2026 European Conference on Computing in Construction
Corfu, Greece
July 12-15, 2026

FABRICATION-AWARE GENERATIVE DESIGN FOR HYBRID PREFABRICATION USING STABLE DIFFUSION XL

Hang Zhang[1], Jianpeng Cao[2], Anton Savov[1], and Benjamin Dillenburger[1]
[1]ETH Zurich, Zurich, Switzerland
[2]The University of Hong Kong, Hong Kong S.A.R., China

Abstract

Existing generative design approaches for floorplan generation rarely incorporate fabrication constraints, limiting their applicability in industrialized construction. This study presents a fabrication-aware generative design framework for hybrid prefabrication based on Stable Diffusion XL. The framework integrates spatial, functional, and manufacturing constraints to balance design flexibility and production efficiency. A two-stage training strategy generates both conceptual layouts and module-integrated solutions. Experiments demonstrate the generation of boundary-compliant layouts and standardized module allocation under diverse boundary conditions and text-based requirements. An interactive workflow supports convergence from conceptual designs to fabrication-aligned solutions, bridging early-stage architectural intent with downstream fabrication requirements.

Introduction

While data-driven layout generation has accelerated early-stage concept exploration, architectural workflows frequently treat downstream fabrication as an afterthought. Recent surveys highlight significant advancements across rule-based, optimization-driven, and data-driven floorplan synthesis, yet they consistently underscore bottlenecks regarding constraint satisfaction and practical constructability (Weber et al., 2022). In parallel, modular building automation has progressed by leveraging BIM-centric data representations and specialized configurators to explicitly systematize modules alongside their spatial relationships (Liu et al., 2024). Merging these high-capacity generative priors with standardized prefabrication rules presents a clear opportunity for industrialized construction.

Yet existing approaches remain split. Mainstream generative floorplan frameworks, such as HouseDiffusion, (Shabani et al., 2023; Zhang & Zhang, 2025), operate primarily on idealized, rectangular room abstractions and do not account for prefabricated assemblies. Conversely, modular-layout methods, such as CoGAN (Ghannad & Lee, 2022) and graph-based BIM pipelines (Gan, 2022b), enforce modular structure but often offer limited design variety and are not integrated with generative frameworks.

To bridge this gap, the goal of this research is to develop a Fabrication-aware generative design framework for prefabrication that accepts predefined modules as input and incorporates them into the customized floor plan generation. The framework aims to increase the adoption of standard modules within a boundary-compliant layout, enabling efficient exploration of feasible configurations while preserving production efficiency in industrialized construction.

Literature review

This section reviews computational methods for automatic floor plan generation. The references are limited to industrialized construction. In addition to top-down and bottom-up methods, relevant BIM-based parametric methods are also included. To maintain the high quality, the analyzed studies are only sourced from journal publications.

Top-down methods

The top-down approach follows a decomposition workflow, where a floor plan is divided into discrete modular units, or referred to as modularization. Lin et al. formulate the modularization process by first segmenting the original space into smaller regions and then merging them to obtain an optimal layout under shape and size constraints. The generation is evaluated via two indicators, number of modules and the diversity of modules with different dimensions (Lin et al., 2025). The generation achieved optimal modular key plan convergence in 50 iterations, with automated runtimes averaging only 7.4 seconds. Cao et al. translate the decomposition process as the decomposition process as matching predefined modules to the given floor plan. The difference compared to Lin's approach is that the floor plan is fixed, while the prefabrication strategy, including panelized units and modular units, is determined by evaluating the construction cost and durations (Cao et al., 2024). The study demonstrated that shifting to a fully volumetric strategy over a fully panelized one can accelerate construction schedules by up to 23%, although it concurrently inflated material and shipping costs by roughly 22%.

Bottom-up methods

The bottom-up approach starts with the definition of standard modules and establishes the generative rules for module placement. Guided by objective functions, the optimal layout is created. Zheng et al. develops a generative design method for the floor layout of modular buildings. The method uses room module, corridor module and stair module as building blocks. The meta-heuristic rules are provided by geometric and functional relationship between module units. The objective functions include project cost and layout penalties (Zheng et al., 2024). The generation could escape local minima to discover globally optimized volumetric configurations within 42 generations (under 870 seconds). Sheijani et al. adopt a bottom-up approach to automate layout generation for office buildings. The generation is framed as identifying the most inclusive path for placing modules (Sheijani et al., 2024). The results show that the approach can condense the standard front-end schematic design phase from a typical timeframe of multiple weeks down to just 5 to 7 days.

