MSD: A Benchmark Dataset for Floor Plan Generation of Building Complexes

Casper van Engelenburg[1(✉)], Fatemeh Mostafavi[1], Emanuel Kuhn[1], Yuntae Jeon[2], Michael Franzen[3], Matthias Standfest[4], Jan van Gemert[1], and Seyran Khademi[1]

[1] Delft University of Technology, Delft, The Netherlands
c.c.j.vanengelenburg@tudelft.nl
[2] Sungkyunkwan University, Seoul, Korea
[3] Kaiserslautern, Germany
[4] Gersau, Switzerland

Abstract. Diverse and realistic floor plan data are essential for the development of useful computer-aided methods in architectural design. Today's large-scale floor plan datasets predominantly feature simple floor plan layouts, typically representing single-apartment dwellings only. To compensate for the mismatch between current datasets and the real world, we develop **Modified Swiss Dwellings** (MSD) – the first large-scale floor plan dataset that contains a significant share of layouts of multi-apartment dwellings. MSD features over 5.3K floor plans of medium- to large-scale building complexes, covering over 18.9K distinct apartments. We validate that existing approaches for floor plan generation, while effective in simpler scenarios, cannot yet seamlessly address the challenges posed by MSD. Our benchmark calls for new research in floor plan machine understanding. Code and data are open.

Keywords: Benchmark Dataset · Floor Plan Generation · Diffusion Models

1 Introduction

A floor plan is a 2D horizontal projection of a building's floor, effectively conveying the layout of its inherent spatial components, such as areas, doors, and walls. Developing floor plans is a primary task in architectural design, and is a time-consuming and expensive operation – it is an informal optimization of multi-variable space functionality, concerning various constraints (*e.g.*, environmental context, regulations, budget).

Recent advancements in deep learning and the accessibility of large-scale floor plan datasets [6,30], led to a large number of techniques for automatically

M. Franzen and M. Standfest—Independent Researcher.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-73636-0_4.

A. Leonardis et al. (Eds.): ECCV 2024, LNCS 15116, pp. 60–75, 2025.
https://doi.org/10.1007/978-3-031-73636-0_4

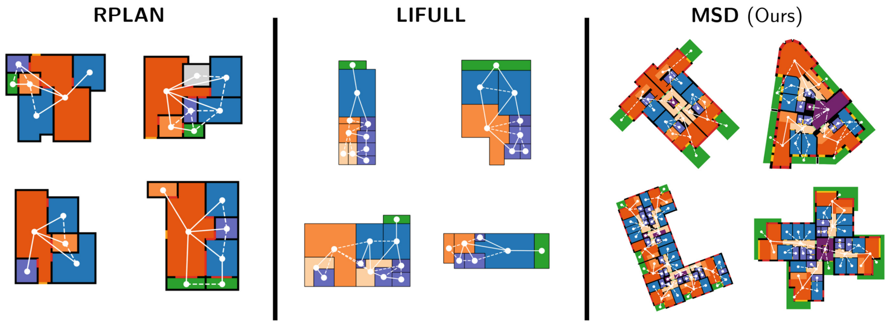

Fig. 1. MSD compared to RPLAN [30] and LIFULL [11]. Rooms are colored on function (*e.g.*, blue for "bedroom"). The functional diagrams (represented as graphs) are drawn on top of the floor plans. MSD (right) significantly differs from RPLAN (left) and LIFULL (middle), as it contains more complex and realistic floor plans.

generating floor plans [16,21,27,30]. The main focus of the current works has been on simple floor plans, mostly of small-scale and single-apartment dwellings. However, the majority of real-world dwellings are more complex, especially those that consist of multiple apartments.

Floor plans of multi-apartment building complexes are very different from single-apartment floor plans. Not only are there an order of magnitude more areas that need to be arranged, but the connectivity *between* apartments plays an essential role as well. Moreover, there are structural constraints on the floor plan design (*e.g.*, staircases, load-bearing walls) that need to remain intact while arranging the space.

