CG-Floor: Centroid-Guided Diffusion for Large-Scale Floorplan Generation

Hongjin Lian[1,†] , Jian Ma[1,†], Hongjie Chen[1], Jia Li[2], Ruizhen Hu[3,*], Yu-Kun Lai[4], Kun Li[1,*]
[1]Tianjin University [2]KEDACOM [3]Shenzhen University [4]Cardiff University
{lianhongjin, jianma, 3021244355, lik}@tju.edu.cn
lijia_jkcpx@kedacom.com ruizhen.hu@gmail.com laiy4@cardiff.ac.uk

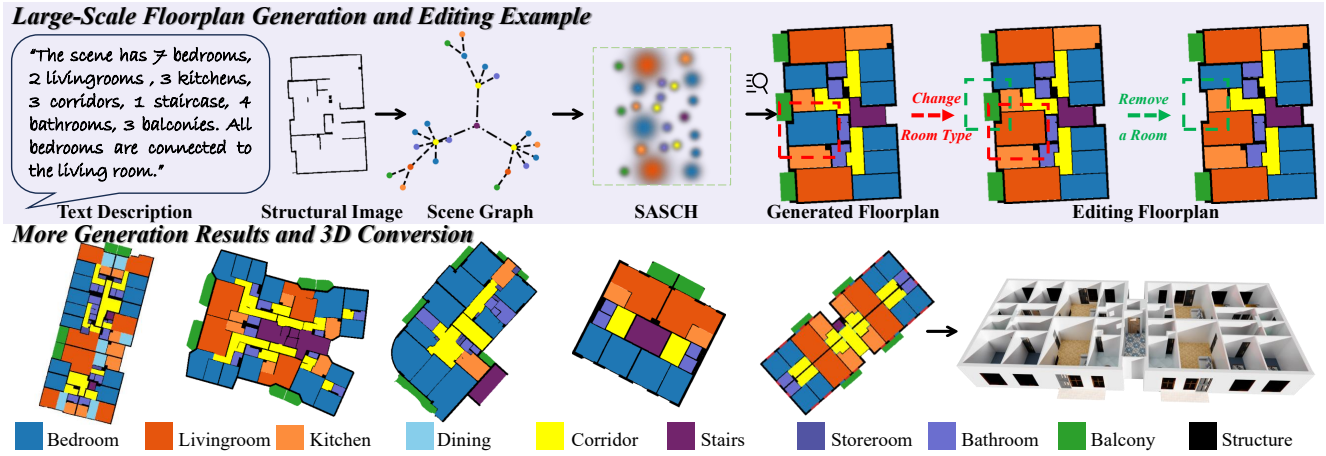

Figure 1. (Top) By inputting a text description or scene graph, our method generates a Size-Aware Semantic Centroid Heatmap to guide the generation of large-scale floorplans and enable editing. (Bottom) Our method produces large-scale floorplans and supports 3D conversion.

Abstract

Large-scale floorplan generation is critical for virtual space planning and architectural simulation. Although existing methods have shown success in generating small-scale floorplans with simple room shapes, they struggle to handle complex room connections and irregular room shapes that arise in large-scale floorplans. In this paper, we propose CG-Floor, a centroid-guided hierarchical framework that explicitly decouples room position and shape generation to address these issues. We first introduce the size-aware semantic centroid heatmap, derived from predicted room centroids and sizes, which provides a structured representation to guide the effective generation of a coarse-to-fine floorplan generator while ensuring semantic alignment. Additionally, we train a vector quantized codebook of floorplans with complex room shapes to capture the diversity of room shapes and employ a latent diffusion transformer to generate large-scale floorplans featuring non-Manhattan room shapes. CG-Floor achieves state-of-the-art performance on the large-scale MSD dataset, and supports 3D floorplan conversion and editing, demonstrating the practicality of our approach. The code is available at https://cic.tju.edu.cn/faculty/likun/projects/CG-Floor.

1. Introduction

The ability to generate controllable and semantically consistent large-scale floorplans with dozens of rooms, featuring significant topological complexity and diverse room shapes, is crucial for realistic simulation of architectural environments, such as multi-unit residences and commercial buildings. However, most existing methods are limited to small-scale floorplans [8, 11, 26] with fewer than 10 rooms, restricting their practical applicability. In this paper, we propose CG-Floor, which can generate large-scale floorplans with complex room shapes and supports 3D conversion and editing, demonstrating practicality, as shown in Figure 1.

Current floorplan generation methods can be broadly divided into rule-based optimization [16, 24] and learning-based approaches [7]. Rule-based optimization relies on manually designed constraint rules and either produces floorplans with simple room shapes or requires long runtime. Learning-based approaches [17] perform well on

† Equal contribution.
* Corresponding author.

small-scale floorplans with simple room shapes; however, when applied to large-scale scenarios, even the two state-of-the-art methods, MHD [10] and UN [9], reveal two major shortcomings. First, the generated floorplans exhibit substantial discrepancies from the input constraints in terms of topological structure (room connectivity) and semantic information (room categories). Second, they struggle to effectively model and generate complex, irregular room shapes. Specifically, MHD [10] represents each room as a polygon and directly outputs its vertex coordinates. However, as the number of rooms and their shape complexity increase, the number of corner points to be predicted grows dramatically. To ensure convergence, it is forced to constrain every room to a rectangle, resulting in severe shape mismatches and inconsistency with structural constraints at the floorplan level. In contrast, UN [9] fuses structural images and connectivity graphs into its pixel-generation pipeline, achieving stronger overall consistency with the structural images. But lacking information like global room positions, its outputs exhibit blurred room boundaries and numerous artifacts, making accurate room segmentation difficult.

