GRE-Diff: Gaussian Room Embeddings for Structured Layout Diffusion

Jing Wang[1], Haoran Xiong[1], Zihao Yan[1], Minglun Gong[2], and Hui Huang[1]*

[1] Guangdong Provincial Key Laboratory of Visual Media and Multidimensional Intelligence, CSSE, Shenzhen University, China
[2] School of Computer Science, University of Guelph, Canada

Abstract. Designing functional and aesthetically coherent floor plans requires exploring a vast space of possible room arrangements, a task that quickly becomes overwhelming for human designers. In this paper, we propose *GRE-Diff*, a controllable and interactive diffusion-based framework that automates the creation and editing of apartment floor plans under user-specified constraints. By combining AI-generated suggestions with real-time, human-in-the-loop editing, the system enables users to specify room types, room counts, boundary shapes, and editing operations through LLM-parsed instructions or GUI-based interaction. It then generates a diverse set of plausible and well-structured designs for refinement. At the core of our approach is Gaussian Room Embedding (GRE), a continuous latent representation that models each room as a spatial Gaussian distribution capturing its location and extent. Extensive experiments on the RPLAN dataset show that *GRE-Diff* produces high-quality, constraint-aware, and editable polygonal layouts, offering a practical step toward bridging AI-driven automation and human creativity in spatial design.

Keywords: Floor Plan · Layout Generation · Controllable Editing

1 Introduction

Designing functional and well-structured floor plans is inherently challenging, as architects must iteratively refine layouts to satisfy complex spatial and regulatory constraints [27,35]. The resulting design space is highly combinatorial, involving numerous topological and geometric configurations that are difficult to explore exhaustively. Recent advances in artificial intelligence have sought to address this challenge through data-driven generative models, ranging from graph-based reasoning [7, 10, 14, 21–23, 29, 31, 34, 40] to diffusion-based layout synthesis [13, 15, 24–26, 37]. These methods demonstrate the promise of AI to explore complex spatial design spaces beyond human capabilities.

However, enabling controllable generation and iterative refinement within a unified generative framework remains challenging. Unlike pixel-level image synthesis, floor plans are structured geometric systems where rooms are mutually

* Corresponding author.

constrained by boundary validity, spatial adjacency, and functional relationships. This structure introduces three key challenges. First, controllable generation under spatial constraints requires simultaneously maintaining geometric validity, functional consistency, and inter-room relationships. Second, robust editing is challenging because modifying one room often propagates structural changes to adjacent spaces, making it difficult to preserve global layout coherence. Third, existing representations rely on discrete masks or vertex sequences, which are sensitive to ordering and coordinate perturbations, leading to unstable optimization and geometrically brittle layouts. These observations suggest that the fundamental difficulty lies in how room geometry is represented within generative frameworks. To overcome the limitations of discrete room representations, we seek a representation that is continuous, less sensitive to valid vertex-order variations, and geometrically robust.

Inspired by probabilistic embedding models [33] and spatially controllable generative representations [8, 18], we introduce a probabilistic latent representation for room geometry. Specifically, we associate each room with a Gaussian spatial embedding that captures its approximate location and spatial extent. We refer to this representation as the **Gaussian Room Embedding (GRE)**. Building upon this representation, we propose *GRE-Diff*, a Gaussian-guided diffusion framework for controllable floor plan generation and editing. Given user-specified constraints, the framework predicts Gaussian room embeddings that guide a diffusion process to generate coherent polygonal layouts. Operating in the continuous GRE space enables unified layout generation and iterative editing within a single diffusion model. Furthermore, the system supports both natural-language instructions and interactive graphical editing, allowing users to refine layouts while maintaining global spatial coherence.

Our main contributions are summarized as follows:

- We introduce Gaussian Room Embeddings, a continuous probabilistic room representation that summarizes room location and spatial extent, reducing sensitivity to valid vertex-order variations in polygonal layouts.
- We propose *GRE-Diff*, a Gaussian-guided diffusion framework that unifies floor plan generation and editing by operating in the GRE latent space and decoding coherent polygonal layouts.
- We design a dual-path conditioning architecture that captures both semantic cues and geometric constraints, enabling controllable and structure-preserving layout synthesis and refinement.

2 Related Work

We mainly review the learning-based methods as they are closely related to our research. Based on the type of input conditions, we categorize existing studies into three groups: boundary-constrained, bubble-constrained, and text-constrained floor plan generation.

2.1 Boundary-constrained generation

Boundary constraints are commonly used in floor plan generation to ensure that the layout adheres to predefined building boundaries or exterior walls, which can be specified by designers or users. Early methods, such as variational autoencoders (VAEs) [2, 16], approached floor plan generation as a raster image synthesis problem, where boundary constraints guided the placement of rooms. RPLAN [36] enhanced layout accuracy and efficiency by generating floor plans from raster images. iPLAN [10] mimicked the human design process by iteratively predicting room locations and partitions. WallPlan [28] employed two alternating modules to predict wall and room functions, but relied on post-processing to obtain complete layouts, which limited output diversity. MaskPLAN [40] improved spatial coherence by refining individual elements and decoupling geometric interdependencies. However, boundary-based methods primarily focus on geometric validity, often overlooking the semantic relationships among rooms. This limitation led to the emergence of bubble-constrained methods, which introduce higher-level spatial reasoning.

2.2 Bubble-constrained generation

Another line of research [19,31] leverage bubble diagrams as abstract spatial constraints, providing room connectivity, adjacency, and relative proportions before geometric realization, usually combined with graph neural networks (GNNs) [7, 14, 34] and generative adversarial networks (GANs) [21, 22, 29]. More recently, vision transformers [23, 32] and diffusion models [6, 24] have been explored for layout synthesis. Graph2Plan [14] used sparse user constraints and bubble diagrams with GNNs to generate coherent floor plans. HouseGAN++ [21] iteratively generated floor plans using a graph-constrained GAN, effectively transforming input graphs into coherent and plausible layouts. HouseDiffusion [26] generated vectorized floor plans by denoising room coordinates through a transformer-based diffusion model with graph constraints. Cons2Plan [13] improved flexibility by predicting connected edges to handle multiple conditional inputs, enhancing diversity in layout generation. GSDiff [15] generated wall junctions and segments while maintaining geometric consistency through structured graphs. Graph-based methods are effective when reliable topology constraints are available, but they often require explicit graph inputs or post-processing, and are less direct for localized editing when users provide sparse or evolving constraints.