To train and evaluate realistic models, we curate a new floor plan dataset, called Modified Swiss Dwellings (MSD), that consists of a large number of complex floor plans. MSD includes precise area annotations, graph attributes, and the essential structural components of the building. Following [5], we define the floor plan generation task as one that is constrained on the functional diagram (represented as a graph) and the necessary structure of the building (represented as a binary image). To benchmark the complexity of MSD, we develop two baseline methods. The first method modifies *HouseDiffusion* [21] by including a wall cross attention module and by integrating it with a graph attention network (GAT) [29]. The second is a segmentation-based approach, integrating a U-Net [19] and a graph convolutional network (GCN) [8].

Our results reveal a significant drop in performance when the two methods are trained and tested on MSD compared to the performance on simpler floor plan data. The increased complexity that MSD brings underscores the need to reassess the current methods for floor plan generation. Our contributions are summarized as follows:

- We develop MSD – a benchmark dataset of floor plans of building complexes. MSD contains 5,372 annotated floor plan images of medium- to large-scale

single- to multi-apartment building complexes, including precise geometrical and topological attributes.
- We benchmark two state-of-the-art frameworks for floor plan generation to validate the complexity of floor plan generation of building complexes on MSD.
- Our evaluations of generated floor plans for the two baseline methods reveal that the floor plan generation of building complexes is a very challenging task and invites researchers to rethink current methods.

2 Related Work

Floor Plan Datasets. Floor plan datasets are used for retrieval [22], reconstruction and structural reasoning [4,6,10,12,25], architectural symbol spotting and wall detection [1,3,26], and floor plan generation [11,30]. For further comparison, we only consider publicly available datasets used in floor plan generation: RPLAN [30] and LIFULL [11] (the part that is publicly available). RPLAN and LIFULL contain 80K+ and 177K+ floor plans of, resp., Japanese and Asian houses. While large in scale, RPLAN and LIFULL have a significant number of shortcomings. First, both datasets only cover single-apartment dwellings with a limited number of areas. Second, floor plans in RPLAN and LIFULL are axis-aligned and entirely Manhattan-shaped layouts, which is at odds with realistic dwellings, which typically contain a significant number of more irregularly shaped rooms. Third, RPLAN and LIFULL do not provide compass orientation, while the direction of the sun is a critical feature in environmental design [14]. In our work, we gather and develop MSD – a big collection of floor plans that addresses the above-mentioned limitations. Specifically, MSD comprises single- *and* multi-apartment dwellings of which a significant part contains irregularly shaped areas as well as building boundaries.

Automated Floor Plan Generation. The goal of floor plan generation is to automatically orchestrate the elements intrinsic to the floor plan (*e.g.* rooms, doors, walls) into a reasonable composition. Rule-based [15,18,31] and learning-based [16,21,27,30] methods exists to do so. We categorize the literature into three distinctive approaches. First, boundary-constrained floor plan generation methods [18,24,30], constrain the generative process on the external walls that separate the interior of the building from the outside. Second, graph-constrained floor plan generation methods [13,16,17,27,31] allow for fine-grained control of the floor plan by constraining the generative process on the functional diagram, which is usually represented as a graph. Instigated by [16], especially graph-constrained floor plan generation of residential houses has led to a broad range of domain-specific network architectures and optimization frameworks: Conv-MPN [32] was introduced to better capture topological and shape-wise features simultaneously; generative adversarial networks (GANs) over graphs to enable graph-structured generation [16,17]; discrete diffusion models [21] to accommodate the denoising of geometrical shapes; transformer GANs [27] to capture both local and global relations across nodes in the graph; etc. Third, boundary- and graph-constrained floor plan generation methods [5,28] allow control over the boundary as well as the graph, a setting that is closest to most real-world design conditions. Besides the graph and boundary, we constrain the generative process on the other necessary structural components (*e.g.*, load-bearing walls).