BIM-based parametric methods

Beyond these two methods, there are BIM-based generative design approaches that accommodate a broader range of building topologies. Gan developed a customized BIM object library where elements with different embodied carbon and cost index could be matched with the model geometry and material information to create the alternative designs (Gan, 2022a). The generative framework by Gan (2022a) can create between 1,000 and 1,500 new design options in approximately 30 minutes. Li et al. proposed a parametric generative design method for precast buildings using BIM and graph convolutional neural networks. The layout is explicitly parameterized through a set of spatial and geometric features, and the resulting designs are evaluated and optimized with respect to wind pressure performance (Li et al., 2024). The method achieves a highly accurate layout prediction rate of 98.79% (R^2) on testing sets for evaluating surface-averaged wind pressure coefficients. However, the BIM-based approach is less efficient during the layout generation, when design exploration is primarily driven by functional program, rather than specific building elements.

In short, the top-down strategy tends to defer module design to the final stage, which may result in numerous irregular modules, thereby reducing manufacturing efficiency and increasing overall project costs. Although prior studies have considered fabrication constraints and modularization efficiency (Cao et al., 2024; Lin et al., 2025) in the post-design phase, they overlook the opportunity for early manufacturer engagement. Delayed engagement might lead to prolonged lead times for procuring building materials before fabrication. In contrast, the bottom-up approach integrates standard modules directly within the generative process, potentially overcoming these limitations (Sheijani et al., 2024; Zheng et al., 2024). However, the generation process is usually driven by a set of rules which limits their scalability and usually consumes longer iterations. Last but not least, the bottom-up approach is not flexible with respect to predefined building boundaries and room adjacency requirements.

Methodology

Dataset preparation

Training data are derived from the SwissDwellings (Van Engelenburg et al., 2025) database, which provides apartment layouts as vector room outlines. Figure 1 summarizes our two-stage preprocessing pipeline. Stage 1 converts each vector layout into a semantic raster target and a set of conditional inputs, including the site context, a partial layout graph, and a structured prompt. Stage 2 augments this dataset with fabrication labels by sampling candidates from the Stage 1 model, filtering invalid instances using geometric checks, and matching them to standardized wet-core modules.

Problem Formulation

We define fabrication-aware floorplan generation as learning the parameters θ of a conditional generative model, $p\theta(F^{img} \mid G^{img}, T_{desc}, T_{fab}, V_{Module}, Mode)$. As illustrated in Figure. 1, given multi-modal inputs that encode spatial, functional, and fabrication constraints, the model synthesizes a floorplan image $F^{img} \in R^{512\times512\times3}$. The objective is to maximize the likelihood of generating coherent, feasible, and architecturally valid designs that are consistent with both functional program and manufacturing realities. The input modalities are defined as follows:

Layout Graph Image (G^{img}) A 2D raster image that encodes the spatial topology as adjacency graph $G^{vec} = (V, E)$, where V denotes functional room values of types and locations, and E denotes the adjacencies to be satisfied. The graph input is rasterized on canvas $\mathbf{G^{img}}$, and it may be partial ($G_{partial}$), specifying only key relationships, or complete ($G_{complete}$), defining the full topology.

Module (V_Module): We predefined five module prototypes with various dimensions, as shown in Module Library in Figure 1.

Floorplan Description (T_desc): A structured prompt of language specification, describing the functional program, including the room types and locations. For example: "Floorplan with a Northeast Bedroom, a West Corridor." The description may be partial or complete, depending on the level of details the user wishes to impose.

Fabrication Information (T_fab): A structured prompt of language specification, describing manufacturing constraints that ground the design in the selected construction system. For example: "use bath-kitchen module, less panels, medium doors, medium windows." This input may also be partial or complete, allowing different degrees of fabrication control.

Mode: A mode switcher that identify whether the model generate layout with only design constraints or generate layout that be aware of fabrication constraints.

