

---

# Point or Line? Using Line-based Representation for Panoptic Symbol Spotting in CAD Drawings

---

Xingguang Wei<sup>1,2\*</sup> Haomin Wang<sup>3,2\*</sup> Shenglong Ye<sup>2</sup>  
Ruifeng Luo<sup>4,8</sup> Yanting Zhang<sup>5</sup> Lixin Gu<sup>2</sup> Jifeng Dai<sup>2,6</sup>  
Yu Qiao<sup>2</sup> Wenhai Wang<sup>2,7</sup> Hongjie Zhang<sup>2†</sup>

<sup>1</sup>University of Science and Technology of China <sup>2</sup>Shanghai AI Laboratory  
<sup>3</sup>Shanghai Jiao Tong University <sup>4</sup>Tongji University <sup>5</sup>Donghua University  
<sup>6</sup>Tsinghua University <sup>7</sup>The Chinese University of Hong Kong  
<sup>8</sup>Arcplus East China Architectural Design & Research Institute Co., Ltd.

The code is available at <https://github.com/WesKwong/VecFormer>

## Abstract

We study the task of panoptic symbol spotting, which involves identifying both individual instances of countable *things* and the semantic regions of uncountable *stuff* in computer-aided design (CAD) drawings composed of vector graphical primitives. Existing methods typically rely on image rasterization, graph construction, or point-based representation, but these approaches often suffer from high computational costs, limited generality, and loss of geometric structural information. In this paper, we propose *VecFormer*, a novel method that addresses these challenges through *line-based representation* of primitives. This design preserves the geometric continuity of the original primitive, enabling more accurate shape representation while maintaining a computation-friendly structure, making it well-suited for vector graphic understanding tasks. To further enhance prediction reliability, we introduce a *Branch Fusion Refinement* module that effectively integrates instance and semantic predictions, resolving their inconsistencies for more coherent panoptic outputs. Extensive experiments demonstrate that our method establishes a new state-of-the-art, achieving 91.1 PQ, with Stuff-PQ improved by 9.6 and 21.2 points over the second-best results under settings with and without prior information, respectively—highlighting the strong potential of line-based representation as a foundation for vector graphic understanding.

## 1 Introduction

Panoptic symbol spotting refers to the task of detecting and classifying all symbols within a CAD drawing, including both countable object instances (*e.g.*, windows, doors, furniture) and uncountable stuff regions (*e.g.*, walls, railings) [1, 2, 3]. This capability is crucial in CAD-based applications, serving as a foundation for automated design review and for generating 3D Building Information Models (BIM). However, spotting each symbol, which typically comprises a group of graphical primitives, remains highly challenging due to factors such as occlusion, clutter, appearance variations, and severe class imbalance across different symbol categories.

Earlier approaches to this problem either rasterize CAD drawings and apply image-based detection or segmentation methods [1, 4], or directly construct graph representations of CAD drawings and leverage GNN-based techniques [5, 6, 7]. However, both paradigms incur substantial computational costs, particularly when applied to large-scale CAD drawings. To better handle primitive-level

---

\*Equal Contribution

†Corresponding author: [nju.zhanghongjie@gmail.com](mailto:nju.zhanghongjie@gmail.com)

data, recent methods treat CAD drawings as sets of points corresponding to graphical primitives and leverage point cloud analysis for symbol spotting. For example, SymPoint [8] represents each primitive as a point with handcrafted features, encoding attributes such as primitive type and length. However, this manually defined representation is restricted to four predefined primitive types (line, arc, circle, and ellipse) and struggles to accommodate the more complex and diverse shapes frequently encountered in real-world CAD drawings. In contrast, the recent CADSpotting [9] forgoes explicit primitive types by densely sampling points along each primitive and representing each point using only its coordinate and color. Although this design eliminates reliance on primitive types, it lacks geometric structure and primitive-level awareness, which may hinder the model’s ability to delineate symbol boundaries, resolve overlapping symbols, and capture structural configurations essential for accurate symbol spotting.

In this work, we propose *VecFormer*, a Transformer-based [10] model built on a *line-based representation* that serves as an expressive and type-agnostic formulation for vector graphical primitives. It employs line sampling to generate a sequence of line segments along each primitive, with each line represented by its intrinsic geometric attributes and associated primitive-level statistics, forming a compact and informative feature set. Figure 1 illustrates a visual comparison of different primitive representations. SymPoint [8] encodes each primitive as a single point, which is too coarse to capture the fine-grained structures, especially for long primitives commonly found in stuff regions, leading to degraded performance. To ensure a fair comparison, we adopt the same sampling density across sampling-based methods. As shown in Figure 1, unlike CADSpotting [9] which suffers from blurred symbol boundaries, our line-based VecFormer yields results with clearer structure and better alignment to ground-truth, demonstrating higher geometric and structural fidelity. This more compact yet expressive representation is also better suited for Transformer-based architecture, which is sensitive to input sequence length. Further discussion on sequence length across different representations is detailed in Appendix C.

Additionally, inspired by OneFormer3D [11], we adopt a dual-branch Transformer decoder to guide the representation learning of vector graphical primitives, leveraging its strong multi-tasking capability to jointly model instance- and semantic-level information. To produce a more coherent panoptic output, we further propose a lightweight, training-free post-processing module, termed *Branch Fusion Refinement* (BFR), which combines predictions from the instance and semantic branches through confidence-based fusion. This refinement enhances label consistency, mitigates mask fragmentation, and improves the overall coherence of panoptic symbol predictions.

To summarize, our main contributions are:

- (1) We introduce *VecFormer*, a novel approach that leverages a type-agnostic and expressive *line-based representation* of vector graphical primitives, instead of traditional point-based methods, leading to more accurate and efficient panoptic symbol spotting.
- (2) We propose a *Branch Fusion Refinement* (BFR) module that effectively integrates instance and semantic predictions via confidence-based fusion, resolving their inconsistencies for more coherent panoptic outputs, yielding a performance gain of approximately 2 points in panoptic quality (PQ) on the FloorPlanCAD [1] dataset.
- (3) We conduct extensive experiments on the FloorPlanCAD [1] dataset, where our *VecFormer* achieves a PQ of 91.1, setting a new state-of-the-art in the panoptic symbol spotting task. Notably, it improves Stuff-PQ by 9.6 and 21.2 points over the second-best results under settings with and without prior information, respectively, underscoring its superior performance and robustness in real-world CAD applications.

## 2 Related Work

### 2.1 Panoptic Image Segmentation

Panoptic segmentation [12] aims to unify semantic [13, 14, 15, 16, 17] and instance segmentation [18, 19, 20, 21] by assigning each pixel both a class label and an instance ID, effectively covering both *things* (countable objects) and *stuff* (amorphous regions). Early approaches predominantly relied on CNN-based architectures [22, 23, 24, 25], which, while effective, often required separate branches for different segmentation tasks. Recent advancements have seen a shift towards Transformer-based models, which offer unified architectures for various segmentation tasks. Notably, Mask2Former [26]

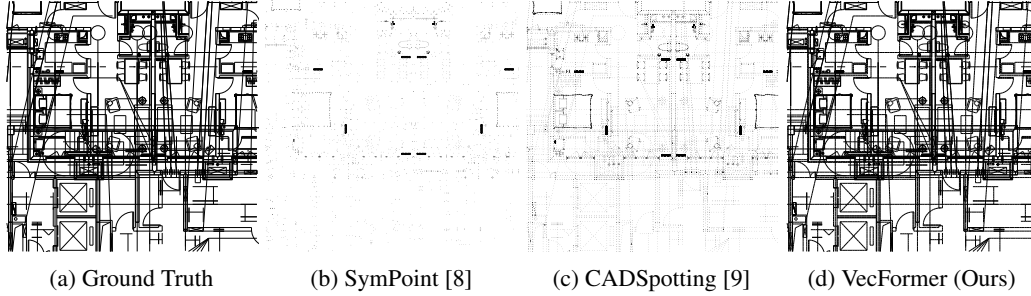


Figure 1: **Visualization of primitive representations.** Compared to the blurry visual representations of point-based methods (b, c), our line-based approach (d) more closely reflects the ground truth drawing (a). As the data is in vector format, please feel free to zoom in to observe finer differences. Additional comparisons are provided in Appendix A and Appendix B.

unifies panoptic, instance, and semantic segmentation using masked attention. SegFormer [17] improves efficiency with hierarchical encoders and lightweight decoders. OneFormer [20] further introduces task-conditioned training to jointly handle multiple segmentation tasks. Despite these successes in raster image domains, pixel-centric segmentation models face challenges when applied to vector graphics tasks, such as Panoptic Symbol Spotting in CAD drawings. Their reliance on dense pixel grids overlooks the inherent structure of vector primitives, making it difficult to capture precise geometric relationships, maintain topological consistency, and resolve overlapping symbols. These limitations hinder performance in structured, symbol-rich vector environments.

