HouseDiffusion: Vector Floorplan Generation via a Diffusion Model with Discrete and Continuous Denoising [Supplementary Material]

Mohammad Amin Shabani, Sepidehsadat Hosseini, Yasutaka Furukawa
Simon Fraser University
{mshabani, sepidh, furukawa}@sfu.ca

The supplementary document provides 1) ablation studies on the hyperparameters (e.g., the number of steps and the target output of the model); 2) more experimental results on the non-manhattan RPLAN dataset; 3) more analysis on our discrete denoising and AnalogBits [1]; and 4) additional qualitative examples of our system and competing methods. Please also see the supplementary video for more examples.

A. Diffusion Hyperparameters

Target output: In the diffusion process, we can train the network to predict either x_0 or ϵ. While both options are mathematically similar, we found that training to predict ϵ works better. Table 1 shows the results. This is because directly predicting the final coordinates (x_0) in the early reverse steps is a lot more challenging than predicting the per-step noise ϵ. We can see the same pattern in overfitting on a single data when the network can learn significantly faster when predicting x_0 comparing to ϵ, and we found it also consistent with the results of AnalogBits [1] when comparing the results on CIFAR-10 and IMAGENET datasets.

Number of steps and the noise schedule: Table 2 shows the sensitivity of our method to the noise schedule and the number of steps. Using a smaller number of steps leads to larger jumps between different time steps, which is harder to predict for the model, and using a larger number of steps needs more capacity to learn the distribution states of all steps. Similarly, using the cosine schedule proposed by Nichol and Dhariwal [4] works better by preventing abrupt changes comparing the Linear schedule proposed by Ho et al. [2].

Table 1. Training the network to predict ϵ works better in our task comparing with x_0 as the output of the model in each step.

Network's output	Diversity (↓)	Compatibility (↓)
x_0	121.2±1.9	4.2±0.0
ϵ	38.8±0.9	2.2±0.0

Table 2. Results of hyper-parameter tuning on the number of steps and noise schedule for our diffusion model. We fix our experiments to 1000 steps with cosine noise schedule [4].

Noise Schedule	Number of steps	Diversity (↓)	Compatibility (↓)
Linear	1000	10.5±0.1	2.7±0.04
Cosine	100	19.0±0.8	3.1±0.0
	1000	9.5±0.1	2.5±0.0
	4000	12.0±0.3	3.2±0.03

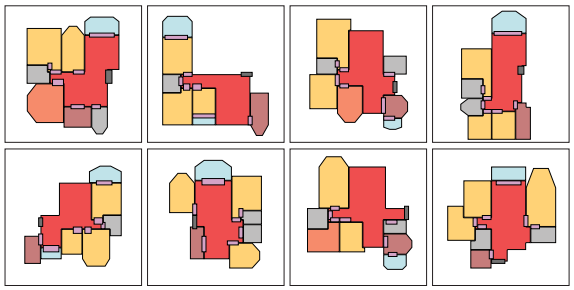

Figure 1. Examples of the created non-manhattan RPLAN dataset.

B. Non-Manhattan RPLAN

To further evaluate our method, we create a non-manhattan version of RPLAN by randomly adding two additional corners to an outer edge of a floorplan. Concretely, for each room in the house, we break the wall with the highest distance from the center of the house. We instead add two additional corners with a random distance from the center of the room in one axis and center of the wall in the other axis of 2D coordinates. We keep the new corners with a probability of 50% if they do not cause an overlap with the other rooms of the house. Figure 1 shows examples of our non-manhattan version of RPLAN.

C. Analysis of AnalogBits

Chen *et al*. [1] proposed AnalogBits which directly uses the continuous formulation of diffusion models to denoise a

