GRAPH2PIX:

A Generative Model for Converting Room Adjacency Relationships into Layout Images

ZHEN HAN[1], XIAOQIAN LI[2], YE YUAN[3] and RUDI STOUFFS[4]
[1,2,3]*School of Architecture, Tianjin University*
[1,2,4]*Department of Architecture, National University of Singapore*
[1]*zhen.han@u.nus.edu, 0000-0002-9547-9063*
[2]*lixiaoqian_95@tju.edu.cn, 0009-0009-0160-6817*
[3]*yyarch@tju.edu.cn, 0000-0002-2277-5806*
[4]*stouffs@nus.edu.sg, 0000-0002-4200-5833*

Abstract. With the advancement of mathematics and computer science, deep learning-based generative design for floorplans has increasingly garnered attention among researchers. This study proposes a graph-based deep learning model, graph2pix (G2P) to synthesize floorplans guided by user-defined constraints. By incorporating room area and type information into the nodes of the graph, G2P can generate floorplans tailored to specific user requirements. It contains three sub-models: the Translator, Generator, and Discriminator. The Translator serves as the foundational layer, interpreting and mapping room information into a coherent building boundary. Following this, the Generator takes the helm, synthesizing this information to formulate a preliminary floorplan layout. This layout is further refined and evaluated by the Discriminator, ensuring that the final output maintains a high degree of fidelity both to the user's constraints and to architectural feasibility. Our empirical investigation, utilizing metrics such as "area error" and "adjacency error", underscores the model's exceptional ability to achieve user tasks with a high degree of accuracy and efficiency. These findings underscore the potential of G2P as a transformative tool in the domain of automated architectural design.

Keywords. Floor Plan Generative Design, Graph to Image Generative, Graph Neural Network, Room Information Addition, Conditional Generative Adversarial Network.

1. Introduction

The architectural design process is an intricate and labour-intensive endeavour, governed by a detailed design brief that provides essential information. This necessitates architects to engage in planning of various architectural elements including room layouts, floor plans, and façade designs. Equipped with foundational data such

ACCELERATED DESIGN, Proceedings of the 29th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2024, Volume 1, 139-148.

as the types and areas of rooms, architects are tasked with the precise delineation of the size, shape, and spatial relationships of each room. These elements are strategically arranged to create a layout that is both coherent and functional, ensuring the alignment of architectural design with the envisaged utility and aesthetic goals. In the early stages of architectural design, architects are often required to generate multiple initial proposals based on the requirements of a detailed design brief. Subsequently, through a process of meticulous refinement and modification, a selection of these proposals is chosen for further detailed design.

In the realm of generative design assisted by computational methods, researchers typically encode established norms and principles into algorithmic directives, enabling Computer-Aided Design (CAD) systems to generate architectural layouts that conform to recognized best practices. Utilizing computer's advanced computational power enables rapid provision of numerous design options for architects, significantly boosting design efficiency. In recent years, with the advancement of artificial intelligence, and more specifically, AI-generated content, there has been a growing interest in using deep learning algorithms for the generation of architectural floorplans. Researchers have employed Generative Adversarial Networks (GANs) to create building floorplans based on architectural outlines (Wu et al., 2019), (Hu et al., 2020), (Sun et al., 2022). Although this approach can produce highly detailed models, it diverges from the traditional design practices and workflows followed by architects. Architects typically arrange and combine rooms based on their area and interconnections, rather than first delineating an outline, and then filling in different rooms accordingly. Therefore, deep learning models for floorplan generation should ideally only incorporate the most fundamental requirement information as input: number of rooms, room types, room areas, etc. Models that use architecturally processed information like building outlines as input may not effectively alleviate the design burden of architects.

To address these challenges, some researchers have shifted to using different deep learning models, employing more primitive information such as architectural function bubble diagrams or adjacency graphs, rather than architectural outlines, for floorplan generation. Figure 1 shows the input and output of the deep learning models built by different researchers.

