Synthesizing Commercial Floorplans via a Controllable Diffusion Framework

Wenming Wu[1,2][0000−0002−0640−8520], Yuntao Wang[1][0009−0003−2827−5727], and Liping Zheng[1(✉)][0000−0001−5071−9628]

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, Anhui, P.R. China
wwming@hfut.edu.cn
wyt@mail.hfut.edu.cn
zhenglp@hfut.edu.cn

[2] Anhui Province Key Laboratory of Industry Safety and Emergency Technology (Hefei University of Technology), Hefei 230601, Anhui, P.R. China

Abstract. Automatic generation of commercial floorplans can significantly improve spatial design efficiency and customer experiences. Traditional manual methods are labor-intensive, costly, and prone to inconsistency. While deep learning methods have shown promise, their application to commercial scenarios faces challenges, including limited datasets and insufficient controllability. To address these issues, we introduce a diffusion-based generative framework tailored specifically for commercial floorplan synthesis. We first create a synthetic dataset of diverse commercial floorplans using traditional geometric approaches. Then, leveraging a diffusion-based architecture, our method generates high-quality, realistic commercial floorplans. A subsequent vectorization step converts generated images into practical vector formats. Additionally, our approach supports practical constraints such as boundary masks, path constraints, and bubble diagrams, enabling precise and flexible control. Experiments demonstrate that our method outperforms existing generative models quantitatively and qualitatively. Our work represents a pioneering effort in applying diffusion-based generative methods to commercial floorplan design, providing a flexible and efficient solution for spatial designers.

Keywords: Commercial floorplan · Diffusion model · Constrained generation.

1 Introduction

Layout design plays a crucial role across various domains in modern society, particularly in architecture, urban planning, interior design, and industrial produc-

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62372152), the Fundamental Research Funds for the Central Universities of China (Grant No. PA2025GDSK0035), and the Open Project Program of the State Key Laboratory of CAD&CG (Grant No. A2412), Zhejiang University. Liping Zheng (zhenglp@hfut.edu.cn) is the corresponding author.

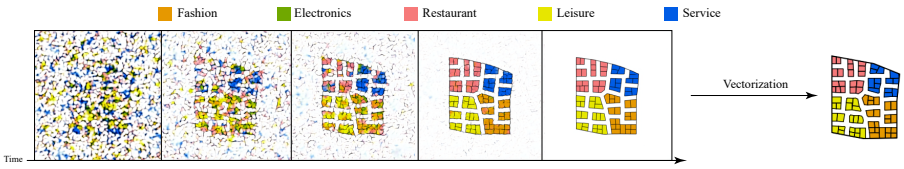

Fig. 1. Overview of the proposed diffusion-based framework. The framework synthesizes commercial floorplans by progressively denoising random noise through a diffusion process over multiple timesteps. Distinct colors represent different functional zones. The final pixel-level layout is subsequently converted into a vectorized representation.

tion. Based on the scale, layout generation tasks can be categorized into small-scale [25,24,32], mid-scale [1,17,35], and large-scale scenarios [28,26,2]. Mid-scale layouts encompass settings such as shopping malls, supermarkets, and hospitals, among others. For the mid-scale commercial environment, efficient spatial layout design has become core to enhancing scenario value, with its importance spanning scenarios such as shopping centers, supermarkets, and public service spaces. Therefore, the automatic generation of commercial floorplans holds significant practical and economic value.

Traditional commercial floorplan design, reliant on manual labor, is time-consuming, costly, and prone to inconsistent quality outcomes due to subjective interpretations and variations in experience among designers. Although some computational optimization-based automatic methods [36,7,40,5] have been introduced to address these issues, they often suffer from complex procedures, limited automation, and poor generality across different commercial scenarios, restricting their practical usability and application scope. Recent advances in deep learning have drastically transformed multiple fields, including residential building floorplan design [37,15,33,14], and urban planning [11,29,38]. Deep generative models [10,13] have demonstrated remarkable capabilities in automating complex spatial design tasks. By learning optimal patterns from large datasets, these models can efficiently generate high-quality layouts automatically, significantly reducing manual effort and improving consistency. Despite these potential advantages, applying deep learning methods to commercial floorplan design still faces challenges. First, there remains a lack of publicly available datasets specifically tailored for commercial environments, severely limiting model training and generalization capabilities. Second, existing deep learning methods developed for residential or urban environments often fail to effectively capture the unique spatial partitioning and functional requirements of commercial floorplans. Finally, commercial design tasks often require strong controllability to meet specific practical constraints and user intents common in design projects.

