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

Supplementary Material

1. Overview

This supplementary material presents additional results to complement the main manuscript. We first introduce our relabeling pipeline in Sec. 2 and show some bi-layout annotation examples in Sec. 3. In Sec. 4, we provide more qualitative comparisons with the state-of-the-art (SoTA) methods. In Sec. 5, we show more examples of our ambiguity detection under different scenarios to validate the robustness of our method. We conduct additional ablation studies in Sec. 6 to compare our method in more comprehensive settings. Finally, we show the limitations in Sec. 7 and provide some future research directions in Sec. 8.

2. Semi-automatic Relabeling

We introduce our semi-automatic relabeling pipeline for annotating the second type of layout on the MatterportLayout [3] dataset as follows and shown in Fig. 1:

(a) Given the original annotations from the MatterportLayout [3] dataset. We check each column of the panorama, if there are more than two annotations in the same column, we define it as the occlusion part. As shown in Fig. 1(a), blue line indicates the original annotation, and the dashed lines highlight the occlusion region.

(b) Next, we take the original annotation and project it to the bird's-eye view floorplan coordinate, aligning it with the center of the camera. As shown in Fig. 1(b), the isolated red point indicates the center of the camera.

(c) After obtaining the annotation on the floorplan coordinate, we categorize the corners as either *visible* or *invisible*, representing whether the corners can be seen from the center of the camera or not. We find the closest *visible* points in the occlusion region as our candidate corners. As shown in Fig. 1(c), the red boxes indicate our candidate visible corners.

(d) Once we have our candidate corners, we generate several annotation proposals using these points. As shown in Fig. 1(d), the red lines are our annotation proposals based on the candidate corners.

(e) We select the best proposal, which should provide a clear boundary between different rooms. Note that this is the only step that needs a human decision. As shown in Fig. 1(e), we manually choose the proposal to separate the two rooms in this case.

(f) Finally, we project these newly defined corners back to the panorama view, creating our relabeled annotation for the panorama. As shown in Fig. 1(f), green line indicates the relabeled annotation.

We introduce our semi-automatic relabeling pipeline with the above steps, which clearly explain how we relabel the MatterportLayout [3] dataset. With this relabel pipeline, we can generate the *enclosed* type of annotations from the *extended* type of annotations and use these new labels with the original labels to train our Bi-Layout model.