In addition, floorplan generation quality is also tightly coupled with the choice of representation. Vector-based methods [8, 17] like MHD offer intuitive precision in small scenes but suffer combinatorial explosion of the number of rooms and per-room vertices in large-scale settings, making optimization more difficult. Conversely, pixel-based approaches [27] like UN naturally accommodate complex non-Manhattan geometries and real-world architectural nuances, such as load-bearing walls within rooms or merged spaces without explicit boundaries, that vector methods oversimplify by assuming all room edges are walls. This property also enhances 3D realism in downstream scene reconstruction. Therefore, pixel representation provides a more flexible and expressive foundation for large-scale floorplan generation. Nevertheless, even with pixel-based modeling, jointly modeling room connectivity and shape details remains challenging, necessitating decoupled prediction of room positions and shapes.

Topological complexity is the primary challenge in large-scale floorplan generation. As room numbers increase sharply into the dozens, connectivity evolves from simple adjacency to densely connected graphs. Previous works [6, 21] often entangle topology (room connectivity) and geometry (room shape) within a single generative process, which makes it difficult to globally capture and manage spatial relationships. Consequently, when applied to large-scale scenarios, their generated floorplans tend to exhibit semantic inconsistencies and misplaced rooms. To address this, we propose an explicit decoupling paradigm for topology and geometry, first focusing on topology modeling. We first employ a Large Language Model (LLM) to translate concise textual inputs into a scene graph, providing relational information. Furthermore, we introduce the Size-Aware Semantic Centroid Heatmap (SASCH), which integrates room categories as well as global positional and size information. We leverage a Graph-Transformer to jointly predict both centroid positions and room sizes, constructing a strong structural representation. By modeling critical information like inter-room relationships, positions, and sizes while temporarily abstracting away shape information, this approach more effectively ensures semantic alignment between input constraints (such as room size and relative position) and the generated floorplan, thereby robustly resolving the topological complexity in large-scale scenarios.

The second challenge is room shape complexity. In large-scale floorplans [23], room outlines no longer adhere to strict axis-aligned boundaries (non-Manhattan structure) and exhibit high diversity and irregularity. Existing methods [21] either generate only simple rectangles or lack dedicated modules to capture such irregular shape diversity. To tackle irregular shape modeling, we introduce a VQ-VAE-based discrete codebook to model complex room shapes. By encoding diverse room-contour patterns into a compact set of discrete latent tokens, the codebook captures rich geometric variability while maintaining structured representation. Building upon this, a vector-quantized diffusion transformer is trained to sample from the codebook, enabling the generation of large-scale floorplans with irregular, non-Manhattan room shapes. This explicit shape modeling significantly enhances the realism and diversity of the generated floorplans.

In summary, our main contributions are as follows:

- We design CG-Floor, a centroid-guided hierarchical framework that explicitly decouples topology (room connectivity) and geometry (room contours). Through a coarse-to-fine pipeline, it enables the generation of large-scale and semantically coherent floorplans.
- We introduce the Size-Aware Semantic Centroid Heatmap (SASCH), which integrates room categories as well as global positional and size information to ensure strong semantic alignment between input constraints and the generated floorplan.
- We build a discrete codebook using VQ-VAE to represent the floorplans with irregular room shapes, and utilize a vector-quantized diffusion transformer to generate large-scale floorplans with diverse room shapes.
- We show that CG-Floor not only achieves state-of-the-art performance on the MSD dataset, but is also effective across both large and small scale scenarios, and supports 3D scene conversion and editing (node swapping, addition/removal), demonstrating high practical value.

2. Related Work

Rule-Based Floorplan Generation. Some studies consider floorplan generation as a type of optimization problem, aim-

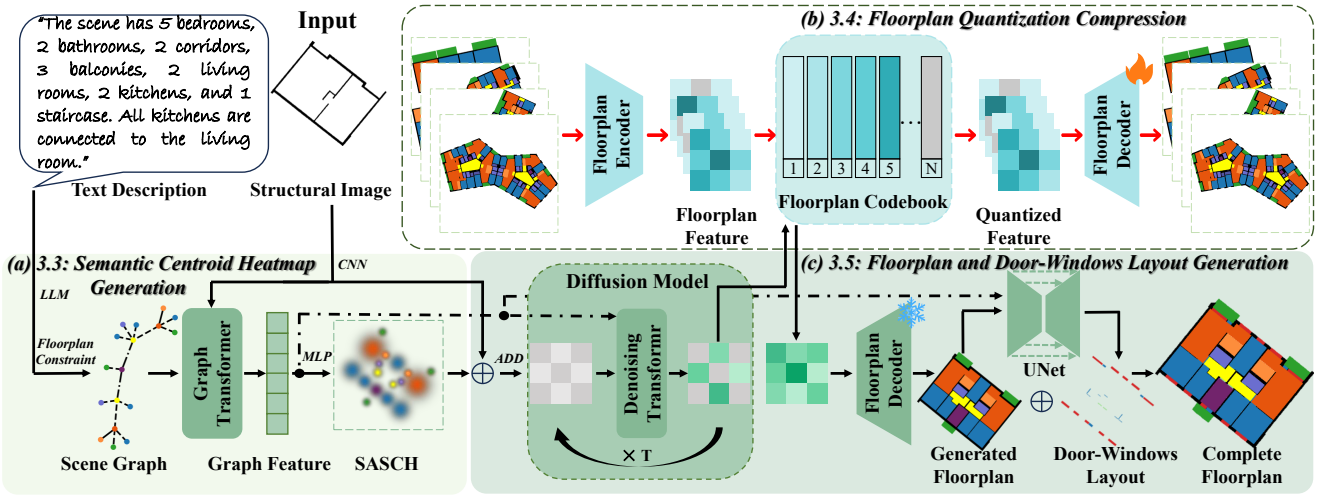

Figure 2. The detailed architecture of the proposed CG-Floor: (a) a graph transformer that integrates information from both the connectivity graph and the structural image to predict room centroid locations and sizes, which are then transformed into the SASCH; (b) a discretized codebook trained to compress large-scale floorplans with complex room shapes; and (c) a diffusion transformer in the latent space to generate floorplans, with a UNet model to generate door and window layouts.

ing to identify reasonable layout solutions through predefined constraints and rules. For example, Wu et al. [25] formalized the floorplan generation problem as a mixed integer quadratic programming (MIQP) problem and introduced a coarse-to-fine hierarchical generation framework to meet high-level constraints. Similarly, Shekhawat et al. [18] integrated graph-theoretical analysis with optimization techniques to construct dimensioned floorplans. More recently, CVTLayout [24] developed a multi-level space partitioning framework that leverages Centroidal Voronoi Tessellation (CVT) to automatically generate commercial space layouts. However, these methods exhibit a strong dependence on the configuration of constraints: insufficient constraints may result in impractical layouts, while overly restrictive constraints can lead to conflicts, yielding no feasible solutions. Furthermore, although the generated results adhere to the preset rules, they often lack flexibility and diversity, deviating from real-world scenarios.