In this paper, we propose a novel framework (Fig. 1) that leverages state-of-the-art diffusion models [13] to automatically synthesize high-quality commercial floorplans, integrated with robust controllability and practical usability. Specifically, [35] is leveraged to synthesize large-scale, structured layout datasets. Subsequently, we employ a diffusion-based generative framework designed ex-

plicitly for high-resolution image synthesis, effectively capturing the complex spatial structures inherent in commercial floorplans. Generated pixel-based layouts are further converted into high-quality vector representations through a vectorization post-processing step. Additionally, our proposed model explicitly incorporates multiple practical constraints, such as boundary masks, path constraints, and bubble diagram constraints, allowing users to precisely control and customize generated floorplans according to specific functional and commercial requirements. This paper presents the first systematic and implemented deep learning-based approach specifically designed for commercial floorplan generation, utilizing a synthetic dataset of commercial layouts. Our method not only fills a critical research gap in this domain but also establishes a foundational framework for applying deep learning to automated spatial design in commercial environments. This pioneering work marks a significant step forward for the field. We incorporate spatially detailed constraints, enabling precise zoning and circulation control. Our method demonstrates strong constraint adherence, outperforming general approaches. These are not reused designs but targeted adaptations for real-world scenarios. Our contributions are twofold:

- We introduce a diffusion-based generative framework tailored for commercial floorplan synthesis. Trained on structured synthetic datasets, the model effectively captures complex spatial semantics and structures.
- We incorporate multiple design constraints—such as spatial boundaries, pedestrian flow paths, and functional zoning diagrams—into the generative pipeline, achieving high controllability and adaptability.

2 Related work

2.1 Commercial floorplan generation

Commercial floorplan generation involves not only the functional zoning of spaces but also the geometric arrangement of architectural units. Due to the lack of publicly available datasets, most existing approaches rely on traditional rule-based or optimization-based methods. Wu et al. [36] formulates the layout generation task as a Mixed-Integer Quadratic Programming (MIQP) problem and proposes a hierarchical framework for generating interior layouts, applicable to scenarios such as shopping malls and office buildings. However, the method requires numerous manually designed parameters, limiting its scalability and generalization to more complex layouts. To incorporate human behavioral factors, Feng et al. [7] integrates crowd simulation into the layout synthesis process, generating crowd-aware commercial floorplans. Similarly, Zhang et al. [40] introduces a parameterized method that follows commercial floorplan design principles by organizing object groupings into configurable patterns. Additionally, Hua et al. [17], though originally targeting urban-scale layouts, propose an integer programming model that can be adapted to commercial environments such as shopping centers by adjusting configuration parameters. Overall, existing commercial floorplan generation methods rely on manual rules and heuristic strategies, lacking generalizability and data-driven flexibility for complex environments.

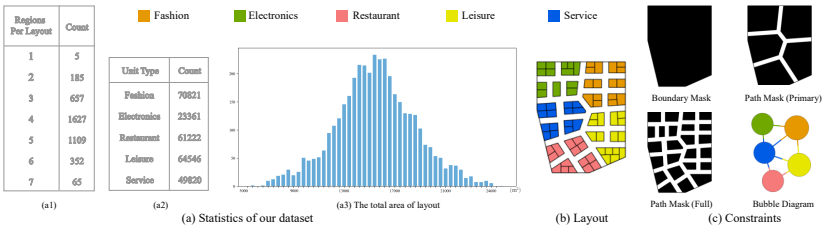

Fig. 2. Our dataset. (a) Statistics on the occurrence of region number per layout (a1), each unit type (a2), and the number of layouts according to the total area (a3). (b) Example from our synthetic dataset of a commercial floorplan. (c) Each sample includes four constraint images.

2.2 Diffusion models for layout generation

There are many generative models in deep learning methods [19,4,13], among which diffusion models have emerged as powerful generative frameworks in various fields. VQGAN [4] combines the efficiency of convolutional neural networks with the expressiveness of transformer architectures by learning a discrete codebook of visual tokens. It enables high-resolution image synthesis through strong compression while preserving perceptual quality. Since the introduction of Denoising Diffusion Probabilistic Models (DDPM) by [13], diffusion-based methods have achieved state-of-the-art results in tasks such as text-to-image generation [39], image restoration [9], and residential floorplan synthesis [31,41]. [18] further explored layout-to-image synthesis using diffusion. The Transformer-based Diffusion architecture (DiT) proposed by [27] significantly enhanced image generation performance and was later extended to other domains, including image editing [6] and human motion generation [8], demonstrating the model's broad applicability. Diffusion models are also applied to the generation of residential floor plans [31,14,16]. To the best of our knowledge, no prior work has explored deep generative frameworks tailored specifically for commercial floorplan design. Our work is the first to address this gap, introducing a structure-aware, constraint-driven diffusion model for generating realistic and controllable commercial floorplans.