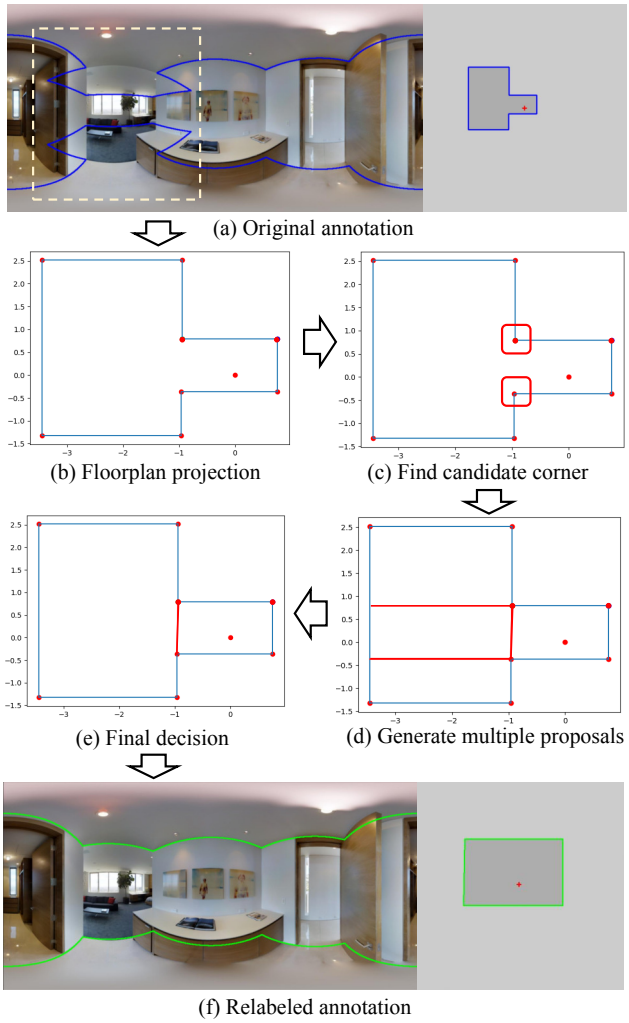

Figure 1. **Our relabeling pipeline**. Blue line in (a) and Green line in (f) represent the **original annotation** and **our relabeled annotation**, respectively. The layout boundaries are shown on the left, and their bird's-eye view projections are on the right. The dashed lines in (a) highlight the occlusion region in the original label.

3. Bi-layout Annotations

We show some bi-layout annotation examples in both the MatterportLayout [3] and ZInD [1] dataset.

(a) Original annotations (b) Relabeled annotations

Figure 2. **Bi-layout annotations on the MatterportLayout [3] dataset.** Blue and Green lines indicate original and relabeled annotations.

(a) Visible annotations (b) Raw annotations

Figure 3. **Bi-layout annotations on the ZInD [1] dataset.** Blue and Green lines indicate *visible* and *raw* annotations.

MatterportLayout We present some cases of our relabeled annotations for the MatterportLayout [3] dataset. The original annotations in Fig. 2(a) are from the original dataset labels. Our relabeled annotations are shown in Fig. 2(b). Based on our definition, we relabel the *extended* type of annotation to the *enclosed* type of annotation.

ZInD We show some cases of two types of annotations officially provided by ZInD [1] dataset. The *visible* annotations are shown in Fig. 3(a), and the *raw* annotations are shown in Fig. 3(b), which corresponds to our *extended* type and *enclosed* type, respectively.

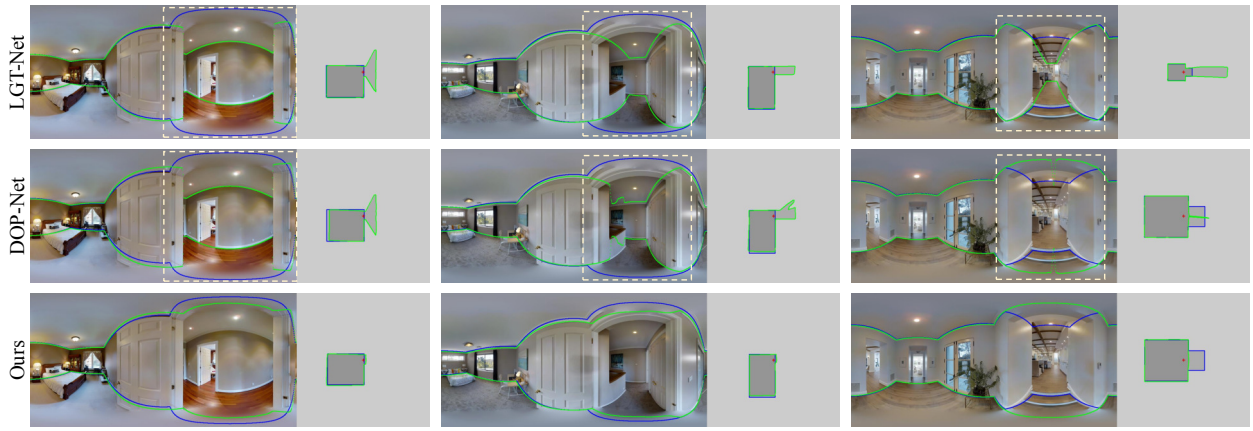

Figure 4. **More qualitative results on the MatterportLayout [3] dataset.** Blue and Green represent ground truth labels and predictions, respectively. The boundaries of the room layout are on the left, and their bird's-eye view projections are on the right. We show our *disambiguate* results, which effectively address the ambiguity issue, while the SoTA methods struggle with the ambiguity, as highlighted in dashed lines.