## 2.2 Panoptic Symbol Spotting

The panoptic symbol spotting task, first introduced in [1], aims to simultaneously detect and classify architectural symbols (*e.g.*, doors, windows, walls) in floor plan computer-aided design (CAD) drawings. While earlier approaches [2] primarily addressed instances of countable *things* (*e.g.*, windows, doors, tables), Fan *et al.* [1], inspired by [12], extended the task to include semantic regions of uncountable *stuff* (*e.g.*, wall, railing). To support this task, they introduced the FloorPlanCAD benchmark and proposed PanCADNet as a baseline, which combines Faster R-CNN [27] for detecting countable *things* with Graph Convolutional Networks [28] for segmenting uncountable *stuff*. Subsequently, Fan *et al.* [4] proposed CADTransformer, utilizing HRNetV2-W48 [29] and Vision Transformers [30] for primitive tokenization and embedding aggregation. Zheng *et al.* [6] adopted graph-based representations with Graph Attention Networks [31] for instance- and semantic-level predictions. Liu *et al.* [8] introduced SymPoint, exploring point-based representations with handcrafted features, later enhanced by SymPoint-V2 [32] through layer feature encoding and position-guided training. Recently, CADSpotting [9] densely samples points along primitives to generate dense point data for feature extraction and employs Sliding Window Aggregation for efficient panoptic segmentation of large-scale CAD drawings. Although point-based representations are widely adopted in existing state-of-the-art methods [8, 32, 9], they exhibit notable limitations in complex and densely annotated CAD drawings, including redundant sampling, loss of geometric continuity, and reduced ability to distinguish adjacent or overlapping symbols, as shown in Figure 1.

## 3 Method

In this section, we first describe how heterogeneous vector graphic primitives are converted into a unified line-based representation. We then present the panoptic symbol spotting framework built upon this representation. Finally, we introduce our post-processing optimization strategy, Branch Fusion Refinement. An overview of the entire pipeline is shown in Figure 2.

### 3.1 Line Sampling

Existing point-based representations [8, 32, 9] suffer from limited geometric continuity, structural expressiveness, and generality across diverse primitive types. To address these issues, we propose

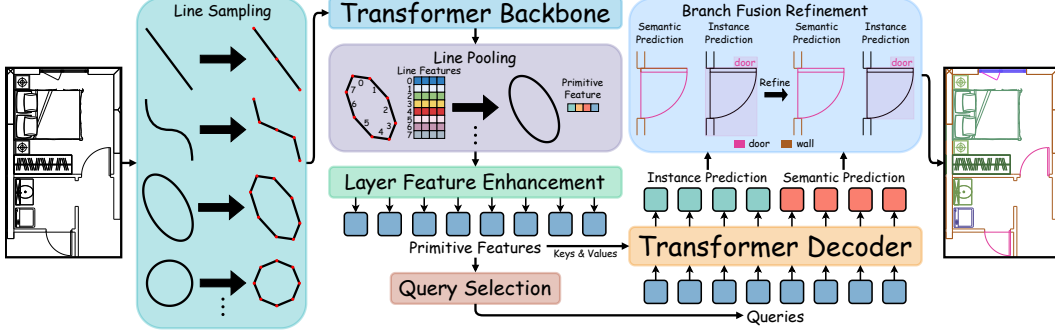


Figure 2: **Overview of VecFormer.** Given a CAD drawing, VecFormer first applies line sampling to build a line-based representation of primitives. A Transformer backbone is then used to extract line-level features, which are subsequently aggregated into primitive-level features. Next, these primitive-level features are enhanced by a Layer Feature Enhancement module and fed into a Transformer decoder for joint instance and semantic prediction. Finally, a Branch Fusion Refinement module integrates both branches to produce the final panoptic symbol spotting result.

*Line Sampling*, a line-based approximation that encodes primitives as sequences of line segments, enabling unified and geometry-preserving modeling of heterogeneous vector graphics.

Specifically, given a vector primitive with a unique identifier  $j$ , we first convert it into a vector path  $\gamma_j(t) : [0, 1] \rightarrow \mathbb{R}^2$ . Then, we perform uniform or dynamic path sampling over its parameter interval to generate a sequence of sampled points  $\mathcal{P}_j = \{\mathbf{p}_i = \gamma_j(t_i) \mid i = 1, \dots, K\}$ . Here,  $0 = t_1 < t_2 < \dots < t_K = 1$ , where the number of samples  $K$  and the sampling parameters  $t_i$  can be dynamically adjusted based on geometric features such as the length and curvature of the primitive.

For simplicity, we adopt a uniform sampling strategy defined as:  $t_i = \frac{i-1}{K-1}$ , and use a hyperparameter called the *sampling ratio* to control the number of samples  $K$ . Specifically, for line primitives, we initially set  $K = 2$ ; for all other types of primitives, we initially set  $K = 9$ . Given a sampling ratio  $\alpha_{\text{sample}}$ , we constrain the maximum allowable distance between adjacent sample points to be no greater than  $\alpha_{\text{sample}} \cdot \min(\text{width}, \text{height})$ , where *width* and *height* denote the dimensions of the CAD drawing. If this condition is violated, we iteratively increase the number of samples by setting  $K \leftarrow K + 1$  until the constraint is satisfied.

Next, adjacent sampling points are pairwise connected to construct a sequence of line segments:

$$\mathcal{L}_j = \{\mathbf{l}_i = (\mathbf{p}^s, \mathbf{p}^e) \mid \mathbf{p}^s = \mathbf{p}_i, \mathbf{p}^e = \mathbf{p}_{i+1}, i = 1, \dots, K-1\}, \quad (1)$$

which approximates the geometric features of the original primitive.

### 3.2 Panoptic Symbol Spotting via Line-based Representation

The process of panoptic symbol spotting via the line-based representation consists of three main stages: first, using a backbone to extract line-level features; second, pooling the line-level features into primitive-level features; and third, utilizing a 6-layer Transformer decoder to generate instance proposals and semantic predictions.

#### 3.2.1 Backbone

We choose Point Transformer V3 (PTv3) [33] as our backbone for feature extraction due to its excellent performance in handling unordered data with irregular spatial distributions.

Given a sampled line  $\mathbf{l}_i$ , with its starting point  $\mathbf{p}^s = (x_1, y_1)$  and endpoint  $\mathbf{p}^e = (x_2, y_2)$ , the primitive ID  $j$  indicates the primitive to which the line segment belongs, and the layer ID  $k$  indicates the layer on the CAD drawing where the primitive is located. We will now describe how to convert it into the position vector  $\text{coord}_i \in \mathbb{R}^3$  and the corresponding feature vector  $\text{feat}_i \in \mathbb{R}^C$  ( $C$  is the dimensionality of the feature vector) suitable for input to the PTv3 backbone.

**Normalization.** The initial step involves the normalization of the raw line features to a standardized range of  $[-0.5, 0.5]$ .



For the starting point  $\mathbf{p}^s = (x_1, y_1)$ , the normalization is performed as follows:

$$\mathbf{p}^s = (x_1, y_1) = \left( \frac{x_1 - x_{\text{origin}}}{\text{width}} - 0.5, \frac{y_1 - y_{\text{origin}}}{\text{height}} - 0.5 \right). \quad (2)$$

In this formulation, the coordinates  $(x_{\text{origin}}, y_{\text{origin}})$  denote the origin of the coordinate system employed in the CAD drawing. The terms *width* and *height* denote the dimensions of the CAD drawing. The normalization for the endpoint  $\mathbf{p}^e$  is achieved through an analogous transformation.

For the normalization of layer ID, let  $k_{\min}$  and  $k_{\max}$  represent the minimum and maximum layer ID values observed within the CAD drawing, respectively. The normalized layer ID  $k$  is then calculated as:

$$k = \frac{k - k_{\min}}{k_{\max} - k_{\min}} - 0.5. \quad (3)$$

**Line Position.** To simultaneously capture both the position information and the layer information, we use the midpoint of the line  $(c_x, c_y)$  for the first two dimensions and the layer ID  $k$  for the third dimension:

$$\text{coord}_i = (x_i, y_i, z_i) = (c_x, c_y, \text{id}) = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}, k \right). \quad (4)$$

**Line Feature.** We set the dimensionality  $C = 7$ , and define the line feature  $\text{feat}_i \in \mathbb{R}^7$  as:

$$\text{feat}_i = (l, d_x, d_y, c_x, c_y, c_x^p, c_y^p). \quad (5)$$

Here,  $l = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$  represents the length of the line. The terms  $d_x = (x_1 - x_2)/l$  and  $d_y = (y_1 - y_2)/l$  denote the unit vectors for displacement in the  $x$  and  $y$  directions, respectively. The coordinates  $(c_x, c_y)$  specify the midpoint of the line. These features are chosen because any point on the line can be expressed as:  $(x, y) = (c_x + td_x, c_y + td_y)$ ,  $t \in [-\frac{l}{2}, \frac{l}{2}]$ , which provides a parametric representation of the line segment based on its center and unit vector.

