^{H_b\times W_b\times D}$, where $H_b \times W_b$ is number of quantization bins and D is embedding dimension; additional details are provided in the supplementary material. Specifically, the decoder is provided with L coordinate tokens, which are denoted by $f_{poly} \in \mathbb{R}^{L\times D}$. Learnable anchors, denoted by $v_{anc} \in \mathbb{R}^{L\times 2}$, are introduced to avoid direct regression of continuous coordinate values. Instead, the model learns residuals relative to these anchors. We demonstrate that the use of learnable anchors yields significant performance gains in Section 4.

Decoder Architecture. The decoder contains an autoregressive

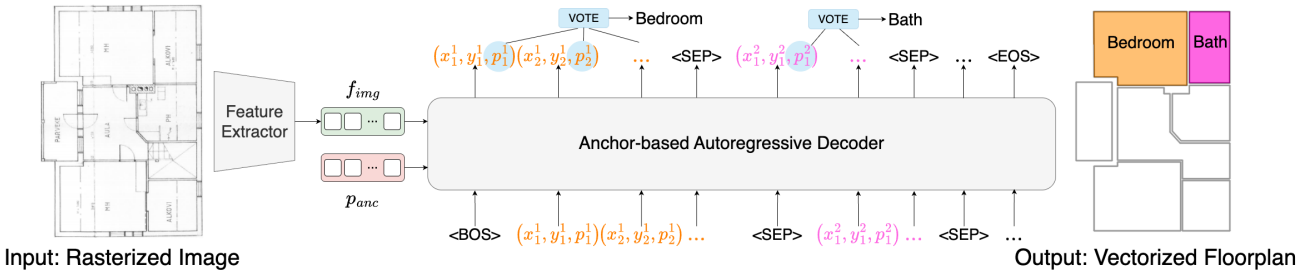

Figure 2: **Method Overview**. Given a rasterized floorplan image (left), our approach converts it into vectorized format, represented as a labeled polygon sequence, separated using special <SEP> tokens. The main architectural component of our framework is an anchor-based autoregressive decoder, which predicts the next token given image features (f_{img}), learnable anchors (v_{anc}) and the previously generated tokens; see Section 3.2 for additional details. Above, we visualize the first two labeled polygons predicted (colored in orange and pink, respectively).

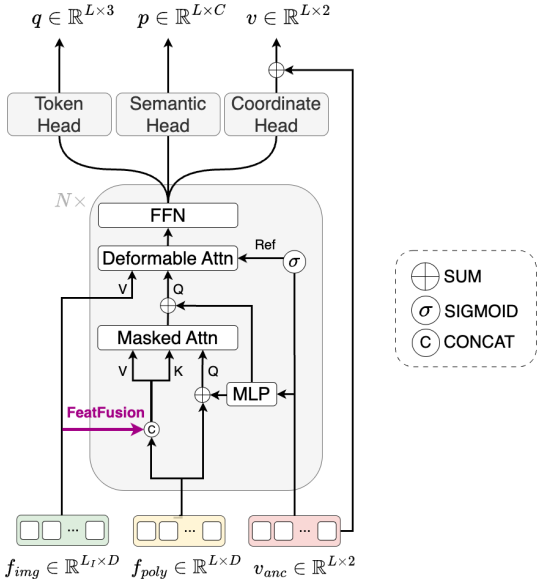

Figure 3: Illustration of our anchor-based autoregressive decoder.

block that contains three different layers: masked attention, deformable attention, and a feed-forward network layer. In the masked attention layer, a causal mask is applied to ensure that each token can only attend to its preceding tokens, reinforcing a left-to-right generation bias [VSP*17]. As shown in Fig. 3, the triplet of query (Q), key (K), and value (V) vectors is derived from the sequence of coordinate tokens. The query vector includes additional positional embeddings from the introduced anchors, while the key and value vectors are derived from a fused feature vector of shape $[L_I + L, D]$. This fused vector combines image features from the encoder with coordinate-token embeddings through tensor concatenation, referred to as *FeatFusion* (highlighted in purple in Figure 3). We find that this early fusion is crucial for precise coordinate regression. Intuitively, the image features act as a prefix that each token can attend to, providing additional contextual information during decoding.

Subsequently, the output vectors from the preceding masked attention layer serve as queries in a deformable attention module. This module, first introduced in [ZSL*21], is an efficient attention-based mechanism that—given a feature map and a set of reference points—for each query, only attends to a small set of sampling points around each reference point, rather than the entire feature map. In our autoregressive decoder, this mechanism allows for attending to a sparse set of relevant spatial positions in the image feature map f_{img}. Specifically, input anchor points are first normalized to [0,1] using a sigmoid function. The deformable attention layer then takes in the query vector and predicts offsets relative to these normalized anchor points using a linear layer. These offsets are added to the anchor points to produce sampling points, allowing the attention mechanism to focus on informative regions of image features. As previously mentioned, the anchor points are learnable parameters that are randomly initialized and learned jointly with the network weights.

Finally, the decoder module contains three lightweight heads on top of the last autoregressive block: a token head for predicting token types, a semantic head for predicting semantic labels, and a coordinate head for predicting 2D corner coordinates. The coordinate head essentially produces residual outputs which are combined with the learnable anchors for producing continuous coordinate values, as illustrated in Figure 3.

3.3. Training and Inference Details

Our method is supervised using three different loss functions: a coordinate regression loss, a token-type classification loss, and a semantic classification loss.

