No More Ambiguity in 360° Room Layout via Bi-Layout Estimation

Yu-Ju Tsai[1,3*] Jin-Cheng Jhang[2*] Jingjing Zheng[3] Wei Wang[3]
Albert Y. C. Chen[3] Min Sun[2,3] Cheng-Hao Kuo[3] Ming-Hsuan Yang[1]
[1]UC Merced [2]National Tsing Hua University [3]Amazon
{ytsai17, myang37}@ucmerced.edu {frank890725}@gapp.nthu.edu.tw
{zhejingj, wweiwan, aycchen, minnsun, chkuo}@amazon.com

Abstract

Inherent ambiguity in layout annotations poses significant challenges to developing accurate 360° room layout estimation models. To address this issue, we propose a novel Bi-Layout model capable of predicting two distinct layout types. One stops at ambiguous regions, while the other extends to encompass all visible areas. Our model employs two global context embeddings, where each embedding is designed to capture specific contextual information for each layout type. With our novel feature guidance module, the image feature retrieves relevant context from these embeddings, generating layout-aware features for precise bi-layout predictions. A unique property of our Bi-Layout model is its ability to inherently detect ambiguous regions by comparing the two predictions. To circumvent the need for manual correction of ambiguous annotations during testing, we also introduce a new metric for disambiguating ground truth layouts. Our method demonstrates superior performance on benchmark datasets, notably outperforming leading approaches. Specifically, on the MatterportLayout dataset, it improves 3DIoU from 81.70% to 82.57% across the full test set and notably from 54.80% to 59.97% in subsets with significant ambiguity.

1. Introduction

Room layout estimation from a single 360° image has received significant attention due to the availability of cheap 360° cameras and the demonstration of visually pleasing room pop-ups. It also plays a vital role in indoor 3D scene understanding [3, 15, 32, 38] as the room layout constrains the space where objects are placed and interact. Its performance has improved significantly over the years, where the gain comes from better algorithms design [17, 29, 31, 34], and more challenging data collected [7, 45]. Despite the progress, the task formulation of predicting a *single* layout given a *single* 360° image has never been changed.

(a) Enclosed Type Annotation

(b) Extended Type Annotation

Figure 1. **Inherent ambiguity in the MatterportLayout [45].** Blue and Green represent ground truth annotations and predictions from the SoTA models, respectively. The layout boundaries are shown on the left, and their bird's-eye view projections are on the right. We define two types of layout annotation: (a) **enclosed type** encloses the room. (b) **extended type** extends to all visible areas. The dashed lines underscore the ambiguity in the dataset labels.

Annotating a *single* layout for each 360° image is, in fact, an ambiguous task. For instance, consider the two images with openings in Fig. 1, where the ground truth (GT) annotation stops at openings and encloses the nearest room in Fig. 1(a) while extending to all visible areas inside the opening in Fig. 1(b). Notably, even within the same dataset, there are variations in how opening regions are annotated. We observe that this ambiguity issue in annotation is prevalent across most datasets, and there is a lack of a clear definition of how to annotate ambiguous regions. Furthermore, state-of-the-art (SoTA) methods often overlook this ambiguity issue, leading to inherent inaccuracy during training. As a result, existing methods may predict in a manner opposite to the GT, as shown in Fig. 1. In this paper, we define two main types of layout annotations: ***enclosed*** and ***extended***. The former

*Equal contribution

Figure 2. **Comparions of our Bi-Layout model and the SoTA models.** Blue and Green indicate ground truth labels and predictions, respectively. For each image, the layout boundaries are shown on the left, while their bird's-eye view projections are shown on the right. Our Bi-Layout model can predict two extremely different types of layouts (*enclosed* and *extended*), addressing the ambiguity issue that the SoTA methods struggle with.

stops at ambiguous regions and encloses the room, whereas the latter extends to all visible areas inside the opening.

To address the confusion arising from the ambiguity issue in model training, as shown in Fig. 3, we propose a novel Bi-Layout model to simultaneously predict both *enclosed* and *extended* layouts for each image. Our model consists of three components: a shared feature extractor, two separate global context embeddings, and a shared feature guidance module. Two separate global context embeddings are learned to encode all context information related to the corresponding layout type. The shared feature guidance module guides the fusion of the shared image feature with our two embeddings separately through cross-attention. Specifically, we use the image feature as the query and each global context embedding as the key and value. When queried by the image feature, the global context embedding can inject layout type-related context information into the image feature. This results in enhanced alignment of the image feature with the corresponding layout prediction type.

Our model design introduces two key innovations. First, contrary to the standard Query-Key-Value setting employed in DETR [2] and other high-level tasks [4, 5, 20, 30, 37, 42], where embeddings serve as queries to retrieve relevant information from the image feature, we invert this relationship and employ our image feature as the queries. This unconventional design allows the image feature to be guided by our embedding, which represents the global information of a specific layout type. To the best of our knowledge, we are the first to develop a query-based model for room layout estimation, inherently designed for predicting multiple layouts. The second innovation lies in the efficiency of our model, which can predict two layouts with minimal additional model size overhead. For bi-layout estimations, naive approaches either train two distinct models with identical architectures on different labels or train a single model by sharing the feature extractor but separating other components. However, the former method doubles the model size and training time, while the latter lacks compactness and grapples with interference in simultaneously learning two layout types. In contrast, our model is not only the smallest, achieved by sharing both the feature extractor and the guidance module, but it also avoids interference issues by employing separate global context embeddings to guide feature fusion for different layout types. As shown in Fig. 2, our model is able to predict two extremely different layouts.