Furthermore,  $(c_x^p, c_y^p)$  indicates the geometric centroid of the primitive. This centroid is determined by calculating the average of the midpoint coordinates from all lines sampled within the primitive  $j$  to which the specific line belongs:

$$(c_x^p, c_y^p) = \left( \frac{\sum_{\mathbf{l}_i \in \mathcal{L}_j} c_x}{|\mathcal{L}_j|}, \frac{\sum_{\mathbf{l}_i \in \mathcal{L}_j} c_y}{|\mathcal{L}_j|} \right). \quad (6)$$

### 3.2.2 Line Pooling

To obtain primitive-level features, we apply *Line Pooling*, which combines max and average pooling over line-level features within each primitive, effectively preserving geometric information and enhancing feature richness.

For each primitive  $j$ , line features  $\mathbf{f}_i \in \mathbb{R}^C$  from  $\mathbf{l}_i \in \mathcal{L}_j$  are aggregated via both max and average pooling, whose results are summed to produce the final primitive feature  $\mathbf{F}_j \in \mathbb{R}^C$ :

$$\mathbf{F}_j = \mathbf{F}_j^{\max} + \mathbf{F}_j^{\text{avg}} = \max_{\mathbf{l}_i \in \mathcal{L}_j} \mathbf{f}_i + \frac{1}{|\mathcal{L}_j|} \sum_{\mathbf{l}_i \in \mathcal{L}_j} \mathbf{f}_i. \quad (7)$$

### 3.2.3 Layer Feature Enhancement

Inspired by SymPoint-V2 [32], we adopt a *Layer Feature Enhancement* (LFE) module in our method. Specifically, we aggregate the features of primitives within the same layer using average pooling, max pooling, and attention pooling, and fuse the resulting layer-level context back into each primitive feature. This fusion enhances the model’s ability to capture intra-layer contextual dependencies and improves the semantic discrimination of similar primitives.

### 3.2.4 Query Decoder

Motivated by OneFormer3D [11], we initialize the queries using a *Query Selection* strategy, which is widely adopted in state-of-the-art 2D object detection and instance segmentation methods [34, 35, 36].

Subsequently, a six-layer Transformer decoder performs self-attention on the queries and cross-attention with key-value pairs derived from primitive features. The decoder outputs are then passed to an *Instance Branch* for generating instance proposals and a *Semantic Branch* for producing semantic predictions.

**Query Selection.** With the primitive features  $\mathcal{L} \in \mathbb{R}^{N \times C}$  derived from the previous stage, where  $N$  denotes the number of primitives and  $C$  is the dimensionality of each feature vector, the Query Selection strategy randomly selects a proportion  $\alpha_{\text{select}} \in [0, 1]$  of the primitive features to initialize the queries  $\mathcal{Q} \in \mathbb{R}^{M \times C}$ , with  $M = \alpha_{\text{select}} \cdot N$  representing the number of queries. Following the configuration in OneFormer3D [11], we set  $\alpha_{\text{select}} = 0.5$  during training to reduce computational cost, which also serves as a form of data augmentation. During inference, we set  $\alpha_{\text{select}} = 1.0$  in order to preserve the complete information of the CAD drawings.

**Instance Branch.** In this branch, each query embedding is mapped to a  $K + 1$  dimensional space as class label logits, where  $K$  denotes the number of classes and an extra  $+1$  for the background predictions. Simultaneously, we use an einsum operation between the query embedding and the primitive features to generate the instance mask.

**Semantic Branch.** This branch aims to produce dense, per-primitive semantic predictions. We project the output queries from the decoder into a  $K + 1$  dimensional space as semantic logits. The prediction for each query is assigned to the primitive that was selected to initialize the query during the Query Selection process, thereby providing semantic label of each primitive.

### 3.2.5 Loss Function

To jointly optimize instance and semantic predictions, we adopt a composite loss function:

$$L_{\text{total}} = \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{bce}} L_{\text{bce}} + \lambda_{\text{dice}} L_{\text{dice}} + \lambda_{\text{sem}} L_{\text{sem}}. \quad (8)$$

Here,  $L_{\text{cls}}$  is a cross-entropy loss for instance classification,  $L_{\text{bce}}$  and  $L_{\text{dice}}$  [37, 38] are used for instance mask prediction to balance foreground-background accuracy and mask overlap, respectively.  $L_{\text{sem}}$  denotes the cross-entropy loss for semantic segmentation. The weights  $\lambda_{\text{cls}}, \lambda_{\text{bce}}, \lambda_{\text{dice}}, \lambda_{\text{sem}}$  control the influence of each term.

### 3.3 Branch Fusion Refinement

To effectively integrate information from both the Semantic Branch and the Instance Branch, we propose a post-processing strategy named *Branch Fusion Refinement* (BFR). This method consists of three steps: *Overriding*, *Voting*, and *Remasking*.

**Overriding.** This step is primarily designed to resolve conflicts between instance predictions and semantic predictions at the per-primitive level. Given a primitive  $p_i$ , the semantic branch outputs a semantic label  $l_{\text{sem}}(p_i) \in \{1, \dots, K + 1\}$  and a corresponding confidence score  $s_{\text{sem}}(p_i) \in [0, 1]$ . Meanwhile, if  $p_i$  is assigned to  $N$  instance proposals, each such proposal provides an instance label  $l_{\text{inst}}^j \in \{1, \dots, K + 1\}$  and an associated confidence score  $s_{\text{inst}}^j \in [0, 1]$ , where  $j \in \{1, \dots, N\}$  indexes the proposals that include  $p_i$ .

To resolve the conflict, we compare the semantic and instance confidence scores. If the highest instance score for  $p_i$  is greater than the semantic score, i.e.,  $\max_j s_{\text{inst}}^j(p_i) > s_{\text{sem}}(p_i)$ , then the semantic prediction for  $p_i$  is overridden by the instance label and score of the highest-confidence proposal:

$$l_{\text{sem}}^{\text{refined}}(p_i) = l_{\text{inst}}^{j^*}(p_i), \quad s_{\text{sem}}^{\text{refined}}(p_i) = s_{\text{inst}}^{j^*}(p_i), \quad \text{where } j^* = \arg \max_{j \in \{1, \dots, N\}} s_{\text{inst}}^j(p_i). \quad (9)$$

If no instance score exceeds the semantic score, the original semantic prediction is retained.

**Voting.** Given an instance proposal that contains  $M$  primitives, its instance label is refined based on the most frequently occurring semantic class among those primitives. Formally, the instance label  $l_{\text{inst}}$  for this proposal is refined as:

$$l_{\text{inst}} = \arg \max_{k \in \{1, \dots, K\}} \sum_{i=1}^M \mathbb{I}(l_{\text{sem}}(p_i) = k), \quad (10)$$

where  $\mathbb{I}(\cdot)$  is the indicator function that returns 1 if the condition is true and 0 otherwise. This majority voting strategy ensures that the instance label aligns with the dominant semantic context of its constituent primitives.

**Remasking.** For each primitive  $p_i$ , if it belongs to an instance mask  $\mathcal{M}_{\text{inst}}$ , but its semantic label  $l_{\text{sem}}(p_i)$  disagrees with the instance’s majority-voted label  $l_{\text{inst}}$ , it is removed from the mask:

$$p_i \in \mathcal{M}_{\text{inst}} \quad \text{and} \quad l_{\text{sem}}(p_i) \neq l_{\text{inst}} \quad \Rightarrow \quad p_i \notin \mathcal{M}_{\text{inst}}. \quad (11)$$

This operation effectively eliminates label contamination in the mask caused by prediction inconsistencies, thereby improving the purity and semantic consistency of the instance segmentation results.

## 4 Experiments

### 4.1 Dataset and Metrics

FloorPlanCAD [1] dataset consists of 11,602 diverse CAD drawings of various floor plans, each annotated with fine-grained semantic and instance labels. We follow the official data split, which includes 6,965 samples for training, 810 for validation, and 3,827 for testing. The annotations cover 30 thing classes and 5 stuff classes.

Following [1, 4], we use the Panoptic Quality (PQ) defined on vector graphics as our main metric to evaluate the performance of panoptic symbol spotting. The Panoptic Quality (PQ) serves as a comprehensive metric that simultaneously evaluates the recognition correctness and segmentation accuracy of symbol-level predictions in vector graphics. A graphical primitive is denoted as  $e = (l, z)$ , where  $l$  is the semantic label,  $z$  is the instance index. A symbol is represented by a collection of primitives and is defined as  $s = \{e_i \in J \mid l = l_i, z = z_i\}$ , where  $J$  is a set of primitives. The metric is defined as:

$$PQ = \frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \times \frac{\sum_{(s_p, s_g) \in TP} \text{IoU}(s_p, s_g)}{|TP|} = \frac{\sum_{(s_p, s_g) \in TP} \text{IoU}(s_p, s_g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}. \quad (12)$$

Here,  $s_p = (l_p, z_p)$  is the predicted symbol, and  $s_g = (l_g, z_g)$  is the ground truth symbol.  $|TP|$ ,  $|FP|$ , and  $|FN|$  represent the number of true positives, false positives, and false negatives, respectively. A predicted symbol is matched to a ground truth symbol if and only if  $l_p = l_g$  and  $\text{IoU}(s_p, s_g) > 0.5$ . The IoU between two symbols is defined as:

$$\text{IoU}(s_p, s_g) = \frac{\sum_{e_i \in s_p \cap s_g} \log(1 + L(e_i))}{\sum_{e_j \in s_p \cup s_g} \log(1 + L(e_j))}, \quad (13)$$

where  $L(e)$  denotes the length of a geometric primitive  $e$ .