Coordinate loss. For the coordinate loss, we use a L1 loss to measure the difference between the predicted coordinates $\hat{v}$ and the ground-truth spatial coordinates v, across all L tokens (*i.e.*, corners) in the sequence:

$$\mathcal{L}_{coord} = \frac{1}{L}\sum_{l=1}^{L} \mathbf{m}_l |\hat{v}_l - v_l|, \tag{1}$$

This loss is computed only over non-padded tokens, using an additional mask $\mathbf{m}$ to exclude irrelevant positions. The same masking strategy is applied to the other losses described below.

Token-type loss. As defined as in Section 3.1, we consider three token classes: <CORNER>, <SEP>, and <EOS>. The model is trained

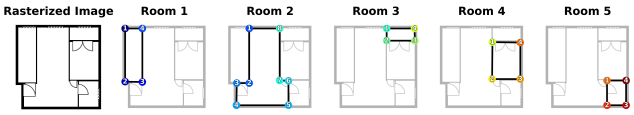

Figure 4: Given an input rasterized image, our method performs sequential corner prediction. We visualize earlier corners in cooler colors (predictions are enumerated per room). As illustrated above, within each room, corners are predicted in counterclockwise order.

to classify individual token into one of these categories using a standard cross-entropy loss:

$$\mathcal{L}_{token} = \frac{1}{L}\sum_{l=1}^{L} \mathbf{m}_l \mathrm{CE}(\hat{q}_l, q_l), \tag{2}$$

where $\hat{q}_l$ is the predicted probability distribution over three token types, and q_l is the ground-truth one-hot vector for the l-th token.

Semantic loss. We supervise prediction of semantic labels using a cross-entropy loss defined for each token:

$$\mathcal{L}_{sem} = \frac{1}{L}\sum_{l=1}^{L} \mathbf{m}_l CE(\hat{p}_l, p_l), \tag{3}$$

where $\hat{p}_l$ is the predicted probability distribution over C predefined room classes, and p_l is the one-hot vector representing the ground-truth room class for the l-th token in the sequence.

The total training loss is:

$$\mathcal{L} = \lambda_{coord} * \mathcal{L}_{coord} + \lambda_{token} * \mathcal{L}_{token} + \lambda_{sem} * \mathcal{L}_{sem}, \tag{4}$$

where λ_{coord}, λ_{token} and λ_{sem} are weighting coefficients. To effectively leverage the left-to-right generation bias, we perform a left-to-right ordering of the polygon sequence during training, where rooms are ordered by top-left coordinates using top-to-bottom, left-to-right scanning priority. As illustrated in our experiments, this significantly enhances our autoregressive learning framework.

At inference, Raster2Seq predicts tokens sequentially till a <EOS> token is obtained. To predict semantic room labels, we aggregate token-level predictions using a majority voting strategy. Specifically, the room label for each polygon sequence is determined by first selecting the class with the highest probability at each token, and then taking the most frequently predicted class across the sequence. Figure 4 provides a visualization of the sequential room prediction process, illustrating how the model maintains a left-to-right generation pattern. Additional details are provided in the supplementary material.

4. Experiments

In this section, we first describe the experimental setup and the baselines we compare our method against (Section 4.1). We then present our main quantitative results (Section 4.2), followed by both a qualitative comparison (Section 4.3). Finally, we present an ablation study of our proposed method (Section 4.4) and discuss several limitations of our approach (Section 4.5).

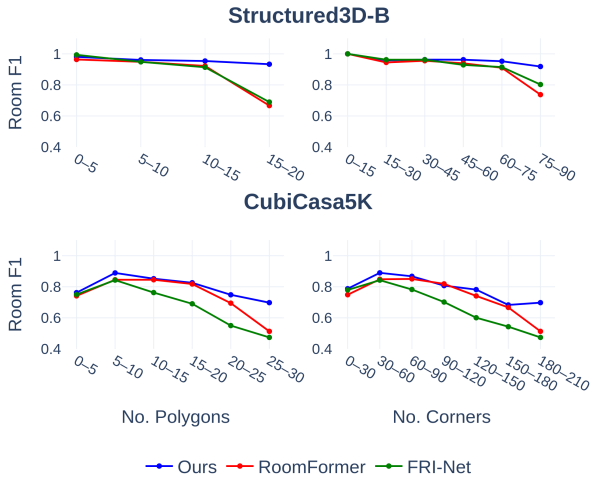

Figure 5: Performance vs. floorplan complexity—as approximated by the total number of polygons (left) and the total number of corners (right). As illustrated above over Structured3D-B (top) and CubiCasa5K (bottom), our approach yields larger gains as the floorplan complexity increases.

4.1. Experimental Setup

Datasets. We conduct experiments on four datasets: Structured3D [ZZL*20], Cubicasa5K [KYH*19], Raster2Graph [HWS*24], and WAFFLE [GAMAE25]. Structured3D is a 3D point cloud dataset containing 3,000/250/250 training/validation/test samples, annotated with 16 room types. CubiCasa5K is a raster-based floorplan dataset with 4,199/399/399 training/validation/test samples, annotated with 11 classes. Raster2Graph has 9,803/500/499 training/validation/test samples, annotated with 12 classes. WAFFLE contains 20K real-world floorplan images scraped from the Internet. As this dataset only contains approximately 100 annotated samples, we only evaluate zero-shot generalization capabilities on this data.