3 Method

3.1 Representation

A commercial floorplan can be formally represented as a set of regions, denoted as $G = \{R_i\}$, where each region $R_i = \{P_i, t\}$ consists of a polygonal shape $P_i = \{v_i\}$ and a functional type label t. The polygon P_i is defined by a set of vertices $\{v_i\}$, and the label t specifies the functional category of the region, such as retail, storage, or service, which is typically visualized using distinct colors in the floorplan. All regions are non-overlapping and collectively describe the

occupied functional space within the floorplan boundary. The remaining areas within the boundary that are not occupied by any defined region are considered unassigned or unoccupied and are treated as passable path spaces, facilitating movement and circulation throughout the environment.

3.2 Dataset

To support our research on commercial floorplan generation, we construct a synthetic dataset tailored to this domain. Due to the lack of publicly available commercial floorplan datasets, we build upon the RPLAN dataset [37] by extracting polygonal spatial boundaries. To simulate realistic commercial layouts with irregular shapes, we remove selected vertices to introduce non-rectilinear contours. We then apply CVTLayout [35] to generate diverse and semantically meaningful commercial floorplans based on these modified boundaries. This pipeline yields 4,000 synthetic commercial floorplans. As shown in Fig. 2(a), each sample includes five components (Fig. 2(b-c)):

- **Layout image:** RGB image (256×256) showing the full layout with color-coded functional zones.
- **Boundary mask:** Binary mask indicating spatial boundaries.
- **Path mask (Primary):** Binary mask showing main navigable paths.
- **Path mask (Full):** Binary mask showing all passable areas.
- **Bubble diagram:** RGB image (256×256) showing high-level functional zoning and adjacency.

Although synthetic data differs from real-world data, it is meticulously designed to capture the structural and functional characteristics of commercial layouts. This design ensures both its representativeness and generalizability, making it a reliable proxy for training and evaluation purposes. We define five representative commercial functional categories: *Fashion*, *Electronics*, *Restaurant*, *Leisure*, and *Service*. These categories are deliberately chosen to align with typical commercial configurations and consumer behavior patterns, thereby enhancing the dataset's practical relevance and applicability. Note that our framework is not limited to these and is easily extensible.

3.3 Network architecture

LayoutVAE To reduce computational complexity and facilitate effective generative modeling, we first train a Variational Autoencoder (VAE), named LayoutVAE, to compress high-dimensional input floorplan images into low-dimensional latent representations. Specifically, we employ an AutoencoderKL [22] framework as our backbone, consisting of an encoder and a decoder. The parameters of the pre-trained LayoutVAE model will be frozen during the subsequent training process of the diffusion model. We follow standard VAE training using pixel-wise MSE and KL divergence to learn compact latent vectors.

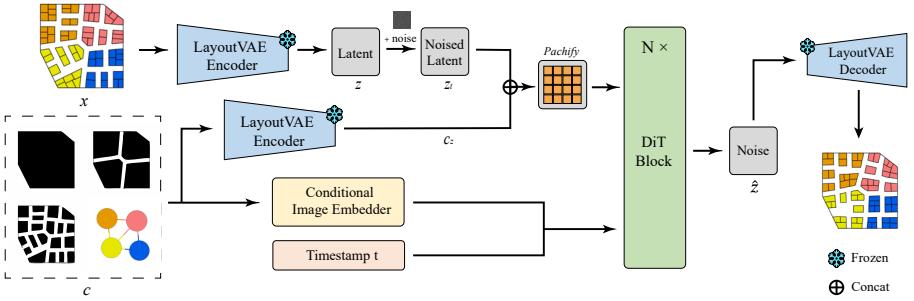

Fig. 3. The pipeline of our diffusion-based generative model. The layout image and the conditional image are taken as inputs. The layout image is first encoded into the latent space by the encoder and then noised. For the conditional image, on one hand, its features are extracted by an image encoder, combined with the timestamp, and fed into the DiT-block; on the other hand, it is also encoded into the latent space by the encoder to align with the layout image's latent space. Subsequently, the conditional image's latent features are concatenated with the noisy layout latent features along the channel dimension. Following DiT [27], we divide images into 16×16 patches, embed them as tokens, and input them into the Transformer. Then, they are fed into the DiT-block, and finally, the output of the model is reconstructed into a floorplan.

Diffusion framework Fig. 3 illustrates the basic model framework during training. This framework uses the DiT model [27] as its backbone, modeling the diffusion process as a Markov chain. In the training process, we take a floorplan image x as input, which contains the image layout information of interest. We then compress the layout image x into the latent space through a pretrained LayoutVAE encoder to obtain its latent representation z. Next, we randomly sample noise from a standard normal distribution, determine the maximum timestep T, and select a positive integer t from $[0, T]$ as the timestep, computing the noisy layout feature map z_t at this timestep.