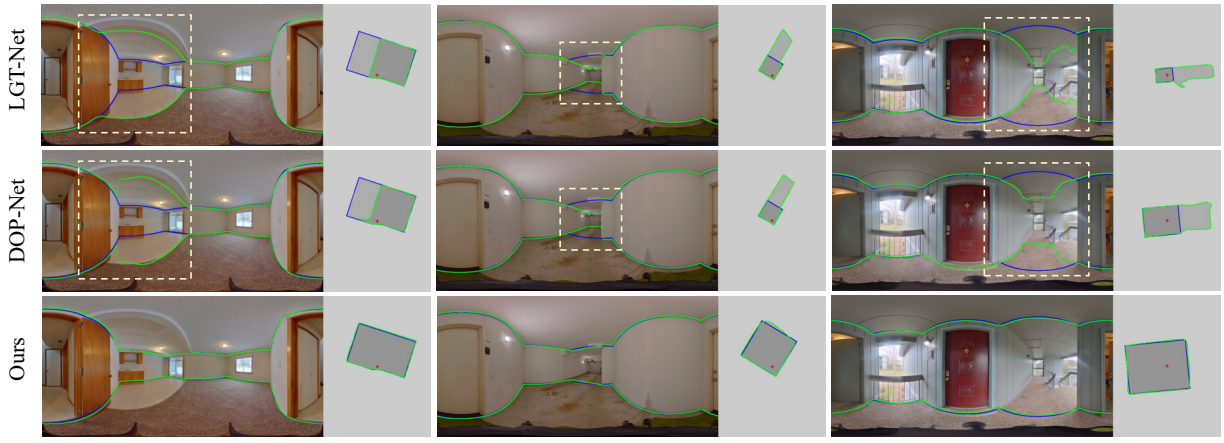

Figure 5. **More qualitative results on the ZInD [1] dataset.** Blue and Green represent ground truth labels and predictions, respectively. The boundaries of the room layout are on the left, and their bird's-eye view projections are on the right. We show our *disambiguate* results, which effectively address the ambiguity issue, while the SoTA methods struggle with the ambiguity, as highlighted in dashed lines.

4. Comparisons with SoTA

We show more qualitative results on the MatterportLayout [3] dataset in Fig. 4 and the ZInD [1] dataset in Fig. 5. Our Bi-Layout model can effectively address the ambiguity issue that the SoTA methods struggle with.

5. Ambiguity Detection

We show more qualitative results and several scenarios of ambiguity detection in Fig. 6. In (a) and (b), we provide more examples to demonstrate that our Bi-Layout model can accurately detect ambiguous regions as the GT shows. In (c), we offer a normal case where there is no ambiguous region in the image, and our model can predict two identical predictions. In (d), we show a special case where the GT does not indicate the ambiguous regions as it should be (i.e., GT itself has ambiguity), and our model can still successfully identify them, showing the capability of our method to address the inherent ambiguity issue in the dataset.

6. Ablation Studies

Global Context Embedding. To show the effectiveness of our proposed *Global Context Embedding* and *Shared Feature Guidance Module*, we conduct the experiment using a ***single*** global context embedding for our feature guidance module and only generate a ***single*** prediction as the conventional methods present (i.e., a single branch version of our proposed method). We compare our single branch with