We also introduce a new metric termed as *disambiguate* metric to resolve ambiguities in the annotations of test data. It calculates the Intersection over Unions (IoU) of both predicted layouts with the ground truth and selects the higher IoU for evaluation. This is an effective way to quantitatively measure the benefit of our Bi-Layout estimation without manually correcting ambiguous annotations during testing. Another noteworthy feature of our Bi-Layout model is its ability to detect ambiguous regions with reasonable precision and recall by comparing two predictions. Our method exhibits superior performance on benchmark datasets, surpassing SoTA methods. On MatterportLayout [45], it enhances 3DIoU from 81.70% to 82.57% on the entire test set and notably from 54.80% to 59.97% in subsets with substantial ambiguity.

The main contributions of this work are:

- We clearly identify layout ambiguity issues in existing datasets and propose a *disambiguate* metric to measure the accuracy with multiple predictions effectively.
- We propose a novel Bi-Layout model that utilizes two global context embeddings with a shared feature guidance module to generate multiple layout predictions while keeping the model compact.
- We evaluate our method with extensive experiments and prove it outperforms SoTA methods in all settings, showing that our Bi-Layout model effectively resolves the layout ambiguity issues.

2. Related Work

360° room layout estimation. In 360° room layout estimation, prior methods follow the Manhattan World as-

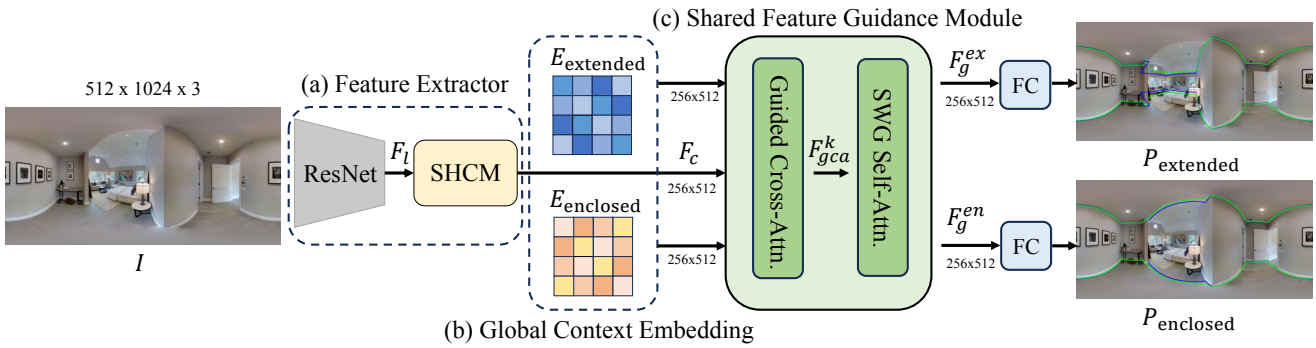


Figure 3. **Our Bi-Layout network architecture.** (a) **Feature extractor**: It processes a panoramic image I using ResNet-50 to extract multi-scale features F_l, and then feed those feature into the Simplified Height Compression Module (SHCM) to produce the final compressed feature F_c. (b) **Global Context Embedding**: It consists of two learnable embeddings E_k, each designed to capture and encode the contextual information inherent in the corresponding type of layout labels. (c) **Shared Feature Guidance Module**: It consists of two components: Guided Cross-Attention and SWG Self-Attention. It guides the fusion of compressed feature F_c with the global context embedding E_k to generate feature F_g^k ($k \in [\textit{extended}, \textit{enclosed}]$) more aligned for the corresponding layout type. Finally, we use fully connected (FC) layers to map F_g^k to horizon-depth and room height, which are further converted to boundary layouts (P_{extended} and P_{enclosed}).

sumption [6]. For instance, LayoutNet [44] predicts corner and boundary probability maps directly from panoramas. Dula-Net [36] predicts 2D-floor plane semantic masks from equirectangular and perspective views of ceilings. Zou *et al*. presents improved versions, LayoutNet v2, and Dula-Net v2 [45], demonstrating enhanced performance on cuboid datasets. Fernandez *et al*. adopts equirectangular convolutions (EquiConvs) [8] for generating corner and edge probability maps. HorizonNet [31] and HoHoNet [32] simplify layout estimation by employing 1D representations and employing Bi-LSTM [14, 27] and multi-head self-attention mechanisms [33] to establish long-range dependencies. LED^2-Net [34] reformulates layout estimation as predicting the depth of walls in the horizontal direction. AtlantaNet [24] predicts room layouts by combining projections of the floor and ceiling planes. DMH-Net [41] transforms panorama into cubemap [9] and predicts the position of intersection lines with learnable Hough Transform Block [40]. LGT-Net [17] employs self-attention transformers [33] to learn geometric relationships and capture long-range dependencies. DOP-Net [29] disentangles 1D feature by segmenting features into orthogonal plane representation, and uses GCN [18] and transformer [33] to refine the features.

These methods are designed only to predict a single layout, which often encounters challenges posed by the inherent ambiguity in dataset labels, resulting in suboptimal performance. In contrast, our Bi-Layout model addresses this issue by generating two distinct layout predictions through the innovative integration of global context embeddings and our shared feature guidance module design.