The model can accept images such as boundaries, paths, and bubble diagrams as conditions c. These conditional images have the same shape as the layout plan and provide additional information to help the model generate images that better meet requirements. The DiT framework fuses features of the conditions through implicit encoding. Considering the feature extraction needs of conditional images, we use ResNet-18 as the Conditional Image Embedder, which effectively extracts the features of the conditional images. ResNet-18 extracts features from resized 256×256 constraint images, producing (64, 32, 32) feature maps. For timestep embedding, it is encoded by an MLP and concatenated with constraint features along the channel dimension before entering the DiT-block module. Constraint images are encoded into the latent space using LayoutVAE, then concatenated with the layout latent to ensure consistency. This combination allows the model to consider both temporal and conditional information, enhancing its representational capability. During the generation of layout plans, constraints from image conditions focus more on local features. To better uti-

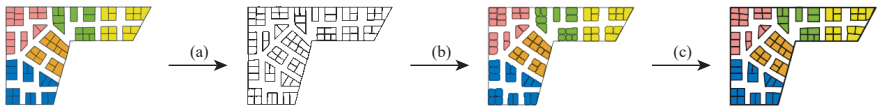

Fig. 4. Illustration of the vectorization post-processing pipeline. (a) Mask extraction and morphological processing; (b) Contour detection and semantic classification; (c) Polygon approximation and simplification.

lize these local features, we adopt channel-wise concatenation to enhance the model's perception of local information, as described in [20,34]. Specifically, we align the image conditions with the latent space of z_t through the LayoutVAE encoder to obtain c_z. We then combine the condition c_z with the layout's latent representation z_t via channel-wise concatenation, which allows the model to integrate more information across channels and enrich the feature representation. Assuming $z_t = (C_1, H, W)$ and $c_z = (C_2, H, W)$, where $C1, C2$ represents the number of channels, H represents the height, and W represents the width. the combined z_t can be expressed as

$$z_t = (C_1 + C_2, H, W) \tag{1}$$

Subsequently, we feed it into the DiT-block module. The module captures information through a multi-head self-attention mechanism, ultimately producing the prediction result $\hat{z}$. Finally, the model's output $\hat{z}$ is reconstructed from the latent space into a floorplan through the LayoutVAE decoder. To measure the difference between the model's prediction and the ground truth, we use mean squared error (MSE) as the loss function, which effectively guides model training and optimizes the model to minimize prediction errors.

3.4 Vectorization

We convert raster floorplans generated by our diffusion model into scalable vector formats via a three-step pipeline (Fig. 4):

- **Mask extraction and morphological processing.** Extract binary masks from the raster image to delineate spatial boundaries. Apply morphological operations (erosion, dilation, opening, closing) to remove generation artifacts and noise.
- **Contour detection and semantic classification.** Detect closed contours and their hierarchical relationships. Classify contours via color analysis, mapping dominant colors to semantic labels using a predefined lookup table.
- **Polygon approximation and simplification.** Approximate contours using the Douglas-Peucker [3] algorithm and alignment techniques to generate clean, editable vector representations suitable for architectural design workflows.

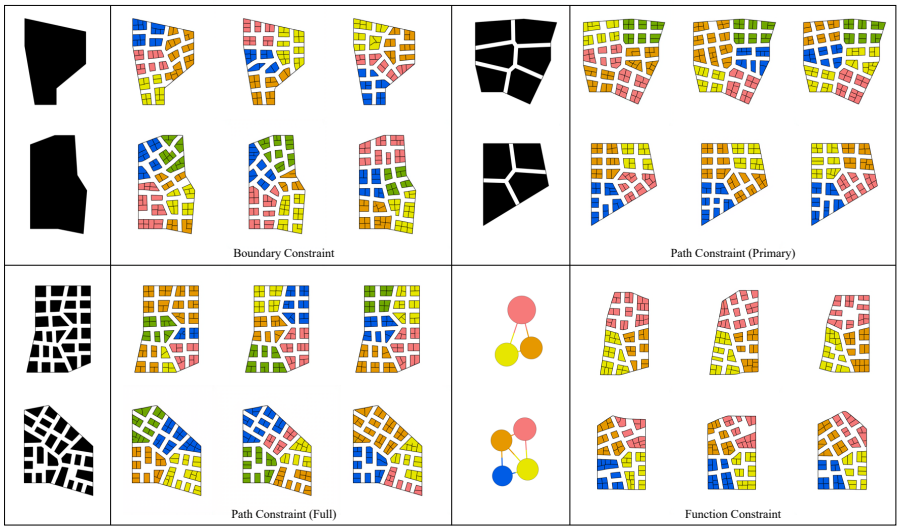

Fig. 5. Examples of constrained layout generation.

3.5 Constrained generation

To meet diverse requirements in real-world commercial scenarios, we incorporate multiple forms of image-based constraints, including boundary masks, primary path masks, full path masks, and bubble diagram constraints. These constraints enable precise control over the generated layouts, significantly enhancing their practical applicability. Fig. 5 demonstrates the results generated by our method under different constraints. We use constraints to improve control and functional layout. Unconditional generation often lacks structural guidance, resulting in overlaps or poor flow, which limits its applicability. Our results show that constraints significantly enhance structural integrity and layout quality. Our framework supports basic interactive design by allowing users to modify constraints like boundaries or bubble diagrams to control layout generation. For conflicting constraints, we plan to use a constraint validation or prioritization mechanism (e.g., giving boundary constraints higher priority).