For Structured3D, existing work [YKSE23] use the projection of 3D point clouds along the vertical axis as input images. Since our focus is on raster-to-vector floorplan reconstruction, we convert the Structured3D samples into binary raster images using the ground-truth annotations, yielding images resembling typical floorplans which are used to train our method; please see the supplementary for additional details. We refer to this converted dataset as Structured3D-B for convenience. Some CubiCasa5K images contain multiple floorplans, so we preprocess them into separate images, increasing the dataset size from 5,000 to 6,281 samples (5,267 train / 503 val / 511 test). We use a fixed resolution of 256×256 for all datasets in all experiments.

Metrics. We follow the evaluation protocol used by prior work [SRFL21], focusing on geometric and semantic metrics obtained from matching model predictions with the ground truth annotations. Three evaluation criteria are Room, Corner, and Angle where each criterion is evaluated using Precision, Recall, and F1 score. Specifically, we first match each ground-truth room with the best-predicted

Method	Room			Corner			Angle			Room Semantic			Window & Door		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
HEAT	95.3	94.1	94.7	81.8	87.4	84.5	77.0	82.3	79.6	-	-	-	-	-	-
PolyRoom	99.4	98.5	98.9	**99.0**	93.1	96.0	**94.7**	89.3	91.9	-	-	-	-	-	-
FRI-Net	97.5	95.4	96.5	88.5	82.6	85.4	86.2	80.5	83.3	-	-	-	-	-	-
RoomFormer	95.8	94.4	95.1	93.0	90.5	91.7	84.4	82.1	83.2	74.7	73.8	74.2	95.0	93.1	94.1
Ours	**99.6**	**99.7**	**99.6**	98.9	**97.7**	**98.3**	93.3	**92.2**	**92.7**	**76.9**	**76.9**	**76.9**	**98.5**	**98.5**	**98.5**

Table 1: Quantitative evaluation on the *Structured3D-B* test set [ZZL*20], where the input image is a binary floorplan image (as further detailed in Section 4.1). Best results are in bold.

Method	Room			Corner			Angle			Room Semantic			Window & Door		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
HEAT	79.9	76.6	78.2	56.2	51.4	53.7	33.8	31.0	32.3	-	-	-	-	-	-
FRI-Net	82.1	72.7	77.1	**69.2**	40.1	50.8	**51.8**	30.0	**38.0**	-	-	-	-	-	-
RoomFormer	84.7	82.3	83.5	58.1	53.1	55.5	35.7	32.6	34.1	63.8	62.3	63.0	**80.8**	76.3	**78.5**
Ours	**89.3**	**88.0**	**88.7**	61.0	**57.8**	**59.4**	38.4	**36.4**	37.4	**64.4**	**63.2**	**63.8**	78.9	**76.7**	77.8

Table 2: Quantitative evaluation on the *CubiCasa5K* test set [KYH*19].

room based on Intersection over Union (IoU), and use these matched pairs to compute evaluation metrics at three levels: room, corner, and angle. For room-level evaluation, a match is considered valid if the IoU exceeds 0.5. For corner and angle evaluation, which are point-wise metrics, we follow the protocol of [SRFL21] by computing the L2 distance and the oriented angle between predicted and ground-truth corners. A corner is considered correctly recovered if the distance is within 10 pixels and the angle difference is less than 5 degrees. For semantic label evaluation, room type predictions are additionally used for finding matches. For WAFFLE, we report room prediction performance using IoU score to access zero-shot performance on the segmentation task. Additional metrics are reported in the supplementary material.

Baselines. We mainly utilize HEAT [CQF22], RoomFormer [YKSE23], FRI-Net [XXH*24]–models originally designed for point-cloud density maps—for conducting a quantitative evaluation, finetuning these models to perform floorplan reconstruction from rasterized floorplan inputs. We also compare our method against Raster2Graph [HWS*24] on raster-to-vector conversion task using their provided dataset.

4.2. Quantitative Evaluation

Performance on Rasterized Datasets. We compare the performance over the raster-to-vector conversion task across three datasets (see Tables 1 to 3). Overall, our method achieves state-of-the-art performance on both structural metrics (RoomF1 and CornerF1) and semantic metrics (RoomSeemF1 and WindowDoorF1). We note that unlike our method that directly optimizes token-level semantic predictions, RoomFormer dilutes semantic information by averaging irrelevant corners within uniform-length sequences. Raster2Graph, on the other hand, introduces unnecessary complexity through its four-neighbor classification scheme for corner points, which may also propagate errors. Regarding window and door predictions, our approach achieves performance comparable to Roomformer. As discussed further in Section 4.5, we believe additional performance gains can be achieved with more tailored architectural modifications.

Model Robustness To Floorplan Complexity. Figure 5 shows the Room F1 performance of RoomFormer, FRI-Net, and our model across varying numbers of polygons and corners on the Structured3D-B and CubiCasa5K datasets. Our method consistently demonstrates greater robustness as floorplan complexity increases. While both models perform similarly on simpler cases, RoomFormer and FRI-Net exhibit a notable performance drop in complex scenes with over 15 polygons or 150 corners. In contrast, our method maintains stable accuracy, demonstrating better scalability and robustness to dense layouts.