3 Dataset of Modified Swiss Dwellings (MSD)

MSD is derived from the Swiss Dwellings dataset (SD) [23]. We carefully cleaned and curated SD by taking the following steps:

- **Feature removal.** All non-floorplan geometries are removed (*e.g.* "bathtub", "stairs"; see the full list in the suppl. mat.).
- **Residential-only filtering.** We remove floor plans that include non-residential-like geometric details (*e.g.* areas categorized as "waiting room", "dedicated medical room"; see the full list in the suppl. mat.). This led to the removal of 2,305 (16.6%) floor plans.
- **Near-duplicate removal.** Many floor plans that come from the same building stem from the same plan ID [23] (see suppl. mat. for details on ID nesting in SD). Floor plans with the same plan ID are based on the same layout, indicating that the spatial arrangements are nearly identical. Hence, we sample only one-floor plan per plan ID to drastically reduce the amount of near-duplicates. Specifically, we sample the floor plan with the lowest elevation. This led to the removal of 4,395 (31.6%) floor plans.
- **Medium- to large-scale filtering.** Floor plans are removed that contain fewer than 15 areas. In addition, every floor plan should have at least two "Zone 2"-categorized areas. This led to the removal of 1,541 (11.1%) floor plans.

Additional steps for cleaning and filtering are provided in the suppl. mat., leading to the removal of an extra 388 (2.8%) floor plans. Ultimately, the amount of floor plans in MSD is 5,372.

Categorization and Labeling. We refer to an area as any well-defined part in a floor plan that a person could walk in or through (*e.g.*, bedroom, corridor, balcony.) Each area has three attributes: 1) the shape of the room (represented as polygon), 2) a room type category, and 3) a zoning type category. The zoning types (or zones) are based on the categorization made in [7]: "zone1" refers to a private space, "zone2" to a public space, "zone3" to a service space, and "zone4" to an outside place.

Image Extraction. The floor plan images are made by 'drawing' the floor plan's corresponding geometries on a single-channel image canvas. The room category labels are represented as integers (*e.g.* 0 for "living room"). The coordinates x (east-to-west) and y (south-to-north) are defined in meters. The floor plans are mostly centered around $(x, y) = (0, 0)$. To retain the information of the original coordinates within the image representation, two extra channels are added on top of the image (see suppl. mat. for details).

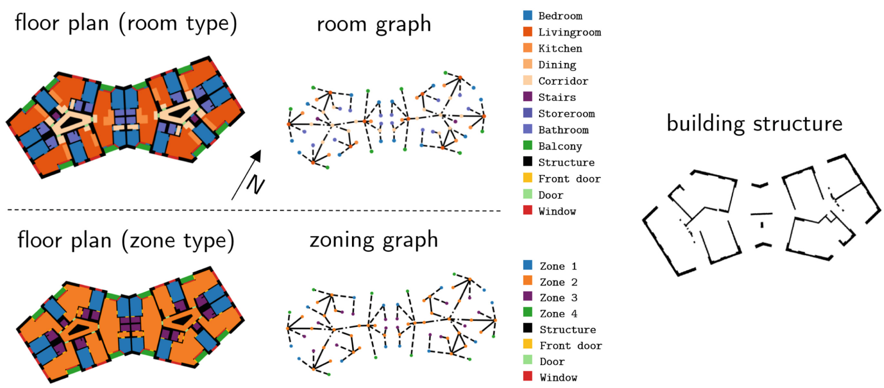

Fig. 2. Representation of floor plan data. MSD contains three types of data. (**Left**) The floor plan is the complete layout of the floor of the building, including the position and shape of the rooms, doors, and windows. The areas are labeled by color (see legends). There are two category systems: 1) category is based on the type of the area ('room type'), in which each area has a room type (*e.g.*, "bedroom"); and 2) category is based on the zone of the area ('zone type'), in which each areas has a zone type (*e.g.*, "zone1"). (**Middle**) The associated room and zoning graphs are depicted on the right of the floor plans. The node colors are equivalent to the colors of the floor plan. The position of each node is taken as the centroid of the area that the node represents. (**Right**) A binary image of the necessary structural components of the floor plan.