4 Experiment

Our proposed model is implemented based on PyTorch and optimized using the Adam optimizer [21]. The model is trained and tested on an NVIDIA GeForce RTX 3090. To standardize the evaluation criteria, we divide the dataset into a training set containing 3,000 samples and a test set containing 1,000 samples, and extract layouts and their corresponding constraints from the test set. For each constraint, we generate 5 commercial floorplans, resulting in a total of 5,000 floorplans for quantitative analysis. To accelerate image generation, we

Table 1. Quantitative analysis of constrained generation.

Constraint Method	Constraint	FID ↓	Precision ↑	Recall ↑
Non-Concat	Boundary	53.949	0.698	0.604
Channel-Concat	Boundary	**10.525**	**0.8226**	**0.847**
Non-Concat	Path (Primary)	32.609	0.691	0.803
Channel-Concat	Path (Primary)	**7.009**	**0.942**	**0.938**
Non-Concat	Path (Full)	11.586	0.863	0.884
Channel-Concat	Path (Full)	**5.633**	**0.996**	**0.998**
Non-Concat	Function	**7.695**	0.807	0.800
Channel-Concat	Function	12.304	**0.868**	**0.803**

use $T = 250$ as the time steps for model inference in the experiments. For our method, floorplan generation takes approximately 2.8 seconds, followed by around 1 second for the vectorization process. This is significantly faster than CVTLayout [35], which requires about 20 seconds to generate a single layout. In terms of model size, LayoutVAE contains about 1.6 million parameters, whereas DiT has around 140 million parameters.

We conduct a quantitative evaluation of the generated results using several widely used metrics in generative tasks: FID (Fréchet Inception Distance) [12], Precision and Recall [23]. FID is primarily used to calculate the feature vectors of generated image data and real image data. A lower score means the generated results are closer to real images. Precision represents the proportion of generated images that belong to the real distribution. A higher value means the details and structure of generated images are closer to real images, with higher quality. Recall reflects the proportion of the real image distribution covered by generated images. A higher value indicates that generated images cover more categories or styles of real data, showing stronger diversity.

4.1 Ablation studies

In the DiT model, it extracts input condition features and fuses constraint information using an attention mechanism. We refer to the approach as **Non-Concat**. We adopt **Channel-Concat** to enhance the model's perception of local conditional information, combining constraint information as additional channels with the original input. Therefore, we analyze and evaluate Non-Concat and Channel-Concat with different constraints. The latter improves generation quality and better satisfies spatial constraints, validating the core innovation.

Table 1 shows the results evaluated using multiple metrics. Layouts generated with constraints improve model performance to a certain extent, with different combinations of constraints and methods producing differentiated effects. Introducing constraints enhances the generated samples' performance across all metrics, indicating that constrained generation improves distribution matching, fidelity, and diversity to varying degrees. The model's performance improves as constraints become finer, achieving the best results when using the full path