Cross-evaluation Results. We perform a cross-evaluation experiment across different train-test dataset configuration. We evaluate performance using metrics reported previously, using RoomF1 for the Cubicasa and Raster2Graph dataset and IoU for WAFFLE. Results are reported in Figure 6. As shown, our method demonstrates the strongest generalization performance across various settings, including both same-dataset and cross-dataset evaluations, outperforming other baselines by a large margin. In particular, we observe significant gaps over WAFFLE test set between our method and the counterparts, further demonstrating its robustness on complex and unseen floorplan samples.

Performance on Structured3D Density Maps. We conduct a comparison on the standard Structured3D benchmark, providing our model with density map inputs for both training and testing. As illustrated in Tab. 4, our method generally outperforms existing baselines on key geometric metrics such as Room and Angle. Although FRI-Net achieves competitive performance with our method

Method	Room			Corner			Angle			Room Semantic		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
HEAT	**98.0**	93.9	95.9	81.2	78.2	79.7	51.9	49.9	50.9	-	-	-
FRI-Net	94.9	88.4	91.5	**86.6**	62.1	72.3	63.2	45.3	52.8	-	-	-
RoomFormer	92.0	91.8	91.9	74.8	74.3	74.5	51.2	50.9	51.1	79.6	79.5	79.5
Raster2Graph	97.1	93.0	95.0	79.9	76.8	78.3	**68.6**	66.0	**67.3**	85.2	81.7	83.4
Ours	97.2	**96.8**	**97.0**	80.4	**80.1**	**80.3**	66.7	**66.5**	66.6	**85.3**	**84.9**	**85.1**

Table 3: Quantitative evaluation on the *Raster2Graph* test set [HWS*24].

Method	Room			Corner			Angle		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
MonteFloor [SRFL21]	95.6	94.4	95.0	88.5	77.2	82.5	86.3	75.4	80.5
HEAT [CQF22]	96.9	94.0	95.4	81.7	83.2	82.5	77.6	79.0	78.3
PolyRoom [LZM*24]	98.9	97.7	98.3	**94.6**	86.1	**90.2**	89.3	81.4	85.2
FRI-Net [XXH*24]	**99.5**	**98.7**	**99.1**	90.8	84.9	87.8	**89.6**	**84.3**	**86.9**
RoomFormer [YKSE23]	97.9	96.9	97.5	89.4	85.5	87.4	83.2	79.7	81.4
RoomFormer (w/ semantic)	95.3(-2.6)	93.5(-3.4)	94.4(-3.1)	85.7(-3.7)	81.8(-3.7)	83.7(-3.7)	78.0(-5.2)	74.5(-5.2)	76.2(-5.2)
Ours	99.0	98.4	98.7	92.0	87.1	89.4	84.7	80.3	82.5
Ours (w/ semantic)	99.1	98.6	98.8	92.1	**88.1**	90.0	86.1	82.5	84.2
FRI-Net + PD [XXH*24]	**99.6**	98.6	99.1	**94.2**	88.2	91.1	**91.9**	86.7	**89.2**
RoomFormer + PD [CDF23]	98.7	98.1	98.4	92.8	**89.3**	91.0	90.8	**87.4**	89.1
Ours + PD	99.4	**98.9**	**99.2**	93.2	89.2	**91.2**	91.0	87.2	89.0

Table 4: Quantitative evaluation on the *Structured3D* test set [ZZL*20], where the input is a density map generated from top-view projection of the 3D point cloud. In the bottom rows, we report performance using PD [CDF23], a recent refinement method. As illustrated above, our method demonstrates competitive performance on this benchmark, and is compatible with existing refinement methods, which enable further performance gains.

FeatFusion	Anchor	Ordering	Room F1	Corner F1	Angle F1
			94.1	91.1	82.0
✓			96.3	93.7	82.6
✓	✓		97.4	95.3	86.0
✓	✓	✓	99.6	98.3	92.7

Table 5: Ablation studies, evaluating the effect of our *FeatFusion* mechanism, the learnable tokens, and performing a left-to-right ordering of the polygons during training, over the *Structure3D-B* dataset.

when using density maps, performance on image inputs is generally lower (see Tabs. 1 to 3). We hypothesize that FRI-Net's reliance on disentangled representations of raw line primitives makes it less robust to the diverse structural and appearance variations present in RGB floorplans compared to the homogeneous nature of density maps. We also report performance using PD [CDF23]–a polygon refinement approach. Our method achieves state-of-the-art performance, demonstrating its compatibility with advanced post-processing techniques. Regarding semantic metrics, RoomFormer exhibits a significant performance drop of 2–5 points when semantic room types are included. In contrast, our model maintains consistent performance under the same conditions, indicating the efficacy of our method for semantic incorporation.

4.3. Qualitative Results

We provide visual comparisons with the RoomFormer model over CubiCasa5K test samples in Fig. 10 and WAFFLE images in Fig. 12. In both cases, the models are trained over the CubiCasa5K train set. These figures illustrate superior visual quality compared to the RoomFormer baseline. In particular, we observe that the RoomFormer model often yields "short-cut" triangular polygons (*e.g.*, leftmost example in Fig. 10), while our model allows for more accurately reconstructing the floorplan's structure.

In Fig. 11, we compare our method with Raster2Graph on their dataset. As clearly seen, our method achieves superior reconstruction quality compared to other counterparts. Notably, Raster2Graph often fails to recover complete floorplan structures, while our approach remains robust across diverse layouts. Note that we only show the structural predictions (without semantics) both here and in the zero-shot generalization experiment over WAFFLE (Fig. 12), as semantic annotations often vary across different datasets. In particular, doors and windows may have different appearances (as illustrated, for instance, by the qualitative results in Fig. 9 and Fig. 10).