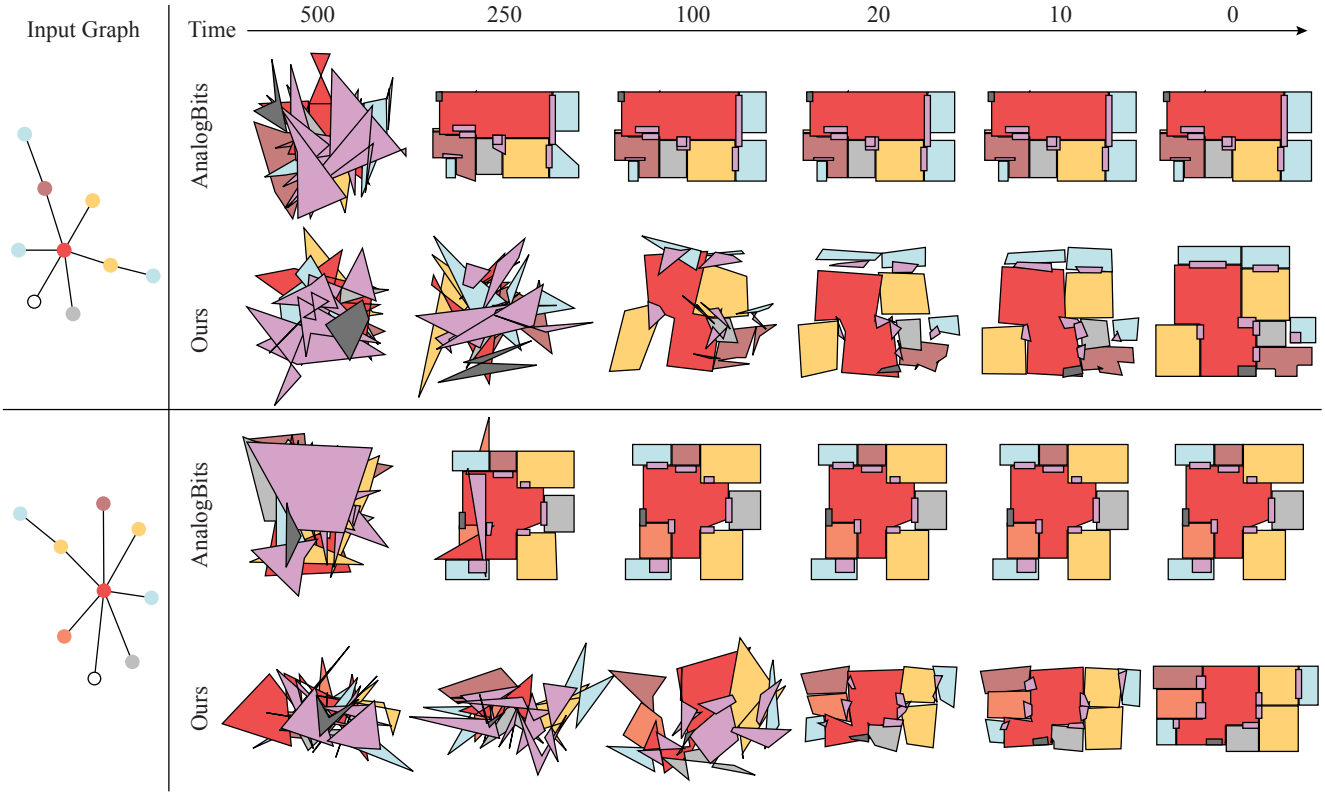

Figure 2. Comparison of our continuous and discrete denoising with AnalogBits based on a given bubble diagram. Our method benefits from all of the steps to refine the generated floorplan while AnalogBits has almost no changes on the final prediction in at least the last 100 steps meaning that the network predicts a simple mapping of input values to 0 and 1 based on a threshold.

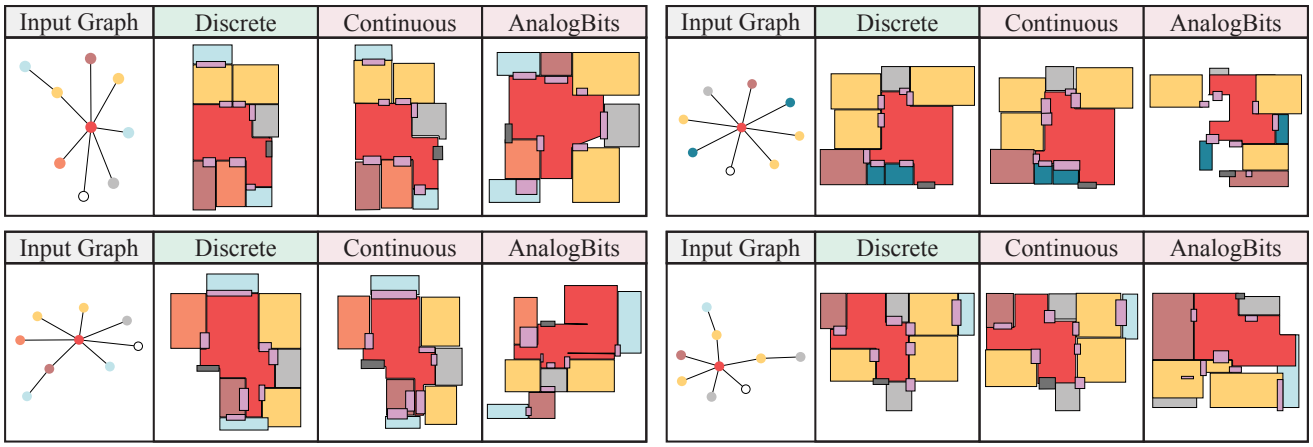

Figure 3. Additional comparison of generated floorplans by our method in both discrete and continuous denoising compared with the results of using AnalogBits for our model. While the results of AnalogBits have significantly lower quality, our model gets high-quality results in continuous head, and the discrete head further improves the results to provide a precise location for each corner.

random Gaussian sample to binary values (-1 and 1 in practice) and convert them directly to discrete values by simple thresholding. In Figure 2 we visualized different steps of the denoising process obtained by the formulation of AnalogBits compared to our continuous and discrete denoising. We visualize the intermediate steps by considering them as the final output (i.e., using thresholding and binary-to-decimal mapping). The generated floorplans of AnalogBits