Learning-based Floorplan Generation. Owing to their rapid inference and superior realism, data-driven deep-learning methods [3, 6] have become the prevailing paradigm for floorplan synthesis. The RPLAN [26] project not only provides a comprehensive dataset but also proposes a two-stage generation pipeline. Graph2Plan [7] converts both the graph and building envelope into floorplans that respect spatial and boundary constraints. WallPlan [19] represents floorplan as a "wall graph" labeled by room types, casting generation as a graph-generation task. In parallel, HouseGAN [12], HouseGAN++ [13], and HouseDiffusion [17] explore GAN and diffusion-model techniques for floorplan synthesis.

More recently, Tang et al. [21] introduced a novel Graph-Transformer GAN that integrates graph convolutional layers and transformer blocks to capture both local and global node interactions, thereby generating graph-constrained floorplans. In addition, GSDiff [8] adopts a wall-point prediction approach, connecting points with edges and extracting rooms from the resulting graph, demonstrating good performance on a small-scale dataset. Other recent works [15] explored the use of large language models to interpret natural language inputs and produce initial layout proposals, which are then refined into final floorplans via conditional diffusion models. Despite these advances, current data-driven methods are generally limited to scenes with fewer than 10 rooms.

Most existing methods rely on graphs [20] or wall constraints [19], with only a few [2, 11] explore text input. However, text input generally requires detailed spatial descriptions, which are impractical for complex, large-scale scenes. To address this, we introduce an optional text-to-graph pipeline: users provide only coarse parameters (room categories and counts), and an LLM infers room connectivity to generate a scene graph using statistical constraints, simplifying the input process. Additionally, our framework supports automatic conversion from floorplans to 3D scenes, facilitating richer visualization and downstream tasks.

3. Method

3.1. Overview

We introduce CG-Floor, a novel hierarchical framework explicitly designed for large-scale indoor floorplan generation.

Previous methods either produce room layouts directly [17] or sequentially (one-by-one) predict room centroids [5, 26] before generating room layouts. However, these paradigms often fail to fully capture the complex global context and inter-room relationships inherent in large-scale floorplans. Sequential prediction, in particular, is highly susceptible to cumulative errors [22] and often neglects room size information during the initial stage. In stark contrast, our approach addresses these challenges by modeling the global context in a single forward pass, decoupling the generation process into topological and geometric stages, and crucially, incorporating predicted room size information alongside centroids to create a richer, more powerful topological anchor.

Core Framework. As illustrated in the overall pipeline (Figure 2), the core innovation of our method lies in the synchronous prediction of all room centroids and their sizes, which serve as a powerful topological anchor to align the global structure with the input constraints. This step effectively decouples the topological complexity (room connectivity and relative positions) from the subsequent geometric complexity (irregular room shapes), thereby reducing the overall problem difficulty in large-scale scenarios.

Our hierarchical framework consists of three primary components (detailed in Sections 3.3–3.5):

- Semantic Centroid Heatmap Generation (Figure 2(a)): This module processes the input constraints (scene graph and structural images) to predict the topological structure (centroids and sizes of all rooms) and encodes them into a Size-Aware Semantic Centroid Heatmap (SASCH), ensuring strict alignment with the input scene graph.
- Floorplan Quantization Compression (Figure 2(b)): To handle complex, non-Manhattan room shapes efficiently, we pre-train a VQ-VAE to construct a discrete codebook, allowing for the compression and high-fidelity representation of floorplan geometry in a compact latent space.
- Floorplan and Door-Window Layout Generation (Figure 2(c)): A Vector-Quantized Diffusion Transformer generates the final floorplan image, guided by the SASCH and the discrete codebook. An auxiliary module then decouples the generation of detailed door and window layouts.

3.2. Controllable Input and 3D Extension

To ensure controllability and computational efficiency, we represent the scene as a scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes semantic room nodes and $\mathcal{E}$ encodes spatial-semantic relationships between them. This graph-based representation provides a unified interface for both *generation* and *editing*: users can flexibly adjust the graph structure to update the resulting floorplan accordingly.

While scene graphs are the native input to our framework, manually creating them is impractical for large-scale scenes. To address this, we also support an optional pipeline that uses an LLM [14] to convert text inputs that specify room categories, quantities, and optionally partial adjacency requirements into scene graphs during inference.

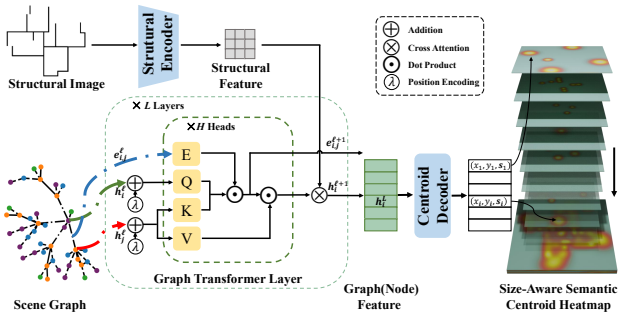

Figure 3. Size-aware semantic centroid heatmap generation. A graph transformer integrates scene-graph connectivity information and structural features to predict room centroids and sizes, which are aggregated into a SASCH.

