IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12227

Graphs to Blueprints: GNN-Powered Floor Plan Modeling

Hrithik P Gowda[1], SN Sreevathsa[2], Gangadhara Gowda KN[3], Sharath SJ[4]

Department of Artificial Intelligence and Machine learning,

Dayananda Sagar Academy of Technology and Management, Bengaluru, India[1-4]

Abstract: We introduce a deep learning system for automatic floorplan generation from layout graphs. Our system combines generative modeling with user-in-the-loop design in which users can add sparse constraints like room numbers, connectivity, and other layout adjustments. The system relies on a core deep neural network that takes an input building boundary and layout graph to generate realistic and constraint-abiding floorplans. The system utilizes a graph neural network (GNN) to encode layout patterns and convolutional neural networks (CNNs) for processing building contours and rasterized floorplan images. The model, trained on RPLAN, a 80K-annotated floorplan dataset, outputs varied floorplan layouts consistent with user inputs. We measure its performance via qualitative and quantitative analysis, ablation experiments, and comparison against state-of-the-art techniques, proving its effectiveness and flexibility in floorplan synthesis with constraints.

Keywords-floorplan generation, layout graph, deep generative modeling

I.INTRODUCTION

Artificial intelligence (AI) and machine learning (ML) are revolutionizing the discipline of architectural design, providing creative solutions that improve efficiency, creativity, and scalability. With increasingly sophisticated architectural projects, AI-based generative models offer architects effective tools to streamline design processes without compromising aesthetic and functional quality. One of the most basic elements of architectural design is the development of floor and building plans, which determine spatial arrangement, functionality, and overall usability. Floorplan generation automation has attracted considerable attention from computer graphics and vision researchers, resulting in improved data-driven modeling methods. Current research has investigated numerous methods, such as raster-to-vector conversion, floorplan reconstruction from 3D scans, and AI-driven layout generation, all of which help in an improved, streamlined, and efficient design process.

In this paper, we present a deep learning-based system for automatic floorplan generation that combines generative modeling with user-in-the-loop design approaches. Conventional design methodologies tend to involve significant manual intervention, making them less scalable and flexible. Our method seeks to overcome these challenges by allowing users to specify high-level design constraints—e.g., room numbers, adjacency relationships, and functional requirements—that inform the generative process. These constraints are most naturally represented as layout graphs, a formal representation of spatial information similar to scene graphs employed in image composition. By incorporating layout graphs, our framework ensures that user-defined preferences are maintained while allowing for the generation of diverse and adaptive floorplans.

A key aspect of our system is the utilization of a large-scale dataset of human-designed floorplans, which serves as the foundation for training our generative model. This dataset allows the model to learn about architectural principles and spatial relationships so that the output designs conform to real-world usability and aesthetic requirements. The framework accommodates both automatic and interactive design processes: users can either let the system generate floorplans automatically or improve the output by modifying the query layout graph. Such versatility renders our methodology applicable to an extensive variety of uses, such as architectural design and urban development, game planning, virtual worlds, and property visualization.

We utilize deep neural networks, viz., graph neural networks (GNNs) and convolutional neural networks (CNNs), in our generative model for processing input restrictions and generating good-quality, vectorized floorplans. The architecture takes a sequential pipeline where the initial layout graph is obtained or created, then passed through deep learning models and further processed to maintain spatial coherence and functional viability. The synthesized floorplans are highly flexible with the ability to investigate different design options while respecting user-specified constraints.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12227

By conducting rigorous qualitative and quantitative analyses, we show the efficacy of our framework in generating architecturally correct and aesthetically appealing floorplans. We also carry out a user study to evaluate the usability and real-world implications of our tool, pointing out its applicability in real-world scenarios. We also carry out an ablation study and comparative analysis with state-of-the-art methods to confirm the efficiency and accuracy of our approach. By combining AI-powered automation with direct user control, our system closes the gap between computational design and real-world architectural workflows, providing a versatile tool for designers, architects, and urban planners.

II.RELATED WORKS

Our work is a part of the larger class of generative modeling for structured configurations. Computer graphics research has provided many techniques to generate structured layouts in recent years, such as document and clipart layouts, urban layouts like street networks, and procedurally generated game levels . Here we concentrate on organized arrangement techniques more immediately connected to our work, such as indoor scene synthesis, floorplan generation, and image composition.