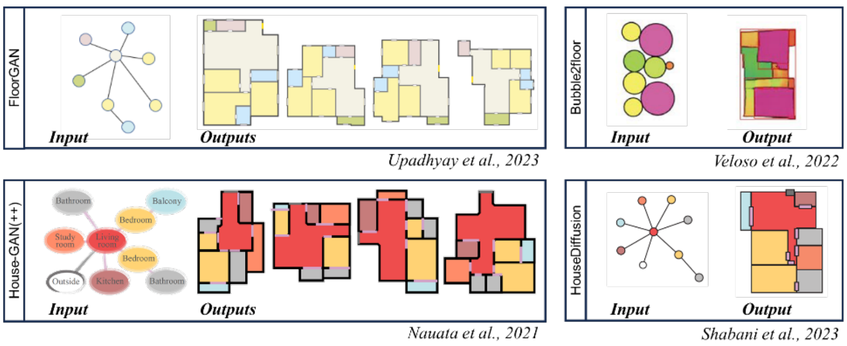


Figure 1. Inputs and outputs of deep learning models used in related literature

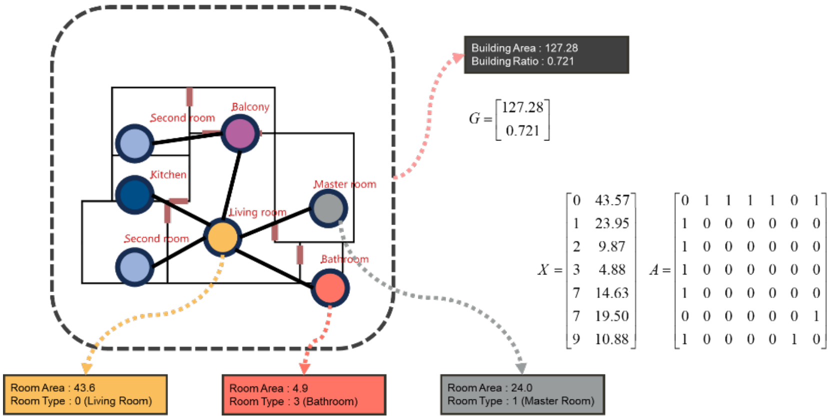
As shown in Figure 1, Veloso et al. (2022) continued to utilize the pix2pix framework, transforming architectural function bubble diagrams into images to serve as model inputs for generating floorplans. These diagrams represent the intended function and approximate size of each room, allowing architects to provide input without having to draw precise outlines; they only need to simply sketch approximate room locations to obtain feasible floor layouts. Nauata et al. (2020 & 2021) represented rooms adjacency information as graphs and employed Graph Neural Networks (GNNs) to generate architectural floorplans (vector floorplan) that comply with the constraints of these adjacencies, further reducing the workload of architects and enabling the generated floorplans to reflect the connectivity between rooms. Upadhyay et al. (2023) combined GNNs with GANs to develop a deep learning model that also takes rooms adjacency information as inputs and produces architectural floorplan images (in image format). Shabani et al. (2023) proposed a method for generating vector floor diagrams through diffusion models, utilizing a transformer architecture. This architecture controls attention masks based on input graphic constraints and generates architectural floorplans (vector floor diagrams) directly through a process of discrete and continuous denoising.

This study builds upon the foundation of room adjacency information based on graphs by embedding additional information such as room area and room type into the nodes of the graph. Compared to existing studies, this research introduces a method that ensures generated floor plans simultaneously meet adjacency relationships and area requirements, addressing the issue of existing models' inability to quantitatively control the generated room areas. Furthermore, this approach obviates the need for architects to input building outlines, better aligning with the design process. Consequently, this enhancement significantly improves the applicability of the model. The resulting floorplans are more controllable and more closely aligned with the needs of architects, reducing the workload associated with design processes.