2.3 Text-constrained Generation

Recent advances in multi-modal models have introduced natural language as a conditioning signal for floor plan generation [5]. Several works enable human-in-the-loop design by translating textual or sketch-based prompts into spatial constraints, allowing interactive layout generation and refinement [17,25]. Other approaches leverage large language models (LLMs) to generate structured layout descriptions that guide subsequent geometric synthesis, such as HouseLLM [42]

and FP-LLaMa [39]. These methods demonstrate the potential of natural-language-driven design interfaces, but still struggle to achieve precise control over local geometry and maintain spatial coherence during layout generation.

3 Method

3.1 Gaussian Room Embedding

Following prior floor plan generation works [4, 26], we represent each room in the layout as a polygonal region. Each room i is represented as a polygon $x_0^i = [v_1^i, \ldots, v_N^i]$, where $v_j^i \in \mathbb{R}^2$ denotes the j-th vertex, ordered clockwise to form a closed shape. This polygon representation provides a precise vector description of room geometry and serves as the final layout produced by our model. Motivated by the representation challenges discussed in Sec. 1, we introduce **Gaussian Room Embedding (GRE)**, a continuous probabilistic representation that provides a structured room-level latent description of polygonal geometry. Specifically, each room is associated with a two-dimensional Gaussian distribution $g_i = \mathcal{N}(\mu_i, \sigma_i^2 I)$, where the mean $\mu_i \in \mathbb{R}^2$ represents the room centroid and σ_i is a scalar room-level scale parameter. In this work, we use an isotropic Gaussian parameterization, where $\sigma_i^2 I \in \mathbb{R}^{2\times 2}$ is not a full covariance matrix but a compact approximation of the room's spatial extent. Intuitively, μ_i determines the room location, while σ_i summarizes its coarse geometric scale and spatial influence within the layout. GRE does not replace the explicit polygon representation. Instead, it serves as a compact room-level spatial state that initializes and guides the diffusion process, reducing sensitivity to valid vertex-order variations and coordinate perturbations compared with directly diffusing ordered polygon vertices. The final output remains an explicit polygonal layout decoded by the denoising process.

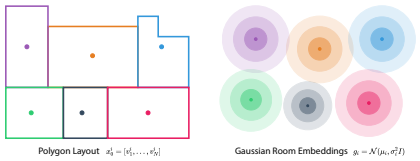

Fig. 1: Polygon layout and its Gaussian Room Embedding. GRE provides a continuous room-level spatial state that initializes and guides polygon diffusion, while the final output remains an explicit polygonal layout.

3.2 Gaussian-guided Diffusion Framework

Building upon the Gaussian Room Embedding representation introduced in Sec. 3.1, we formulate floor plan generation as a Gaussian-guided diffusion process. The proposed framework first predicts structured Gaussian room embeddings from user conditions and then generates polygonal room layouts through a conditional diffusion model.

Diffusion Formulation. Given the Gaussian room embeddings predicted by *GuidanceNet*, we generate room polygons through a diffusion-based denoising

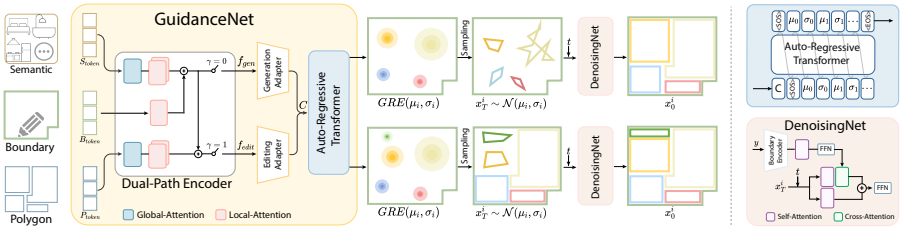

Fig. 2: Inference process of *GRE-Diff*. *GuidanceNet* encodes semantic (S_{token}), boundary (B_{token}), and polygon (P_{token}) conditions to predict Gaussian room embeddings (μ_i, σ_i). Samples drawn from $\mathcal{N}(\mu_i, \sigma_i^2 I)$ serve as diffusion initialization and are iteratively refined by *DenoisingNet* under boundary constraints to generate vectorized layouts x_0^i. The resulting layouts can be re-encoded as polygon tokens, enabling iterative refinement and interactive editing.

process. Instead of initializing the diffusion process from isotropic Gaussian noise, we sample the initial state of each room from its Gaussian embedding: $x_T^i \sim \mathcal{N}(\mu_i, \sigma_i^2 I)$. At each time step $t \in [0, T]$, $\mu_t^i = (1 - \sqrt{\overline{\alpha}_t})\mu_i$, $\sigma_t^i = (\sqrt{1 - \overline{\alpha}_t})\sigma_i$. The denoising process estimates the previous state x_{t-1}^i from the current noisy state x_t^i through the transition distribution $p(x_{t-1}^i | x_t^i)$. The update rule is:

$$x_{t-1}^i = \frac{1}{\sqrt{\alpha_t}} \left(x_t^i - \mu_t^i - \frac{1 - \alpha_t}{1 - \overline{\alpha}_t} \sigma_t^i \cdot \boldsymbol{D}(x_t, i, y) \right) + \mu_t^i + \sigma_t z^i,$$

where α_t is the noise attenuation coefficient that controls the intensity of noise removal at each step, and $z^i \sim \mathcal{N}(0, I)$ introduces stochasticity for sampling. Here, $D(x_t, i, y)$ represents *DenoisingNet*, whose architecture is described in the next section, and predicts the noise residuals conditioned on structural guidance y, which includes (1) a *boundary image* defining the apartment contour, and (2) *anchor masks* indicating user-specified fixed regions during interactive editing. This conditional formulation enables the network to generate geometrically valid and semantically coherent layouts.

GuidanceNet GuidanceNet consists of four modules: a multi-conditional encoder that tokenizes user inputs, a dual-path encoder for semantic–geometric feature fusion, task-specific adapters for generation and editing modes, and an autoregressive transformer that sequentially predicts GRE.