Meanwhile, the model output is defined as follows:

Floorplan Image (F): A high-resolution raster image of a complete floorplan. The generated plan must jointly satisfy the constraints imposed by all input modalities. Specifically, it should:

(1) Respect the spatial topology encoded by the layout graph image G.

(2) Fulfill the functional program (room types and locations) described in T_desc.

(3) Comply with the fabrication constraints, including clear placement of predefined modules (e.g., bathroom, kitchen) and compatibility of remaining spaces with the stated construction system (e.g., timber panels).

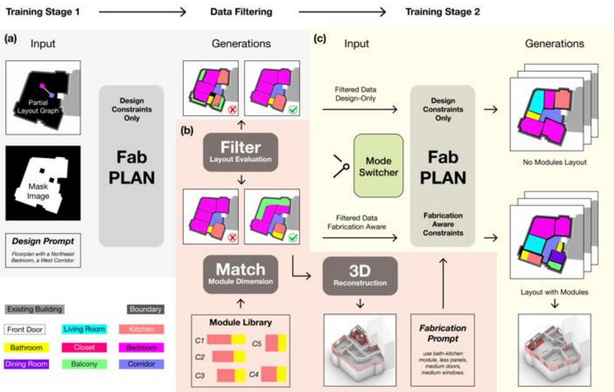

Figure 1: Framework of Fab-aware GD Framework

Fab-aware GD Framework using SDXL

As also described in Figure 1, we propose a Fab-aware GD framework built on Stable Diffusion XL (SDXL) (Podell et al., 2023), where the UNet, the core denoising network responsible for transforming random noise into coherent images, is fine-tuned with hybrid conditioning to support both design exploration and fabrication aware generation. To maintain general design capability while enforcing fabrication constraints, we introduce a mode-switching strategy. When under design mode, the model is conditioned on **G** and **T_desc**, encouraging broad layout diversity within the boundary and adjacency constraints. When under fabrication-aware mode, the model is additionally conditioned on the module specification **V_Module** and **T_fab**, biasing generation toward layouts that can accommodate standardized modules and compatible panel logic. This dual-path conditioning prevents overfitting to modular templates while enabling reliable module-aware outputs.

During fine-tuning, both modes are sampled within each training epoch using filtered design-only data and synthetically enriched fabrication-aware data. This yields a single UNet that can be toggled at inference time to produce either unconstrained conceptual layouts or fabrication-ready layouts that explicitly integrate standardized modules. In effect, the framework unifies early-stage architectural exploration and downstream manufacturing feasibility within a single SDXL-based model.

Data Preparation and Two-Stage Training Pipeline

To generate layouts that are both spatially valid and compatible with fabrication constraints, as demonstrated in Figure 2, we adopt a two-stage training pipeline with dataset filtering and synthetic fabrication-aware enrichment.

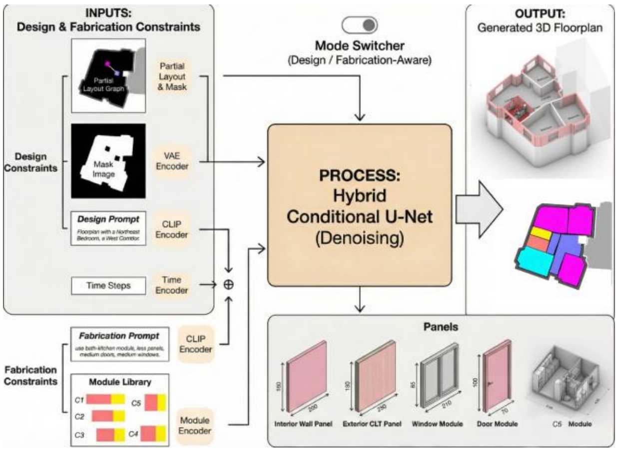

Figure 2: Data Preparation and Two-Stage Training Framework

Training Stage 1: Design-Centric Initialization

We first train the base model with design constraints only to establish robust spatial priors. The model receives partial layout graphs (room types, locations, and adjacency), mask images (the region model should denoise), and structured prompt design prompts. It then generates candidate floorplans without fabrication inputs.

