Supplementary Materials for: End-to-end Graph-constrained Vectorized Floorplan Generation with Panoptic Refinement

Jiachen Liu[1†], Yuan Xue[2†], Jose Duarte[3], Krishnendra Shekhawat[4], Zihan Zhou[5], and Xiaolei Huang[1]

[1] College of Information Sciences and Technology, Penn State University
[2] Department of Electrical and Computer Engineering, Johns Hopkins University
[3] College of Arts and Architecture, Penn State University
[4] Department of Mathematics, BITS Pilani
[5] Manycore Tech Inc.

1 Implementation details

We implemented our proposed method using PyTorch [4] and ran all experiments on an NVIDIA RTX 3090 GPU. We set up the codebase based on the official implementation of LayoutTransformer [1]. The AdamW [2] optimizer was applied with a constant learning rate 3×10^{-4}. We trained our model end-to-end for 20 epochs with batch size 128. We applied student forcing to generate the draft floorplan sequence from scratch during testing, starting from the $\langle BoS \rangle$ token. Specifically, we used greedy decoding with top-k probabilities, then sampled from the probability distribution to obtain discrete coordinates. We set $k = 5$ in all experiments. All feature dimensions used in the proposed network were set to be 128. Based on experimental results, we chose 5 refinement iterations as the default setting in the refining stage.

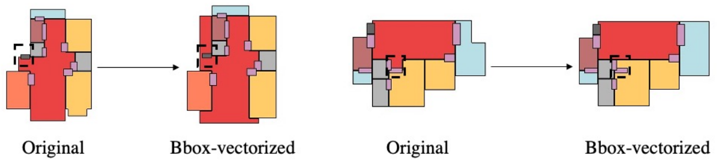

Fig. 1: Examples of noisy data introduced by bbox vectorization.

2 Model size and speed comparison

As shown in Table 1, compared with the two other representative works [1,3], although our model has more parameters, it achieves a good trade-off among

[†] These authors contributed equally to this work.

inference time, memory consumption, and generation performance. Although transformer models are parameter-heavy, our method is only slightly slower than [1] because the proposed panoptic refinement does not add much overhead compared to the autoregressive generation process.

Table 1: Number of parameter and speed comparison among different methods.

Methods	Parameters / M	Speed / FPS ↑	Memory / G ↓
LayoutTransformer [1]	1.27	5.65	1.25
HouseGAN++ [3]	0.64	2.83	1.6
Ours	2.51	4.22	1.27

3 User interaction and design customization

Our vectorized floorplan generation method allows flexible user interaction and design customization. For user interaction, our method generates multiple outputs with single input for user to choice from, which was greatly appreciated by design professionals during our user study. Our framework also allows users to easily specify input coordinates for certain rooms to customize size and location (see Fig. 2(b) below). Further, for design customization, the vectorized representation makes it easy to customize the size or location of rooms by directly modifying generated room coordinates (see Fig. 2(c)).

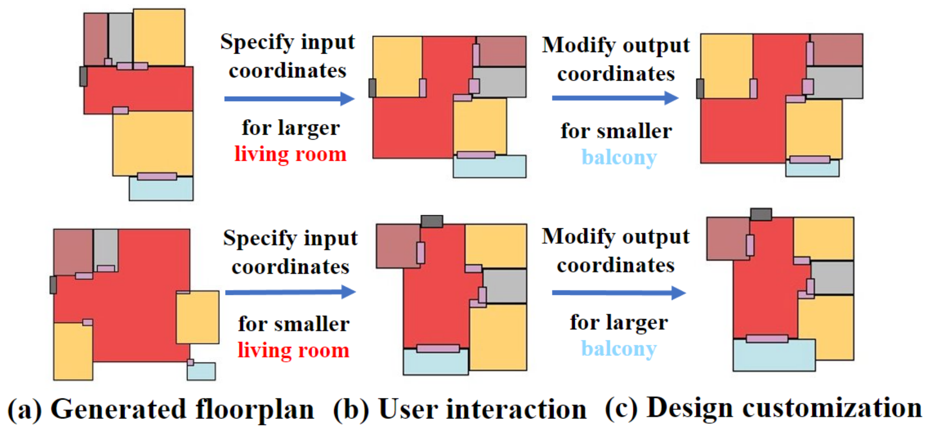

Fig. 2: Examples on user interaction and design customization.

References

1. Gupta, K., Lazarow, J., Achille, A., Davis, L.S., Mahadevan, V., Shrivastava, A.: Layouttransformer: Layout generation and completion with self-attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1004–1014 (2021)
2. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
3. Nauata, N., Hosseini, S., Chang, K.H., Chu, H., Cheng, C.Y., Furukawa, Y.: House-gan++: Generative adversarial layout refinement networks. arXiv preprint arXiv:2103.02574 (2021)
4. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)