### 4.2 Implementation Details

During training, we adopt the AdamW [39] optimizer with a weight decay of 0.05. The initial learning rate is set to 0.0001, with a warm-up ratio of 0.05, followed by cosine decay applied over 20% of the total training epochs. The model is trained for 500 epochs with a batch size of 2 per GPU on 8 NVIDIA A100 GPUs. To improve model generalization, we apply several data augmentation strategies during training, including random horizontal and vertical flips with a probability of 0.5, random rotations, random scaling within the range [0.8, 1.2], and random translations up to 10% of the CAD drawing size along both axes. Furthermore, we empirically set the loss weight as  $\lambda_{\text{cls}} : \lambda_{\text{bce}} : \lambda_{\text{dice}} : \lambda_{\text{sem}} = 2.5 : 5.0 : 5.0 : 5.0$ .

### 4.3 Quantitative Evaluation

**Panoptic Symbol Spotting.** We compare our method with existing approaches on FloorPlanCAD [1] for panoptic symbol spotting, as shown in Table 1a. Our method achieves the highest Panoptic Quality (PQ) across *Total*, *Thing*, and *Stuff* categories, demonstrating superior and more balanced performance. Existing methods tend to perform better on *Thing* than *Stuff* categories, revealing an imbalance in recognition. For example, SymPoint [8] scores 84.1 in Thing-PQ but only 48.2 in

Table 1: Quantitative evaluation results

(a) Panoptic symbol spotting results on FloorPlanCAD [1] dataset. A dash (–) indicates that the method does not support this setting or that the result is not reported in the original paper.

| Method                  | w/o Prior          |                    |                     | w/ Prior           |                    |                    |
|-------------------------|--------------------|--------------------|---------------------|--------------------|--------------------|--------------------|
|                         | PQ                 | PQ <sub>th</sub>   | PQ <sub>st</sub>    | PQ                 | PQ <sub>th</sub>   | PQ <sub>st</sub>   |
| PanCADNet [1]           | 59.5               | 65.6               | 58.7                | -                  | -                  | -                  |
| CADTransformer [4]      | 68.9               | 78.5               | 58.6                | -                  | -                  | -                  |
| GAT-CADNet [6]          | 73.7               | -                  | -                   | -                  | -                  | -                  |
| SymPoint [8]            | 83.3               | 84.1               | 48.2                | -                  | -                  | -                  |
| SymPoint-V2 [32]        | 83.2               | 85.8               | 49.3                | 90.1               | 90.8               | 80.8               |
| CADSpotting [9]         | -                  | -                  | -                   | 88.9               | 89.7               | 80.6               |
| DPSS [40]               | 86.2               | 88.0               | 64.7                | 89.5               | 90.4               | 79.7               |
| <b>VecFormer (Ours)</b> | <b>88.4</b> (+2.2) | <b>90.9</b> (+2.9) | <b>85.9</b> (+21.2) | <b>91.1</b> (+1.0) | <b>91.8</b> (+1.0) | <b>90.4</b> (+9.6) |

(b) Primitive-level semantic quality. wF1: length-weighted F1.

| Method | GAT-CADNet [6] | SymPoint [8] | SymPoint-V2 [32] | <b>VecFormer (Ours)</b> |
|--------|----------------|--------------|------------------|-------------------------|
| F1     | 85.0           | 86.8         | 89.5             | <b>93.8</b> (+4.3)      |
| wF1    | 82.3           | 85.5         | 88.3             | <b>92.2</b> (+3.9)      |

Stuff-PQ. In contrast, our method achieves more balanced results and shows a marked advantage in the *Stuff* classes, in particular, surpassing the current state-of-the-art method, SymPoint-V2 [32], by 9.6 in Stuff-PQ.

To reflect real-world conditions where detailed annotations (such as layers) are often unavailable, we evaluate current mainstream methods without using prior information. As shown in Table 1a, existing state-of-the-art methods exhibit strong reliance on prior, particularly for *Stuff* categories. Specifically, SymPoint-V2 [32] and DPSS [40] suffer significant performance drops in Stuff-PQ when evaluated without prior, decreasing by 31.5 and 15 points, respectively. In contrast, our method VecFormer consistent performance across both settings by using primitive IDs instead of layer IDs as  $z$ -coordinate of position vector, i.e., use  $\mathbf{coord}_i = (c_x, c_y, j)$ , but not  $\mathbf{coord}_i = (c_x, c_y, k)$  described in subsection 3.1. As shown in Table 1a, VecFormer achieves a PQ of 88.4 and 90.9 in the *Total* and *Thing* categories, outperforming the second-best methods by 2.2 and 2.9, respectively. For the more challenging *Stuff* category, VecFormer demonstrates particularly strong performance, achieving a PQ of 85.9 with a notable gain of 21.2 over the second-best result.

These results demonstrate that VecFormer maintains excellent generalization and robustness even without relying on prior information, making it more suitable for practical deployment in real-world CAD scenarios.

**Primitive-Level Semantic Quality.** We assess the model’s semantic prediction performance for each graphical primitive by computing the F1 and wF1 score. As summarised in Table 1b, our VecFormer consistently surpasses all prior methods, achieving an improvement of 4.3 in F1 and 3.9 in wF1, compared to SymPoint-V2 [32]. The qualitative results are shown in Figure 3. For more qualitative studies, please refer to Appendix D.

#### 4.4 Ablation Studies

**Impact of Sampling Strategy.** As shown in Table 2a, line sampling outperforms point sampling in both settings—with and without prior information. The point sampling variant omits line-specific features  $(l, d_x, d_y)$ , leading to inferior results, confirming the superiority of line-based representations for vector graphic understanding.

**Choice of Sampling Ratio.** As shown in Table 2b, reducing the sampling ratio  $\alpha_{\text{sample}}$  from 0.1 to 0.01 steadily improves performance, with the best PQ (91.1) at  $\alpha_{\text{sample}} = 0.01$ —also yielding peak PQ<sub>th</sub> and PQ<sub>st</sub> scores. Further reduction to  $\alpha_{\text{sample}} = 0.005$  slightly degrades performance while

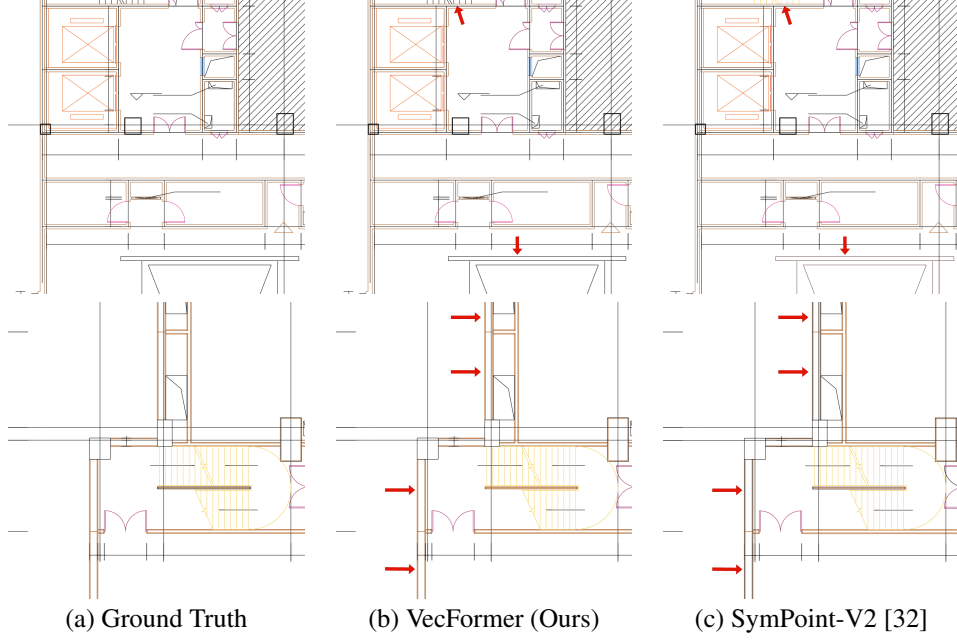


Figure 3: Qualitative comparison of primitive-level semantic quality between VecFormer and SymPoint-V2. Each row shows a representative example, with (a) Ground Truth annotations, (b) predictions from our VecFormer, and (c) predictions from SymPoint-V2. As shown, VecFormer provides more accurate and consistent semantic predictions across various graphical primitives.

increasing computational cost, making  $\alpha_{\text{sample}} = 0.01$  the optimal trade-off between accuracy and efficiency.