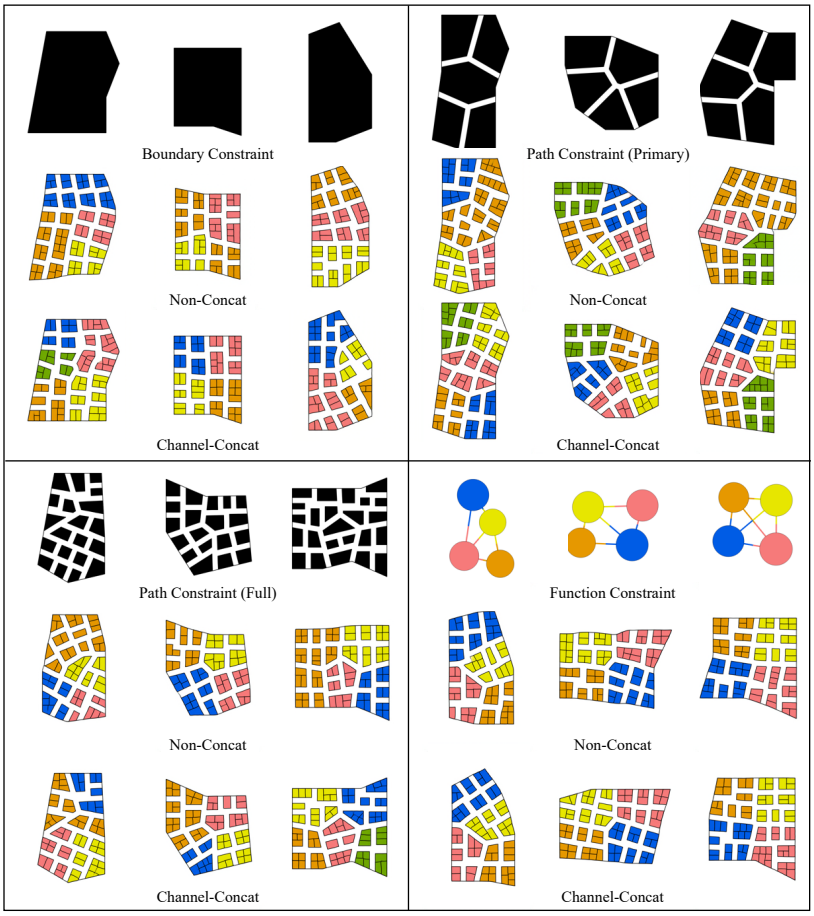

Fig. 6. Ablation analysis of constrained generation.

constraint for generation. Channel concatenation explicitly increases the input dimensions, transforming constraints into spatial features directly computable by the model. This enhances the model's perception of local features and reduces potential attenuation or ambiguity of constraint signals during feature fusion in Non-Concat. When generating with boundary constraint, path constraint (primary), and path constraint (full), Channel-Concat outperforms in all metrics. For generations with function constraints, while Channel-Concat slightly lags in FID, it achieves better performance in Precision, Recall. Overall, considering all quantitative metrics, Channel-Concat is more suitable for constrained generation in this work compared to Non-Concat.

Fig. 6 demonstrates the results generated by different constraint methods and conditions. For strong constraints like boundary constraints, Non-Concat cannot strictly adhere to the input constraints. Taking the boundary constraint as an

Table 2. Quantitative comparisons with baseline generative models.

Method	Constraint	FID ↓	Precision ↑	Recall ↑
VQGAN [4]	Unconstraint	139.196	0.003	0.038
Ours	Unconstraint	**17.876**	**0.676**	**0.689**
Pix2Pix [19]	Function	128.737	0.047	0.070
LDM [30]	Function	14.843	**0.875**	0.666
Ours	Function	**12.304**	0.868	**0.803**

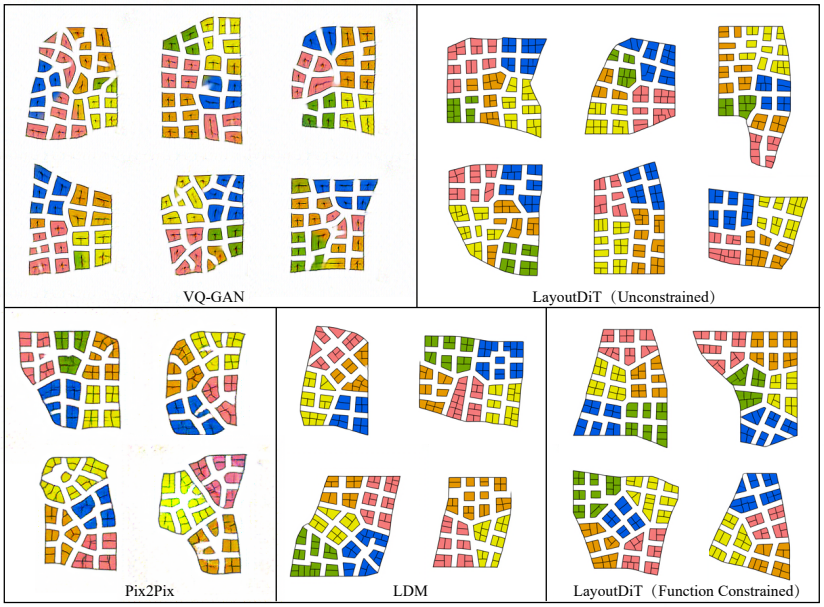

Fig. 7. Comparison of floorplans generated by different methods.

example, although the results generated by Non-Concat are similar in shape to the input boundary mask, they are not completely consistent, whereas Channel-Concat results exactly match the boundary shape. Taken together, Channel-Concat outperforms Non-Concat and is more conducive to precise layout control.

4.2 Comparision

Our work introduces the first deep learning framework specifically for commercial floorplan synthesis, addressing a largely unexplored area. Due to the lack of dedicated work, we evaluate our method by comparing it with representative GAN, autoregressive, and diffusion models for image generation, as well as the recent CVTLayout [35]. In this section, we conduct a comparative analysis of our

method with commonly used generative methods such as Pix2Pix [19], VQGAN [4], and LDM [30]. For constrained generation, we adopt the function constraint as a representative case for evaluation.