Graph Extraction. An (access) graph is an attribute of a floor plan and explicitly models the connections (edges) between the areas (nodes). We use an algorithmic approach to extract the graphs from the room shapes. The procedure is as follows:

- **Edge types.** We define three types of edge connectivity: 1) "passage" when one can walk from one area to the other without a door in between; 2) "door" if two areas are connected by a door; and 3) "front door" if two areas are connected by a front door.
- **Edge development.** We iterate over all possible area pairs and create an edge between the two areas if 1) either the polygons that define the shapes of the areas are close enough (≤ 0.04 m) – in this case the edge type is "passage" – or 2) there is a door for which the polygon that defines the door's shape is close enough to both area shapes (≤ 0.05 m) – in this case the edge type is either "door" or "front door" depending on the type of door.
- **Node development.** We include all necessary geometric and semantic information as node attributes. "centroid": is the center of the area. "geometry": is an array of the 2D coordinates representing the shape of the room as a polygon. "roomtype": is an integer representing the room category of the area. "zonetype": an integer representing the zoning category of the area.

- **Room and zoning graph.** We define the room graph as the graph including only "roomtype" node attributes, and the zoning graph as the graph including only "zonetype" node attributes. Figure 2 depicts both room and zoning graphs.

Structure Extraction. Structural elements of the building are often regarded as fixed beforehand. The structural elements in floor plans include load-bearing walls and columns. Accordingly, the structural elements of the floor plans are extracted and regarded as a part of the input to frame a plausible design problem. The criteria to distinguish the structural walls from the regular separating walls are based on the base wall thickness. The base wall thickness for each floor plan is the 60% quantile of the full set of existing walls' thickness given a floor plan. Any wall with a thickness larger than the base thickness value is then regarded as a load-bearing wall. Similar to load-bearing walls, columns are regarded as fixed as well. Hence, all geometric details categorized as "columns" are appended to the building structure.

3.1 Comparison to Other Datasets

We compare MSD to RPLAN [30] and LIFULL [11]. Table 1 accompanies the findings that are given next.

Origin. RPLAN and LIFULL contain floor plans that originate from, resp., the Asian and Japanese houses. MSD, on the other hand, is the first large-scale and detailed floor plan dataset originating from Europe, specifically Switzerland. While investigating the differences between Asian and European floor layouts is in itself an interesting endeavor (and goes beyond this paper), with region-specific floor plan datasets, machine learning algorithms can assist in designing buildings that cater to specific cultural preferences or comply with local building codes. We are actively extending the current dataset, to dwellings from other regions in Europe as well.

Realism. The vectorized floor layouts in LIFULL are extracted from the original image dataset [11] using the vectorization algorithm proposed in [10]. This vectorization algorithm is not necessarily error-proof and reaches on average an accuracy ($1 - \mathrm{IoU}$) of 88.5 and 94.7 for predicting the room shapes and wall junctions, resp. (Table 1 in [10]). Furthermore, floor plans in RPLAN and LIFULL are re-oriented to make them axis-aligned, arguably for easy processing and use. MSD, on the contrary, retains the orientation of the floor plans – a feature of significant importance to the quality of the floor plan design.

Complexity. RPLAN and LIFULL comprise floor plans of isolated apartments; therefore, the data do not contain information about the relations between the distinct apartments. MSD is the first large-scale floor plan dataset of multi-apartment dwellings; hence, the connections between the distinct apartments are explicitly modeled (Table 1, column 5). Moreover, rooms in RPLAN and LIFULL are Manhattan-shaped. In the real world, rooms are often more diverse

Table 1. MSD compared to RPLAN [30] and LIFULL [11]. **Complexity** is measured by the average number of corners per room (c_3), rooms per unit (c_4), and units per floor (c_5). Additionally, c_6 indicates a *significant* share (>5%) of NM-shaped rooms. **Information** is measured per the existence of room labels (room type and zoning type in c_7 and c_8, resp.) and of doors (c_9). **Size**: c_{10} and c_{11} provide the *total* number of rooms and units in the datasets, resp. Note that the *total* number reflects the reduced dataset sizes when near-duplicates are removed. **Diversity** is measured as the entropy, H_g (see Eq. (1)), over the distribution of graphs (c_{12}). MSD sets a new standard as a more complex and realistic floor plan dataset. Note that c_i is column i.