Multiple layout hypotheses. Several prior studies [10, 12, 13, 19, 25, 26, 28, 35, 39] have employed multiple hypotheses in their methods for estimating room layouts. The fundamental concept behind these methods involves leveraging vanishing points, edges, or other pertinent information to generate several rays or boxes as potential layout hypotheses. Through various scoring function designs, one of the hypotheses can be selected as the prediction that best fits the room. In contrast, our method generates two distinct layout predictions, which can also be viewed as having two hypotheses. However, the key disparity lies in the fact that the previous methods only have one hypothesis defining the correct geometry. On the contrary, both of our predictions are meaningful and offer two different geometries, *extended* and *enclosed* layouts, allowing for flexibility in choosing the suitable one based on the specific requirements of different use cases.

Query-based vision transformer. Transformers [33] have exhibited considerable efficacy in various high-level computer vision tasks, including object detection [2, 43], segmentation [4, 5], tracking [20, 42], and floorplan reconstruction [30, 37]. The standard transformer decoder utilizes feature embeddings as queries to extract relevant features from the image feature, which acts as both the key and value. Unlike the standard query-based transformer, our proposed design utilizes the image feature as the query with our global context embedding as the key and value. This unique design allows our model to predict two distinct layouts, a departure from prior methods that predict only a single layout.

3. Inherent Ambiguity in Labeled Data

We systematically examine instances of low IoU in the SoTA methods [17, 29, 34] using MatterportLayout dataset [45]. Our analysis identifies two types of ambiguity. First, when a *enclosed* type GT label is given, the SoTA methods predict regions located outside of that designated room (See Fig. 1(a)). Conversely, when a *extended* type GT label is given, the SoTA methods concentrate on the room where the camera is positioned (See Fig. 1(b)). These findings underscore inherent ambiguity within the testing GT labels. Moreover, since

the SoTA models will predict either *enclosed* or *extended* types of layouts, the same ambiguity is likely to be within the training GT labels as well. This presents a substantial challenge for single-prediction-based SoTA methods.

4. Method

To address the ambiguity issue in the dataset labels, we introduce our Bi-Layout model as shown in Fig. 3, which can generate two types of layout predictions P_{extended} and P_{enclosed}. Our model mainly consists of three modules: feature extractor (Sec. 4.1), global context embedding (Sec. 4.2), and shared feature guidance module (Sec. 4.3). We describe each component and the training objectives (Sec. 4.4) used for training our Bi-Layout model in the following sections.

4.1. Feature Extractor

The feature extractor in our Bi-Layout model is shown in Fig. 3(a). We follow previous works [17, 29, 31, 34] to use ResNet-50 [11] architecture to extract 2D image features F_l of 4 different scales from the input panorama I. For each feature scale, we modify the module from [31] as *Simplified Height Compression Module* (SHCM) to compress the features along the image height direction and generate 1D feature of the same dimension $\mathbb{R}^{N \times \frac{D}{4}}$, where N is the width of feature map and D is the feature dimension. Finally, we concatenate these features from different scales to generate the final compressed feature $F_c \in \mathbb{R}^{N \times D}$, where $N = 256$ and $D = 512$.

In contrast to previous works [17, 31, 34] setting $D = 1024$, our design reduces model parameters. By pruning our feature representation, we enhance the model's efficiency without compromising its effectiveness. To assess this design choice, we present detailed ablation studies in Sec. 5.5.

4.2. Global Context Embedding

Once the compressed feature F_c is extracted, we introduce a novel and learnable embedding mechanism termed as *Global Context Embedding*. This mechanism captures and encodes the overarching contextual information in a specific layout label, as illustrated in Fig. 3(b). We employ two learnable embeddings $E_k \in \mathbb{R}^{N \times D}$ where $k \in [\textit{extended}, \textit{enclosed}]$, one for *extended* and the other one for *enclosed* type. During training, these embeddings learn and encode diverse contextual information associated with different types of layout annotation. Moreover, they play a vital role in providing the dataset's global context information when queried by the compressed image feature F_c via cross-attention in our shared feature guidance module (refer to Sec. 4.3). By infusing our compressed feature F_c with this rich layout type-related embedding E_k, we generate diverse and meaningful predictions (P_{extended} and P_{enclosed}), each aligned with a distinct global context of the dataset label.

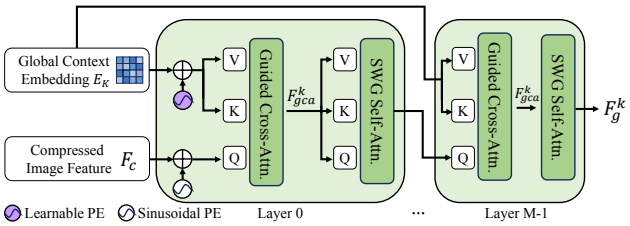


Figure 4. **Our Shared Feature Guidance Module architecture (SFGM).** It consists of two blocks: Guided Cross-Attention and SWG Self-Attention. The module has $M = 8$ layers, and the structure of each layer is identical. Given the compressed image feature F_c and global context embedding E_k, we first apply the sinusoidal and learnable positional encoding, respectively. With the compressed feature F_c as the query $\mathbf{Q}$ and our global context embedding E_k as both the key $\mathbf{K}$ and value $\mathbf{V}$, our guided cross-attention generates the feature F^k_{gca}, and it is served as $\mathbf{QKV}$ inputs to SWG self-attention. This process will repeat and further refine the output feature with our global context embedding to generate the final guided feature F^k_g.