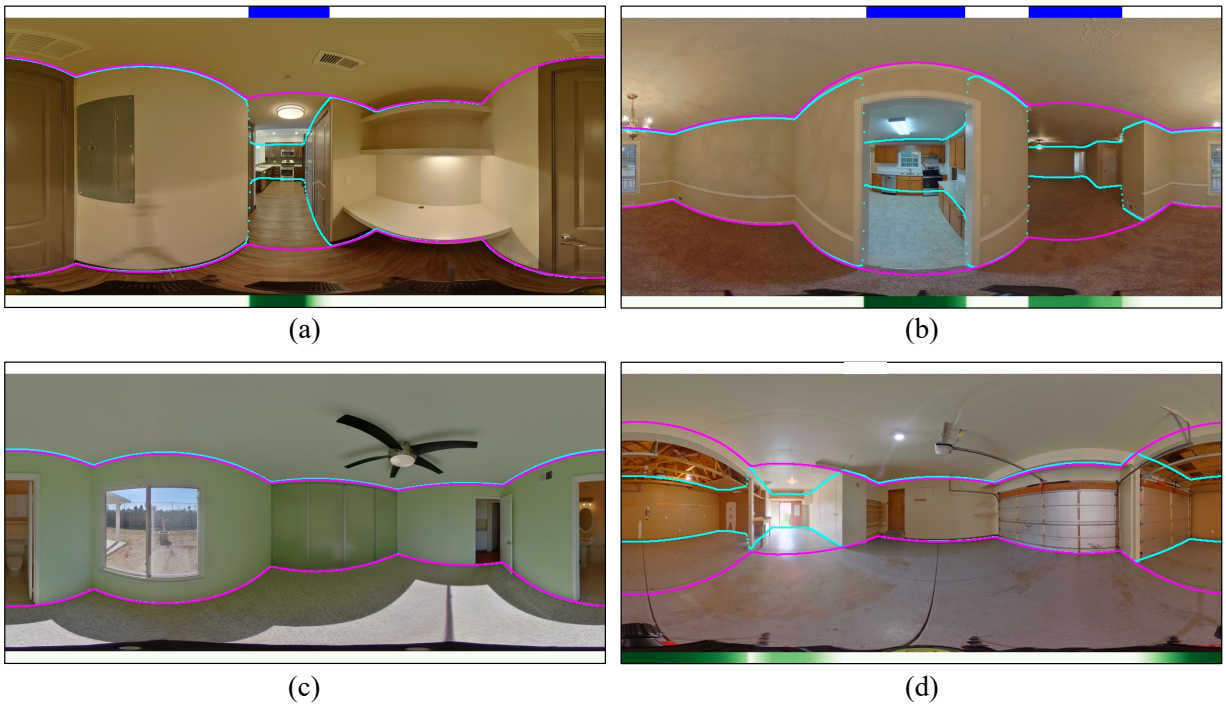

Figure 6. **Qualitative results and different scenarios for ambiguity detection.** Blue and Green on the top and bottom rows per image represent ground truth and predicted confidence, respectively. Cyan and Magenta lines are our *extended* and *enclosed* type layout predictions. In (a) and (b), our Bi-Layout model can accurately detect ambiguous regions as the GT shows. In (c), our model is able to predict two identical predictions when there is no ambiguous region in the image. In (d), we show a special case where the GT does not indicate the ambiguous regions as it should be (i.e., GT itself has ambiguity), and our model can still successfully identify them. Note that (d) is the example we show in the main manuscript where the SoTA methods fail, which corroborates the ambiguity in this image.

		Full set		Subset	
Method	# Params	2DIoU(%)	3DIoU(%)	2DIoU(%)	3DIoU(%)
LGT-Net [2]	136 M	83.52	81.11	53.17	50.54
Our single branch	102 M	**84.09**	**81.78**	**58.65**	**56.23**

Table 1. **Global Context Embedding for a single branch.** We conduct the experiment on both the full set and subset of the MatterportLayout [3] dataset. We choose LGT-Net [2] as our baseline method to compare the effectiveness of our Global Context Embedding design.

LGT-Net [2] since the proposed components are built on top of its architecture. In Table 1, our single branch outperforms the baseline method on both the full set and subset of the MatterportLayout [3] dataset, showing the effectiveness of our Global Context Embedding design.

Model size comparison. As discussed in the main manuscript, many model variations can let the SoTA method predict two layouts. We additionally show the *two-head* version of the baseline model, which shares the feature extractor and transformer parts, and simply add the other prediction head to generate the second type of layout. We compare all the model variations in Table 2. Although the *two-head* version model decreases the model parameters, the performance is degraded significantly due to the naive model design. The comparison with these model variations shows the effectiveness and compactness of our Bi-Layout model.