Dataset	Origin	Complexity				Information			Size		Diversity
		$\frac{\text{corners}}{\text{room}}$	$\frac{\text{rooms}}{\text{unit}}$	$\frac{\text{units}}{\text{floor}}$	NM	function	zoning	doors	# rooms	# units	H_g
LIFULL	Asia	4.54	8.15	1.00	✗	✓	✗	✗	**489.4.3K**	**61.3K**	7.79
RPLAN	Asia	5.04	6.67	1.00	✗	✓	✗	✓	161.8K	24.2K	4.56
MSD (ours)	Europe	**8.68**	**8.75**	**3.52**	✓	✓	✓	✓	163.5K	18.5K	**8.02**

in shape (*i.e.*, non-Manhattan (NM)). MSD retains the imposed shapes of the rooms, even if the shapes are NM (Table 1, column 6). Additionally, rooms in MSD consist of more corners (Table 1, column 3).

Information. Where RPLAN and LIFULL comprise floor plans in either image (RPLAN) or vectorized (LIFULL) format, MSD explicitly represents the floor plans in an image, vectorized, *and* graph formats. On top of the full floor plan layouts, MSD contains the corresponding structural necessary components, represented as binary images. In addition to room type labels, MSD provides the zoning category of each room as well. Floor plans in LIFULL do not contain door information. MSD contains (as does RPLAN) the geometric information of the doors as well – a necessary feature to better understand and exploit the interconnectivity between spaces.

Size and Diversity. RPLAN (∼81K) and LIFULL (∼124K) have significantly more floor plans than MSD (∼5K) (column 8 in Table 1). The total number of *rooms* is much closer though: ∼165K for MSD vs. ∼539K and ∼1010K for RPLAN and LIFULL, resp. However, RPLAN and LIFULL contain a serious amount of near-duplicates. We measure the number of near-duplicates in a floor plan dataset by measuring the MIoU between pairs of floor plans and removing those that exceed a certain threshold, which we set to 0.87, equivalent to the procedure in [2]. While the number of near-duplicates in MSD is negligible (<1%), those in RPLAN and LIFULL are not: 70% for RPLAN and 50% for LIFULL. When filtering out the near-duplicates from the original RPLAN and LIFULL datasets, the number of rooms in MSD and RPLAN are approximately equal, while the number in LIFULL still remains significantly larger. In terms of the topology of the room graphs, MSD is notably more diverse than RPLAN and slightly more diverse than LIFULL. The diversity is measured as the entropy over the distribution of graphs:

$$\mathrm{H}_g = -\sum_{g \in \mathcal{G}} p_g(g) \log p_g(g) \quad (1)$$

where $\mathcal{G}$ is the set of distinct un-attributed room graphs in the dataset and $p_g(g)$ is the probability of a floor plan having a corresponding room graph equal to graph g. $p_g(g)$ is numerically approximated through the dataset. The entropy over the graph distributions for MSD, RPLAN, and LIFULL are resp., 8.02, 4.56, and 7.79, which reveals that MSD is most diverse in terms of the connectivity.

In summary, MSD is a large floor plan dataset of European (Switzerland) building complexes and exceeds other datasets in layout complexity and graph diversity. MSD is, above all, the first big dataset that makes explicit the inter-relations between spatially-connected apartments. Figure 1 further reveals the noteworthy differences between MSD and other floor plan datasets.

4 Floor Plan Generation Task

We set our task as a real-world design formulation by bridging the schematic design (spatial zoning) to the detailed design (floor layout). Similar to [5], we formulate the floor plan generation task as a multi-modal constrained optimization problem with the following in- and outputs:

- **Input 1.** The building structure indicates where the necessary structural components are positioned. The building structure is represented as a set of geometries or as a binary image.
- **Input 2.** The zoning graph defines the connectivity of areas and is represented as a graph with category-attributed nodes and edges that indicate the zoning classes.
- **Output.** The floor plan which is either represented as an image with pixel values that indicate the room classes or a room graph with geometry- and category-attributed nodes that indicate the shape and room category. We include both representations to enable the use of different model architectures such as convolutional neural networks (that need images) and graph neural networks (that need graphs).