2. Problem Formulation

In the standard architectural design process, clients provide architects with a brief containing specific design parameters, such as total building area, types and numbers of rooms, and the area of each room. Initially, architects create function bubble diagrams based on the client's requirements. Subsequently, each 'bubble' is assigned area information and transformed into various shapes. Finally, through a process of design exploration and iteration, these are converted into architectural floorplans. To comprehensively incorporate the requirements detailed in the design brief into the deep learning model, enabling controlled generation of floorplans with respect to area and room adjacency relationships, this study amalgamates the concepts of GNNs and GANs. A novel deep learning architecture is proposed, encompassing Translator, Generator, and Discriminator components, aimed at accomplishing production-level tasks.

INPUT: The input to the model is a building bubble diagram, which is represented as a graph where a node encodes a room with its room type and room area. Room adjacency is depicted by edges connecting the respective nodes, providing a graph-based representation of the floorplan's layout. The building's total area and the building ratio, which is the footprint area as a proportion of the total lot area, are also provided to inform the generative model of the overall scale and density of the structure (shown in Figure 2). The model input is formalized as three matrices: a feature matrix X encoding room areas and types, a sparse adjacency matrix A representing room



connectivity, and a global matrix G detailing the building's area and aspect ratio. These matrices provide a structured representation for the generative model to produce building floorplans.

Figure 2. The information contained in the input and its expression

OUTPUT: The output of the proposed model is a detailed architectural floorplan, rendered as a 256x256 pixel image. In this discretized spatial representation, distinct colours correspond to different room types, facilitating immediate visual discrimination of the various functional areas within the building. This color-coding scheme is predefined (shown in Figure 3) and consistent across all generated floorplan images, ensuring uniformity and ease of interpretation. The resolution of the output image has been chosen to balance the need for detail against computational efficiency, providing sufficient granularity to discern room boundaries and configurations while maintaining a manageable image size for model processing.

Figure 3. Color-coding scheme (dataset from RPLAN: Wu et al., 2019)

3. Technical Innovations

In this study, the input contains room and building information, whereas the output is a pixelated image representation. Given the variability in room count per building, the input matrices are of variable dimensions, a format not inherently accommodated by traditional GANs. As shown in Figure 4, to address this, this study introduces a Graph Neural Network-GAN (GNN-GAN) architecture that facilitates the mapping from graphs of arbitrary size to structured images. Also, it is noteworthy that the input and output exhibit substantial differences in scale, a factor that could degrade the quality of the generated images due to insufficient feature extraction, potentially resulting in outputs that do not meet to the input constraints.

To address these challenges, this study introduces a novel network architecture, named "Translator", designed to encode room and building information and predict the corner points of the layout boundary. Subsequently, these predicted boundaries, along with the encoded information, are fed as conditional inputs to both the Generator and Discriminator of a GAN framework. Experimental results indicate that this hierarchical, phased output approach effectively mitigates issues stemming from the significant disparity in data scale between inputs and outputs.