Multi-conditional Encoding. As illustrated in Fig. 2, GuidanceNet encodes multi-modal user conditions into structured Gaussian representations that capture room position, size, and type. It transforms input conditions into three types of tokens: room semantics (S_{token}), apartment boundaries (B_{token}), and geometric polygons (P_{token}), respectively. S_{token} encodes functional intent derived from the LLM-parsed JSON that specifies room types and quantities (*e.g.*, three bedrooms and one kitchen). B_{token} describes the apartment contour using 64 uniformly sampled points, providing a global geometric constraint that

bounds the layout. P_{token} represents existing room geometries with vertex-level precision and acts as a spatial prior for refinement.

Dual-path Encoder. As illustrated in Fig. 2, the dual-path encoder jointly processes semantic, boundary, and polygonal conditions through complementary attention pathways. Specifically, S_{token} and P_{token} are independently processed by separate global attention modules, each modeling long-range spatial dependencies within its respective token domain. No cross-attention is applied between these branches, allowing functional semantics and geometric priors to be learned independently before feature fusion. Meanwhile, boundary tokens B_{token} are processed by a local attention branch that focuses on fine-grained geometric alignment along the apartment contour. The overall encoded feature representation is formulated as:

$$C = \text{Adapter}\big(\text{Enc}(S_{token}) + \text{Enc}(B_{token}) + \gamma\text{Enc}(P_{token})\big),$$

where $\text{Enc}(\cdot)$ denotes the corresponding attention-based encoder for each token type, and $\text{Adapter}(\cdot)$ denotes the task-specific adapter for either generation or editing. The coefficient γ modulates the contribution of polygonal cues, with $\gamma = 0$ for generation and $\gamma = 1$ for editing. This design enables GuidanceNet to maintain a unified semantic–geometric latent space while adapting its representation to layout generation and editing.

Autoregressive Transformer. To sequentially generate room representations conditioned on semantic and geometric context, we employ an autoregressive transformer. At each step, the transformer predicts the Gaussian parameters of the i-th room based on the previously generated ones:

$$(\mu_i, \sigma_i) = \text{ARTrans}\big(C,\, \langle\text{SOS}\rangle,\, \langle\mu_{<i}, \sigma_{<i}\rangle\big),$$

where $\langle\mu_{<i}, \sigma_{<i}\rangle$ denotes the sequence of previously predicted parameters, and C provides global semantic–geometric conditioning. Starting from $\langle\text{SOS}\rangle$, the model auto-regressively outputs $(\mu_0, \sigma_0), (\mu_1, \sigma_1), \ldots, (\mu_N, \sigma_N)$, where each pair (μ_i, σ_i) defines the spatial center and extent of the i-th room. This formulation models spatial dependencies among rooms, including adjacency, functional grouping, and relative scaling, thereby preserving coherent layout structure at both global and local levels.

DenoisingNet DenoisingNet serves as the geometric decoder of *GRE-Diff*, transforming Gaussian embeddings from GuidanceNet into explicit polygonal room layouts through a conditional diffusion process. As shown in Fig. 2, DenoisingNet employs a dual-attention transformer tailored for vectorized floor plan representation and refinement. A boundary encoder [11] extracts geometric constraints and fuses them with latent room embeddings through cross-attention, while a parallel self-attention pathway captures inter-room spatial dependencies. The fused features are then processed through feed-forward layers to predict $D(x_t, i, y)$, guiding the denoising trajectory toward structurally consistent configurations. During editing, additional anchor masks within y enable localized resampling and refinement while maintaining the global topology.

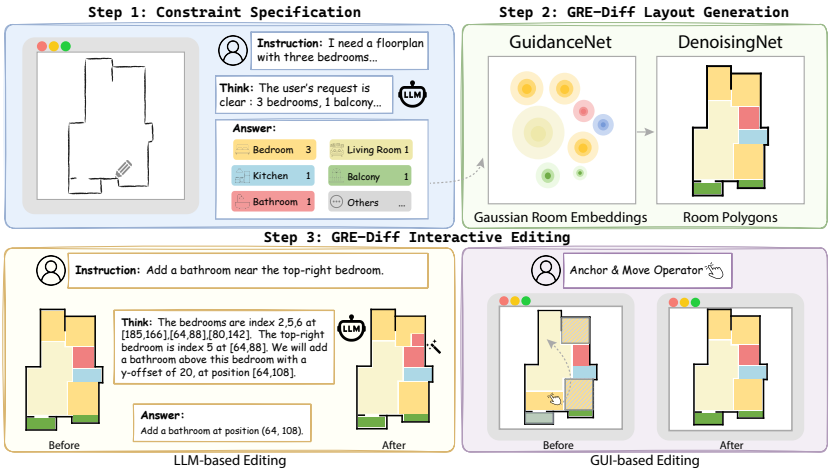

Fig. 3: Overview of *GRE-Diff*, a controllable framework for floor plan generation and editing. Given apartment boundaries and room constraints, *GuidanceNet* predicts Gaussian room embeddings that guide the diffusion process, while *DenoisingNet* generates coherent polygonal layouts. The framework supports both language-based and GUI-based editing for interactive layout refinement.

3.3 User-guided Layout Generation and Editing

As illustrated in Fig. 3, *GRE-Diff* consists of three stages: constraint specification, layout generation, and dual-mode editing.

Step 1: Initial Constraint Specification. Users describe layout requirements using natural language (e.g., "three bedrooms and one kitchen"). A large language model (LLM), such as Kimi-K2 [30], converts the request into a structured layout specification that defines room categories and quantities. Users can also provide apartment boundaries by sketching or uploading floor outlines, which determine the spatial extent of the layout. Compared with prior methods [14, 40] that rely on manually defined topological graphs or attributes, our interface enables more flexible and intuitive control through natural-language instructions. Additional details are provided in the supplementary material.

Step 2: Layout Generation. These structured conditions are subsequently processed by *GRE-Diff*, which integrates two core modules: *GuidanceNet* and *DenoisingNet*. Specifically, *GuidanceNet* encodes the multi-modal tokens into Gaussian embeddings that capture both spatial and semantic priors, while *DenoisingNet* iteratively refines them through conditional diffusion to produce coherent polygonal layouts that strictly satisfy user constraints.