As shown in Fig. 7, in the unconstrained case, VQGAN produces blurred and grainy structure division within the regions and fails to construct clear shape and structural features. In contrast, our method achieves a clearer division in detail. As shown in Table 2, VQGAN performs poorly on core metrics such as FID. The high FID value reflects a significant difference between the generated samples and the real distribution, and there are also deficiencies in the quality and diversity of the generated samples. When generating with a function constraint, the results generated by Pix2Pix do not capture the features of the layout regions well in terms of shape and perform poorly on all metrics. LDM, with its latent space diffusion mechanism, outperforms the earlier methods on all metrics. Both LDM and our approach yield high-quality generated results. LDM stands out in the Precision metric, indicating a strong ability to follow function constraints, but its low Recall value suggests that there is still room for improvement in the diversity of the generated samples. Our method achieves the best results on most metrics. Although it is slightly inferior to LDM in the Precision metric, it achieves a better balance between distribution matching, generation quality, and diversity. Overall, the targeted design of our method enables it to demonstrate superior comprehensive performance in the commercial floorplan generation task, especially in the generation with constraint conditions.

Comparison with CVTLayout We further compare our method with the results generated by CVTLayout [35], a representative traditional optimization-based approach. CVTLayout requires a fixed clipping process during path generation, so the paths generated by this method have certain limitations. Especially in cases of extreme or narrow spatial conditions, it may lead to overly small regions or narrow shapes in the layout, as shown in the area marked by the red border in Fig. 8(a). In contrast, the diffusion-based generative model demonstrates greater flexibility and can effectively handle irregular and challenging spatial constraints. Our generation time (3.8s) is much faster than CVTLayout (20s), and Fig. 8 shows our model's robustness under irregular boundaries, showing its ability to handle extreme constraints. Unlike CVTLayout, our method also supports bubble diagram constraints, showing stronger controllability.

5 Conclusion

This paper proposes a diffusion-based framework for generating commercial floorplans. We create a synthetic dataset to train and evaluate the model, leveraging diffusion models' generative power to produce high-quality, high-resolution layouts. Our method incorporates practical constraints (boundaries, paths, and functional requirements) to enable precise control over generation, thereby enhancing flexibility in design. Our model has several limitations. It struggles with complex or conflicting constraints such as irregular boundaries, and currently

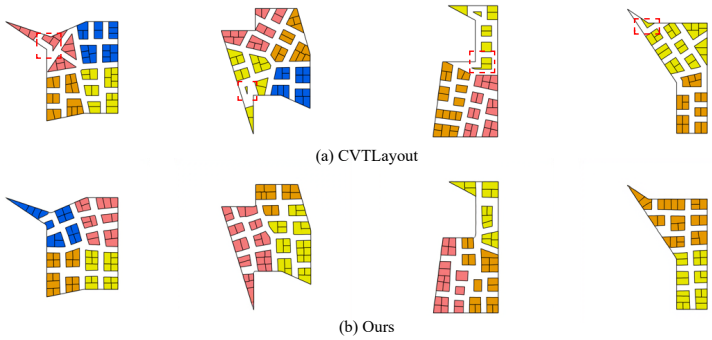

Fig. 8. Comparison between our diffusion-based method and CVTLayout.

supports only polygon-based layouts rather than curved or free-form shapes. Since real-world datasets are unavailable, we rely on synthetic data that, while capturing spatial logic and customer flow, simplifies complex layouts and omits practical factors such as fire safety and utilities. Moreover, business logic and user behavior are beyond our scope: we only simulate customer flow through path constraints without modeling aspects like dwell time or visibility. There are still many areas worthy of exploration and improvement in future research. The framework is modular and extensible, offering a plug-and-play solution for multimodal generation. With additional components (e.g., vision–language encoders), it can seamlessly support diverse modalities such as text and speech. We can develop cross-modal constraint inputs for fine-grained generation control. Additionally, we can integrate user feedback into an interactive design framework that dynamically aligns outputs with practical needs, improving applicability.

References

1. Berseth, G., Haworth, B., Usman, M., Schaumann, D., Khayatkhoei, M., Kapadia, M., Faloutsos, P.: Interactive architectural design with diverse solution exploration. IEEE transactions on visualization and computer graphics **27**(1), 111–124 (2019)
2. Chen, Z., Song, P., Ortner, F.P.: Hierarchical co-generation of parcels and streets in urban modeling. Computer Graphics Forum **43**(2), e15053 (2024)
3. DOUGLAS, D.H., PEUCKER, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica **10**(2), 112–122 (1973). https://doi.org/10.3138/FM57-6770-U75U-7727
4. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
5. Fahmy, S.A., Alablani, B.A., Abdelmaguid, T.F.: Shopping center design using a facility layout assignment approach. In: 2014 9th International Conference on Informatics and Systems. pp. ORDS–1–ORDS–7 (2014)
6. Feng, K., Ma, Y., Wang, B., Qi, C., Chen, H., Chen, Q., Wang, Z.: Dit4edit: Diffusion transformer for image editing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 2969–2977 (2025)