Indoor Scene Synthesis

Indoor scene synthesis is the process of creating realistic furniture and object arrangements in a specified room. Initial approaches depended on pre-defined placement constraints and physics-based simulation. For example, Xu et al proposed an approach that employs pseudo-physics to place objects according to pre-defined constraints. More recently, Merrell et al proposed an interactive system that proposes furniture placements by integrating user constraints and interior design rules.

As data-driven methods developed, more complex methods were implemented. Fisher et al used Bayesian networks to generate scenes from example setups, and Fisher et al. further improved this method with the prediction of action maps from input scans via learned models. Zhao et al. presented a method which generates scenes according to object interaction in example settings.

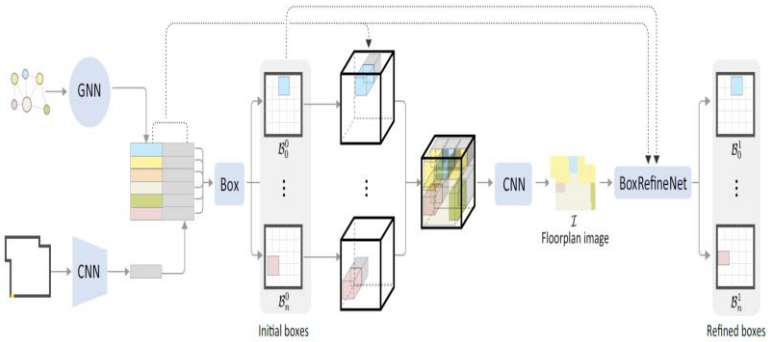
Although indoor scene synthesis is related to floorplan generation, the main difference occurs in the goals of design: scene synthesis seeks to place objects inside given room boundaries, whereas floorplan generation requires dividing a building layout into individual rooms under functional and architectural constraints.

Floorplan Generation

Floorplan generation seeks to generate functional building layouts from input constraints, including room adjacencies, sizes, and building boundaries in general. Early research in this area was largely based on optimization and procedure-driven approaches. Arvin and House employed spring-system representations to produce layouts approximating pre-specified design goals. Merrell et al utilized stochastic optimization combined with Bayesian networks to produce residential layouts from top-level constraints. Later research, e.g., by Rosser et al, further developed this idea by combining user-specified building profiles and room details. Furthermore, Rodrigues et al investigated evolutionary techniques for the optimization of building floor plans considering constraints.

More recently, Wu et al proposed a mixed-integer quadratic programming (MIQP) method to produce optimized interior plans, further pushing the limit of constraint-based layout generation. Some approaches have ventured beyond typical residential floorplans; for instance, Bao et al gave an exterior building floorplan exploration approach, while Feng et al proposed a spatial layout optimization framework for large spaces, including shopping malls and transportation centers, from pedestrian simulation data.

In contrast to such methods, based on strong priors or limited machine learning models (e.g., Bayesian networks), our method uses deep learning to learn implicit architectural constraints from training data and thus has a more flexible and efficient floorplan generation process.



IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12227

Deep Learning for Layout Generation

The use of deep learning in layout generation has made great strides. Wu et al presented a deep model that produces residential floorplans by predicting room positions and wall locations given an input building outline. The model, which was trained with a large dataset of labeled floorplans, effectively translates its predictions into vectorized representations of the floorplan. Its key limitation, though

Graph-Based Generative Models

New means of representing structured data in deep learning models have been offered through recent developments in graph generative models. GRAINS and PlanIT are some of the me, is that there is little user control over the output, other than defining the building boundary.

Our approach varies from the above methods as it allows for fine-grained user control when generating floorplans. Rather than depending on solely implicit constraints acquired through data, we enable users to specify room adjacencies and important spatial properties through layout graphs. Such a hybrid method guarantees generated floorplans follow learned design rules as well as user-specified constraints. Further, our approach incorporates graph retrieval methods, drawing upon existing architectural work to influence the generation process, leading to more contextually consistent layouts.

Methods that use graph-based representations to generate indoor scenes.