Step 3: Dual-Mode Editing. To refine a generated layout, *GRE-Diff* supports interactive editing through either LLM-based instructions or polygon-level manipulation via a GUI. The framework provides five basic editing operators: *Add*, *Delete*, *Move*, *Repurpose*, and *Anchor*. The first four operators modify room attributes such as location or type, whereas *Anchor* fixes selected rooms during

editing. *GRE-Diff* does not regenerate the entire layout from scratch. Instead, user-anchored rooms are enforced as hard constraints through a fixed-room index mask: after each denoising step, their polygon vertices are copied back from the input layout. The remaining editable rooms are resampled under the predicted GRE priors, the boundary condition (y), and the current layout context. This mechanism enables localized refinement while preserving fixed structures and maintaining global spatial coherence.

4 Results and Evaluation

4.1 Experimental Setup

Dataset. We use the RPLAN dataset [36], which contains over 80,000 vectorized floor plans with detailed room types, boundaries, and structural annotations. Its diverse residential layouts reflect real-world complexity and are widely adopted in prior works [14, 36]. For our experiments, we select six room types: living room, bedroom, kitchen, bathroom, dining room, and balcony. The data is split into 72,709 samples for training and 8,079 for testing.

Metrics. Following prior works [9, 13, 15], we evaluate our method across seven dimensions: *Distribution similarity* (FID [12], KID [3], MMD [1, 9]), *Generative diversity* (COV [9]), *Generative novelty* (LPIPS [41]), visual realism, *Conditional controllability* (BC, RC, $F1_{BC\text{-}RC}$), *Editing Effectiveness* (Tiny-ROE [38]), and *Efficiency* (parameters, inference time). Metrics are averaged over five runs using 512 randomly sampled test layouts. For *functional validity*, the overlap-free rate measures the percentage of generated layouts without positive-area intersections between any pair of room polygons. The reachability rate evaluates whether the room connectivity graph forms a single connected component, where rooms are treated as nodes and derived door connections are treated as edges. These metrics complement BC and RC by assessing whether the generated layouts are not only constraint-satisfying but also spatially functional. Additional details are provided in the supplementary materials.

Training. *GRE-Diff* is trained for 300 epochs on eight NVIDIA Quadro GV100 GPUs with a batch size of 40 per GPU. We use the AdamW optimizer [20] with a learning rate of 2×10^{-4}, applying linear warm-up followed by cosine decay. Further details of the forward and reverse diffusion processes and loss formulations are provided in the supplementary materials.

4.2 Evaluation of Generation Capability

Quantitative Evaluation with SOTAs. We evaluate the generation performance of *GRE-Diff* against seven state-of-the-art methods, including iPLAN [10],

Table 1: Comparison with prior floor plan generation and editing methods. A checkmark (✓) indicates the available conditioning type (boundary, bubble, room type). The best cells are highlighted in light green and the second best in light blue. Our method attains the best overall similarity scores and the highest controllability with competitive efficiency among diffusion-based models.

Method	Bound.	Bubble	RType	Similarity (↓)			Diversity (↑)	Controllability (↑)			Efficiency (↓)	
				FID	$KID_{(\times 10^{-3})}$	MMD	COV	BC	RC	$F1_{BC\text{-}RC}$	Param (M)	Time (S)
Graph2Plan [14]	✓	✓	✓	4.81	1.65	0.124	94.92%	**100.00%**	83.11%	90.78%	**8**	**0.41**
iPLAN [10]	✓		✓	12.63	12.20	0.148	75.11%	47.53%	85.65%	60.75%	46	1.61
HouseDiffusion [26]		✓	✓	6.55	2.83	0.193	76.36%	–	93.35%	–	27	8.76
WallPLAN [28]	✓			6.73	4.46	0.149	58.40%	99.01%	–	–	106	0.72
MaskPLAN [40]	✓		✓	23.54	22.84	0.190	67.18%	**100.00%**	20.12%	33.50%	998	2.68
GSDiff [15]	✓			4.55	1.13	0.118	86.13%	91.61%	–	–	137	0.67
ChatHouseDiff. [25]	✓	✓	✓	5.21	1.31	0.145	83.01%	99.80%	53.91%	70.01%	82	1.15
Ours	✓			**4.36**	**0.96**	**0.107**	**96.09%**	98.44%	**100.00%**	**99.21%**	60	0.91

Graph2Plan [14], HouseDiffusion [26], WallPLAN [28], MaskPLAN [40], GSDiff [15], and ChatHouseDiffusion [25]. All models are evaluated under identical constraint conditions to ensure fair comparison. HouseDiffusion is evaluated without boundary constraints, as its original formulation does not support explicit boundary conditioning. Integrating boundary constraints would require non-trivial architectural modifications beyond the scope of the original model.

Table 1 shows that *GRE-Diff* achieves the best overall performance among the compared methods. It obtains the lowest FID of 4.36, improving over GSDiff, Graph2Plan, ChatHouseDiffusion, and HouseDiffusion by 4.2%, 9.4%, 16.3%, and 33.4%, respectively. It also achieves the lowest KID (0.96×10^{-3}) and MMD (0.107), indicating strong distributional similarity to real layouts. In terms of generative diversity, *GRE-Diff* attains the highest coverage of 96.09%, showing its ability to explore diverse layout variations. For controllability, it achieves a high BC of 98.44% and the best RC of 100.00%, resulting in the highest $F1_{BC\text{-}RC}$ score of 99.21%. With 60M parameters, *GRE-Diff* generates one layout in 0.91s, faster than iPLAN and ChatHouseDiffusion and comparable to WallPLAN.

We further note that Graph2Plan relies on a retrieval-assisted refinement pipeline to obtain its final layouts. As a diagnostic comparison, removing this non-neural refinement increases its FID to 20.10 and reduces RC to 50.19%, with BC and $F1_{BC\text{-}RC}$ dropping to 0.00%. This suggests that its final performance depends heavily on post-hoc refinement. In contrast, *GRE-Diff* directly synthesizes vectorized polygonal layouts within a unified diffusion framework, while also supporting local editing under user-specified constraints.