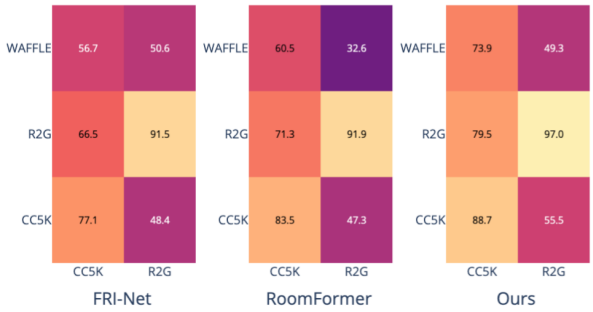

Figure 6: Cross-evaluation heatmaps showing performance across training (rows) and test (columns) dataset combinations, with lighter colors denoting higher performance. R2G and CC5K denote Raster2Graph and CubiCasa5K datasets, respectively. Our method exhibits strong generalization across different evaluation settings, substantially outperforming FRI-Net and RoomFormer.

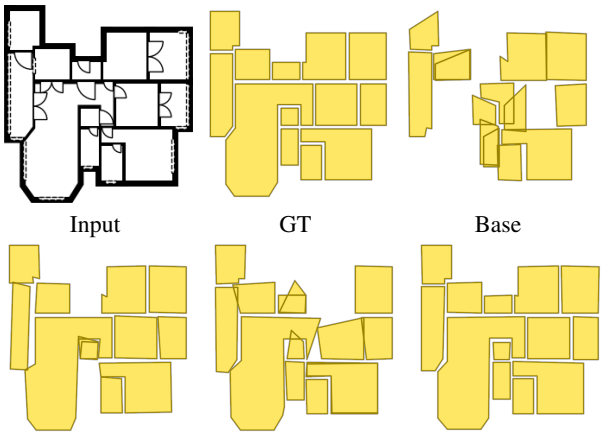

Figure 7: Ablation results over a sample from the Structure3D-B test set. As illustrated above, incorporating our proposed components significantly improves geometric reconstruction accuracy and alignment with the groundtruth.

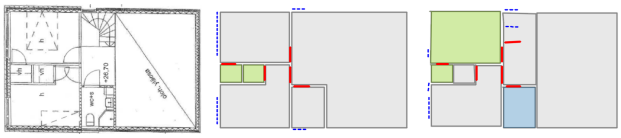

Figure 8: Limitation example, illustrating that our method may generate windows and doors inside rooms. Red line denotes a door and a dashed line denotes a window.

4.4. Ablations

We conduct extensive ablations, evaluating the effect of various components in our framework, on the Structure3D-B dataset. For simplicity of the ablations, we report only the F1 scores for three geometric criteria—Room, Corner, and Angle—with all models trained for 1,350 epochs. As a result, the reported performance in this section does not reflect the best results our model is capable of achieving.

Table 5 highlights the impact of three key components—FeatFusion, which merges polygon and image features, the learnable anchors, and the left-to-right ordering of polygons in the sequence—on floorplan reconstruction performance; qualitative results over a single sample are provided in Figure 7. Incorporating FeatFusion alone yields a notable improvement, *e.g.*, increasing Room F1 from 94.1 to 96.3. Integration of the learnable anchor further boosts performance, with Room F1 reaching 97.4. Further combining the ordering constraint on input training data gives the best overall performance, with a nearly perfect Room F1 and a 6-point improvement for angle metric, illustrating the importance of the left-to-right ordering for effectively training our model. Overall, these results demonstrate the effectiveness of our proposed architectural components. Additional ablations are reported in the supplementary.

4.5. Limitations

While our approach achieves strong performance in both geometric reconstruction and generalization, we find that performance over less prevalent semantic structures such as doors and windows can be further refined. As shown in Fig. 8, the model occasionally fails to accurately localize windows and doors, resulting in artifacts such as cross-over windows. Future work can investigate tailored architectural changes to better accommodate other element types, potentially modeling these elements separately from room entities.

5. Conclusion

In this work, we proposed to frame raster-to-vector floorplan conversion as a sequence-to-sequence task. We introduced a framework that predicts vectorized representation as labeled polygon sequence. The driving mechanism of our framework is an anchor-based autorgressive decoder, that learns to predict the next corner token conditioned on previously generated corners. Technically, our decoder introduces several architectural components, such as the integration of learnable anchors and the *FeatFusion* concatenation operation, enabling for effectively learning the generation of complex polygon sequences. Our experiments demonstrate that our approach outperforms prior work targeting similar tasks across various geometric and semantic metrics.

Raster2Seq demonstrates promising generalization performance to *in-the-wild* Internet data, representing a step towards the goal of modeling historical buildings, defined by hand-drawn floorplans. Future work can incorporate mechanisms that further improve results on out-of-distribution data, such as appearance-based augmentations. In particular, combining our system with open-vocabulary predictions could potentially allow for reconstructing the rich semantics reflected in diverse real-world floorplans. Finally, as indoor scene structure is inherently hierarchical, future work could explore how to best inject such hierarchical knowledge into the sequence-to-sequence framework—not only over the raster-to-vector task, but also for additional floorplan reconstruction related tasks.