Data Filtering and Module Matching

Stage 1 outputs are screened to remove invalid or low-quality layouts. The retained plans are passed to a module dimension matching procedure that fits standardized components from a module library (C1–C5, e.g., bathroom and kitchen units). The selection of five module prototypes is motivated by the geometric variation of standard residential wet-core units, ranging from compact square layouts to elongated narrow configurations. These modules represent bathrooms and kitchens, which are among the most service-intensive and costly elements in prefabricated housing. While the number of possible module types is not inherently limited, expanding the library would increase the complexity of module embedding and training, whereas practical industrial systems typically rely on a relatively small family of standardized products. The five-module library therefore provides a balanced test set that captures representative variation while remaining computationally and practically tractable."

For successful matches, the system reconstructs the layout in 3D and automatically generates fabrication prompts describing the applied module and panel constraints. This yields a synthetic fabrication-aware dataset pairing geometry with explicit manufacturing information.

To ensure high-quality training data for the diffusion model, raw generated layouts are passed through an automated rule-based validation pipeline. A generated layout is classified as invalid or low-quality if it fails any of the following four objective criteria:

- Room Shape Convexity: Deformed or highly irregular room polygons are rejected using a compactness index (ratio of room area to its axis-aligned bounding box area). Layouts falling below a mean threshold of 0.85 are discarded.
- Aspect Ratios: Rooms with extreme, unbuildable proportions (e.g., long, thin slithers) are flagged by checking side-length ratios.
- Connectivity: A room-contact graph is built by analyzing polygon intersections. A layout is invalid if a room has no feasible spatial connection to the rest of the dwelling (e.g., a bathroom that is isolated and cannot be accessed from a corridor or living room).
- Essential Room Types: To be valid, a residential layout must contain a core set of functional zones: a kitchen, a bathroom, and a living or sleeping area.
- Module Feasibility: Bathroom, kitchen, and combined total area ratios are computed against the module's specifications. A module is considered feasible if all three ratios fall within a 15% tolerance threshold.

Training Stage 2: Fabrication-Aware Refinement

After assembling the filtered design-only dataset and the synthetically enriched fabrication-aware dataset, we fine-tune the SDXL UNet with a dual-path strategy controlled by a mode switcher. (1) Mode A (design-only): The model is conditioned on the layout graph image and design prompt (and boundary masks, if applicable). This path preserves spatial generality and layout diversity, and prevents overfitting to modular templates. (2) Mode B (fabrication-aware): The training target is the same floorplan image, but the model is additionally conditioned on module specifications and fabrication prompts. This path enables dimensional feasibility and module accommodation. Mini-batches are routed between the two modes using a fixed sampling policy, yielding a single unified model. At inference, the same UNet can be toggled to generate either concept-level layouts or module-integrated layouts that better satisfy fabrication constraints.

Results

We evaluate the proposed fabrication-aware generative framework in three experiments: (1) controlled comparisons between design-only and fabrication-aware outputs under identical boundary and partial inputs, (2) robustness to different boundary conditions and partial-input completeness, and (3) iterative, user-in-the-loop customization including module freezing. All results are generated from partial inputs to reflect early-stage design practice.

Fabrication-Aware Generation

Figure 3 compares design-only and fabrication-aware generation using the same boundary condition. Design-only outputs satisfy the partial topology and program intent, but do not explicitly allocate space for standardized modules, often implying partition or opening patterns that are difficult to align with fabrication systems. With fabrication-aware conditioning enabled, the model consistently introduces module-compatible allocations and clearer service clustering while maintaining boundary compliance and the intended spatial logic.

Using identical inputs, we further test prompt-level controllability within fabrication mode across three variants: baseline module integration, reduced panel/door/window complexity, and increased panel/door/window richness. The results show that the model can shift between assembly-simplified and architecture-enriched solutions without breaking module fit or topology.