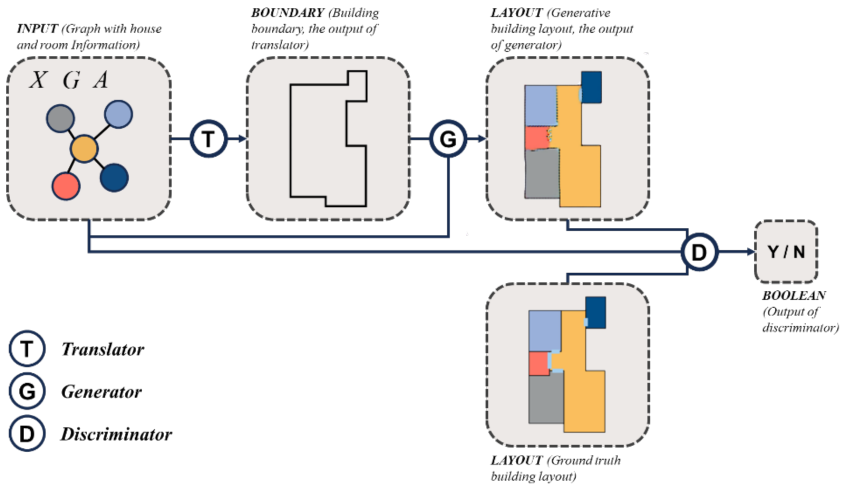


Figure 4. Overall framework of the proposed GNN-GAN model

3.1. FRAMEWORK OF THE TRANSLATOR

As shown in Figure 5, the Translator (T) is composed of an encoder and multiple decoders. The encoder, leveraging a graph attention mechanism coupled with Long Short-Term Memory networks (LSTM), processes the input room and building information (X, A, G) to generate a coded information. This information, along with the LSTM's hidden state, is then fed into the decoders, as illustrated in the left part of Figure 6. The decoding process is iterative and recursive, with each decoder outputting the position of a single vertex of the building contour until the contour is complete. A standard decoder unit, as shown in the lower right of Figure 6, includes an LSTM layer that receives the encoded information and hidden state from the encoder, as well as the prediction from the preceding decoder (with the initial decoder receiving a zero value).

The point set outputted by the Translator is utilized to render a 256x256x1 binary image, with pixel values of 0 or 1, delineating the architectural contour. Concurrently, an Intersection over Union (IoU) Loss, calculated between the generated contour and the ground truth, is employed as the loss function for the Translator, enabling the optimization of the network towards precise contour delineation.

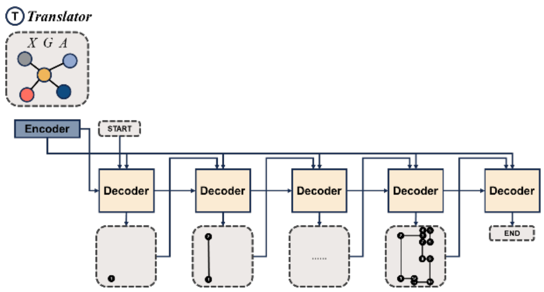


Figure 5. Framework of the Translator

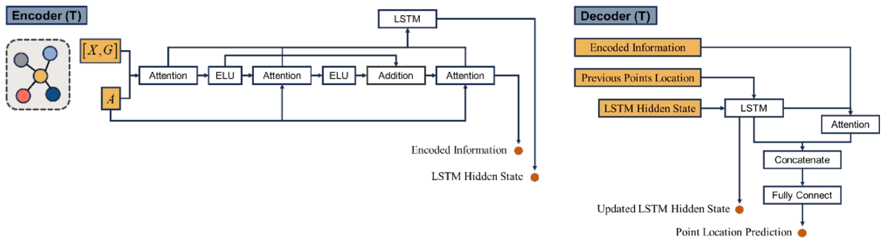


Figure 6. Framework of the encoder and decoder in the Translator

3.2. FRAMEWORK OF THE GENERATOR AND DISCRIMINATOR

Figure 7 illustrates the framework of the Generator (G) and Discriminator (D) within our proposed GAN model. Both G and D are equipped with an encoder-decoder structure. For the Generator, its encoder processes the input room and building information (X, A, G), encoding them into a compressed latent space. The encoded information is then passed through to the decoder, which will also receive the 256*256*1 building boundary image. In the case of the G, the decoder's task is to construct a plausible building layout, which takes the form of a 256*256*3 RGN image representing the floorplan. For the D, the decoder assesses whether the generated layout is authentic (Y) or not (N), based on the learned representations. This adjudication is predicated on discriminative features learned during training, enabling the Discriminator to effectively differentiate between genuine and synthesized designs.