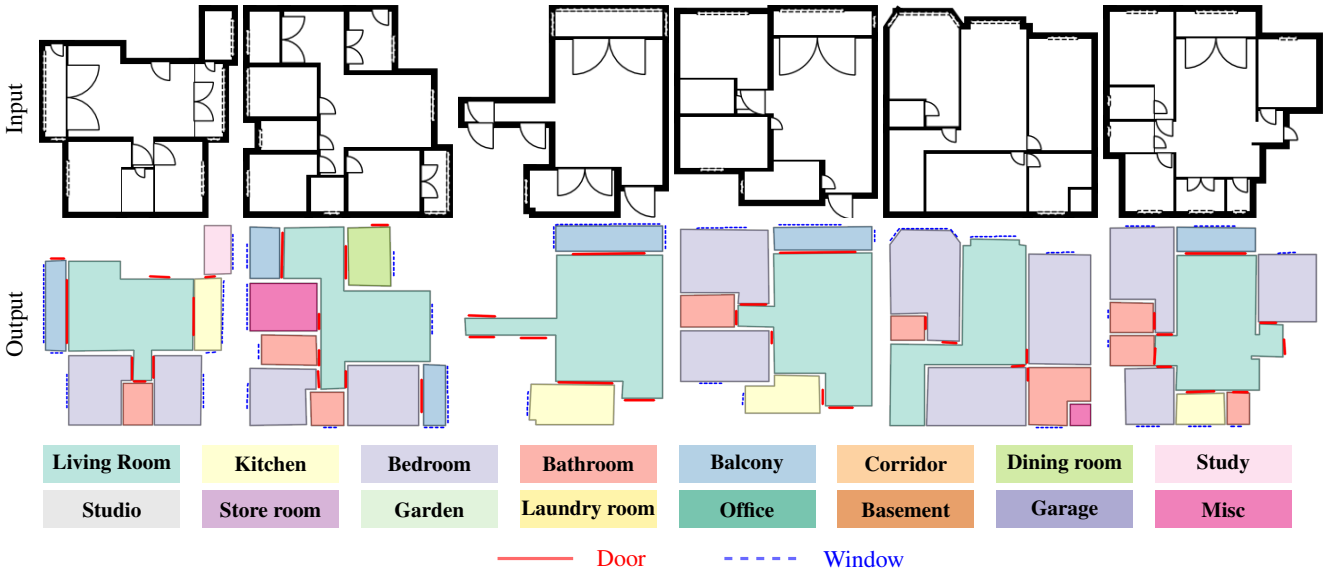

Figure 9: *Raster2Seq* reconstruction results on Structured3D.