4.3. Shared Feature Guidance Module

Building upon the compressed image feature F_c and the global context embeddings E_k, as shown in Fig. 3(c), we present an innovative component called *Shared Feature Guidance Module* (SFGM). This module can effectively guide the fusion of the image feature with the target global context embedding. Specifically, we share the compressed image feature F_c and use different global context embeddings E_k (one at a time) as the inputs for our shared feature guidance module $SFGM(\cdot)$ to generate corresponding guided feature $F^k_g \in \mathbb{R}^{N \times D}$, denoted as:

$$F^k_g = SFGM(F_c, E_k),\ k \in [\textit{extended},\ \textit{enclosed}]. \quad (1)$$

Our shared feature guidance module consists of *Guided Cross-Attention* and *SWG Self-Attention* as the building blocks, and the details of the architecture are shown in Fig. 4.

The standard cross-attention setting in DETR [2] or other high-level tasks [4, 5, 20, 30, 37, 42] uses embeddings as the query Q to retrieve relevant information from the corresponding image feature, which acts as both the key $\mathbf{K}$ and value $\mathbf{V}$ in order to generate the target outputs. In our scenario, if we adopt the standard $\mathbf{QKV}$ setting, the shared image feature F_c alone may not carry sufficient information to distinguish between the two types of distinct layouts. Therefore, we reverse this relationship and use our global context embeddings E_k to learn from corresponding labels, guiding the image feature F_c to generate the desired layout types.

As shown in Fig. 4, we use the compressed feature F_c as the query $\mathbf{Q}$ and our global context embedding E_k as both the key $\mathbf{K}$ and value $\mathbf{V}$ in our guided cross-attention

$GCA(\cdot)$:

$$\begin{aligned} \mathbf{Q} &= (F_c + PE_{\text{sin}})\mathbf{W}_q, \quad \mathbf{K} = (E_k + PE_{\text{learn}})\mathbf{W}_k, \\ \mathbf{V} &= (E_k + PE_{\text{learn}})\mathbf{W}_v, \quad F^k_{gca} = GCA(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \end{aligned} \tag{2}$$

where $F^k_{gca} \in \mathbb{R}^{N \times D}$ is the output of our guided cross-attention block. We apply different positional encoding strategies for these features, utilizing learnable positional encoding [2] $PE_{\text{learn}} \in \mathbb{R}^{N \times D}$ for our global context embedding E_k and sinusoidal positional encoding [1, 21] $PE_{\text{sin}} \in \mathbb{R}^{N \times D}$ for the compressed image feature F_c. Each feature is then multiplied by its respective learnable weights $\mathbf{W}_{q/k/v} \in \mathbb{R}^{D \times D}$. This unique design choice enables us to enrich the image feature F_c by effectively incorporating our global context embedding E_k. Subsequently, this enriched feature F^k_{gca} is served as $\mathbf{QKV}$ inputs to the SWG self-attention module [17] for further enhancement. As demonstrated in [17], the SWG self-attention module can effectively establish local and global geometric relationships within the room layout. Then, the process of guided cross-attention and SWG self-attention is repeated several times to refine the image feature with our context embeddings to generate the final guided feature F^k_g, as shown in Fig. 4.

By employing this novel cross-attention design, we achieve an enriched and context-aware guided feature representation F^k_g that is subsequently utilized for generating our Bi-Layout predictions (P_{extended} and P_{enclosed}), each possessing distinct and valuable properties. This flexibility in our method enables us to provide diverse layout predictions tailored to different global context embeddings E_k and input panorama features F_c. Built on this architectural design, our model can be compact and efficient to generalize to more label types by increasing global context embeddings E_k.

4.4. Training Objective

After obtaining the target feature F^k_g, we follow [17, 29] using fully connected (FC) layers to map the feature F^k_g to horizon-depth $d_k = \{d^i_k\}^N_{i=1}$ and room height h_k, where N is the width of the feature map. We can apply the explicit transformation to convert the horizon depth and room height to layout boundaries P_k on the panorama. We further convert column-wise depth d^i_k into depth normal n^i_k and gradient of normal g^i_k, $k \in [\textit{extended}, \textit{enclosed}]$.

The loss functions for depth, normal, gradient, and room height are defined as follows:

$$\begin{aligned} \mathcal{L}^k_{\text{depth}} &= \frac{1}{N}\sum_{i \in N} |d^i_k - d^i_{\text{gt}}|, \ \mathcal{L}^k_{\text{normal}} = \frac{1}{N}\sum_{i \in N}(-n^i_k \cdot n^i_{\text{gt}}), \\ \mathcal{L}^k_{\text{gradient}} &= \frac{1}{N}\sum_{i \in N} |g^i_k - g^i_{\text{gt}}|, \ \mathcal{L}^k_{\text{height}} = |h_k - h_{\text{gt}}|, \end{aligned} \tag{3}$$

where $d^i_{\text{gt}}, n^i_{\text{gt}}, g^i_{\text{gt}}$, and h_{gt} denote the ground truth of depth, normal, gradient, and room height respectively. We calculate the L1 loss for depth loss, gradient loss, height loss, and cosine similarity for normal loss. Our branch loss $\mathcal{L}_k$ is:

$$\mathcal{L}_k = \lambda_d \mathcal{L}^k_{\text{depth}} + \lambda_n \mathcal{L}^k_{\text{normal}} + \lambda_g \mathcal{L}^k_{\text{gradient}} + \lambda_h \mathcal{L}^k_{\text{height}}, \tag{4}$$

where $k \in [\textit{extended}, \textit{enclosed}]$ and the final loss $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{extended}} + \mathcal{L}_{\text{enclosed}}$. We set $\lambda_d = 0.9$, $\lambda_n = 0.1$, $\lambda_g = 0.1$ and $\lambda_h = 0.1$ for both branches to balance the model weight.