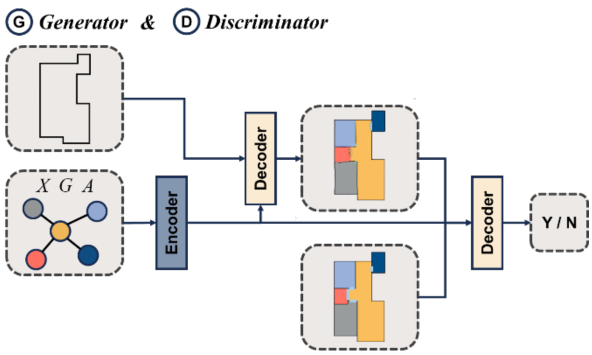


Figure 7. Framework of the Generator and Discriminator

It is important to note that both the G and D leverage a shared parameterization within their respective encoders. This parameter sharing is strategic, ensuring that both G and D utilize identical feature representations derived from the input data. By employing this architecture, we enhance the Discriminator's capacity to discern between genuine and synthesized designs. The shared encoder framework not only facilitates a more efficient learning process by reducing the number of free parameters within the system but also encourages a more nuanced feature extraction, leading to improved generalization of the Discriminator. This shared parameterization underscores the synergetic learning dynamics between G and D, integral to the adversarial training regimen that underpins the GAN's ability to generate compelling architectural layouts.

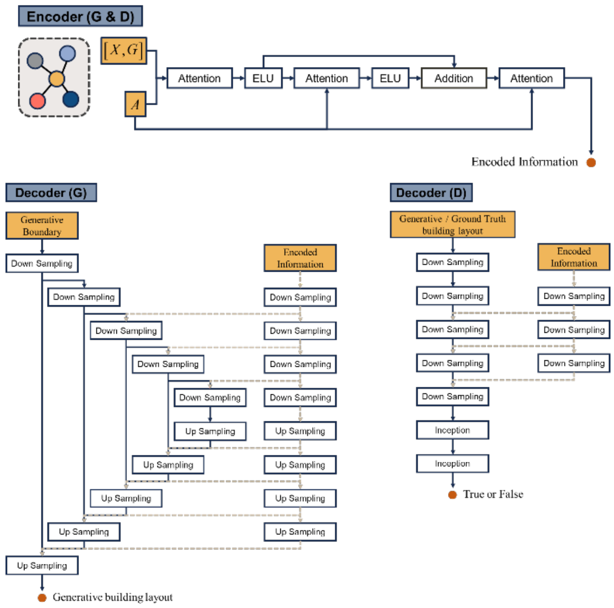


Figure 8. Framework of the encoder and decoder in Generator and Discriminator

Analogous to the Translator (T) network, the encoders within both the G and D incorporate a graph attention-based network structure. This design enables the encoding of room and building information into a latent space, effectively capturing the complex interdependencies and features present in the input data. However, distinct from T, the encoding processes in G and D do not involve sequential data output, and therefore, their architectures omit the LSTM layers. The exclusion of LSTM layers from G and D's encoders is aligned with the nature of their decoding tasks, which do not require processing temporal or ordered data, thus streamlining the architecture for the specific demands of generating and discriminating static images (shown in Figure 8 upper part).

Figure 8 also showcases the distinct frameworks of the decoders for the G and D, positioned in the lower left and lower right of the figure, respectively. The decoder of G is structured as a multimodal U-net, which utilizes the building boundary generated by the Translator (T) along with the encoded information as inputs to synthesize the floorplan. On the other hand, the decoder of D adopts a multimodal Patchnet design, which processes the encoded information in conjunction with either the generated floorplan from G or the ground truth data to execute its adjudication function. This configuration allows D to determine the veracity of the floorplans by examining localized patches, providing a granular assessment of authenticity.

4. Experimental Result

This study uses MATLAB for implementation and a workstation with Xeon CPUs and Nvidia Tesla V100-32GB. The GNN-GAN model uses ADAM optimizer and is trained for 120k iterations. The learning rates of the translator, generator and discriminator are 0.0001, 0.00001, 0.00001, respectively. The batch size is 10. And the model uses leaky-ReLUs ($\alpha = 1$) for all activate functions.