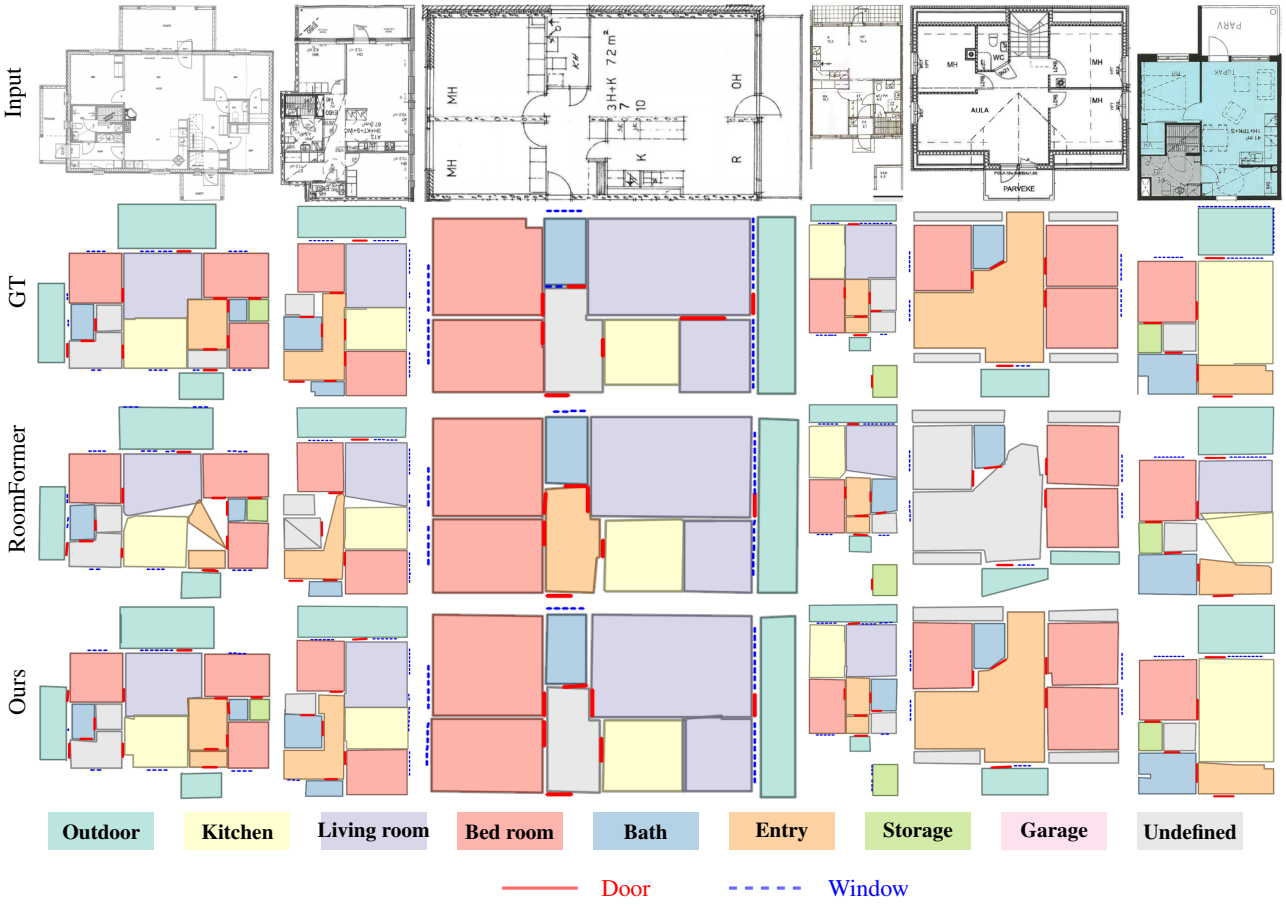

Figure 10: Qualitative results on the CubiCasa5K dataset, comparing Raster2Seq to the RoomFormer model.

Figure 11: Qualitative comparison with Raster2Graph on their dataset. Our method achieves more accurate floorplan reconstructions in comparison to their model, which often produces incomplete results.

Figure 12: Qualitative comparison with RoomFormer, over WAFFLE floorplan images (both models are trained on CubiCasa5K). As illustrated above, our model exhibits stronger generalization capabilities over the structures of real-world Internet data. Building names from left-to-right: Church of Saint James, the Greater in Rovny, Teltow Canal Power Station, Church of Saint Nicholas, Imkerhaus, Palais du Louvre, Palmer Mansion.

References

[ALKF18] ACUNA D., LING H., KAR A., FIDLER S.: Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2018), pp. 859–868. 3

[ALWD11] AHMED S., LIWICKI M., WEBER M., DENGEL A.: Improved automatic analysis of architectural floor plans. In *2011 International conference on document analysis and recognition* (2011), IEEE, pp. 864–869. 2

[AXHJ*24] AVETISYAN A., XIE C., HOWARD-JENKINS H., YANG T.-Y., AROUDJ S., PATRA S., ZHANG F., FROST D., HOLLAND L., ORME C., ET AL.: Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision* (2024), Springer, pp. 247–263. 3

[CDF23] CHEN J., DENG R., FURUKAWA Y.: Polydiffuse: Polygonal shape reconstruction via guided set diffusion models. *Advances in Neural Information Processing Systems 36* (2023), 1863–1888. 7

[CF14] CABRAL R., FURUKAWA Y.: Piecewise planar and compact floorplan reconstruction from images. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), IEEE, pp. 628–635. 2

[CLWF19] CHEN J., LIU C., WU J., FURUKAWA Y.: Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 2661–2670. 2

[CQF22] CHEN J., QIAN Y., FURUKAWA Y.: Heat: Holistic edge attention transformer for structured reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 3866–3875. 2, 3, 6, 7

[CSBC20] CORNIA M., STEFANINI M., BARALDI L., CUCCHIARA R.: Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 10578–10587. 3

[CSL*21] CHEN T., SAXENA S., LI L., FLEET D. J., HINTON G.: Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852* (2021). 3

[CSL*22] CHEN T., SAXENA S., LI L., LIN T.-Y., FLEET D. J., HINTON G. E.: A unified sequence interface for vision tasks. *Advances in Neural Information Processing Systems 35* (2022), 31333–31346. 3

[DLHAL*14] DE LAS HERAS L.-P., AHMED S., LIWICKI M., VALVENY E., SÁNCHEZ G.: Statistical segmentation and structural recognition for floor plan interpretation: Notation invariant structural element recognition. *International Journal on Document Analysis and Recognition (IJDAR) 17*, 3 (2014), 221–237. 2

[GAMAE25] GANON K., ALPER M., MIKULINSKY R., AVERBUCH-ELOR H.: Waffle: Multimodal floorplan understanding in the wild. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2025), IEEE, pp. 1488–1497. 1, 5

[HWS*24] HU S., WU W., SU R., HOU W., ZHENG L., XU B.: Raster-to-graph: Floorplan recognition via autoregressive graph prediction with an attention transformer. In *Computer Graphics Forum* (2024), vol. 43, Wiley Online Library, p. e15007. 2, 3, 5, 6, 7

[KYH*19] KALERVO A., YLIOINAS J., HÄIKIÖ M., KARHU A., KANNALA J.: Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis. In *Image Analysis: 21st Scandinavian Conference, SCIA 2019, Norrköping, Sweden, June 11–13, 2019, Proceedings 21* (2019), Springer, pp. 28–40. 1, 5, 6

[LDC*23] LIU J., DING H., CAI Z., ZHANG Y., SATZODA R. K., MADHEVAN V., MANMATHA R.: Polyformer: Referring image segmentation as sequential polygon generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023), pp. 18653–18663. 3

[LPX*19] LI M., PATIL A. G., XU K., CHAUDHURI S., KHAN O., SHAMIR A., TU C., CHEN B., COHEN-OR D., ZHANG H.: Grains: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG) 38*, 2 (2019), 1–16. 3

[LSK*15] LIU C., SCHWING A. G., KUNDU K., URTASUN R., FIDLER S.: Rent3d: Floor-plan priors for monocular layout estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3413–3421. 2

[LTL*24] LI T., TIAN Y., LI H., DENG M., HE K.: Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems 37* (2024), 56424–56445. 3