Table 2: Ablation studies on sampling strategy, sampling ratio, BFR, and prior information.

| (a) Ablation studies on sampling strategy |          |             |                  |                  | (b) Ablation studies on sampling ratio |             |                  |                  |
|---|----------|-------------|------------------|------------------|--|-------------|------------------|------------------|
| Prior                                     | Strategy | PQ          | PQ <sub>th</sub> | PQ <sub>st</sub> | Ratio                                  | PQ          | PQ <sub>th</sub> | PQ <sub>st</sub> |
| w/o                                       | Point    | 87.8        | 89.9             | 85.4             | 0.1                                    | 90.4        | 91.1             | 89.7             |
|   | Line     | <b>88.4</b> | <b>90.9</b>      | <b>85.9</b>      | 0.05                                   | 90.6        | 91.3             | 89.7             |
| w/  | Point    | 89.8        | 90.0             | 89.5             | 0.01                                   | <b>91.1</b> | <b>91.8</b>      | <b>90.4</b>      |
|   | Line     | <b>91.1</b> | <b>91.8</b>      | <b>90.4</b>      | 0.005                                  | 90.4        | 91.0             | 89.8             |

| (c) Ablation studies on BFR |             |                  |                  | (d) Ablation studies of prior information |       |     |             |                  |                  |
|-----------------------------|-------------|------------------|------------------|---|-------|-----|-------------|------------------|------------------|
| Method                      | PQ          | PQ <sub>th</sub> | PQ <sub>st</sub> | Base                                      | Layer | LFE | PQ          | PQ <sub>th</sub> | PQ <sub>st</sub> |
| w/o BFR                     | 89.2        | 90.4             | 88.1             | ✓   |       |     | 88.4        | 90.9             | 85.9             |
| w/ BFR                      | <b>91.1</b> | <b>91.8</b>      | <b>90.4</b>      | ✓   | ✓     |     | 90.2        | 91.7             | 88.4             |
| Gain                        | (+1.9)      | (+1.4)           | (+2.3)           | ✓   | ✓     | ✓   | <b>91.1</b> | <b>91.8</b>      | <b>90.4</b>      |

**Effects of the Branch Fusion Refinement Strategy.** We conduct controlled experiments to evaluate the effectiveness of the proposed Branch Fusion Refinement (BFR) strategy. As shown in Table 2c, incorporating BFR significantly boosts performance across all metrics, demonstrating its essential role in improving prediction accuracy and robustness.

**Effects of Prior Information.** As shown in Table 2d, replacing the primitive ID  $j$  with the layer ID  $k$  in the position vector boosts PQ from 88.4 to 90.2, highlighting the value of layer priors. Adding the Layer Feature Enhancement (LFE) module further improves PQ to 91.1, demonstrating that structural priors and LFE together enhance geometric understanding.

## 5 Conclusions

We present *VecFormer*, a novel method that employs an expressive and type-agnostic line-based representation to enhance feature learning for vector graphical primitives by preserving geometric continuity and structural relationships, which are critical for symbol-rich vector graphics. To unify instance- and semantic-level predictions from a dual-branch Transformer decoder, we propose the *Branch Fusion Refinement* (BFR) module, which resolves inconsistencies and improves panoptic quality. A current limitation lies in the use of uniform line sampling for simplicity, which may underperform in regions of high geometric complexity. Future work will explore a geometry-aware dynamic sampling strategy to better adapt to diverse structural patterns in vector graphics. To the best of our knowledge, the proposed method does not pose any identifiable negative societal risks.

## Acknowledgements

This work was supported by Shanghai Artificial Intelligence Laboratory, the National Key R&D Program of China (No. 2022ZD0161301), the Shanghai Committee of Science and Technology (No. 22YF1461500), and the National Natural Science Foundation of China (No. 62206046).

## References

- [1] Zhiwen Fan, Lingjie Zhu, Honghua Li, Xiaohao Chen, Siyu Zhu, and Ping Tan. Floorplancad: A large-scale cad drawing dataset for panoptic symbol spotting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10128–10137, 2021.
- [2] Alireza Rezvanifar, Melissa Cote, and Alexandra Branzan Albu. Symbol spotting for architectural drawings: state-of-the-art and new industry-driven developments. *IPSN Transactions on Computer Vision and Applications*, 11:1–22, 2019.
- [3] Alireza Rezvanifar, Melissa Cote, and Alexandra Branzan Albu. Symbol spotting on digital architectural floor plans using a deep learning-based framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 568–569, 2020.
- [4] Zhiwen Fan, Tianlong Chen, Peihao Wang, and Zhangyang Wang. Cadtransformer: Panoptic symbol spotting transformer for cad drawings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10986–10996, 2022.
- [5] Xinyang Jiang, Lu Liu, Caihua Shan, Yifei Shen, Xuanyi Dong, and Dongsheng Li. Recognizing vector graphics without rasterization. *Advances in Neural Information Processing Systems*, 34:24569–24580, 2021.
- [6] Zhaohua Zheng, Jianfang Li, Lingjie Zhu, Honghua Li, Frank Petzold, and Ping Tan. Gat-cadnet: Graph attention network for panoptic symbol spotting in cad drawings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11747–11756, 2022.
- [7] Bingchen Yang, Haiyong Jiang, Hao Pan, and Jun Xiao. Vectorfloorseg: Two-stream graph attention network for vectorized roughcast floorplan segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1358–1367, 2023.
- [8] Wenlong Liu, Tianyu Yang, Yuhang Wang, Qizhi Yu, and Lei Zhang. Symbol as points: Panoptic symbol spotting via point-based representation. *arXiv preprint arXiv:2401.10556*, 2024.
- [9] Jiazuo Mu, Fuyi Yang, Yanshun Zhang, Junxiong Zhang, Yongjian Luo, Lan Xu, Yujiao Shi, Jingyi Yu, and Yingliang Zhang. Cadspotting: Robust panoptic symbol spotting on large-scale cad drawings. *arXiv preprint arXiv:2412.07377*, 2024.
- [10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [11] Maxim Kolodiaznyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20943–20953, 2024.

- [12] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019.
- [13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.
- [15] Huikai Wu, Junge Zhang, Kaiqi Huang, Kongming Liang, and Yizhou Yu. Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation. *arXiv preprint arXiv:1903.11816*, 2019.
- [16] Liang-Chieh Chen. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- [17] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.
- [18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
- [19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [20] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. One-former: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2989–2998, 2023.
- [21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [22] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019.
- [23] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.
- [24] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7026–7035, 2019.
- [25] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic segmentation. *arXiv preprint arXiv:2011.11675*, 2020.
- [26] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.
- [27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

- [28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [29] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.
- [30] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [32] Wenlong Liu, Tianyu Yang, Qizhi Yu, and Lei Zhang. Sympoint revolutionized: Boosting panoptic symbol spotting with layer feature enhancement. *arXiv preprint arXiv:2407.01928*, 2024.
- [33] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024.
- [34] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.
- [35] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.
- [36] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3041–3050, 2023.
- [37] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.
- [38] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *Proceedings of the European conference on computer vision (ECCV)*, pages 562–578, 2018.
- [39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [40] Ruifeng Luo, Zhengjie Liu, Tianxiao Cheng, Jie Wang, Tongjie Wang, Xingguang Wei, Haomin Wang, YanPeng Li, Fu Chai, Fei Cheng, et al. Archcad-400k: An open large-scale architectural cad dataset and new baseline for panoptic symbol spotting. *arXiv preprint arXiv:2503.22346*, 2025.



## Appendix

### A Detailed Visual Comparisons across Different Representations.

We present additional fine-grained visualizations to facilitate a more detailed comparison of different representations. As illustrated in Figure 4, our proposed line-based representation demonstrates closer visual alignment with the ground truth than point-based methods (e.g., SymPoint [8], CADSpotting [9]), effectively preserving geometric continuity and structural integrity across a variety of primitive types.