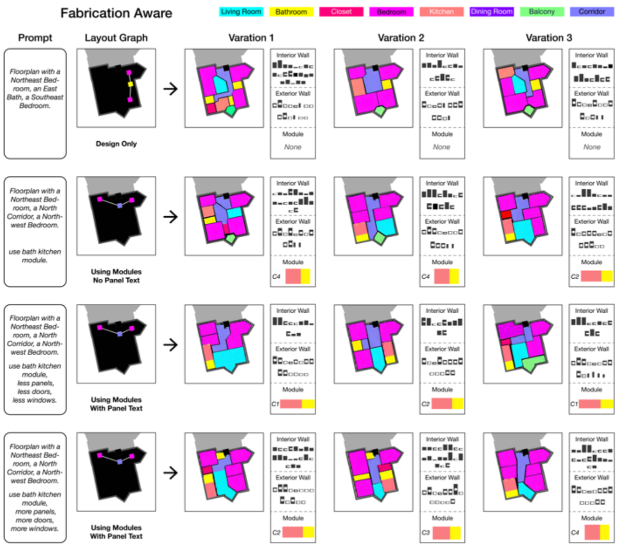

Figure 3: Design-only vs fabrication-aware outputs with identical boundary and partial inputs

Various Partial Inputs

As illustrated in figure 4, we evaluate fabrication-aware generation across diverse boundary conditions and varying levels of partial input. For both regular and irregular envelopes, the model preserves boundary compliance and maintains plausible module placement aligned with the provided adjacency cues. When the partial input is sparse, the module specification acts as a stabilizing prior, producing a broad set of feasible configurations. As the partial input becomes more detailed, the output distribution narrows and more consistently reflects user intent. These results indicate that Fab-aware generation remains robust under both geometric variability and incomplete early-stage requirements.

Iterative Customization

Figure 5 demonstrates an interactive workflow in which users progressively refine partial inputs rather than specifying all constraints upfront. Starting from an initial partial graph and boundary, the model produces multiple candidate solutions. The user then selects a preferred outcome and adjusts the partial input to steer the next iteration.

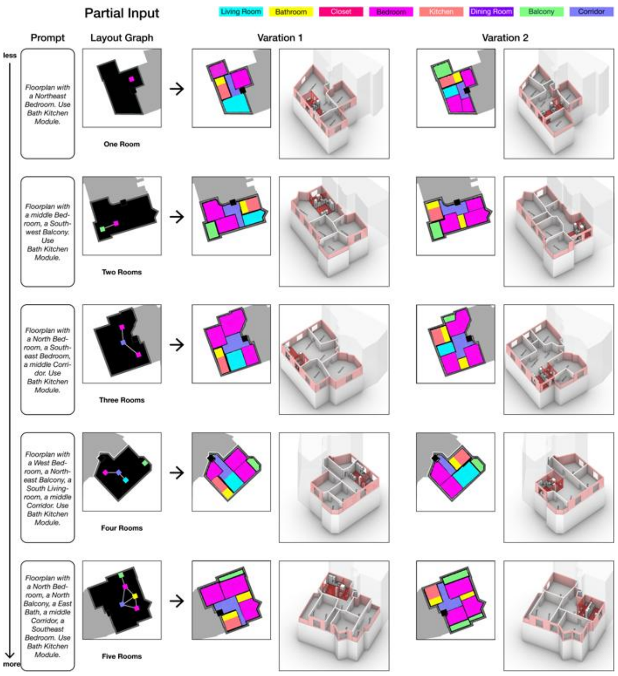

Figure 4: Fabrication-aware results across different boundaries and increasing partial-input completeness.

Modifying constraints directly shapes the design space. For instance, adding constraints yields more precise, less diverse outcomes; editing constraints corrects local functional issues while preserving global structure; and removing constraints increases diversity under the same boundary and module rules. Finally, users can freeze accepted module positions and dimensions, prompting the model to regenerate surrounding spaces while respecting the locked design decisions. This supports gradual convergence from conceptual exploration to fabrication-aligned layouts.