5. Experiments

We conduct our experiments on a single NVIDIA RTX 4090 GPU and implement the proposed method with PyTorch [22]. For training, we use a batch size of 12 and set the learning rate to 1×10^{-4}. We select Adam as our optimizer, adhering to its default configurations. For the data augmentation, we use the technique proposed in [31], including left-right flipping, panoramic horizontal rotation, luminance adjustment, and panoramic stretch.

5.1. Datasets

MatterportLayout. MatterportLayout [45] contains 2295 samples labeled by Zou *et al*. [45]. However, as we analyzed in Sec. 3, this dataset has annotation ambiguity, and many images with ambiguity are annotated with the *extended* type. Hence, we propose a semi-automatic procedure (provide the details in supplementary material) to re-annotate *enclosed* type labels from the ambiguous *extended* ones. We re-annotate 15% of the labels in the whole dataset. Note that these new labels of *enclosed* type plus the remaining 85% of original labels will be used to train our Bi-Layout model's *enclosed* branch. In contrast, all original labels will be used to train the *extended* branch. For a fair comparison with SoTA methods, we use the original label and the same testing split for evaluation.

ZInD. ZInD [7] dataset is currently the largest dataset with room layout annotations. It provides both *raw* and *visible* labels, which is similar to our defined *enclosed* and *extended* types, respectively. Besides, ZInD separates the data into *simple* and *complex* subsets based on whether the images have contiguous occluded corners. Therefore, we have two variants of ZInD in our experiments: **(a) ZInD-Simple** represents the *simple* subset and consists of 24882, 3080, and 3170 panoramas for training, validation, and testing splits. **(b) ZInD-All** represents the whole dataset with 50916, 6352, and 6352 panoramas for each split. It has complex opening regions, resulting in more severe ambiguity issues. Therefore, it can better evaluate the robustness of different methods for handling the ambiguity issue. In both ZInD dataset variants, we use the *raw* and *visible* labels to train our *enclosed* and *extended* branches, respectively, and follow the SoTA methods to test on *raw* labels for a fair comparison.

(a) Full set		MatterportLayout [45]		ZInD-Simple [7]		ZInD-All [7]	
Method	# Params	2DIoU(%)	3DIoU(%)	2DIoU(%)	3DIoU(%)	2DIoU(%)	3DIoU(%)
LED^{2}Net [34]	82 M	82.37	80.05	90.20	88.34	82.31	80.28
LGT-Net [17]	136 M	83.52	81.11	91.77	89.95	84.07	82.09
DOP-Net [29]	137 M	84.11	81.70	91.94	90.13	83.92	81.87
Ours (equivalent branch)	102 M	84.56	82.05	92.07	90.25	84.90	82.96
Ours (disambiguate)	102 M	**85.10**	**82.57**	**92.79**	**90.95**	**86.21**	**84.22**
(b) Subset		MatterportLayout [45]		ZInD-Simple [7]		ZInD-All [7]	
Method	# Params	2DIoU(%)	3DIoU(%)	2DIoU(%)	3DIoU(%)	2DIoU(%)	3DIoU(%)
LED^{2}Net [34]	82 M	53.57	51.12	45.31	44.10	48.76	47.35
LGT-Net [17]	136 M	53.17	50.54	53.20	52.00	50.89	49.58
DOP-Net [29]	137 M	57.13	54.80	51.55	50.26	50.92	49.46
Ours (equivalent branch)	102 M	59.85	57.08	55.09	53.76	54.22	52.78
Ours (disambiguate)	102 M	**62.81**	**59.97**	**62.10**	**60.63**	**60.20**	**58.53**

Table 1. **Full set and Subset evaluation. Equivalent branch** represents the output, which is trained with the same label as baseline methods. **Disambiguate** is our proposed metric.

5.2. Disambiguate Metric

If we already know the layout type of each test image, we can use this information to select the output from the corresponding branch for evaluation. However, we find that the test data has annotation ambiguity; even the *raw* labels in ZInD are still not exempt from this issue.

To address the above issue and demonstrate our model's capability to handle the annotation ambiguity, we introduce a new metric, termed the *disambiguate* metric, as follows:

$$IoU_{\text{disambiguate}} = \sum_{i=0}^{S} \arg\max_{P_k^i \in \mathbb{P}} IoU(P_{\text{gt}}^i,\ P_k^i), \qquad (5)$$

where $P_k^i \in \mathbb{P},\ k \in [\textit{extended},\ \textit{enclosed}]$ denotes layout predictions from both branches and P_{gt}^i denotes the ground truth layout. We first calculate the Intersection over Union (IoU) between each prediction and ground truth (GT) for each image and then select the higher IoU for averaging all samples. This is because the higher IoU serves as the disambiguate prediction and represents the most suitable prediction when encountering ambiguity.