Existing methods exhibit different trade-offs between geometric validity, room-level controllability, diversity, and post-processing dependency. Graph2Plan depends on retrieval and boundary alignment, achieving high controllability but low diversity. iPLAN and MaskPLAN use progressive room localization and attribute decomposition with heavy post-processing, often producing over-dispersed layouts or empty regions. WallPLAN and GSDiff generate wall-line polygons via snapping and alignment, improving local precision but reducing layout diversity and failing to meet room-level constraints. HouseDiffusion and ChatHouseDiffusion omit post-processing entirely, resulting in disconnected or inconsistent room

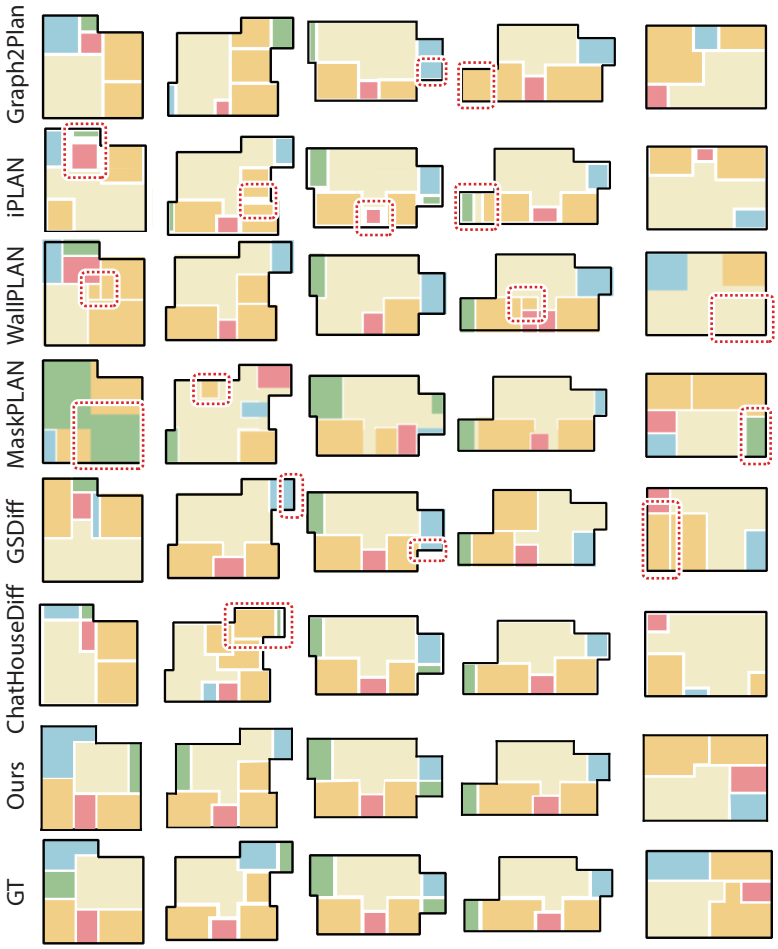

Fig. 4: Qualitative comparison of floor plan generation. The red dashed regions highlight failure cases in existing methods, often arising from inconsistencies in room numbers, functional mismatches, or inaccessible layouts. *GRE-Diff* generates structurally coherent and functionally consistent results, closely resembling GT layouts.

geometries. In contrast, *GRE-Diff* embeds both boundary and room constraints directly into the diffusion dynamics, ensuring structural correctness, maintaining high diversity, and achieving real-time efficiency within a compact model.

Qualitative Evaluation with SOTAs. To better illustrate the quantitative superiority, Fig. 4 visualizes representative results from different methods. Graph2Plan and WallPLAN often fail to generate all required rooms, resulting in incomplete layouts. iPLAN and ChatHouseDiffusion produce unintended gaps that break spatial continuity and hinder accessibility. MaskPLAN frequently ex-

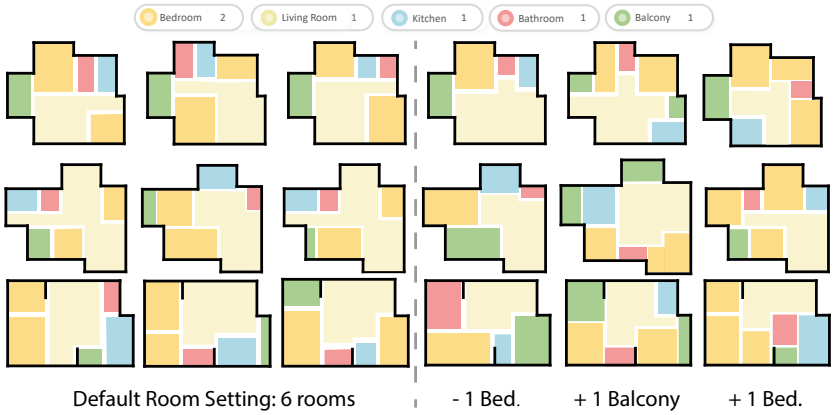

Fig. 5: Results of floor plans generated under various configurations. This figure shows the network's adaptability in generating layouts with consistent room configurations and its flexibility in adjusting room arrangements within a fixed boundary, underscoring the impact of load-bearing wall constraints. The network autonomously optimizes room layouts while maintaining spatial organization and functionality.

hibits discontinuous or fragmented boundaries, further reducing geometric consistency. In contrast, *GRE-Diff* preserves complete room structures, produces smooth and closed boundaries, and maintains balanced spatial organization, consistent with its strong numerical performance. More qualitative results are presented in the supplementary materials.

Diversity. As illustrated in Fig. 5, we examine the controllable diversity of *GRE-Diff* under various input conditions. (1) With identical functional requirements (*e.g.*, one living room, one balcony, one bathroom, and two bedrooms), the model generates structurally diverse yet functionally consistent layouts, demonstrating its stochastic generative capability (the first three columns). (2) When room types vary while the building boundary remains fixed, the model adaptively reorganizes the layout to preserve functional logic, *e.g.*, aligning balconies with exterior walls and placing living rooms at the center (last three columns). (3) Under structural constraints such as fixed load-bearing walls, our method naturally respects these limitations and maintains spatial organization (the last row). These results collectively demonstrate that *GRE-Diff* achieves both high generative diversity and controllable structural consistency.

Realism. We conducted a user study to evaluate the quality of our generated floor plans against Graph2Plan and ground truth (GT) layouts. Using boundaries, room types, and counts extracted from GT as constraints, we assessed 30 floor plans divided into two groups: one without room location input (compared to GT), and one with location input (compared to Graph2Plan). Participants, unaware of layout sources, selected their preferred design from ran-

domized options. We invited 58 computer science graduate students, collecting 1,740 responses. Preference counts were: GT/Ours/Cannot tell — 342/479/49; Graph2Plan/Ours/Cannot tell — 242/568/60. These results indicate that participants preferred our layouts over GT in 60.69% of cases, and over Graph2Plan in 70.12%, demonstrating the strong realism of our generated floor plans.