4.1 Evaluation Metrics

To measure the performance of the models, we compute both the visual and topological similarities between the predicted and ground truth floor plans. The ordered sets of target and predicted floor plans are, resp., denoted as $\mathcal{Q} = \{Q_i\}_{1,\ldots,N}$ and $\mathcal{K} = \{K_i\}_{1,\ldots,N}$, in which N is the size of the test set.

Mean Intersection-over-Union. The rooms of a floor plan must have the correct shape and location. To measure the performance at the pixel level, the Mean Intersection-over-Union (MIoU) between $\mathcal{Q}$ and $\mathcal{K}$ is used. Equivalent to [2], MIoU across all classes $c \in \mathcal{C}$ is computed by:

$$\mathrm{MIoU}(\mathcal{Q};\mathcal{K}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c\in\mathcal{C}} \frac{R_c(Q_i)\bigcap R_c(K_i)}{R_c(Q_i)\bigcup R_c(K_i)}, \tag{2}$$

where $R_c(\cdot)$ is the function that outputs the region in the image occupied by c.

Graph Compatibility. It is similarly important that the topology of the predicted floor plan's composition closely matches that of the ground truth. To measure the consistency between the predicted and target graph, we compute the graph compatibility between the graph extracted from the predicted floor plan and the target graph. [2,16,17,21,27] measure the compatibility based on a graph edit distance (GED) [20]. The output of the floor plan generation task, however, hinders the practical use of the GED in our case, because doors are not predicted in our setting. Similar to the graph extraction algorithm used in the making of MSD, we extract the room graphs of the predicted floor plan by looping through all the pairs of different areas of a given floor plan. We assign an edge whenever the minimum distance between the areas is less or equal to a buffer. The compatibility is computed by checking whether the edges from the target graph are retained in the predicted graph:

$$\text{Compatibility}(\mathcal{Q};\mathcal{K}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\left|\mathcal{E}_i^k\right|}\sum_{e\in\mathcal{E}_i^k}\mathbf{1}\left[e\in\mathcal{E}_i^q\right], \tag{3}$$

Extracting graphs from noisy pixel maps is error-prone; hence, we refrain from using it for methods alike (*e.g.*, for UN; see Sect. 5.2). For graph-based methods (*e.g.*, for MHD; see Sect. 5.1), graph extraction from the predicted layouts could lead to errors as well, usually when a predicted layout contains many overlapping rooms. However, we found that such scenarios do not often occur (see suppl. mat. for details). Hence, we deem extracting room edges algorithmically as reasonable. (Note that previous works use a similar algorithmic approach too.)

As mentioned before, previous works use a GED to measure the compatibility [2,16,17,21,27]. Hence, lower scores suggest better methods. We measure a graph *similarity* instead of *distance*. Therefore, a high score (instead of low in the case of GED) positively correlates with performance.

5 Models

We develop two baseline models to benchmark MSD: a diffusion- and segmentation-based approach. Figure 3 provide visual clarifications of both baselines. We also tested the generalizability of *HouseGAN++* [17] and *FLNet* [28], for which the results are provided in the suppl. mat. (Note that most of the details on the model architectures, the training, and pre-processing are given in the suppl. mat.).

5.1 Modified HouseDiffusion (MHD)

HouseDiffusion (HD) [21] is a state-of-the-art model for graph-constrained floor plan generation. To adapt HD to our task, a cross-attention module is added, which effectively conditions the diffusion process on the building structure. To learn the room graph from the zoning graph, we use a GAT [29], which operates as a pre-processing step to the diffusion process. We coin our method *Modified HouseDiffusion* (MHD). Figure 3 (right) provides all modifications and added modules, which are depicted in blue.