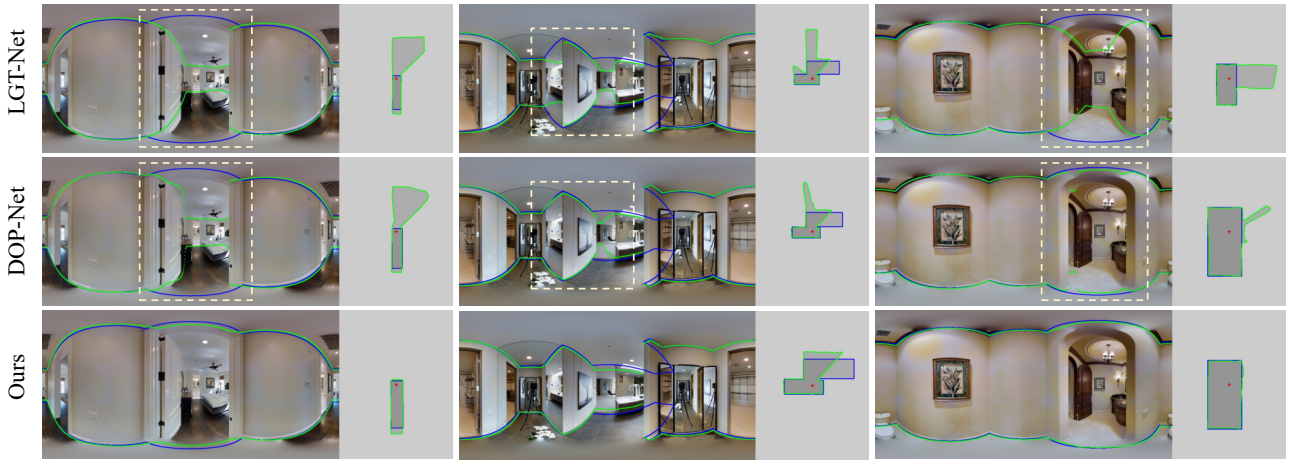
The proposed metric effectively provides a robust and quantitative measure of how a method excels in handling ambiguous scenarios within the dataset without necessitating manual corrections to the ambiguous annotations. In other words, we can use the labels provided by the original dataset to do the evaluation.

5.3. Comparison with State-of-the-Art Methods

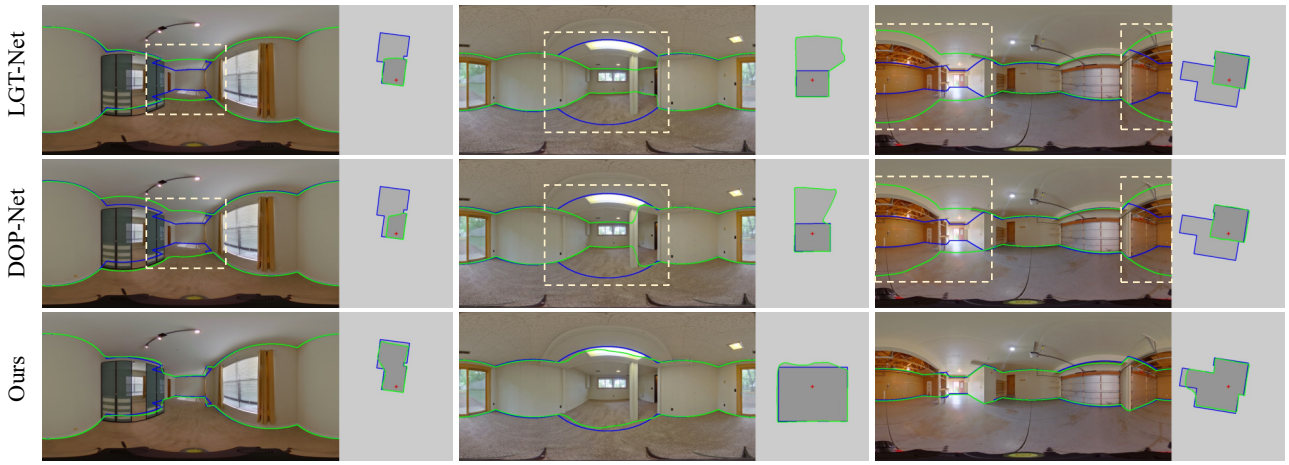
Evaluation settings. Since our model outputs two layouts, we propose to compare our method with the SoTA methods in two ways. **Using the equivalent branch**: We use the output from the branch trained with the same data as other methods. Specifically, we use the output from *extended* branch for the MatterportLayout dataset and the output from *enclosed* branch for ZInD to fairly compare with other methods. **Using both branches with disambiguate metric**: We use the proposed *disambiguate* metric as defined in Sec. 5.2 to evaluate the performance of our method.

Full set evaluation. We present quantitative results on three different datasets in Table 1(a). As the SoTA methods do not experiment on the ZInD-All dataset, we retrain all baseline models based on their official repositories. The results demonstrate that our *equivalent* branch consistently outperforms SoTA methods across all datasets, underscoring the advantages of joint training with bi-layout data. Furthermore, our disambiguated results surpass these benchmarks, affirming the existence of ambiguity in the original annotations. Our Bi-Layout model effectively mitigates this issue by selecting the most appropriate prediction. Notably, in terms of model size, our architecture, despite generating bi-layout predictions, maintains a smaller total parameter size compared to the SoTA models. This underscores the efficiency of our design in achieving superior performance with a more compact model.

Subset evaluation. To highlight the ambiguity issue, we select a subset based on the failure predictions of all the previous SoTA models [17, 29, 34]. For each SoTA model, we find images with the 2DIoU evaluation lower than 0.6. Next, we combine all these failure cases among all the SoTA models to construct the subset. Finally, the subsets consist of 11%, 6%, and 18% of the test data in three datasets, respectively. The quantitative results in Table 1(b) reveal a more pronounced performance gap between our method and the SoTA models, with differences reaching up to 9.28% in 2DIoU on the most ambiguous ZInD-All dataset. We also provide the qualitative results in Fig. 5, where a bird's-



(a) Qualitative comparison on the MatterportLayout [45] dataset.



(b) Qualitative comparison on the ZInd [7] dataset.