7. Feng, T., Yu, L.F., Yeung, S.K., Yin, K., Zhou, K.: Crowd-driven mid-scale layout design. ACM Trans. Graph. **35**(4), 132–1 (2016)
8. Gan, Q., Ren, Y., Zhang, C., Ye, Z., Xie, P., Yin, X., Yuan, Z., Peng, B., Zhu, J.: Humandit: Pose-guided diffusion transformer for long-form human motion video generation. arXiv preprint arXiv:2502.04847 (2025)
9. Garber, T., Tirer, T.: Image restoration by denoising diffusion models with iteratively preconditioned guidance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 25245–25254 (2024)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
11. He, L., Aliaga, D.: Globalmapper: Arbitrary-shaped urban layout generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 454–464 (2023)
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
14. Hong, S., Zhang, X., Du, T., Cheng, S., Wang, X., Yin, J.: Cons2plan: Vector floorplan generation from various conditions via a learning framework based on conditional diffusion models. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 3248–3256 (2024)
15. Hu, R., Huang, Z., Tang, Y., Van Kaick, O., Zhang, H., Huang, H.: Graph2plan: Learning floorplan generation from layout graphs. ACM Transactions on Graphics (TOG) **39**(4), 118–1 (2020)
16. Hu, S., Wu, W., Wang, Y., Xu, B., Zheng, L.: Gsdiff: Synthesizing vector floorplans via geometry-enhanced structural graph generation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 17323–17332 (2025)
17. Hua, H., Hovestadt, L., Tang, P., Li, B.: Integer programming for urban design. European Journal of Operational Research **274**(3), 1125–1137 (2019)
18. Inoue, N., Kikuchi, K., Simo-Serra, E., Otani, M., Yamaguchi, K.: Layoutdm: Discrete diffusion model for controllable layout generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10167–10176 (2023)
19. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
20. Ju, X., Zeng, A., Zhao, C., Wang, J., Zhang, L., Xu, Q.: Humansd: A native skeleton-guided diffusion model for human image generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15988–15998 (2023)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
23. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. Advances in neural information processing systems **32** (2019)

24. Laignel, G., Pozin, N., Geffrier, X., Delevaux, L., Brun, F., Dolla, B.: Floor plan generation through a mixed constraint programming-genetic optimization approach. Automation in Construction **123**, 103491 (2021)
25. Liu, H., Yang, Y.L., AlHalawani, S., Mitra, N.J.: Constraint-aware interior layout exploration for pre-cast concrete-based buildings. The Visual Computer **29**(6), 663–673 (2013)
26. Nishida, G., Garcia-Dorado, I., Aliaga, D.G.: Example-driven procedural urban roads. Computer Graphics Forum **35** (2016)
27. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
28. Peng, C.H., Yang, Y.L., Wonka, P.: Computing layouts with deformable templates. ACM Transactions on Graphics (TOG) **33**(4), 1–11 (2014)
29. Qin, Y., Zhao, N., Sheng, B., Lau, R.W.: Text2city: One-stage text-driven urban layout regeneration. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 4578–4586 (2024)
30. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
31. Shabani, M.A., Hosseini, S., Furukawa, Y.: Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5466–5475 (2023)
32. Shekhawat, K., Upasani, N., Bisht, S., Jain, R.N.: A tool for computer-generated dimensioned floorplans based on given adjacencies. Automation in Construction **127**, 103718 (2021)
33. Sun, J., Wu, W., Liu, L., Min, W., Zhang, G., Zheng, L.: Wallplan: synthesizing floorplans by learning to generate wall graphs. ACM Transactions on Graphics (TOG) **41**(4), 1–14 (2022)
34. Voynov, A., Aberman, K., Cohen-Or, D.: Sketch-guided text-to-image diffusion models. In: ACM SIGGRApH 2023 conference proceedings. pp. 1–11 (2023)
35. Wang, Y., Wu, W., Fei, Y., Zheng, L.: Cvtlayout: Automated generation of mid-scale commercial space layout via centroidal voronoi tessellation. Computers & Graphics p. 104175 (2025)
36. Wu, W., Fan, L., Liu, L., Wonka, P.: Miqp-based layout design for building interiors. In: Computer Graphics Forum. vol. 37, pp. 511–521. Wiley Online Library (2018)
37. Wu, W., Fu, X.M., Tang, R., Wang, Y., Qi, Y.H., Liu, L.: Data-driven interior plan generation for residential buildings. ACM Transactions on Graphics (TOG) **38**(6), 1–12 (2019)
38. Xie, H., Chen, Z., Hong, F., Liu, Z.: Citydreamer: Compositional generative model of unbounded 3d cities. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9666–9675 (2024)
39. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)
40. Zhang, S.K., Liu, J.H., Li, Y., Xiong, T., Ren, K.X., Fu, H., Zhang, S.H.: Automatic generation of commercial scenes. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 1137–1147 (2023)
41. Zheng, G., Zhou, X., Li, X., Qi, Z., Shan, Y., Li, X.: Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22490–22499 (2023)