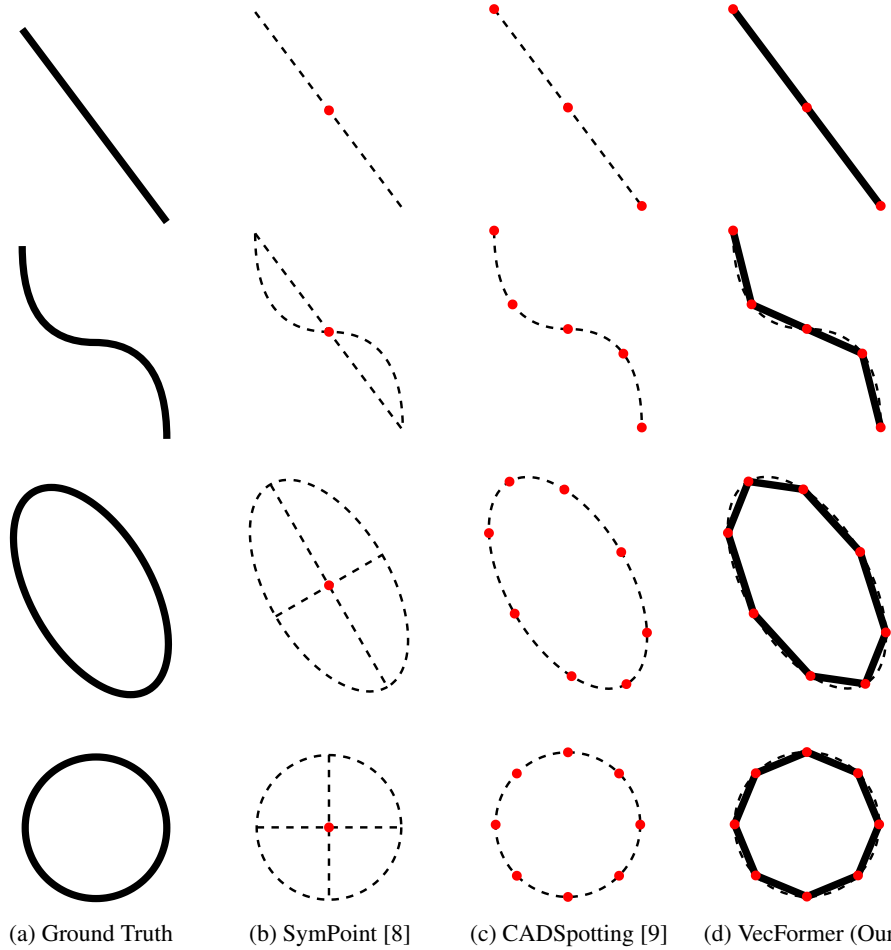


Figure 4: Visualization of how different representations perform on different primitives.

## B Visual Comparison under Varying Sampling Ratios

To further investigate the impact of sampling density on representation quality, we conduct a comparative analysis across different representations under varying sampling ratios  $\alpha_{\text{sample}}$ . As shown in Figure 5, our line-based representation consistently maintains higher geometric fidelity and structural coherence, even under lower sampling densities. In contrast, point-based representations tend to suffer from fragmentation and loss of continuity as the sampling ratio decreases.

These visual results highlight the robustness of our approach in preserving essential geometric and topological features, suggesting its suitability for vector graphics tasks where structural integrity is critical under constrained sampling conditions.

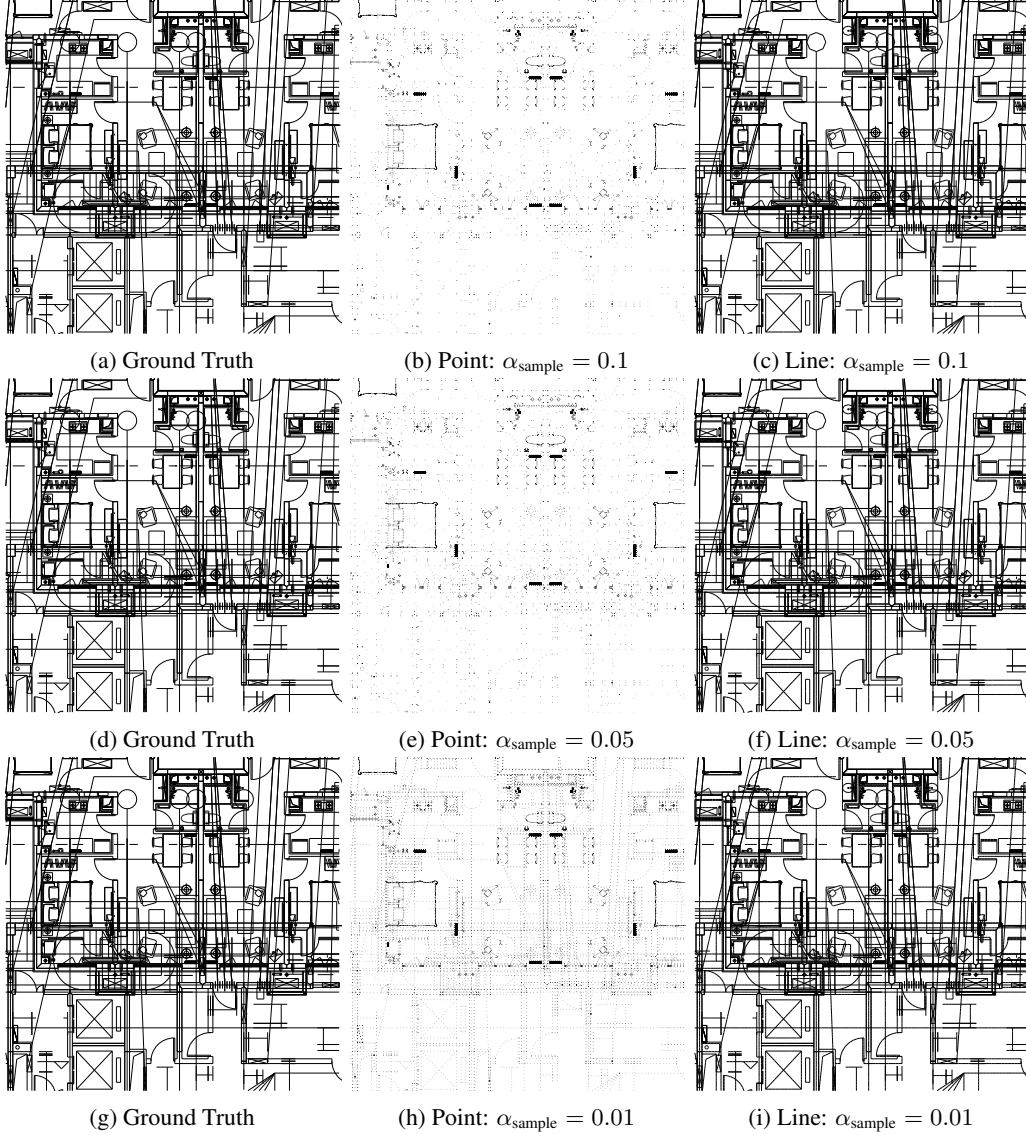


Figure 5: Visual comparison of the effects of varying sampling ratios on different representations. Since the data is in vector format, zooming in allows for a detailed examination of the differences between representations.

## C Sequence Length Analysis of Point- and Line-based Representations

This section analyzes the differences in sequence length between point-based and line-based representations on FloorPlanCAD [1] dataset.

We begin by configuring the line-based representation with a sampling ratio of  $\alpha_{\text{sample}} = 0.01$ , consistent with our experimental setup. For the point-based representation, we set  $\alpha_{\text{sample}} = 0.001$ , which yields a similar sampling density to that used in CADSpotting [9], although the sampling strategies differ. As illustrated in Figure 6, this setting results in CADSpotting, the point-based method, producing sequences that are approximately 8 times longer than our line-based counterpart. Despite the significantly shorter sequence length, our method achieves higher Panoptic Quality (PQ), as demonstrated in the main results (Table 1a).

To ensure a fair comparison, we further evaluate both representations under the same sampling ratio of  $\alpha_{\text{sample}} = 0.01$ . Even in this setting, the line-based representation yields approximately 15% fewer tokens than the point-based representation. Moreover, ablation results in Table 2a confirm that our approach not only reduces sequence length but also achieves superior performance.

These findings underscore the efficiency and representational strength of the line-based approach: by encoding primitives through fewer yet structurally meaningful elements, it preserves geometric fidelity while enhancing learning effectiveness. This compact, structure-aware design leads to more accurate segmentation and improved overall performance, making line-based representation a more effective and scalable solution for vector graphic understanding.

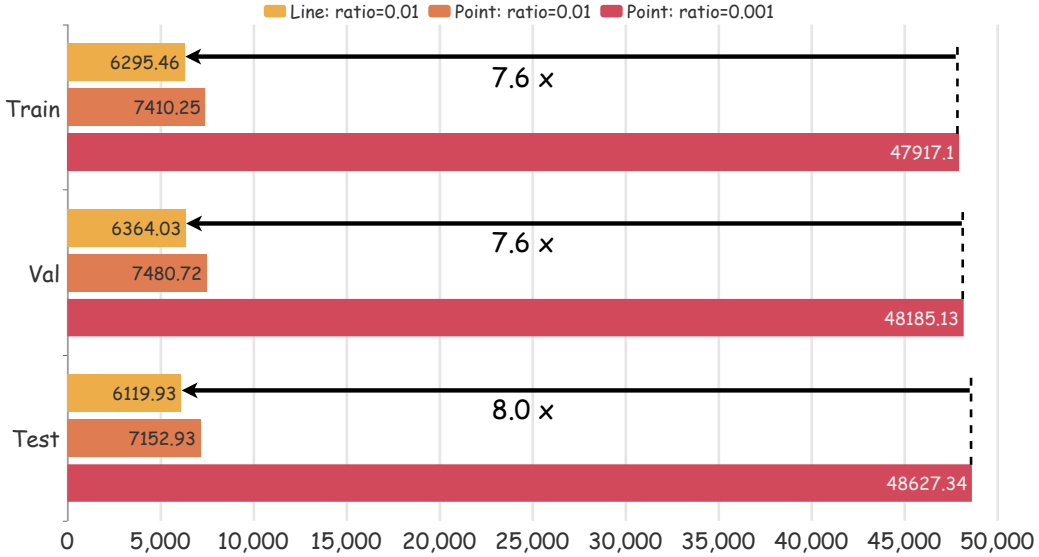


Figure 6: Comparison of sequence lengths between point-based and line-based representations on FloorPlanCAD [1] dataset. The vertical axis indicates different dataset splits, while the horizontal axis represents the average sequence length of each representation across these splits.