While such approaches effectively produce structured layouts, they mainly project rectangular room boundaries and fixed wall locations. In contrast, our method accommodates arbitrary rectilinear building perimeters, enabling more varied and plausible architectural arrangements. To that end, we use graph neural networks (GNNs) to encode spatial relations and improve generated layouts using an alignment step to maintain architectural consistency.

Image Composition from Scene Graphs

Scene graph-based image synthesis has emerged in computer vision as a way of producing structured visual content. Graph convolutional networks (GCNs) with adversarial learning were proposed in influential works like Johnson et al. to produce images from scene graphs. Ashual and Wolf built upon this by decoupling object appearance modeling from layout generation to achieve better image synthesis quality.

Our work draws inspiration from these approaches by utilizing graph-based representations to guide generative processes. However, unlike image synthesis, which focuses on blending objects within a scene, our problem involves spatial partitioning and floorplan layout optimization. The key challenge lies in ensuring that generated layouts adhere to both geometric constraints and functional requirements, necessitating a tailored approach that extends beyond standard scene graph models.

III.OVERVIEW

This paper introduces a new deep learning architecture for floorplan generation that supports user-in-the-loop design personalization. The system combines user constraints, layout retrieval, and neural network-based generation of floorplans to generate customized layouts with respect to architectural design principles.

The system is composed of three major components: initial user input, layout graph retrieval, and floorplan generation. Users first provide a building boundary and constraints like room types, number, and adjacency preferences. The system then extracts suitable layout graphs from the RPLAN dataset in a retrieve-and-adjust paradigm such that the produced floorplans meet real-world design guidelines. Users can also edit these graphs to suit their own specifications.

The network is the focal point of the floorplan generation process. The network learns to retarget the extracted layout graphs into the specified boundary, circumventing structural retargeting difficulties. The network predicts room bounding boxes and renders a raster floorplan image in order to merge overlaps and ensure spatial consistency. A final optimization phase aligns and vectorizes the layout in order to yield a valid and well-structured floorplan.

Through the combination of user-guided customization and deep learning, this system makes it possible to generate various and well-designed floorplans. Future work may include adding other types of constraints, such as functional and accessibility aspects, and expanding the system to include furniture placement and more complex layout adjustments.

IV.LAYOUT GRAPH RECOMMENDATION

We derive layout graphs from the RPLAN dataset [Wu et al. 2019] and scale them to user-specified building boundaries. Users can then edit these graphs to suit constraints even better.Graph Extraction:Floorplans in RPLAN are annotated semantic raster images. Each room exists as a node in a graph, with rooms adjacent to one another connected with edges. Rooms contain information for room type, location (a 5×5 grid coordinate), and relative size (comparing to building area).

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12227

Adjoining pairs were identified by interior doors and the proximity threshold values, while relative spatial positions (left of, above) provide edge connections. More than 80K graphs were derived from 120K floorplans with varied templates provided for user drawings.Graph Retrieval:Constraints such as room types, locations, and adjacencies are specified by users. Graphs are screened according to constraint satisfaction and ordered according to the similarity of their source boundaries with the boundary supplied by the user. Matching involves front door position, employing a turning function [Arkin et al. 1991] to match boundary shapes.Graph Adjustment:Retrieved graphs are readjusted to the user's boundary by rotating them to correspond to front doors and readjusting node locations within a 5×5 grid. Nodes beyond the boundary are moved to the nearest vacant cell, maintaining layout validity. Users can interactively adjust the graph by modifying room nodes and adjacencies. The resulting adjusted graph is applied to produce a tailored floorplan.

V.FLOORPLAN GENERATION

A graph network for floorplan generation. It also describes how the output is processed to generate the final vector-based floorplan.