Figure 5. **Qualitative comparison** on the MatterportLayout [45] (top) and ZInd [7] datasets (bottom). Blue and Green represent ground truth labels and predictions, respectively. The boundaries of the room layout are on the left, and their bird's eye view projections are on the right. We show our *disambiguate* results, which effectively address the ambiguity issue, while the SoTA methods struggle with the ambiguity, as highlighted in dashed lines.

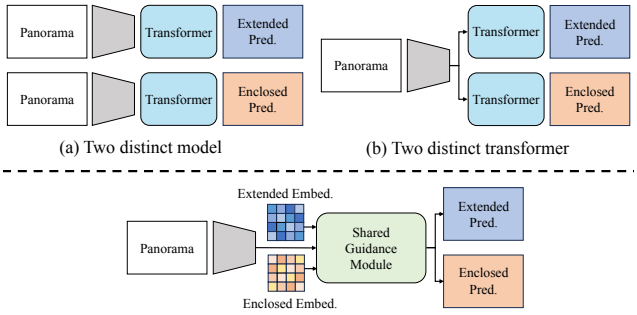
eye view of the predictions vividly illustrates the significant challenges posed by ambiguity. This confirms that layout ambiguity is a key cause for low IoUs of previous methods, and our Bi-Layout estimation is effective in addressing the issue and performs remarkably well in this subset.

5.4. Ambiguity Detection

Our Bi-Layout model can naturally detect ambiguous regions by comparing the per-column pixel difference between two predicted layout boundaries. This per-column pixel difference can serve as our predicted confidence score. If the difference is larger, the column is more likely to be an ambiguous region (i.e., typically an opening room structure). We formulate the detection of ambiguous regions into a binary classification task where GT ambiguous regions are columns with more than 2 pixels' difference between two annotations of the *extended* and *enclosed* types. We predict

Figure 6. **Qualitative results for ambiguity detection.** Blue and Green on the top and bottom rows represent ground truth and predicted confidence, respectively. Cyan and Megenta lines are our *extended* and *enclosed* type layout predictions. With these two predictions, our model can accurately detect ambiguous regions.

columns with more than 10 pixels' difference between predicted layouts as ambiguous regions. We test this method on ZInD as it is the only dataset that provides both types of GT



(a) Two distinct model (b) Two distinct transformer

(c) Our two embedding with shared guidance module

Figure 7. **Model architecture comparison.** We show the different model architecture designs for predicting multiple layouts. Please refer to Table 3 for quantitative comparison.

labels (*i.e., raw* and *visible*) in testing. Our method achieves a reasonably high Precision of **0.82** and Recall of **0.71**. Moreover, the qualitative results in Fig. 6 further demonstrate that our method can indeed detect ambiguous regions. We believe this is particularly useful for applications where the model can highlight ambiguous regions and let the users select suitable predictions for their use cases.

5.5. Abalation Studies

Different feature fusion designs. To fuse the information from image features and our global context embeddings, we conduct comprehensive ablation studies to validate the effectiveness of different designs. The fusion methods include add, concatenation, AdaIn [16], and FiLM [23]. We further investigate two Query-Key-Value (QKV) feature designs in our shared feature guidance module.

The results in Table 2 show that our proposed design significantly outperforms all feature fusion methods and the standard cross-attention setting, where the global context embedding E_k served as query $\mathbf{Q}$. The compressed image feature F_c served as key $\mathbf{K}$ and value $\mathbf{V}$. This demonstrates that our design can effectively utilize contextual information within the embedding E_k to enhance the alignment of the compressed feature F_c with the corresponding layout prediction type.

Comparions of model architectures. As shown in Fig. 7, we compare three model architectures to predict multiple layouts. We conduct the quantitative comparison in Table 3.
Two models: Training two models of the same architecture to predict two layout types is the most straightforward design (Fig. 7(a)). This architecture achieves good performance without interference between learning two types of layout. However, this doubles the model size and training time.
Two transformers: An alternative is to share the feature extractor but separate the transformer and prediction head (Fig. 7(b)). This saves model size a little, but the performance drops significantly as it cannot handle the interference when learning two types of layouts simultaneously.

Design	Q	KV	2D IoU (%)	3D IoU (%)
Add			83.79	81.26
Concat			84.17	81.44
AdaIn [16]			83.28	80.54
FiLM [23]			84.08	81.39
Standard	E_k	F_c	83.34	80.85
Ours	F_c	E_k	**85.10**	**82.57**

Table 2. **Comparison of QKV feature designs.** F_c is the compressed feature, and E_k is our global context embedding. We evaluate different designs of QKV features in the Shared Feature Guidance Module on MatterportLayout [45] with our proposed *disambiguate* metric.

Method	# Params	2DIoU(%)	3DIoU(%)
Two models	272 M	**85.29**	82.72
Two Transformers	203 M	84.35	81.88
Ours (c = 1024)	172 M	85.25	**82.76**
Ours (c = 512)	**102 M**	85.10	82.57

Table 3. **Model size and performance trade-off.** In this part, we only compare to the LGT-Net [17] model variances (i.e., The first two rows) since our model is built on top of its architecture. In the third row, c represents the # channel of the image feature. We conduct these quantitative evaluations on MatterportLayout [45] with our proposed *disambiguate* metric. Our final model strikes the best balance between performance and parameter efficiency.

Our model: Our model is the smallest as we share both the feature extractor and transformer and only separate the lightweight prediction head (Fig. 7(c)). To reduce the interference in learning two layouts, we introduce two learnable global context embeddings, which can inject layout type-related context information into the image feature via cross-attention. Therefore, our model achieves comparable or better performance than others. In addition, there is only a slight performance drop if we further reduce the model size by reducing the compressed feature channel dimension from 1024 to 512. Our final model ($c = 512$) balances performance and parameter efficiency best.