Novelty. We assess perceptual similarity between generated and reference layouts using LPIPS [41]. As shown in Fig. 6, most scores lie within the range [0.03, 0.05], indicating high perceptual alignment, while samples above 0.06 correspond to intentionally diverse geometric configurations. For illustration, we display samples from the 2nd, 4th, 7th, and 9th groups along with their most similar counterparts retrieved from the training set. Visualizations across LPIPS ranges highlight both close resemblance (low LPIPS) and novelty (high LPIPS), demonstrating the model's ability to generate realistic and diverse layouts. These results confirm that *GRE-Diff* preserves structural fidelity while enabling diverse layout variations.

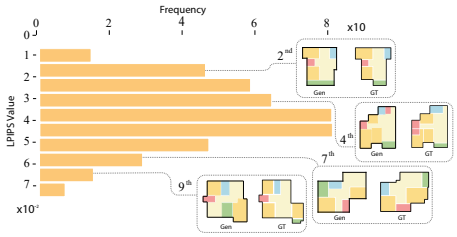

Fig. 6: Novelty analysis using LPIPS. We plot the LPIPS distribution of 500 randomly generated layouts and visualize representative samples from different percentile groups together with their nearest retrieved training counterparts.

Functionality. We further evaluate the functional usability of generated layouts for downstream applications by measuring overlap-free rate and reachability rate on 19,680 generated layouts. A layout is considered overlap-free if no pair of room polygons has a positive-area invalid intersection; 95.56% of the generated layouts satisfy this criterion. For reachability, we use the annotated door labels to construct a room connectivity graph, where rooms are nodes and door connections are edges. A layout is considered reachable if all rooms belong to a single connected component through the labeled door connections. Under this criterion, 96.35% of the generated layouts are reachable. These results indicate that *GRE-Diff* produces largely non-overlapping and door-connected layouts, supporting downstream applications such as furniture arrangement and interior layout refinement.

4.3 Evaluation of Editing Capability

Quantitative Evaluation with SOTAs. As shown in Table 2, our method achieves a 0.00% Tiny-ROE across all operations, indicating that every requested edit is correctly applied. In terms of editing accuracy, our F1 scores reach 86.27% for *Add*, 98.58% for *delete*, and 97.59% for *Anchor & Move*, which outperform

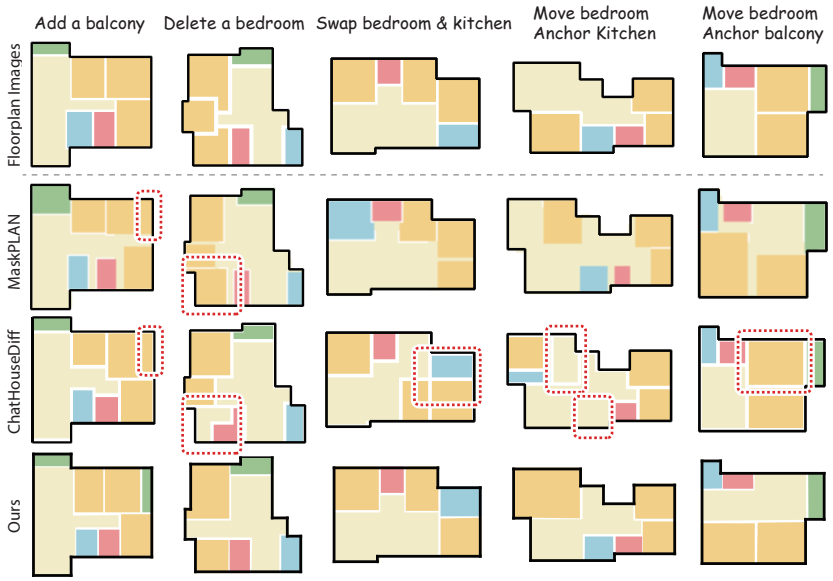

Fig. 7: Qualitative comparison of floor plan editing. We apply editing operations to the original ground truth layout, as shown in the first row. Compared to other methods, our approach provides more precise control over room positions, ensuring that fixed structures (balcony or kitchen) remain unchanged while enabling flexible room relocation. Red dashed boxes mark incorrect edits from other methods.

the best baseline (ChatHouseDiffusion) by +17.8, +40.3, and +24.5 points, respectively. Although MaskPLAN achieves a 0.00% Tiny-ROE, it suffers from much lower F1. These results demonstrate that our model not only executes all edit operations successfully but also maintains superior geometric and semantic consistency across diverse editing types.

Table 2: Editing performance comparison across different operation types (*Add*, *Delete*, and *Anchor & Move*).

Method	Add		Delete		Anchor & Move	
	Tiny-ROE (%) ↓	$F1_{BC\text{-}RC}$ ↑	Tiny-ROE (%) ↓	$F1_{BC\text{-}RC}$ ↑	Tiny-ROE (%) ↓	$F1_{BC\text{-}RC}$ ↑
MaskPLAN	**0.00**	49.47	**0.00**	24.09	**0.00**	66.05
ChatHouseDiffusion	68.49	68.45	10.96	58.26	4.11	73.06
Ours	**0.00**	**86.27**	**0.00**	**98.58**	**0.00**	**97.59**

Table 3: Ablation study on the effect of *Global Attention* (*GA*) and *Local Attention* (*LA*) in *GuidanceNet*.