3.3. Semantic Centroid Heatmap Generation

Since directly predicting room positions and shapes in large-scale floorplans from a scene graph is challenging and prone to structural disarray, we first focus on estimating the centroid positions and sizes of all rooms while temporarily ignoring shape details to reduce complexity. As shown in Figure 2(a) or Figure 3, we employ a graph transformer to predict these centroids and sizes, which are then converted into the SASCH to guide subsequent room shape generation.

Room Centroid and Size Prediction. As shown in the left half of Figure 3, the proposed graph transformer jointly considers the room connectivity relationships encoded in the scene graph $\mathcal{G}$ and the structural information represented by the structural image $\mathbf{I}_{struct}$.

Specifically, the first step is to extract structural features. $\mathbf{I}_{struct} \in \mathbb{R}^{1\times\mathcal{S}\times\mathcal{S}}$ is encoded by the *Structural Encoder* (a 3-layer CNN) into structural feature $\mathbf{f}_{struct} \in \mathbb{R}^{d_e\times\mathcal{S}_l\times\mathcal{S}_l}$.

These features are fed into a graph transformer consisting of L layers, where node and edge features at layer ℓ are iteratively updated to capture higher-order spatial and relational dependencies following [4]:

$$\hat{h}_i^{\ell+1} = O_h^{\ell} \Big\|_{k=1}^{H} \Big(\sum_{j\in\mathcal{N}_i} w_{ij}^{k,\ell} V^{k,\ell} h_j^{\ell} \Big), \quad (1)$$

$$\hat{e}_{ij}^{\ell+1} = O_e^{\ell} \Big\|_{k=1}^{H} \Big(\hat{w}_{ij}^{k,\ell} \Big), \text{ where,} \quad (2)$$

$$w_{ij}^{k,\ell} = \text{softmax}_j(\hat{w}_{ij}^{k,\ell}), \quad (3)$$

$$\hat{w}_{ij}^{k,\ell} = \Big(\frac{Q^{k,\ell} h_i^{\ell} \cdot K^{k,\ell} h_j^{\ell}}{\sqrt{d_k}} \Big) \cdot E^{k,\ell} e_{ij}^{\ell}, \quad (4)$$

where h_j^{ℓ} and e_{ij}^{ℓ} denote node and edge features respec-

tively, $Q^{k,\ell}$, $K^{k,\ell}$, $V^{k,\ell}$, $E^{k,\ell} \in \mathbb{R}^{d_k \times d}$, O_h^ℓ,$O_e^\ell \in \mathbb{R}^{d \times d}$ represent learnable projections, $k = 1$ to H denotes the attention heads, and $\|$ denotes concatenation. Outputs $\hat{h}_i^{\ell+1}$ and $\hat{e}_{ij}^{\ell+1}$ undergo residual connections, layer normalization, and feedforward networks to produce $\hat{\hat{h}}_i^{\ell+1}$ and $e_{ij}^{\ell+1}$.

To fuse visual and graph features, we adopt a cross attention layer to compute the dynamic correlation between $\hat{\hat{h}}_i^{\ell+1}$ and $\mathbf{f}_s$, in order to obtain $h_i^{\ell+1}$.

After L Graph Transformer layers, the final node features h_i^L encapsulate room types, connectivity, and structural information. The *Centroic Decoder* (an MLP) decodes these into centroid coordinates $\hat{\mathbf{p}}_i$ and sizes $\hat{\mathbf{s}}_i$.

Constructing the Semantic Centroid Heatmap. As shown in the right half of Figure 3, we then generate a multi-channel semantic centroid heatmap $\mathbf{H}$ combining centroid locations and sizes with room types. For $\mathcal{C}$ room types, each type k corresponds to a single-channel heatmap $\mathbf{H}_k \in \mathbb{R}^{\mathcal{S}_l \times \mathcal{S}_l}$. In each channel k, every room is represented by a Gaussian kernel centered at its centroid position $\mathbf{p}_i = (x_i, y_i)$ with an adaptive standard deviation σ_i proportional to its size s_i:

$$\mathbf{H}_k(x, y) = \sum_{i \in \mathcal{R}_k} \exp\left(-\frac{(x - x_i)^2 + (y - y_i)^2}{2\sigma_i^2}\right), \quad (5)$$

where $\mathcal{R}_k$ are the room centroids and sizes of room type k. We stack them into SASCH $\mathbf{H} \in \mathbb{R}^{\mathcal{C} \times \mathcal{S}_l \times \mathcal{S}_l}$.

3.4. Floorplan Quantization Compression

In order to model large-scale floorplans with complex room shapes, as illustrated in Figure 2(b), our framework performs compression and quantization of the input floorplan.

Given an input floorplan $F \in \{0, 1\}^{\mathcal{C} \times \mathcal{S} \times \mathcal{S}}$, where $\mathcal{C}$ denotes the number of room categories and $\mathcal{S}$ is the spatial resolution, we employ a VQ-VAE to learn a discrete latent representation of the floorplan. The encoder E maps F into a continuous latent tensor $Z = E(F) \in \mathbb{R}^{d_e \times \mathcal{S}_l \times \mathcal{S}_l}$, where d_e is the embedding dimension and $\mathcal{S}_l$ is the downsampled spatial size. We maintain a learnable codebook $\mathcal{C} = \{ c_i \in \mathbb{R}^{d_e} \}_{i=1}^{\mathcal{N}}$, of N embedding vectors. Each spatial feature vector $z_k \in \mathbb{R}^{d_e}$ in Z is quantized by selecting its nearest codebook entry:

$$c_t = \arg\min_{c_i \in \mathcal{C}} \left\| z_k - c_i \right\|_2, \quad (6)$$

Collecting all quantized vectors yields the quantized latent tensor $Z_q = \{c_t\}$. The decoder D then reconstructs the floorplan $\hat{F} = D(Z_q)$.

3.5. Floorplan and Door-Window Layout Generation

As shown in Figure 2(c), to facilitate the subsequent extraction of room geometries, we first generate a floorplan that exclusively comprises rooms and walls. Subsequently, a U-Net is employed to produce the door and window layout.