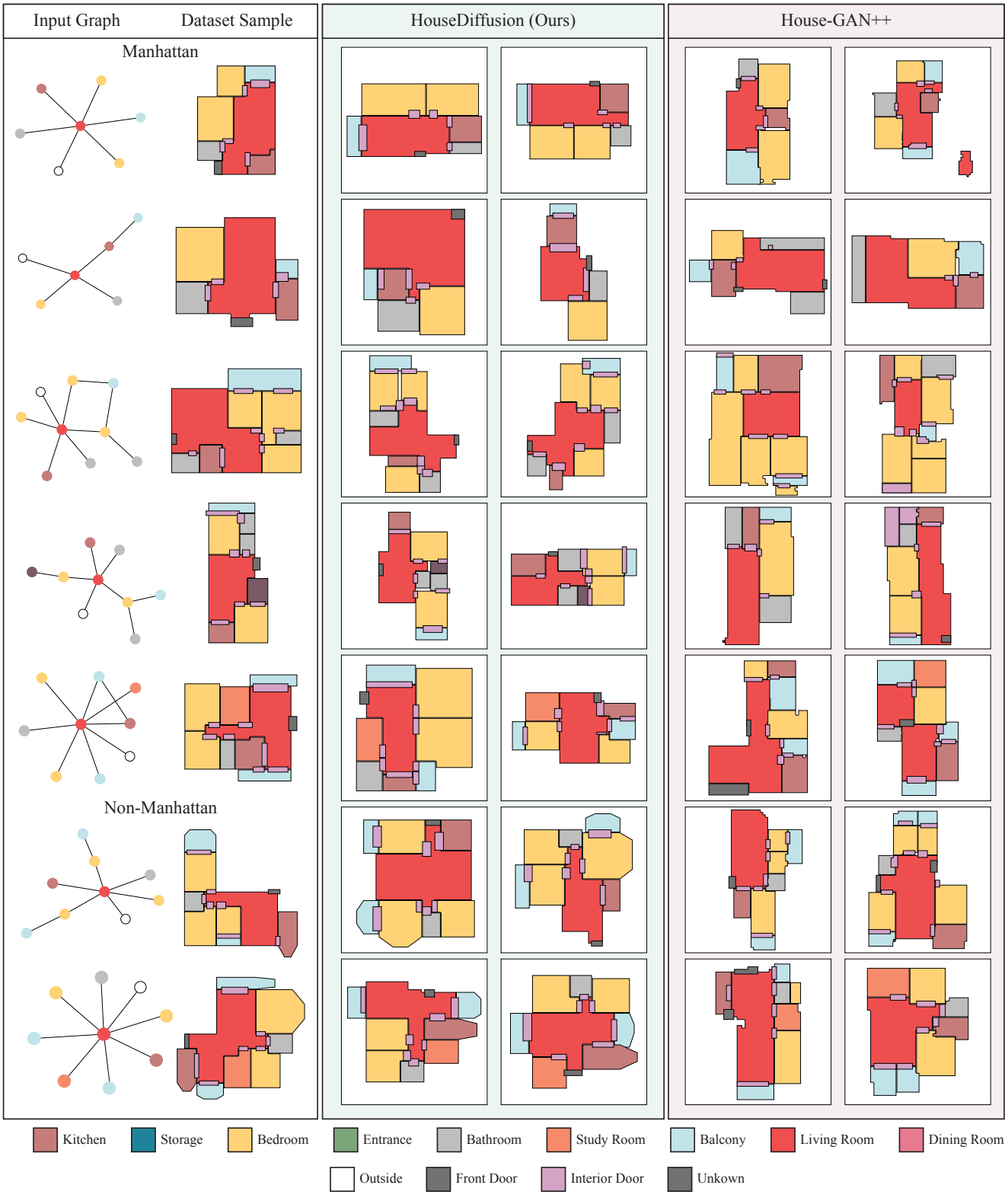

Figure 4. Additional generated floorplan samples against House-GAN++ [3]. Our results look more diverse and higher quality, where the major issue of House-GAN++ is duplicate or missing rooms, ignoring the input constraint.

are roughly the same in the last 100 steps, while our method uses every step to continuously denoise the generated floorplan. We hypothesize that in the case of AnalogBits, when the noise coefficient $\sqrt{(1-\gamma_t)}$ is small (See Equation 1 in the main paper), the network learns a simple mapping of $f : x_t \rightarrow \lfloor x_t + \frac{1}{2} \rfloor$ instead of the dataset distribution, which reduces the final quality of the generated samples.

D. Qualitative Examples

Figure 3 shows additional comparison of our method with AnalogBits and both continuous and discrete outputs of our model. In addition, we have provided more examples of generated floorplans by our method versus the results of House-GAN++ [3] in Figure 4. Please see the provided video in the supplementary for better visualization of the reverse process of our method as well as additional examples.

References

[1] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022. 1

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1

[3] Nelson Nauata, Sepidehsadat Hosseini, Kai-Hung Chang, Hang Chu, Chin-Yi Cheng, and Yasutaka Furukawa. House-gan++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13632–13641, 2021. 3, 4

[4] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 1