GA	LA	FID↓	$KID_{(\times 10^{-3})}$↓	MMD↓	COV↑	BC↑	RC↑	$F1_{BC\text{-}RC}$↑
✗	✓	4.57	0.87	0.153	86.52%	96.67%	99.80%	98.21%
✓	✗	4.49	**0.82**	0.145	86.71%	97.07%	99.41%	98.23%
✓	✓	**4.36**	0.96	**0.107**	**96.09%**	**98.44%**	**100.00%**	**99.21%**

Qualitative Evaluation with SOTAs. Fig. 7 shows our model achieving superior editing consistency and spatial preservation. It maintains fixed structures (*e.g.*, balcony or anchored rooms) while adaptively updating surrounding regions. In contrast, MaskPLAN often suffers from boundary drift and unassigned

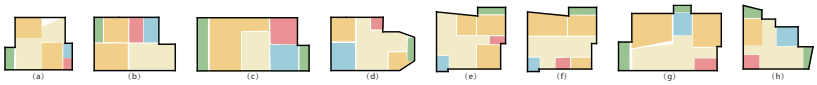

Fig. 8: Representative limitation cases. Examples include (a) uneven vertex density; (b) incorrect room connectivity; (c) semantic mismatch; (d) limited generalization to non-rectilinear boundaries; (e-f) diverse layouts under identical non-Manhattan boundary; (g) a tilted bedroom; and (h) correctly places two balconies along a sloped wall. Despite these, the layouts remain structurally coherent and spatially plausible.

spaces caused by mask misalignment. We additionally implemented the *Move* and *Anchor* operations in ChatHouseDiffusion for a fair comparison; however, the method frequently produces unstable results with fragmented or overlapping rooms. Additional qualitative results and implementation details are provided in the supplementary material.

Automatic Refinement. Beyond user-guided editing, *GRE-Diff* integrates an automatic refinement mechanism based on BC and RC. This system provides real-time feedback to detect and correct issues such as overlaps, voids, or irregular geometries during generation. See the supplement for more details.

4.4 Ablation Studies

Table 3 evaluates the roles of global attention (GA) and local attention (LA) in *GuidanceNet*. Using only LA weakens global guidance and reduces layout plausibility (FID = 4.57). Using only GA better preserves global structure (FID = 4.49) but lacks room-level differentiation, leading to imbalanced room scales and lower COV (86.71%). Combining GA and LA leverages their complementary strengths, achieving the best performance with the lowest FID (4.36) and highest COV (96.09%). Additional ablations, including GRE-guided diffusion variants and robustness analyses, are provided in the supplementary material. Detailed GRE ablations in the supplementary material further show that integrating GRE into the diffusion dynamics is more effective than using GRE only as an external condition, validating its role as a room-level geometric state.

4.5 Limitations Analysis

While *GRE-Diff* achieves strong overall performance, it occasionally fails under complex or incomplete spatial constraints (*e.g.*, irregular boundaries), as shown in Fig. 8. In such cases, the model may produce suboptimal room adjacencies or disproportionate room sizes, especially when boundary geometries deviate markedly from the training distribution. Examples (a–c) correspond to regular boundaries with uneven vertex density or misplaced functional zones, whereas (d–h) depict non-orthogonal or highly irregular contours that can lead to distorted room shapes.

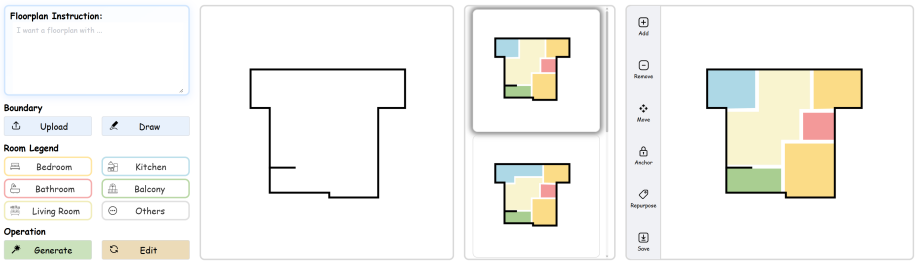

Fig. 9: Snapshot of the intuitive and user-friendly floor plan design interface.

4.6 User Interface

Fig. 9 shows the user interface of the floor plan design system. Users can begin by providing natural-language instructions, while optionally uploading or sketching the apartment boundary (left panel). After clicking the *Generate* button, *GRE-Diff* produces multiple layout candidates that satisfy the given boundary and room-type constraints (center panel). Once a layout is selected, users can refine it interactively through intuitive operations such as *Add*, *Remove*, *Move*, *Anchor*, and *Repurpose* (right panel). The refined floor plan can then be exported in vectorized form by clicking the *Save* button. This interface provides a seamless and user-friendly workflow for iterative floor plan generation and editing.

5 Conclusion and Future Work

We presented *GRE-Diff*, a controllable framework for floor plan generation and editing based on Gaussian Room Embeddings (GRE) and Gaussian-guided diffusion. The proposed design enables stable modeling of room configurations while supporting flexible user interaction. Through its *GuidanceNet* architecture, dual-path encoding, and real-time refinement workflow, *GRE-Diff* produces high-quality, vectorized layouts that remain structurally coherent under diverse user constraints. By supporting both natural-language instructions and direct GUI manipulation, the system bridges automated layout synthesis with intuitive designer control, advancing the capabilities of AI-assisted spatial design. Future work will explore broader architectural typologies and extend the framework toward holistic indoor layout generation, including furniture placement for intelligent design automation.