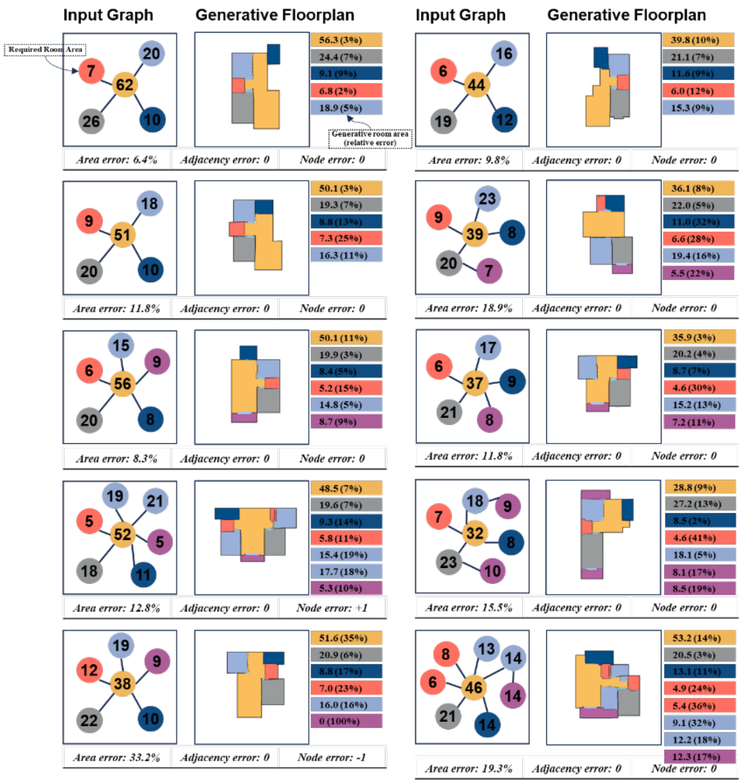


Figure 9. Framework of the encoder and decoder in Generator and Discriminator

"Area error", "Adjacency error" and "Node error" are used to evaluate the performance of the proposed model. The "Area error" represents the relative error between the input demanded area and the generated area. It serves as a measure of the model's precision in conforming to the spatial dimensions dictated by the design criteria:

$$Area\ error = \sum_{i=1}^{n} \left(\left| \hat{R}_i - R_i \right| / R_i \right) / n \tag{1}$$

where n is the number of rooms, R_i is the area of room i, $\hat{R}_i$ is the area of generative room i. The "Adjacency error" and the "Node error", respectively, count the number of rooms in the produced floorplan that fail to comply with the adjacency matrix constraints and the number of missing or redundant rooms, reflecting the model's fidelity to the prescribed room-to-room spatial relationships.

In figure 9, the model demonstrates the capability to generate diverse floorplans tailored to varying requirements. It not only adheres to the specified adjacency relationships among rooms but also responds adaptively to the area demands of individual rooms. Furthermore, the model exhibits a nuanced approach to floorplan generation; for inputs with identical adjacency relationships but differing room areas, it produces distinct floorplans with unique configurations, rather than resorting to mere scaling of room dimensions. This indicates the model's sophisticated understanding of spatial constraints and its ability to generate architecturally viable layouts beyond simple geometric transformations. The ability to produce varied layouts from the same topological constraints underscores the model's potential utility in architectural design, offering a tool that can accommodate a broad spectrum of design scenarios and site-specific conditions.

5. Conclusion

In this study, we propose a novel deep learning model for floorplan generation that takes room adjacency, room area, building area, building ratio as input variables. Departing from traditional pix2pix frameworks that rely on building contours as input, we establish an innovative Graph Neural Network-Generative Adversarial Network (GNN-GAN) architecture. We introduce the "Translator", a component designed for generating building boundaries, effectively addressing the issues of image distortion and non-compliance with input constraints due to the significant magnitude discrepancy between input (building information) and output (floorplan).