Floorplan Generation. Inspired by [1], we use denoising transformer with M layers to perform the discrete absorbing diffusion process to learn the latent vector distribution of floorplan in VQ-VAE.

To integrate structural and room positions, we first concatenate the SASCH $\mathbf{H}$ with structural features $\mathbf{f}_{struct}$, followed by a 1×1 convolution to align with the latent dimension d_e. This conditional tensor is then added to the *initial noise* $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ to form the *noisy latent* Z_T, where T denotes the maximum diffusion timestep. The denoising process learns the reverse distribution $p_\theta(Z_0|Z_t)$, which iteratively recovers the clean latent Z_0.

In order to address the shortcomings of centroid representation (such as the centroid of a C-shaped room being outside the room), we additionally incorporate scene graph information via skip connections from the graph transformer into the denoising diffusion process to better adapt to room connections and category constraints in the scene. So, during iterative denoising, the denoising transformer injects scene graph node features $\{h_i^L\}$ through cross-attention layers inserted every M_s blocks. The discrete absorbing diffusion mechanism accelerates generation through parallel token prediction.

Door-Window Layout Generation. Since the generated floorplan $\hat{F}$ lacks door and window positions, we employ a U-Net to predict the door-window layout $\mathbf{I}_{dw} \in \mathbb{R}^{\mathcal{C}_{dw} \times \mathcal{S} \times \mathcal{S}}$. During U-Net encoding, when features reach the smallest resolution, we integrate graph node features h_i^L via cross-attention.

3.6. Loss Function

This framework is trained in two stages. The first stage is to train floorplan quantization compression, and the second stage is to train floorplan generation.

Quantitative Compression Loss Function. To train the VQ-VAE to both reconstruct accurate floorplans and learn a compact codebook, we use a combination of reconstruction loss and quantization loss.

The reconstruction loss $\mathcal{L}_{\text{rec}}$ is formulated as a cross-entropy loss, the quantization loss can be written as:

$$\mathcal{L}_{\text{VQ}} = \left\| \text{sg}[Z] - Z_q \right\|_2 + \beta \left\| Z - \text{sg}[Z_q] \right\|_2, \quad (7)$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator. The first term in $\mathcal{L}_{\text{VQ}}$ encourages codebook vectors to be closer to the encoder outputs, while the second term encourages the encoder outputs to commit to their assigned codes. The full objective of quantitative compression is

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{VQ}}. \quad (8)$$

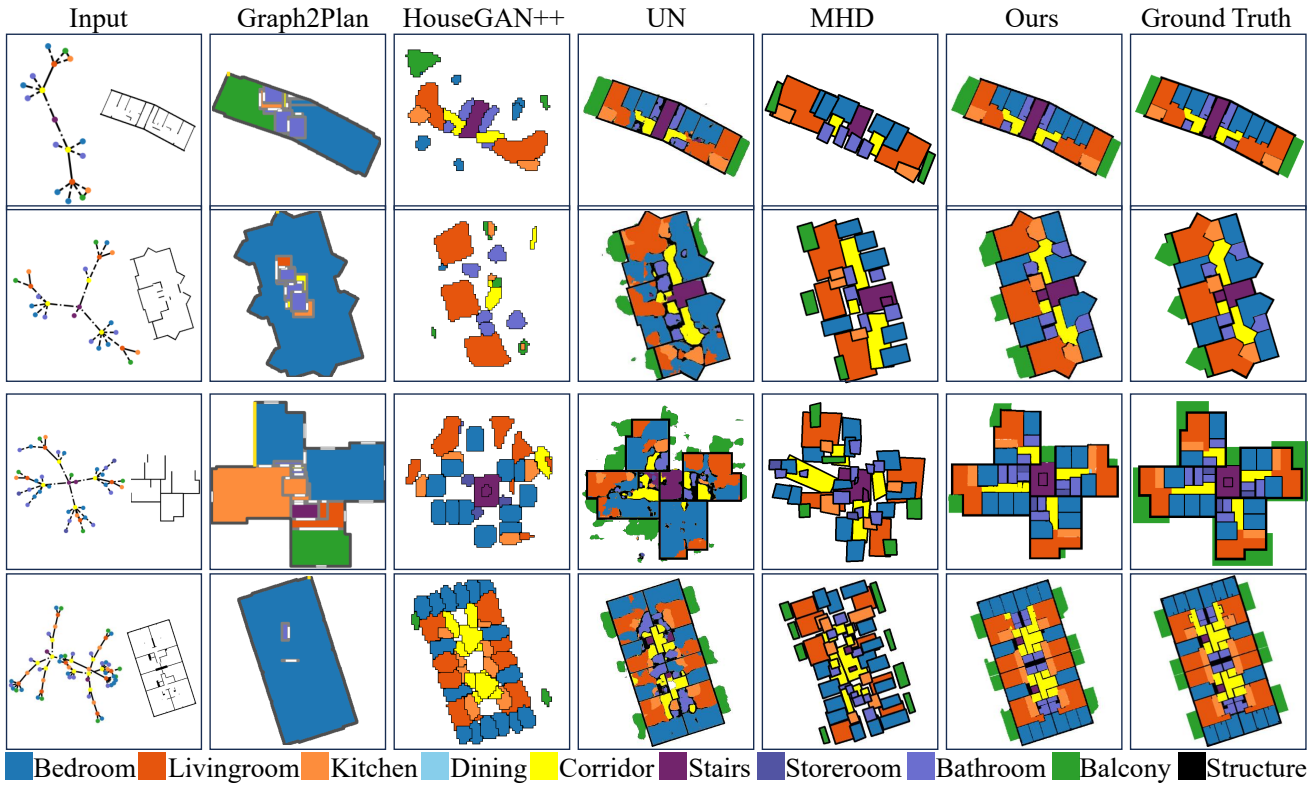

Figure 4. Qualitative comparison of graph-conditioned floorplan generation with representative methods [7, 13, 17, 23]. Our method better preserves room-level structural details and global connectivity, producing floorplans that more closely resemble the ground truth.