Acknowledgments

This work was supported in part by ICFCRT (W2441020), NSFC(62402323), Guangdong Basic and Applied Basic Research Foundation (2023B1515120026), Shenzhen S&T Program (KQTD20210811090044003, KJZD20240903100022028), and Guangdong Provincial Key Laboratory of Visual Media and Multidimensional Intelligence.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International conference on machine learning. pp. 40–49. PMLR (2018)
2. Bao, J., Chen, D., Wen, F., Li, H., Hua, G.: Cvae-gan: fine-grained image generation through asymmetric training. In: Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition. pp. 2745–2754 (2017)
3. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. arXiv preprint arXiv:1801.01401 (2018)
4. Chen, J., Deng, R., Furukawa, Y.: Polydiffuse: Polygonal shape reconstruction via guided set diffusion models. Proc. Conf. on Neural Information Processing Systems **36** (2024)
5. Chen, Q., Wu, Q., Tang, R., Wang, Y., Wang, S., Tan, M.: Intelligent home 3d: Automatic 3d-house design from linguistic descriptions only. In: Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition. pp. 12625–12634 (2020)
6. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Proc. Conf. on Neural Information Processing Systems **34**, 8780–8794 (2021)
7. Dupty, M.H., Dong, Y., Leng, S., Fu, G., Goh, Y.L., Lu, W., Lee, W.S.: Constrained layout generation with factor graphs. In: Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition. pp. 12851–12860 (2024)
8. Epstein, D., Park, T., Zhang, R., Shechtman, E., Efros, A.A.: Blobgan: Spatially disentangled scene representations. In: Proc. Euro. Conf. on Computer Vision. pp. 616–635. Springer (2022)
9. Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., Fidler, S.: Get3d: A generative model of high quality 3d textured shapes learned from images. In: Proc. Conf. on Neural Information Processing Systems (2022)
10. He, F., Huang, Y., Wang, H.: iplan: Interactive and procedural layout planning. In: Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition. pp. 7793–7802 (2022)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition. pp. 770–778 (2016)
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. Proc. Conf. on Neural Information Processing Systems **30** (2017)
13. 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)
14. Hu, R., Huang, Z., Tang, Y., Van Kaick, O., Zhang, H., Huang, H.: Graph2plan: Learning floorplan generation from layout graphs. ACM Trans. on Graphics **39**(4), 118–1 (2020)
15. Hu, S., Wu, W., Wang, Y., Xu, B., Zheng, L.: Gsdiff: Synthesizing vector floorplans via geometry-enhanced structural graph generation. In: Proc. AAAI Conf. on Artificial Intelligence. vol. 39, pp. 17323–17332 (2025)
16. Kingma, D.P.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
17. Leng, S., Zhou, Y., Dupty, M.H., Lee, W.S., Joyce, S.C., Lu, W.: Tell2design: A dataset for language-guided floor plan generation. arXiv preprint arXiv:2311.15941 (2023)

18. Li, Y., Li, L., Zhang, Z., Li, X., Wang, G., Li, H., Cun, X., Shan, Y., Zou, Y.: Blobctrl: A unified and flexible framework for element-level image generation and editing. arXiv preprint arXiv:2503.13434 (2025)
19. Liu, J., Xue, Y., Duarte, J., Shekhawat, K., Zhou, Z., Huang, X.: End-to-end graph-constrained vectorized floorplan generation with panoptic refinement. In: Proc. Euro. Conf. on Computer Vision. pp. 547–562. Springer (2022)
20. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
21. Nauata, N., Hosseini, S., Chang, K., Chu, H., Cheng, C., Furukawa, Y.: House-gan++: generative adversarial layout refinement networks. arXiv preprint arXiv:2103.02574 (2021)
22. Nauata, N., Chang, K.H., Cheng, C.Y., Mori, G., Furukawa, Y.: House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In: Proc. Euro. Conf. on Computer Vision. pp. 162–177. Springer (2020)
23. Para, W., Guerrero, P., Kelly, T., Guibas, L.J., Wonka, P.: Generative layout modeling using constraint graphs. In: Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition. pp. 6690–6700 (2021)
24. Ploennigs, J., Berger, M.: Automating computational design with generative ai. Civil Engineering Design **6**(2), 41–52 (2024)
25. Qin, S., He, C., Chen, Q., Yang, S., Liao, W., Gu, Y., Lu, X.: Chathouse-diffusion: Prompt-guided generation and editing of floor plans. arXiv preprint arXiv:2410.11908 (2024)
26. Shabani, M.A., Hosseini, S., Furukawa, Y.: Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In: Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition. pp. 5466–5475 (2023)
27. Sully, A.: Interior Design: conceptual basis. Springer (2015)
28. Sun, J., Wu, W., Liu, L., Min, W., Zhang, G., Zheng, L.: Wallplan: Synthesizing floorplans by learning to generate wall graphs. ACM Trans. on Graphics **41**(4), 1–14 (2022)
29. Tang, H., Shao, L., Sebe, N., Van Gool, L.: Graph transformer gans with graph masked modeling for architectural layout generation. IEEE Trans. Pattern Analysis & Machine Intelligence **46**(6), 4298–4313 (2024)
30. Team, K., Bai, Y., Bao, Y., Chen, G., Chen, J., Chen, N., Chen, R., Chen, Y., Chen, Y., Chen, Y., et al.: Kimi k2: Open agentic intelligence. arXiv preprint arXiv:2507.20534 (2025)
31. Upadhyay, A., Dubey, A., Arora, V., Kuriakose, S.M., Agarawal, S.: Flnet: graph constrained floor layout generation. In: 2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW). pp. 1–6. IEEE (2022)
32. Vaswani, A.: Attention is all you need. Proc. Conf. on Neural Information Processing Systems (2017)
33. Vilnis, L., McCallum, A.: Word representations via gaussian embedding. CoRR **abs/1412.6623** (2014), `https://api.semanticscholar.org/CorpusID:13468104`
34. Wang, L., Liu, J., Zeng, Y., Cheng, G., Hu, H., Hu, J., Huang, X.: Automated building layout generation using deep learning and graph algorithms. Automation in Construction **154**, 105036 (2023)
35. Weber, R.E., Mueller, C., Reinhart, C.: Automated floorplan generation in architectural design: A review of methods and applications. Automation in Construction **140**, 104385 (2022)
36. Wu, W., Fu, X.M., Tang, R., Wang, Y., Qi, Y.H., Liu, L.: Data-driven interior plan generation for residential buildings. ACM Trans. on Graphics **38**(6) (2019)

37. Xu, M., Lou, Y., Gao, X., Zhou, X.: Floorplan-diffusion: Automatic floor plan generation via pre-trained large latent diffusion model. In: Proceedings of the 2025 International Conference on Multimedia Retrieval. pp. 1617–1625 (2025)
38. Ye, Y., He, X., Li, Z., Lin, B., Yuan, S., Yan, Z., Hou, B., Yuan, L.: Imgedit: A unified image editing dataset and benchmark. arXiv preprint arXiv:2505.20275 (2025)
39. Yin, J., Zeng, P., Sun, H., Dai, Y., Zheng, H., Zhang, M., Zhang, Y., Lu, S.: Floorplan-llama: Aligning architects' feedback and domain knowledge in architectural floor plan generation. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 6640–6662 (2025)
40. Zhang, H., Savov, A., Dillenburger, B.: Maskplan: Masked generative layout planning from partial input. In: Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition. pp. 8964–8973. IEEE (2024)
41. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition (2018)
42. Zong, Z., Zhan, Z., Tan, G.: Housellm: Llm-assisted two-phase text-to-floorplan generation. arXiv e-prints pp. arXiv–2411 (2024)