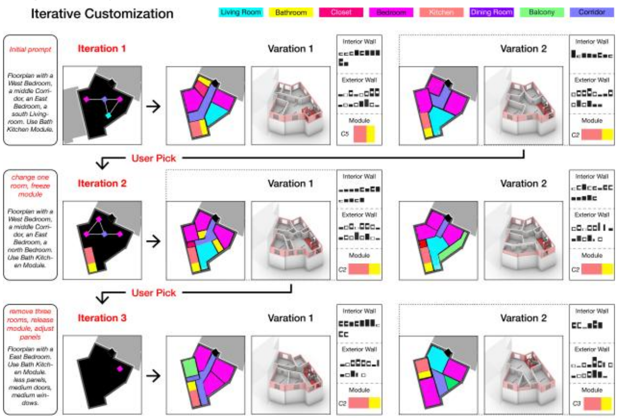

Figure 5: Iterative customization: initial partial input, user selection, add/edit/remove constraints, and module freezing with regenerated layouts.

Evaluation

Our framework is designed for floor plan generation where designers progressively specify site boundaries, partial layouts, and wet-core modules. There is no comparable prior SOTA methods that support the same combination of partial layout input and module-aware generation within a single generator. The framework was quantitatively evaluated on a held-out test split of 1,000 site conditions and prompts. We tracked perceptual visual similarity via Fréchet Inception Distance (FID). A lower FID is better, as it means the generated images are indistinguishable from real drawings. Besides, Module Feasibility is recorded as the percentage of samples whose decoded wet-core matches the specified module.

Table 1 presents a comparative benchmarking of the full framework against three key ablations, quantifying how each component affects architectural validity and module integration. In the default configuration, the full framework achieves the strongest overall performance, attaining a high spatial realism score (FID=5.68) and highest geometry pass rate and module feasibility (85.6%). Deactivating the mode switcher leads to a noticeable deterioration in performance, with module feasibility dropping to 64.2% and FID numbers increasing from 5.68 to 6.54. Omitting the data pipeline's geometric screening yields the poorest performance across the board. Under this scenario, module feasibility drops to its lowest point at 57.8%, while the FID worsens to 8.21. In particular, fine-grained geometric checks like shape convexity drop from 91.8% to 82.6%.

Table 1 Fabrication-aware ablation test

	FID	Mod Fea %	Geometry Pass Rate %			
			Conv	Ratio	Conn	Type
Ours (Full)	5.68	85.6	91.8	85.4	81.4	94.2
w/o Switcher	6.54	64.2	89.4	83.8	69.6	92.7
w/o Filter	8.21	57.8	82.6	77.1	72.4	90.8

Discussion

This study presents a fabrication-aware floorplan generation framework that unifies early-stage spatial reasoning with downstream manufacturability. By fine-tuning the UNet under hybrid conditioning and a mode-switching scheme, the model can generate both concept-level layouts and module-integrated solutions from partial inputs while respecting a fixed boundary. The results indicate that diffusion-based models can serve as a practical bridge between architectural intent and fabrication constraints without relying on brittle, hand-crafted placement rules.

A key contribution is the two-mode training strategy. The design-only path preserves broad generative capability and avoids narrowing the model to a small set of modular templates. The fabrication-aware path introduces explicit module and panel constraints, improving dimensional feasibility and constructability. This separation appears to be critical: without the design-only channel, the model risks reduced diversity and over-regularized outputs; without the fabrication-aware channel, module fit remains inconsistent. The combined approach yields a single deployable model that can be toggled by conditioning, which is advantageous for real design workflows where the level of constraint evolves over time.

The results also highlight the value of the synthetic enrichment pipeline. Module matching and fabrication-prompt generation convert raw design outputs into a structured dataset that injects manufacturing knowledge into training. This offers a scalable alternative to manually annotated fabrication-aware datasets, which are expensive and scarce. The observed robustness across boundaries and partial-input amounts suggests that the enrichment strategy effectively teaches module accommodation as a generalizable behavior rather than a memorized pattern.

However, several limitations remain. First, representing layouts solely as raster outputs may obscure geometric precision needed for detailed coordination, especially for tolerance-sensitive assemblies. Second, the module matching success rate and its dependency on library richness likely influence overall performance; limited module typologies may bias the generator toward a narrower solution space. Third, the current framework infers constructability primarily through textual and library-based constraints rather than explicit structural, MEP routing, code compliance, or egress simulation. These aspects may require additional conditioning signals or post-generation validation.