## D Additional Qualitative Evaluation

This section provides additional qualitative results through visualizations. The color scheme for each category is defined in Figure 7, and further examples are illustrated in Figure 8, Figure 9, and Figure 10.

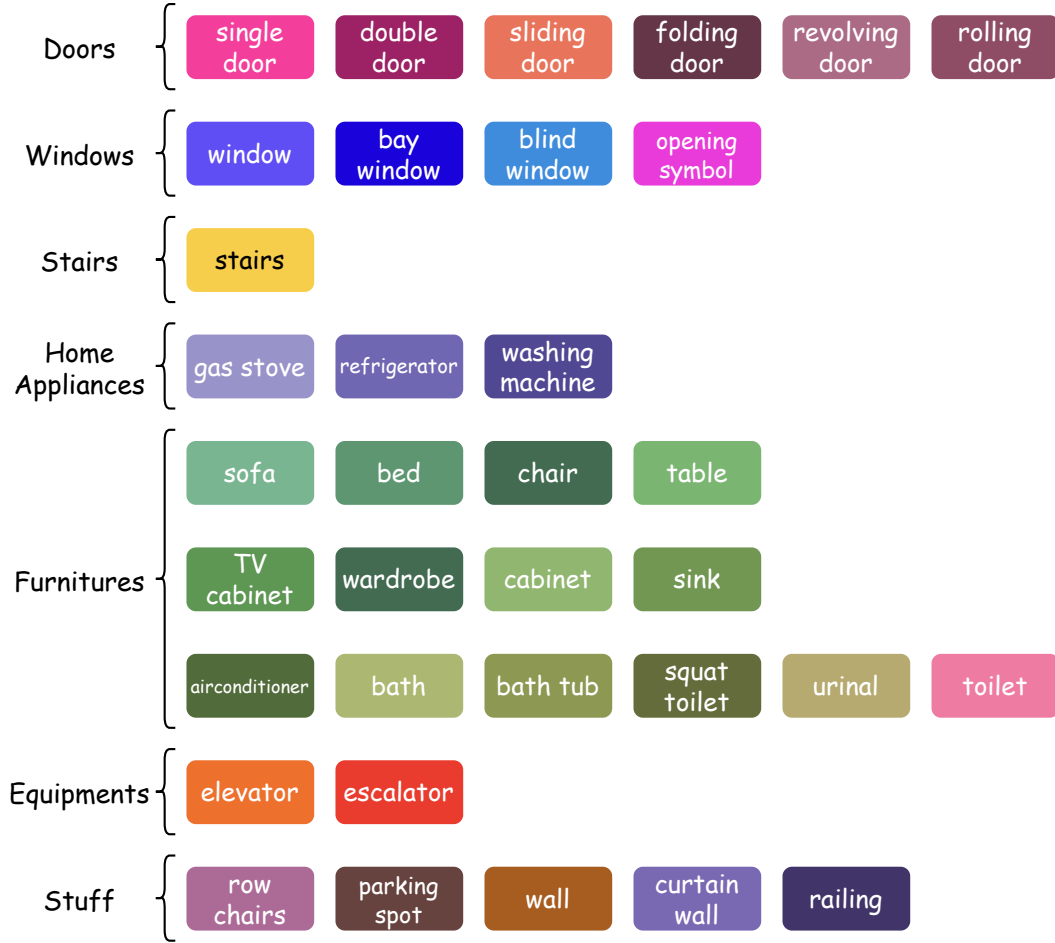


Figure 7: Color map used for category visualization, adapted from [8].

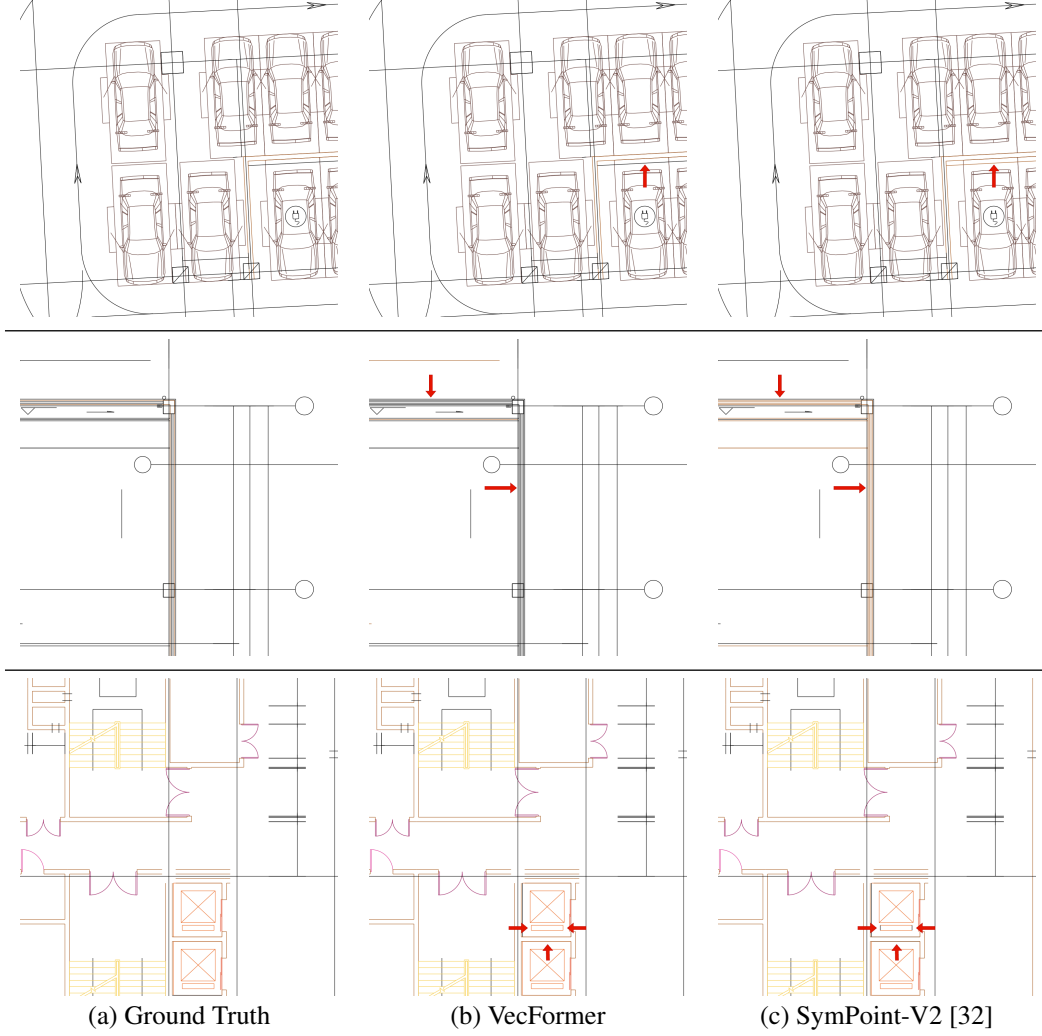


Figure 8: More qualitative comparison of primitive-level semantic quality between VecFormer and SymPoint-V2 [32]. Each row shows a representative example, with (a) Ground Truth annotations, (b) predictions from our VecFormer, and (c) predictions from SymPoint-V2 [32]. As shown, VecFormer provides more accurate and consistent semantic predictions across various graphical primitives.