HouseDiffusion (HD). In HD, floor plans are represented by a set of polygons $P = \{P_1, P_2, ..., P_N\}$, each representing a room or door. Each polygon $P_\bullet$ is defined by a sequence of 2D corner coordinates, $C_{l,m} \in \mathbb{R}^2$ in which l refers to l-th polygon and m the index of the corner. In the forward process, noise is added to the corner coordinates at each timestep t such that at timestep $t = T$ all corner coordinates follow a normal distribution. The corner coordinates at timestep $t = 0$ remain unaltered. The goal of the model is then to learn the reverse process (*i.e.*, to iteratively denoise the noisy corner candidates back). At its core, HD consists of three attention layers with structured masking: 1) CSA, limiting attention among corners in the same room or door, 2) GSA, full stack attention between every pair of corners across all rooms, and 3) RCA, limiting attention between connected room-to-door corner pairs.

Wall-Cross Attention (WCA). MHD expects the building structure to be encoded as a set of wall elements (straight lines) w_i, which are extracted from the binary image by a morphological thinning technique followed by skeletonization (see suppl. mat. for details). Each wall element is converted into a 512-d wall embedding $\hat{w}_i$ by an MLP followed by three multi-head attention modules. To condition the model on the building structure, we add an extra cross-attention module (WCA) between all corner and wall embeddings.

Graph Attention Network (GAT). Instead of changing HD's architecture to denoise a room type for each corner in addition to the coordinates, we separately learned the room types beforehand. We use a GAT [29] to learn the room graph from the zoning graph, by essentially framing the problem as node classification.

Minimum Rotated Rectangle (MRR). In HD, the number of corners is sampled from the known corner count distributions per room type in the training set. In contrast to RPLAN, which typically has between 4 and 10 corners per room, MSD contains many areas with a much larger amount of corners, making it more difficult for the model to appropriately denoise the polygons. In addition to doing experiments with the full set of corners (POL) we approximate the polygons by a minimum rotated rectangle fit (MRR), and subsequently learn to denoise the MRRs instead.

5.2 Graph-Informed U-Net (UN)

We propose a floor plan generation model based on U-Net [19] for direct prediction at the pixel level. At the deepest level of the network, the U-Net is constrained on a graph-level encoding of the zoning graph which is learned by a GCN [8].

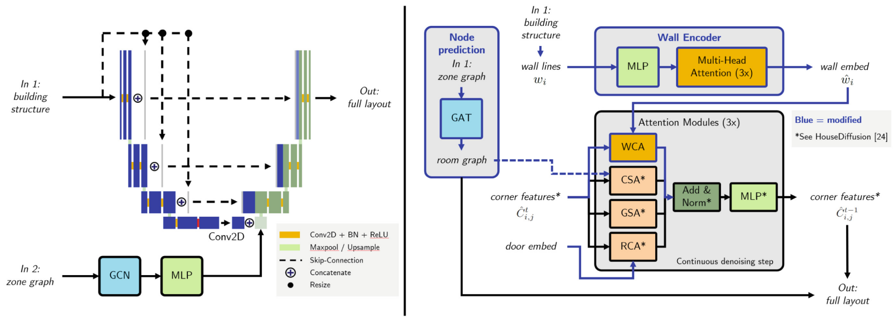

Fig. 3. Baseline methods for floor plan generation. (**Left: UN**) UN takes the building structure (image) as input to the U-Net. The U-Net is composed of an encoder and decoder using the conventional up- and down-sampling 2D convolutions, resp., and includes skip connections between the encoder and decoder feature maps at equivalent feature map scales. A GCN is used to map the zoning graph to a feature vector which is concatenated to the latent space of the U-Net. (**Right: MHD**) A wall encoder is used to map the pre-processed building structure into corresponding wall embeddings. MHD expands HD [21] by introducing an extra attention module (WCA) between the wall embeddings from and corner features of the rooms. A GAT is separately trained to predict the room types from the zoning types, which are used to "color" the full layout. (Color figure online)

U-Net. A U-Net is used to 'segment' the building structure into the floor plan. Essentially, a U-Net is an autoencoder with the addition of skip-connection between the feature maps of the encoder and decoder that share the same feature map resolution. Similar to the original U-Net implementation, we use convolutional layers for both down- and up-sampling. The output of the U-Net is the floor plan image.

Graph Convolutional Network (GCN). A GCN [8] is used to learn a fixed-sized graph-level embedding of the zoning graph. To effectively combine the graph and boundary representations, the graph-level embedding is concatenated to the deepest layer's feature map of the U-Net.