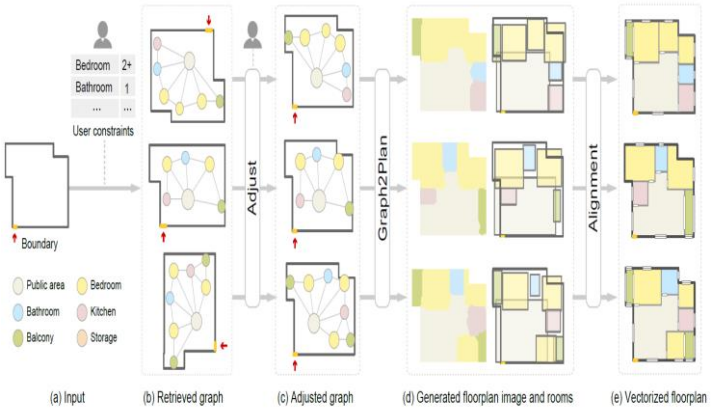
Network Input and Output

Network takes as input:Building boundary (B): A 128 × 128 image with three binary channels for the inside, boundary, and entrance doors.
Layout graph (G): A user-limited graph with nodes (N) and edges (E), denoting rooms and their connectivity.
Every room (i) of the floorplan is a node (ni), having: Room category (ri): A learned 128-dimensional representation encoding 13 types of rooms.
Location (li): A 25-dimensional vector representing the position of the room in a 5 × 5 grid.
Size (si): A 10-dimensional feature vector for room size.
Edges (eij): Represented relations between rooms, selected from 10 possible relations.The output is comprised of:A 128 × 128 floorplan image (I).Two sets of room bounding boxes:Initial boxes (B0i). Advanced boxes (B1i), each of which specifies the position and extent of a room (xi, yi, wi, hi)

Network Structure

Graph Neural Network (GNN): Maps the layout graph (G) into features of the rooms.Boundary Encoder: Finds boundary features of (B), which are fed together with the room features.
Box Network: Predits an early group of room bounding boxes.
Cascaded Refinement Network (CRN): It exploits the predicted room boxes to predict the floorplan image (I). Rooms which are overlapping get merged through their features summed.
BoxRefineNet: Each room's bounding box gets refined utilizing the floorplan image (I) as reference. It performs in the following way:
The entire image is convolved with convolutional layers to produce a feature map.
A Region of Interest (RoI) pooling layer extracts features for every room.
The features are fed through fully connected layers to fine-tune the box's position and size.

VI.ROOM ALIGNMENT



produces a raster floorplan image and one bounding box per room. The misalignments and overlaps are possible though. To counteract them, we adjust room alignments and find the order of drawing via generated floorplan.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12227

Initially, we align room boxes with the building boundary and align adjacent rooms. Every room edge is snapped to the closest boundary edge of the same direction if it's within a threshold ττ. Neighboring rooms are aligned according to their geometric relations in the layout graph—for example, room A's right boundary aligns with room B's left boundary if they are proximal, reducing corners. Updates are executed iteratively and keeping previously refined alignments.

Finally, we find room labels and drawing order for overlapping areas. From the produced floorplan image, we count the pixels in the overlap labeled and give priority to the room with fewer pixels or, if there is a tie, the larger area. A directed graph is formed with nodes for rooms and edges representing precedence constraints. A topological sorting algorithm is used to decide the final drawing order, breaking cycles by eliminating the node of smallest outdegree.

Lastly, windows and interior doors are positioned according to Wu et al. [2019] so that doors link adjacent rooms and windows line up with the exterior walls.

VII.RESULT AND EVALUATION

generates a raster floorplan image and room bounding boxes. Misalignments and overlaps can occur, though. For these to be resolved, we refine room alignments and establish the drawing order from the created floorplan.

To start, we align room boxes with the building boundary and reposition neighboring rooms. Each edge of a room is snapped to the closest boundary edge of the same orientation if it is within a threshold ττ. Overlapping rooms are aligned according to their spatial position in the layout graph—i.e., A's right edge would align with B's left edge if they are close enough to reduce corners. Iteratively applied updates preserve previously improved alignments.

Finally, we establish room labels and the drawing order of overlapping parts. Based on the resulting floorplan image, we number the pixels of labeled pixels in the overlap and give priority to the room having fewer pixels or, if they are equal, the greater area. A directed graph is built in which vertices correspond to rooms, and arcs signify precedence relationships. We apply a topological sort algorithm to establish the ultimate drawing order, breaking cycles by eliminating the vertex of minimum outdegree.

Lastly, windows and internal doors are positioned according to Wu et al. [2019] so that doors link adjacent rooms and windows face exterior walls.

VIII.CONCLUSION AND FUTURE WORKS

In summary, we have described the first user-guided deep learning framework to generate floorplans with interactive and flexible design tuning. Through utilizing layout graphs, users can state their design purposes, fetch floorplan structures relative to their intents, and loop refine these structures to achieve desired outcomes. Our method provides a way of generating varied floorplans from identical input boundaries along with maintaining architecturally learned rules from a wide set of historical layouts.