Conclusion

Overall, our proposed Fab-aware GD framework demonstrates that diffusion models can support boundary-compliant, module-aware floorplan generation from partial inputs, offering a scalable approach to balancing design flexibility with fabrication efficiency. The mode-switching paradigm and synthetic fabrication enrichment provide a practical foundation for hybrid prefabrication workflows where manufacturability must be introduced progressively rather than imposed upfront.

Future work could focus on (1) integrating vector or parametric representations for higher geometric fidelity, (2) expanding module libraries and supporting multi-objective optimization across cost, carbon, and schedule, (3) incorporating explicit code and performance-aware validators in the loop, and (4) evaluating the system with professional designers to quantify usability, time savings, and trust. A promising direction is a closed-loop pipeline where diffusion-based generation, rule-based checking, and module library updates continuously inform one another.

References

Cao, J., Said, H., Savov, A., & Hall, D. (2024). Graph-Based Evolutionary Search for Optimal Hybrid Modularization of Building Construction Projects. Journal of Construction Engineering and Management, 150(8), 04024098. https://doi.org/10.1061/JCEMD4.COENG-14687

Gan, V. J. L. (2022a). BIM-Based Building Geometric Modeling and Automatic Generative Design for Sustainable Offsite Construction. Journal of Construction Engineering and Management, 148(10), 04022111. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002369

Gan, V. J. L. (2022b). BIM-based graph data model for automatic generative design of modular buildings. Automation in Construction, 134, 104062. https://doi.org/10.1016/j.autcon.2021.104062

Ghannad, P., & Lee, Y.-C. (2022). Automated modular housing design using a module configuration algorithm and a coupled generative adversarial network (CoGAN). Automation in Construction, 139, 104234. https://doi.org/10.1016/j.autcon.2022.104234

Li, K., Gan, V. J. L., Li, M., Gao, M. Y., Tiong, R. L. K., & Yang, Y. (2024). Automated generative design and prefabrication of precast buildings using integrated BIM and graph convolutional neural network. Developments in the Built Environment, 18, 100418. https://doi.org/10.1016/j.dibe.2024.100418

Lin, X., Su, P., Lu, W., & Guo, H. (2025). ModulePacking: A Top-Down Generative Design Approach for Modular Key Plans. Journal of Computing in Civil Engineering, 39(1), 04024047. https://doi.org/10.1061/JCCEE5.CPENG-6078

Liu, J., Qiu, Z., Wang, L., Liu, P., Cheng, G., & Chen, Y. (2024). Intelligent floor plan design of modular high-rise residential building based on graph-constrained generative adversarial networks. Automation in Construction, 159, 105264. https://doi.org/10.1016/j.autcon.2023.105264

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., & Rombach, R. (2023). SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2307.01952

Shabani, M. A., Hosseini, S., & Furukawa, Y. (2023). HouseDiffusion: Vector Floorplan Generation via a Diffusion Model with Discrete and Continuous Denoising. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5466–5475. https://doi.org/10.1109/CVPR52729.2023.00529

Sheijani, S. S., Momenaei, A., & Hassanzade, H. (2024). Automating modular open-plan office layouts with performance-based generative design. Automation in Construction, 167, 105692. https://doi.org/10.1016/j.autcon.2024.105692

Van Engelenburg, C., Mostafavi, F., Kuhn, E., Jeon, Y., Franzen, M., Standfest, M., Van Gemert, J., & Khademi, S. (2025). MSD: A Benchmark Dataset for Floor Plan Generation of Building Complexes. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, & G. Varol (Eds.), Computer Vision – ECCV 2024 (Vol. 15116, pp. 60–75). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-73636-0_4

Weber, R. E., Mueller, C., & Reinhart, C. (2022). Automated floorplan generation in architectural design: A review of methods and applications. Automation in Construction, 140, 104385. https://doi.org/10.1016/j.autcon.2022.104385

Zhang, H., & Zhang, R. (2025). Generating accessible multi-occupancy floor plans with fine-grained control using a diffusion model. Automation in Construction, 177, 106332. https://doi.org/10.1016/j.autcon.2025.106332

Zheng, Z., Li, Y., & Torres, J. (2024). An integrated method of automated layout design and optimization for modular construction. Engineering, Construction and Architectural Management, 31(3), 1016–1036. https://doi.org/10.1108/ECAM-04-2022-0329