6. Conclusion

We propose a novel Bi-Layout model to generate two distinct predictions, effectively resolving the layout ambiguity. Most importantly, we introduce a novel embedding mechanism with a shared feature guidance module, where each embedding is designed to learn the global context inherent in each type of layout. Our model strikes a good balance between model compactness and prediction accuracy with these designs. In addition, we propose a *disambiguate* metric to evaluate the accuracy with multiple predictions. On MatterportLayout [45] and ZInD [7] datasets, our method outperforms other state-of-the-art methods, especially on the subset setting with considerable ambiguity.

References

[1] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *ICCV*, 2019. 5

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 3, 4, 5

[3] Xiaoxue Chen, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. Pq-transformer: Jointly parsing 3d objects and layouts from point clouds. *IEEE Robotics and Automation Letters*, 2022. 1

[4] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021. 2, 3, 4

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 2, 3, 4

[6] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *ICCV*, 1999. 3

[7] Steve Cruz, Will Hutchcroft, Yuguang Li, Naji Khosravan, Ivaylo Boyadzhiev, and Sing Bing Kang. Zillow indoor dataset: Annotated floor plans with 360deg panoramas and 3d room layouts. In *CVPR*, 2021. 1, 5, 6, 7, 8

[8] Clara Fernandez-Labrador, Jose M Facil, Alejandro Perez-Yus, Cédric Demonceaux, Javier Civera, and Jose J Guerrero. Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters*, 2020. 3

[9] Ned Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 1986. 3

[10] Abhinav Gupta, Martial Hebert, Takeo Kanade, and David Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. *NeurIPS*, 2010. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[12] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 3

[13] Martin Hirzer, Vincent Lepetit, and PETER ROTH. Smart hypothesis generation for efficient and robust room layout estimation. In *WACV*, 2020. 3

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 3

[15] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *ECCV*, 2018. 1

[16] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 8

[17] Zhigang Jiang, Zhongzheng Xiang, Jinhua Xu, and Ming Zhao. Lgt-net: Indoor panoramic room layout estimation with geometry-aware transformer network. In *CVPR*, 2022. 1, 3, 4, 5, 6, 8

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 3

[19] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *ICCV*, 2015. 3

[20] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *CVPR*, 2022. 2, 3, 4

[21] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018. 5

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 5

[23] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 8

[24] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. Atlantanet: inferring the 3d indoor layout from a single 360° image beyond the manhattan world assumption. In *ECCV*, 2020. 3

[25] Srikumar Ramalingam, Jaishanker K Pillai, Arpit Jain, and Yuichi Taguchi. Manhattan junction catalogue for spatial reasoning of indoor scenes. In *CVPR*, 2013. 3

[26] Yuzhuo Ren, Shangwen Li, Chen Chen, and C-C Jay Kuo. A coarse-to-fine indoor layout estimation (cfile) method. In *ACCV*, 2017. 3

[27] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997. 3

[28] Alexander G Schwing and Raquel Urtasun. Efficient exact inference for 3d indoor scene understanding. In *ECCV*, 2012. 3

[29] Zhijie Shen, Zishuo Zheng, Chunyu Lin, Lang Nie, Kang Liao, Shuai Zheng, and Yao Zhao. Disentangling orthogonal planes for indoor panoramic room layout estimation with cross-scale distortion awareness. In *CVPR*, 2023. 1, 3, 4, 5, 6

[30] Jheng-Wei Su, Kuei-Yu Tung, Chi-Han Peng, Peter Wonka, and Hung-Kuo Chu. Slibo-net: Floorplan reconstruction via slicing box representation with local geometry regularization. In *NeurIPS*, 2023. 2, 3, 4

[31] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *CVPR*, 2019. 1, 3, 4, 5

[32] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *CVPR*, 2021. 1, 3

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

[34] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Led2-net: Monocular 360deg layout estimation via differentiable depth rendering. In *CVPR*, 2021. 1, 3, 4, 6

[35] Huayan Wang, Stephen Gould, and Daphne Roller. Discriminative learning with latent variables for cluttered indoor scene understanding. *Communications of the ACM*, 2013. 3

[36] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In *CVPR*, 2019. 3

[37] Yuanwen Yue, Theodora Kontogianni, Konrad Schindler, and Francis Engelmann. Connecting the dots: Floorplan reconstruction using two-level queries. In *CVPR*, 2023. 2, 3, 4

[38] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. In *CVPR*, 2021. 1

[39] Hao Zhao, Ming Lu, Anbang Yao, Yiwen Guo, Yurong Chen, and Li Zhang. Physics inspired optimization on semantic transfer features: An alternative method for room layout estimation. In *CVPR*, 2017. 3

[40] Kai Zhao, Qi Han, Chang-Bin Zhang, Jun Xu, and Ming-Ming Cheng. Deep hough transform for semantic line detection. *TPAMI*, 2021. 3

[41] Yining Zhao, Chao Wen, Zhou Xue, and Yue Gao. 3d room layout estimation from a cubemap of panorama image via deep manhattan hough transform. In *ECCV*, 2022. 3

[42] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, 2022. 2, 3, 4

[43] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 3

[44] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *CVPR*, 2018. 3

[45] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. Manhattan room layout reconstruction from a single 360ˆ ◦ 360◦ image: A comparative study of state-of-the-art methods. *IJCV*, 2021. 1, 2, 3, 5, 6, 7, 8