Floorplan Generation Loss Function. The total loss of floorplan generation combines the centroid–size prediction, diffusion, and door-window layout losses:

$$\mathcal{L}_{\text{floorplan}} = \lambda_1 \mathcal{L}_{\text{geo}} + \lambda_2 \mathcal{L}_{\text{diffusion}} + \lambda_3 \mathcal{L}_{\text{dw}} \tag{9}$$

where $\mathcal{L}_{\text{geo}}$ denotes the L2 loss on both room centroid positions and sizes, $\mathcal{L}_{\text{dw}}$ denotes the cross-entropy loss for the door-window floorplan, and $\mathcal{L}_{\text{diffusion}}$ uses a reweighted ELBO framework [1] that prioritizes early denoising steps by downweighting later timesteps.

3.7. Application: 3D Scene Generation

We further design a general pipeline to transform the generated floorplan into a 3D scene, enhancing the realism and usability of the output. This pipeline extracts structural elements, constructs 3D geometry, and places semantically appropriate objects guided by a large language model. Detailed implementation is provided in the supplementary materials.

4. Experiments and Results

4.1. Experiment Setup

Implementation details. The experiments were implemented on 4 NVIDIA RTX 4090 GPUs. We first trained a VQ-VAE for floorplan quantization and compression for 1000 epochs, then froze the VAE encoder and decoder to train the floorplan generation model for another 1000 epochs. The floorplan dimension was set to $S = 1024$, with an embedding dimension $d_e = 64$, codebook size $N = 1024$, and loss weights β, λ_1, λ_2, λ_3 are set to 0.25, 1, 1, 1. The diffusion process employed $T = 1000$ timesteps. All implementations used PyTorch with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$), where the VAE learning rate was 4.5×10^{-6} and the generation model learning rate was 2×10^{-4}. The 3D scene generation utilized ChatGPT [14] as the LLM for generating object positions.

Dataset. The proposed method was trained on the MSD [23] dataset, which is currently the only large-scale floorplan dataset containing a significant share of multi-apartment dwellings and contains over 5.3K annotated floorplans of medium-to-large building complexes, following its training and testing set partitioning, and synchronously applies multiple data augmentation strategies to the scene graphs, structure images, and floorplans to enhance the model's generalization capability. These strategies include random rotation, uniform scaling, mirror flipping, and random change of certain node types.

Baselines. Given the current absence of data-driven floorplan generation methods for large-scale indoor envi-

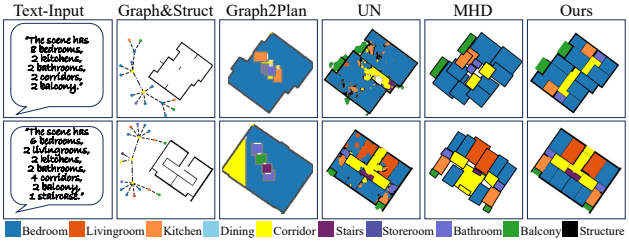

Figure 5. Qualitative comparison of text-conditioned floorplan generation. All methods are compared using the scene graph auto-generated from text by an LLM.

Table 1. Quantitative comparison of graph-conditioned floorplan generation. The best results are in bold. The Consistency scores are missing for some baselines, as graphs cannot be extracted from their output images.

Method	FID(↓)	KID(↓)	Shape-Sim(↑)	Consistency(↑)
Graph2Plan	279.86	277.49	0.56	-
HouseGAN++	160.59	125.70	0.68	-
UN	179.22	180.67	0.41	-
MHD	79.72	63.85	0.65	87.1
Ours	**16.03**	**6.80**	**0.71**	**91.3**

ronments, we establish comprehensive benchmark comparisons against the official baselines introduced in MSD and the most advanced state-of-the-art approaches: Graph-informed U-Net (UN) [9], Modified HouseDiffusion (MHD) [10, 17], HouseGAN++ [13], and Graph2Plan [7]. To ensure fairness in comparison, all models are provided with a unified dual-input setting (scene graph + structure image), and all models were retrained on the MSD dataset. Specific implementation details and more results can be found in the supplementary material.

Evaluation metrics. We designed evaluation metrics targeting three core objectives of large-scale floorplan generation. 1) Visual Quality: We employ Fréchet Inception Distance (FID) and Kernel Inception Distance (KID) to assess the overall visual fidelity of the generated results. 2) Room Shape Similarity: We calculate the similarity between generated and real floorplans for rooms of the same type using Hu Moments, in order to evaluate the plausibility and geometric complexity of room shapes. 3) Topological Consistency: We use the graph compatibility metric from MSD [23] to measure the topological matching between the predicted floorplan and ground truth. The specific calculation method can be found in the supplementary material.

4.2. Qualitative Evaluation

As shown in Figure 4, we qualitatively compare floorplan generation models conditioned on graph inputs. While HouseGAN++ [13] and Graph2Plan [7] perform well on small-scale indoor floorplans, they struggle with large-scale designs, often producing incorrect room counts and weak

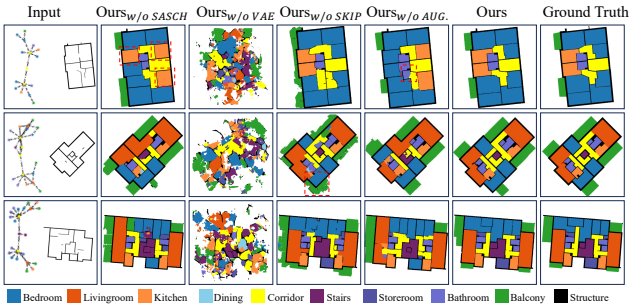

Figure 6. Qualitative comparison of ablation study. Each ablation setting impairs performance to varying extents, confirming the indispensable contributions of each module and augmentation strategy in our approach.

Table 2. Quantitative results of ablation study.