Although it has its benefits, the model has some limitations. The existing model does not include accessibility requirements, functional aspects, or adjacency constraints such as ensuring that certain rooms are not placed next to each other or certain boundary features. Also, interior features such as doors and windows are not modeled explicitly, and the alignment of predicted rooms is based on post-processing and not part of the learning process. In addition, if the extracted layout graph is substantially different from the input boundary, the resulting floorplan might not satisfy all constraints, resulting in overlaps or misplaced rooms.

Future research will address these limitations by increasing the variety of user constraints and improving the structural modeling of synthesized floorplans. These can include support for more user-specified parameters, like support walls, optimizing room alignment in the learning setup, and supporting more sophisticated layout graph manipulations. Additionally, expanding the framework to accommodate user preference-based furniture arrangement and investigating structural learning for optimizing room arrangements will enhance the usability and correctness of our system further. These developments will help in developing a more feature-rich and versatile floorplan generator with more control and accuracy to designers and architects.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 2, February 2025

DOI: 10.17148/IARJSET.2025.12227

REFERENCES

[1]. E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. 1991. An efficiently computable metric for comparing polygonal shapes. IEEE Trans. Pattern Analysis & Machine Intelligence 13, 3 (March 1991), 209–216. https://doi.org/10.1109/34.75509

[2]. Scott A. Arvin and Donald H. House. 2002. Modeling architectural design objectives in physically based space planning. Automation in Construction 11, 2 (2002), 213–225.

[3]. Oron Ashual and Lior Wolf. 2019. Specifying Object Attributes and Relations in Interactive Scene Generation. In Proc. Int. Conf. on Computer Vision.

[4]. Fan Bao, Dong-Ming Yan, Niloy J. Mitra, and Peter Wonka. 2013. Generating and Exploring Good Building Layouts. ACM Trans. on Graphics (Proc. SIGGRAPH) 32, 4 (2013), 122:1–122:10.

[5]. Stanislas Chaillou. 2019. AI + Architecture: Towards a New Approach. Master's thesis. Harvard School of Design.

[6]. Qifeng Chen and Vladlen Koltun. 2017. Photographic image synthesis with cascaded refinement networks. In Proc. Int. Conf. on Computer Vision. 1511–1520.

[7]. Tian Feng, Lap-Fai Yu, Sai-Kit Yeung, KangKang Yin, and Kun Zhou. 2016. Crowd-driven Mid-scale Layout Design. ACM Trans. on Graphics (Proc. SIGGRAPH) 35, 4 (2016), 132:1–132:14.

[8]. Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. 2012. Example-based synthesis of 3D object arrangements. ACM Trans. on Graphics

[9]. (Proc. SIGGRAPH Asia) 31, 6 (2012), 135:1–11. Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric Scene Synthesis for Functional 3D Scene Modeling. ACM Trans. on

[10]. Graphics (Proc. SIGGRAPH Asia) 34, 6 (2015), 179:1–13. Ross Girshick. 2015. Fast R-CNN. In Proc. Int. Conf. on Computer Vision. 1440–1448. Aditya Grover, Aaron Zweig, and Stefano Ermon. 2019. Graphite: Iterative Generative Modeling of Graphs. In Proc. Conf. on Machine Learning.

[11]. Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural Content Generation for Games: A Survey. ACM Trans. Multimedia Comput. Commun. Appl. 9, 1 (2013), 1:1–1:22.

[12]. Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image Generation from Scene Graphs. In Proc. IEEE Conf. on Computer Vision & Pattern Recognition.

[13]. Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. 2019b. Lay- outGAN: Generating Graphic Layouts with Wireframe Discriminators. In Proc. Int. Conf. on Learning Representations (ICLR).

[14]. Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen Or, and Hao Zhang. 2019a. GRAINS:Generative Recursive Autoencoders for INdoor Scenes. ACM Trans. on Graphics 38 (2019).

[15]. Chen Liu, Jiaye Wu, and Yasutaka Furukawa. 2018. FloorNet: A Unified Framework for Floorplan Reconstruction from 3D Scans. In Proc. Euro. Conf. on Computer Vision.203–219.