Our model advances the state of the art by embedding information such as room area and type into the graph nodes, which permits precise control over room dimensions, aligning the generated floorplan more closely with architects' specifications. Furthermore, by incorporating building area and aspect ratio into the global information, architects can manipulate the overall shape of the generated floorplan to better suit site-specific requirements. Experimental validation confirms the efficacy of our proposed model, demonstrating its ability to fulfil professional demands in architectural design with heightened accuracy and relevance.

Despite the advancements introduced by the proposed model, there remain certain limitations that merit attention. Notably, the generated floorplans occasionally lack detail, particularly in the articulation of door placements, which are not rendered with sufficient precision. Additionally, there are instances where the quantity and types of rooms produced do not align with the specified input constraints. These discrepancies highlight areas for further development. Future research endeavours will focus on refining the granularity of the output, enhancing the sophistication of the model's ability to represent minute architectural features. Moreover, a deeper extraction of features from the input data will be crucial to ensure that the generated floorplans adhere more strictly to the given constraints. Addressing these challenges is essential for bridging the gap between automated floorplan generation and the nuanced requirements of architectural design.

Acknowledgements

This study is supported by the National Natural Science Foundation of China (Grant No. 5220080610) and China Scholarship Council grant (No. 202206250074). Part of the research was conducted at the Singapore-ETH Centre, which was established collaboratively between ETH Zürich and the National Research Foundation Singapore. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program. We also would like to acknowledge that computational work involved in this research work is partially supported by National University of Singapore IT's Research Computing group.

References

Hu, R., Huang, Z., Tang, Y., Van Kaick, O., Zhang, H., & Huang, H. (2020). Graph2plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics (TOG)*, 39(4), 118-1. https://doi.org/10.1145/3386569.3392391.

Nauata, N., Chang, K. H., Cheng, C. Y., Mori, G., & Furukawa, Y. (2020). House-gan: Relational generative adversarial networks for graph-constrained house layout generation. *In Computer Vision–ECCV 2020: 16th European Conference,* Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16 (pp. 162-177). Springer International Publishing. https://doi.org/10.1007/978-3-030-58452-8_10.

Nauata, N., Hosseini, S., Chang, K. H., Chu, H., Cheng, C. Y., & Furukawa, Y. (2021). House-GAN++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. *In 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. Nashville, TN, USA (IEEE). 2021: 13627-13636. In CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 13627-13636). https://doi.org/10.1109/CVPR46437.2021.01342.

Shabani, M. A., Hosseini, S., & Furukawa, Y. (2023). 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). https://doi.org/10.1109/CVPR52729.2023.00529.

Sun, J., Wu, W., Liu, L., Min, W., Zhang, G., & Zheng, L. (2022). Wallplan: synthesizing floorplans by learning to generate wall graphs. *ACM Transactions on Graphics (TOG)*, 41(4), 1-14. https://doi.org/10.1145/3528223.3530135.

Upadhyay, A., Dubey, A., Mani Kuriakose, S., & Agarawal, S. (2023, January). FloorGAN: Generative Network for Automated Floor Layout Generation. *In Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD)* (pp. 140-148). https://doi.org/10.1145/3570991.3571057

Veloso, P., Rhee, J., Bidgoli, A., & Ladron, G. (2022). Bubble2Floor: A Pedagogical Experience With Deep Learning for Floor Plan Generation. In *27th International Conference on Computer-Aided Architectural Design Research in Asia: Intelligent and Informed, CAADRIA 2022* (pp. 373-382). The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA). https://doi.org/10.52842/conf.caadria.2022.1.373.

Wu, W., Fu, X. M., Tang, R., Wang, Y., Qi, Y. H., & Liu, L. (2019). Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG)*, 38(6), 1-12. https://doi.org/10.1145/3355089.3356556.