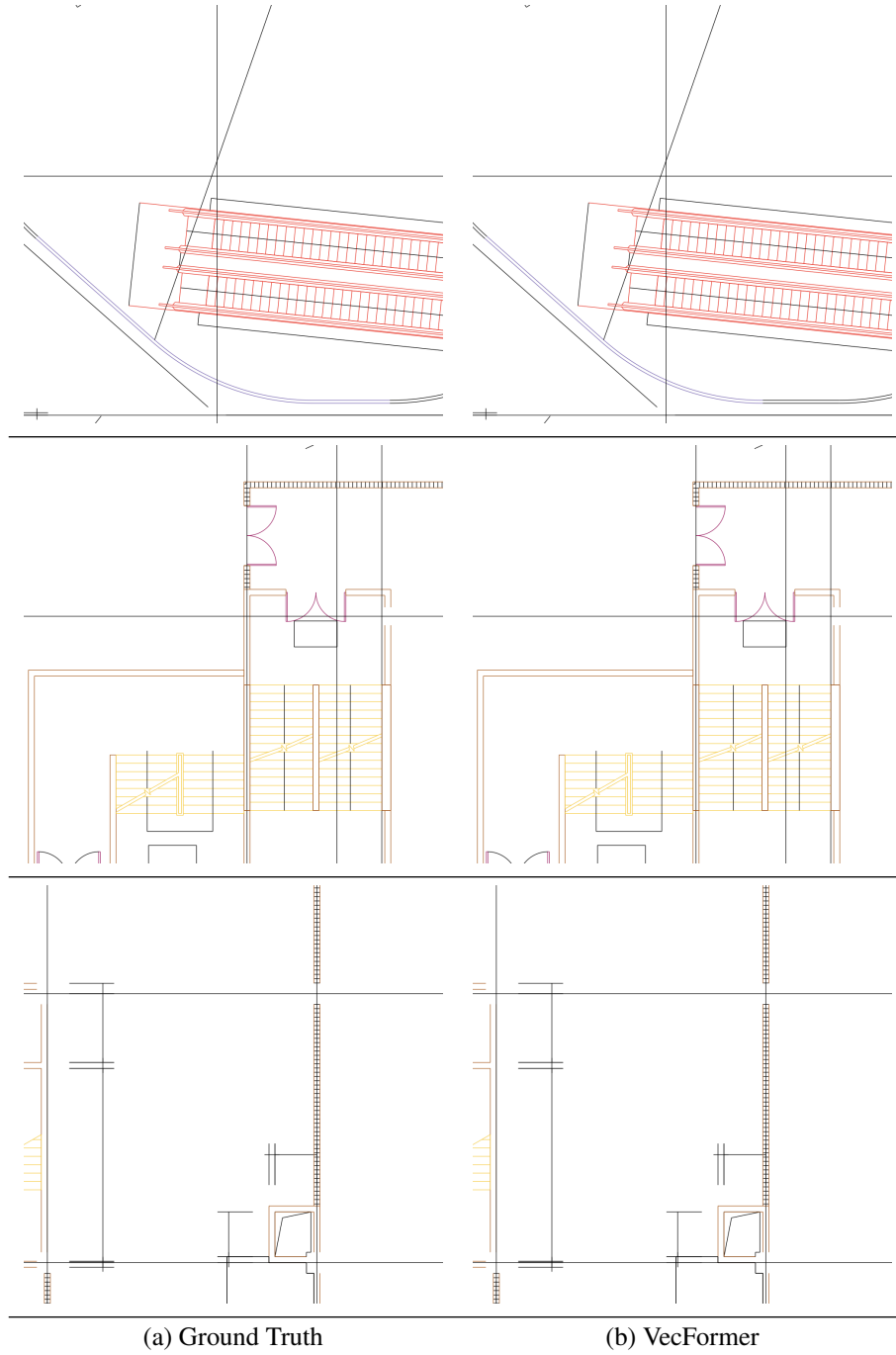


Figure 9: Results of VecFormer on FloorPlanCAD.

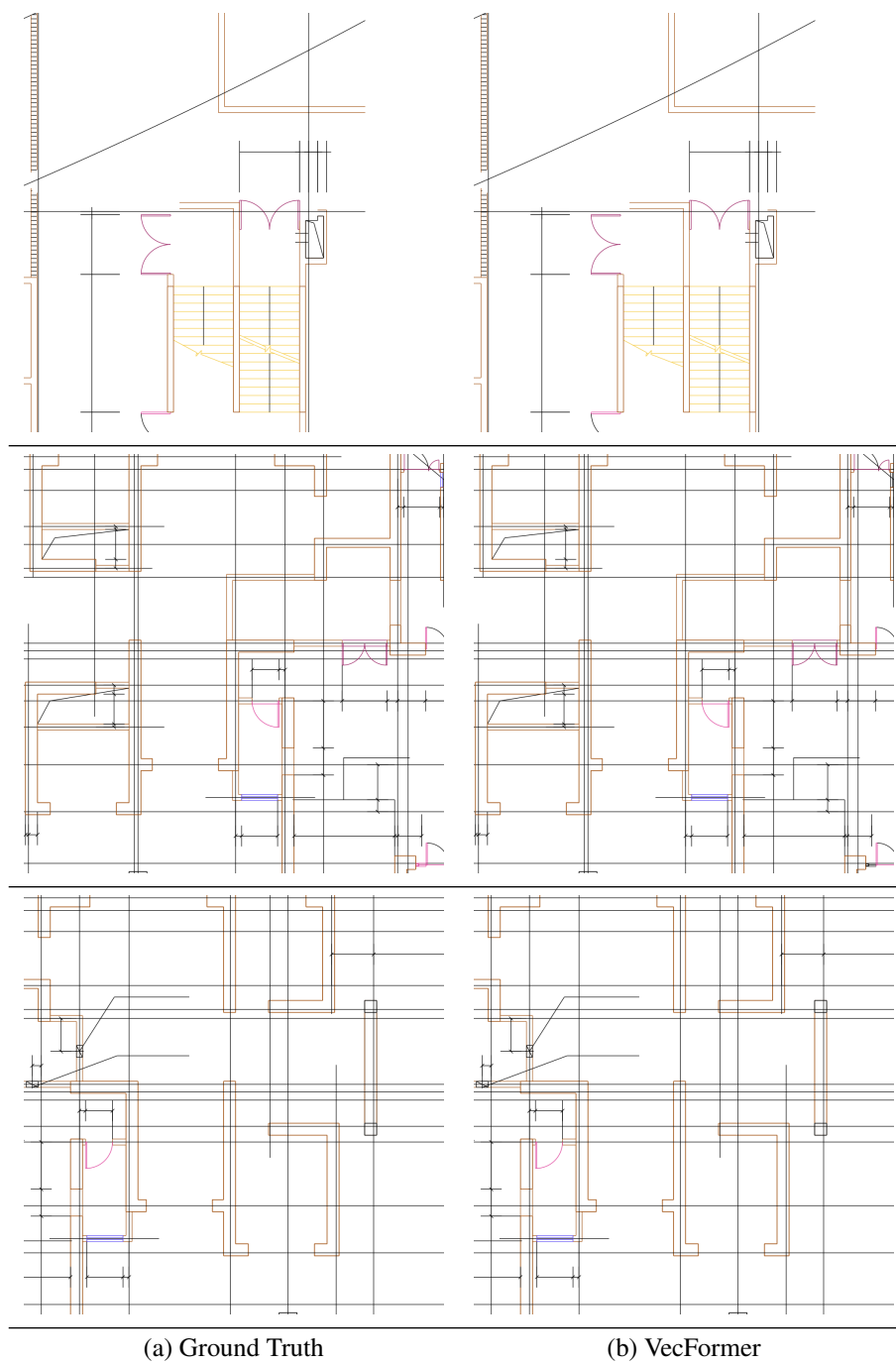


Figure 10: Results of VecFormer on FloorPlanCAD.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction explicitly state the main contributions and the scope of the paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss the limitations in section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)



Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We provide sufficient details in section 3 and section 4 to allow reproduction of the main results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: This study is based on publicly available datasets. The code is available at <https://github.com/WesKwong/VecFormer>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide sufficient implementation details in section 3 and section 4 to facilitate the understanding of our experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: In this submission, we report only the mean performance of each method from a single experimental run. Owing to the high computational overhead, we did not assess variability across different random seeds or data splits. Consequently, neither error bars nor statistical significance analyses are provided.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide detailed descriptions of the compute environment used for our experiments. Details are included in section 4 of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research presented in this paper fully complies with the NeurIPS Code of Ethics. It does not involve any personally identifiable information, human subjects, or sensitive data. The experiments are conducted on publicly available datasets, and no risks of misuse or negative societal impact have been identified. We also provide implementation details to promote reproducibility.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We briefly discuss potential societal impacts in section 5. The proposed method can enhance automation and efficiency in CAD-based design workflows, reducing manual effort and improving accuracy in symbol recognition. We have carefully examined the potential societal impacts of our work and did not identify any foreseeable risks or negative consequences.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release any pretrained models or datasets associated with high misuse risk. Only the training code and evaluation pipeline will be made available for research purposes, which do not pose identifiable risks of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

#### 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly credited all third-party assets used in our work. Specifically, we utilized the FloorPlanCAD dataset, which is licensed under the Creative Commons Attribution-NonCommercial 4.0 License (CC BY-NC 4.0). We have cited the original paper in our paper. All usage complies with the dataset's license terms.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We plan to release our code upon acceptance of the paper. The codebase will include comprehensive documentation, including a README file with installation instructions, usage examples, and detailed comments within the code to facilitate reproducibility.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: Our study does not involve any human participants or crowdsourcing experiments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our research does not involve any human participants or crowdsourcing experiments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### 16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Our research does not involve the use of Large Language Models (LLMs) in the development of our core methodology. Any LLM tools used were solely for writing assistance and did not influence the scientific content of the paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.