[LWF18] LIU C., WU J., FURUKAWA Y.: Floornet: A unified framework for floorplan reconstruction from 3d scans. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 201–217. 2

[LWKF17] LIU C., WU J., KOHLI P., FURUKAWA Y.: Raster-to-vector: Revisiting floorplan transformation. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 2195–2203. 2

[LXC*17] LI J., XU K., CHAUDHURI S., YUMER E., ZHANG H., GUIBAS L.: Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–14. 3

[LZM*24] LIU Y., ZHU L., MA X., YE H., GAO X., ZHENG X., SHEN S.: PolyRoom: Room-aware Transformer for Floorplan Reconstruction. In *European Conference on Computer Vision* (2024). 2, 7

[MBHRS14] MARTIN-BRUALLA R., HE Y., RUSSELL B. C., SEITZ S. M.: The 3d jigsaw puzzle: Mapping large indoor spaces. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13* (2014), Springer, pp. 1–16. 2

[MLVT10] MACÉ S., LOCTEAU H., VALVENY E., TABBONE S.: A system to detect rooms in architectural floor plan images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems* (2010), pp. 167–174. 2

[NCV*24] NGUYEN H. T., CHEN Y., VOLETI V., JAMPANI V., JIANG H.: Housecrafter: Lifting floorplans to 3d scenes with 2d diffusion model. *arXiv preprint arXiv:2406.20077* (2024). 2

[NWC*20] NARASIMHAN M., WIJMANS E., CHEN X., DARRELL T., BATRA D., PARIKH D., SINGH A.: Seeing the un-scene: Learning amodal semantic maps for room navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16* (2020), Springer, pp. 513–529. 2

[PBEPAE20] PATIL A. G., BEN-ELIEZER O., PEREL O., AVERBUCH-ELOR H.: Read: Recursive autoencoders for document layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2020), pp. 544–545. 3

[PKS*21] PASCHALIDOU D., KAR A., SHUGRINA M., KREIS K., GEIGER A., FIDLER S.: Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems 34* (2021), 12013–12026. 2

[RPG*21] RAMESH A., PAVLOV M., GOH G., GRAY S., VOSS C., RADFORD A., CHEN M., SUTSKEVER I.: Zero-shot text-to-image generation. In *International conference on machine learning* (2021), Pmlr, pp. 8821–8831. 3

[SPH*23] SHUM K. C., PANG H.-W., HUA B.-S., NGUYEN D. T., YEUNG S.-K.: Conditional 360-degree image synthesis for immersive indoor scene decoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 4478–4488. 2

[SRFL21] STEKOVIC S., RAD M., FRAUNDORFER F., LEPETIT V.: Montefloor: Extending mcts for reconstructing accurate large-scale floor plans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 16034–16043. 2, 3, 5, 6, 7

[SVL14] SUTSKEVER I., VINYALS O., LE Q. V.: Sequence to sequence learning with neural networks. *Advances in neural information processing systems 27* (2014). 3

[SWL*22] SUN J., WU W., LIU L., MIN W., ZHANG G., ZHENG L.: Wallplan: synthesizing floorplans by learning to generate wall graphs. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–14. 2, 3

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł., POLOSUKHIN I.: Attention is all you need. *Advances in neural information processing systems 30* (2017). 2, 4

[VTBE15] VINYALS O., TOSHEV A., BENGIO S., ERHAN D.: Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3156–3164. 3

[WFU15] WANG S., FIDLER S., URTASUN R.: Lost shopping! monocular localization in large indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 2695–2703. 2

[XBK*15] XU K., BA J., KIROS R., CHO K., COURVILLE A., SALAKHUDINOV R., ZEMEL R., BENGIO Y.: Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (2015), PMLR, pp. 2048–2057. 3

[XXH*24] XU H., XU J., HUANG Z., XU P., HUANG H., HU R.: Fri-net: Floorplan reconstruction via room-wise implicit representation. In *ECCV* (2024). 2, 6, 7

[YKSE23] YUE Y., KONTOGIANNI T., SCHINDLER K., ENGELMANN F.: Connecting the dots: Floorplan reconstruction using two-level queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 845–854. 2, 3, 5, 6, 7

[YXK*22] YU J., XU Y., KOH J. Y., LUONG T., BAID G., WANG Z., VASUDEVAN V., KU A., YANG Y., AYAN B. K., HUTCHINSON B., HAN W., PAREKH Z., LI X., ZHANG H., BALDRIDGE J., WU Y.: Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research* (2022). Featured Certification. URL: https://openreview.net/forum?id=AFDcYJKhND. 3

[ZHY*24] ZHANG S.-K., HUANG J., YUE L., ZHANG J.-T., LIU J.-H., LAI Y.-K., ZHANG S.-H.: Sceneexpander: Real-time scene synthesis for interactive floor plan editing. In *Proceedings of the 32nd ACM International Conference on Multimedia* (2024), pp. 6232–6240. 2

[ZLYF19] ZENG Z., LI X., YU Y. K., FU C.-W.: Deep floor plan recognition using a multi-task network with room-boundary-guided attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 9096–9104. 2, 3

[ZSL*21] ZHU X., SU W., LU L., LI B., WANG X., DAI J.: Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations* (2021). URL: https://openreview.net/forum?id=gZ9hCDWe6ke. 2, 4

[ZZL*20] ZHENG J., ZHANG J., LI J., TANG R., GAO S., ZHOU Z.: Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16* (2020), Springer, pp. 519–535. 5, 6, 7