Image feature dimension. To make the model more compact, we compare different image feature dimensions: 1024, 512, and 256. We experiment on the full set and subset of the MatterportLayout [3] dataset. In Table 3, the feature dimension of 1024 performs the best but it has the largest model size. Our proposed Bi-Layout model with a feature dimension of 512 strikes a good balance between the performance and model size. When it comes to the feature dimension of 256, the performance significantly drops, which means too small feature dimensions are not feasible for our task.

		Full set		Subset	
Method	# Params	2DIoU(%)	3DIoU(%)	2DIoU(%)	3DIoU(%)
Two models	272 M	**85.29**	**82.72**	62.54	**60.04**
Two transformers	203 M	84.35	81.88	59.21	56.80
Two heads	136 M	84.06	81.51	57.97	55.47
Ours (c = 512)	102 M	85.10	82.57	**62.81**	59.97

Table 2. **Model size and performance trade-off.** We conduct the experiment on both the full set and subset of the MatterportLayout [3] dataset and evaluate with our proposed *disambiguate* metric.

		Full set		Subset	
Method	# Params	2DIoU(%)	3DIoU(%)	2DIoU(%)	3DIoU(%)
Ours (c = 1024)	172 M	**85.25**	**82.76**	**63.33**	**60.50**
Ours (c = 512)	102 M	85.10	82.57	62.81	59.97
Ours (c = 256)	80 M	84.47	81.90	60.39	57.83

Table 3. **Different image feature dimensions.** We conduct the experiment on the full set and subset of the MatterportLayout [3] dataset and evaluate with our proposed *disambiguate* metric.

Model	2D IoU (%)	3D IoU (%)
Train from scratch	85.10	82.57
Pretrain on ZInD-Simple	85.52	83.28
Pretrain on ZInD-All	**85.81**	**83.52**

Table 4. **Pretraining effectiveness** on MatterportLayout [3] with our proposed *disambiguate* metric. The pretraining on different types of ZInD [1] datasets indeed helps the model to disambiguate, and the more data for the pretraining stage, the more performance gain it has.

Pretrain with more data. MatterportLayout [3] has limited bi-layout samples, with only 15% of 1647 training images being re-annotated. However, most images in ZInD-Simple [1] (24,882) and ZInD-All [1] (50,916) have both *raw* and *visible* labels. Although ZInD also has ambiguity issues, we believe pretraining on ZInD with extensive and diverse bi-layouts can boost the model's performance on MatterportLayout. Therefore, we train one model from scratch and fine-tune two models pre-trained on ZInD-Simple and ZInD-All, respectively. The results in Table 4 demonstrate that pretraining indeed aids the model in disambiguating, with a more significant performance gain observed when more data is used during the pretraining stage. This suggests that with additional bi-layout annotations, our model has the potential to more effectively address the ambiguity issue.

(a) Extended prediction on extended annotation

(b) Enclosed prediction on enclosed annotation

Figure 7. **Failure case on the MatterportLayout [3] dataset.** 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.

7. Limitations

Our Bi-Layout model also has limitations. As shown in Fig. 7, we provide the predictions from our two branches, which aim to fit *extended* and *enclosed* annotations. We find that the main type of failure case comes from the large opening region, and there is no obvious room boundary to separate the different rooms. To address this difficult sce-

nario, we believe there is a need for more diverse bi-layout training data to ensure our model can learn the corresponding label properties.

8. Future Directions

We provide possible future research directions based on our current proposed method.

Cross-dataset training. Our main manuscript shows that pretraining on the large-scale ZInD [1] dataset can benefit the model performance evaluated on the MatterportLayout [3] dataset. This observation provides the possible direction for cross-dataset training, which may further improve the model performance.

Bi-Layout to multiple layouts. Our Bi-Layout model can generate two types of predictions. Based on our network design, it is possible to extend the number of predictions to more than two predictions. In the future, once the dataset provides multiple types of labels, multiple predictions can be achieved by simply adding more global context embeddings and the corresponding prediction heads. Most importantly, due to our shared feature guidance module design, those additional components for multiple predictions are very lightweight, which can still maintain the compactness of our model.

References

[1] 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, 2, 3, 5, 6

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

[3] 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, 4, 5, 6