Method	FID(↓)	KID(↓)	Shape-Sim(↑)	Consistency(↑)
$w/o\ SCH$	21.62	8.61	0.67	-
$w/o\ VAE$	270.49	349.00	0.64	-
$w/o\ SKIP$	66.92	38.31	0.65	89.7
$w/o\ AUG$	25.95	15.65	0.68	90.1
Ours	**16.03**	**6.80**	**0.71**	**91.3**

spatial connectivity, leading to invalid or fragmented structures, highlighting fundamental gaps in maintaining global structural coherence during upscaling to complex architectural configurations.

UN [9] exhibits structural issues such as indistinct room demarcation and failure to maintain discrete room segmentation. In contrast, MHD [10, 17] offers better spatial definition but is fundamentally constrained by rigid rectangular room assumptions. These constraints limit topological diversity and often lead to fragmented, irrational layouts at scale. In contrast, our proposed method addresses these critical limitations through SASCH representation and a pre-trained discrete codebook, enabling reproduction of both local structural details (e.g., precise room demarcation and shape diversity) and global spatial connectivity. Compared to existing methods, our method achieves high fidelity to the ground truths.

As shown in Figure 5, our method also performs well under text-conditioned settings. Utilizing the scene graph generated by the LLM, our approach produces results that are valid both semantically and structurally.

Moreover, as shown in Figure 7, our method successfully generates reasonable floorplans from two distinct custom text descriptions applied to the same structural image, demonstrating its robustness in in-the-wild settings; minor local artifacts can be easily filtered, and the resulting 3D scenes remain semantically and structurally valid. Additional qualitative results presented in the supplementary material further validate the generalizability and effectiveness of our framework in practical applications.

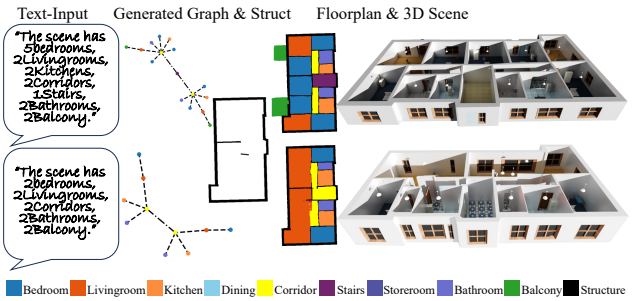

Figure 7. Test results for in-the-wild data. With a fixed structural image, different text inputs generate reasonable floorplans and corresponding 3D scenes.

4.3. Quantitative Evaluation

As shown in Table 1, our method outperforms existing approaches in multiple metrics for large-scale indoor floorplans generation. In terms of visual quality, our approach achieves FID=16.03 and KID=6.80, representing 79.8% and 89.3% reductions compared to the second-best method, MHD. This demonstrates the effectiveness of our SASCH representation in maintaining global structural coherence, addressing the fragmentation issues highlighted in existing methods. While HouseGAN++ and Graph2Plan perform reasonably well on small-scale floorplans, their results degrade significantly in more complex settings. The Shape-Similarity metric (0.71 vs. MHD's 0.65) indicates our method better handles diverse geometries, overcoming the rectangular geometry constraints. This aligns with the qualitative observations, shown in Figure 4, regarding UN's structural irregularities (Shape-Sim=0.41) and Graph2Plan's topological limitations. Our higher Consistency score (91.3 vs. MHD's 87.1) further confirms the model's ability to maintain spatial connectivity. Overall, these results validate our hierarchical design in balancing local accuracy and global structure.

4.4. Ablation Study

To thoroughly investigate the contribution of each module to the final performance, we design the following ablation settings: 1) w/o SASCH: Remove SASCH, generating the floorplan directly from the graph. 2) w/o VAE: Remove the VQ-VAE module, allowing the Diffusion model to directly generate complex room shapes. 3) w/o SKIP: Remove the use of skip connections (dashed lines in the green part of Figure 2) for injecting graph features extracted by the Graph Transformer into the Diffusion and UNet models. 4) w/o AUG: Remove all data augmentation.

Figure 6 and Table 2 present qualitative and quantitative results for four ablation settings. The full model performs best across all metrics. Although the w/o SASCH setting attains the second-best results on FID and KID, it still has notable errors in the number and types of rooms, as shown in the second column of Figure 6, confirming that removing the SASCH leads to semantic inaccuracies. The removal of VQ-VAE (w/o VAE) shows that most scenarios are incapable of generating plausible floorplans. This demonstrates the VAE's critical role in compressing and representing complex room shapes. Removing skip connections (w/o SKIP) or data augmentation (w/o AUG) degrades detail fidelity, resulting in pixel artifacts and unreasonable room shapes (Figure 6, columns 4–5), emphasizing their importance for detail refinement and generalization. Overall, each ablation variant degrades model performance to varying degrees, validating the essential role of each module and augmentation strategy in our approach.

Table 3. User study: proportion of preference of different methods.

Method	GT-Sim	Consistency	Real-Sim
Graph2Plan (A)	0.91%	2.05%	1.36%
HouseGAN++ (B)	0.68%	0.45%	0.68%
UN (C)	1.82%	0.23%	0.91%
MHD (D)	1.59%	4.09%	4.55%
Ours (E)	**95.00%**	**93.18%**	**92.50%**

4.5. User Study

To evaluate our method against other approaches, we additionally conducted a user study focusing on three key aspects of user preference: (1) Ground Truth Similarity (GT-Sim): When ground truth (GT) floorplans are available, which method produces results most similar to GT; (2) Consistency: When GT is not available, which method generates results that best align with the input; (3) Real-World Similarity (Real-Sim): When GT is not available, which method produces results that most closely resemble realistic floorplans. For each metric, we presented users with 5 comparison examples. We collected responses from 88 participants. The results, shown in Table 3, indicate that our method was the most preferred across all metrics, receiving over 90% preference in each case. More details are shown in the supplementary material.

5. Conclusion

This paper presents CG-Floor, a centroid-guided hierarchical framework for large-scale floorplan generation. First, we propose a SASCH representation to impose explicit topological priors and ensure room adjacencies and categories conform to the input constraints. Second, we introduce a discrete codebook of complex room shapes and employ a vector-quantized diffusion transformer to synthesize large-scale floorplans with rich geometric diversity. On the MSD dataset, CG-Floor achieves state-of-the-art performance, and its outputs support editing and seamless 3D conversion, demonstrating strong practical applicability.

Acknowledgements

This work was supported in part by National Key R&D Program of China (2023YFC3082100), National Natural Science Foundation of China (62501416), Science Fund for Distinguished Young Scholars of Tianjin (No. 22JCJQJC00040), Natural Science Foundation of Tianjin (24JCYBJC01300), and the Engineering and Physical Sciences Research Council (No. EP/Y028805/1). For the purpose of open access, the authors have applied a creative commons attribution (CC BY) licence to any author accepted manuscript version arising.

References

[1] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In *European Conference on Computer Vision*, pages 170–188. Springer, 2022. 5, 6

[2] Qi Chen, Qi Wu, Rui Tang, Yuhan Wang, Shuai Wang, and Mingkui Tan. Intelligent home 3D: Automatic 3D-house design from linguistic descriptions only. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 12625–12634, 2020. 3

[3] Mohammed Haroon Dupty, Yanfei Dong, Sicong Leng, Guoji Fu, Yong Liang Goh, Wei Lu, and Wee Sun Lee. Constrained layout generation with factor graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12851–12860, 2024. 3

[4] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021. 4

[5] Feixiang He, Yanlong Huang, and He Wang. iPLAN: Interactive and procedural layout planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7793–7802, 2022. 4

[6] Shibo Hong, Xuhong Zhang, Tianyu Du, Sheng Cheng, Xun Wang, and Jianwei Yin. 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*, pages 3248–3256, 2024. 2, 3

[7] Ruizhen Hu, Zeyu Huang, Yuhan Tang, Oliver Van Kaick, Hao Zhang, and Hui Huang. Graph2Plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics (TOG)*, 39(4):118:1–118:14, 2020. 1, 3, 6, 7

[8] Sizhe Hu, Wenming Wu, Yuntao Wang, Benzhu Xu, and Liping Zheng. GSDiff: Synthesizing vector floorplans via geometry-enhanced structural graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1–10, 2025. 1, 2, 3

[9] Yuntae Jeon, Dai Quoc Tran, and Seunghee Park. Skip-connected neural networks with layout graphs for floor plan auto-generation. *arXiv preprint arXiv:2309.13881*, 2023. 2, 7

[10] Emanuel Kuhn. Adapting HouseDiffusion for conditional floor plan generation on modified Swiss dwellings dataset. *arXiv preprint arXiv:2312.03938*, 2023. 2, 7

[11] Sicong Leng, Yang Zhou, Mohammed Haroon Dupty, Wee Sun Lee, Sam Joyce, and Wei Lu. Tell2Design: A dataset for language-guided floor plan generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 14680–14697, 2023. 1, 3

[12] Nelson Nauata, Kai-Hung Chang, Chin-Yi Cheng, Greg Mori, and Yasutaka Furukawa. House-GAN: Relational generative adversarial networks for graph-constrained house layout generation. In *European Conference on Computer Vision*, pages 162–177. Springer, 2020. 3

[13] Nelson Nauata, Sepidehsadat Hosseini, Kai-Hung Chang, Hang Chu, Chin-Yi Cheng, and Yasutaka Furukawa. House-GAN++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13632–13641, 2021. 3, 6, 7

[14] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022. 4, 6

[15] Sizhong Qin, Chengyu He, Qiaoyun Chen, Sen Yang, Wenjie Liao, Yi Gu, and Xinzheng Lu. Chathousediffusion: Prompt-guided generation and editing of floor plans. *arXiv preprint arXiv:2410.11908*, 2024. 3

[16] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21783–21794, 2024. 1

[17] Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. HouseDiffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5466–5475, 2023. 1, 2, 3, 4, 6, 7

[18] Krishnendra Shekhawat, Nitant Upasani, Sumit Bisht, and Rahil N Jain. A tool for computer-generated dimensioned floorplans based on given adjacencies. *Automation in Construction*, 127:103718, 2021. 3

[19] Jiahui Sun, Wenming Wu, Ligang Liu, Wenjie Min, Gaofeng Zhang, and Liping Zheng. Wallplan: synthesizing floorplans by learning to generate wall graphs. *ACM Transactions on Graphics (TOG)*, 41(4):1–14, 2022. 3

[20] Hao Tang, Zhenyu Zhang, Humphrey Shi, Bo Li, Ling Shao, Nicu Sebe, Radu Timofte, and Luc Van Gool. Graph transformer gans for graph-constrained house generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2173–2182, 2023. 3

[21] Hao Tang, Ling Shao, Nicu Sebe, and Luc Van Gool. Graph transformer gans with graph masked modeling for architectural layout generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4298–4313, 2024. 2, 3

[22] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[23] Casper Van Engelenburg, Fatemeh Mostafavi, Emanuel Kuhn, Yuntae Jeon, Michael Franzen, Matthias Standfest, Jan van Gemert, and Seyran Khademi. MSD: A benchmark dataset for floor plan generation of building complexes. In *European Conference on Computer Vision*, pages 60–75. Springer, 2024. 2, 6, 7

[24] Yuntao Wang, Wenming Wu, Yue Fei, and Liping Zheng. CVTLayout: Automated generation of mid-scale commercial space layout via centroidal voronoi tessellation. *Computers & Graphics*, 127:104175, 2025. 1, 3

[25] Wenming Wu, Lubin Fan, Ligang Liu, and Peter Wonka. MIQP-based layout design for building interiors. *Computer Graphics Forum*, 37(2):511–521, 2018. 3

[26] Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019. 1, 3, 4

[27] Haolan Zhang and Ruichuan Zhang. Generating accessible multi-occupancy floor plans with fine-grained control using a diffusion model. *Automation in Construction*, 177:106332, 2025. 2