Boundary Pre-processing. Inspired by [30], we pre-process the building structure into a 3-channel image, distinguishing the interior (channel 1), the exterior (channel 2), and the original building structure (channel 3). We use Segment Anything [9] to extract the interior and exterior from the building structure.

6 Results

MIoU. To measure the visual similarity, we use (Eq. 2). The polygonal outputs of MHD are drawn on the same image canvas as that of the ground truth. The drawing order matters and is from largest to smallest region size to make sure the smaller areas are not entirely occluded by larger areas.

The average MIoU (Table 2) ranges from 10.9 to 42.4. Intuitively, this means that for all the models, a pixel is more likely to be predicted incorrectly. We conclude that the performance is poor and does not yet comply with the performance standards we would like to see. The apparent mismatch between predictions and targets is further investigated based on some of the generated examples, provided in Fig. 4. Even though the relative positioning of the areas (for MHD models) and pixel regions (for UN models), for most generated floor plans, are predicted quite accurately, the exact locations and precise shapes of the areas and regions are far from accurate yet – explaining, indeed, the low overlap scores.

Table 2. MIoU and compatibility scores for MHD and UN. The best scores across indicated in bold, and we underline the scores that are best within each approach. The scores for different floor plan 'sizes' – based on the range in a number of areas – are provided in the different columns of MIoU and graph compatibility. The graph compatibility scores for the UN-based models are not available (n.a.) because graphs cannot be reasonably extracted from the output images. The vanilla version of UN is denoted as "U-Net", and "(pre)" indicates the use of Segment Anything for pre-processing. MHD considers either the full polygons (POL) or a minimum rotated rectangle fit (MRR). "+WCA" indicates the use of the full-stack attention module between corner and wall embeddings.

Method	MIoU (↑)						Compatibility (↑)					
	avg.	15–19	20–29	30–39	40–49	50+	avg.	15–19	20–29	30–39	40–49	50+
U-Net	32.5	33.4	33.1	32.8	29.7	29.3	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
UN	40.6	44.8	42.9	38.4	32.3	30.4	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
UN (pre)	**42.4**	**45.4**	**45.4**	**40.6**	**35.1**	**32.2**	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
MHD (POL)	10.9	11.6	11.5	10.2	9.8	9.1	80.3	80.1	79.5	81.4	80.4	81.9
MHD (POL) + WCA	17.9	18.6	18.4	17.6	16.2	15.5	71.1	70.5	70.7	71.4	71.9	73.7
MHD (MRR)	11.5	12.2	12.2	11.1	10.2	9.0	**87.1**	**85.9**	**87.3**	**88.0**	**87.5**	**88.6**
MHD (MRR) + WCA	21.8	23.5	22.0	21.0	20.1	17.9	76.2	76.0	75.6	75.8	77.6	79.2

Graph Compatibility. The compatibility is only measured for MHD, because reliably extracting graphs from the floor plans generated by UN is too ambiguous. For an image size equal to 512, the buffer is set to 5 to allow some, but not too much, space between rooms. The compatibility is computed between the extracted room graph of the predicted floor plan and the room graph of the target floor plan by Eq. 3.

Compared to the MIoU, the compatibility scores, ranging between 74.4 and 87.0, are much higher; thus, the topology of the zoning graphs is largely retained in the generated floor plans. Therefore, MHD can reasonably well learn how the rooms should be composed.

UN vs. MHD. Figure 4 shows the qualitative differences between MHD and UN models. One observation is that MHD creates composed shapes in which the distinct areas can be easily separated by the eye. In contrast, UN models create segmented scenes for which it is less visible which set of pixels belongs to which room. The MIoU scores are, however, much higher for the UN models from which we can conclude that the UN models have a better understanding of the placement of specific rooms in relation to the building structure. Indeed, the example outputs of the UN models clearly show that the central regions in the floor plans are usually corridors, that the balconies are placed outside the building structure, and that the kitchen is usually located close to the living room – characteristics that are to lesser extend present in the floor plans generated by MHD.

The discrepancy in the performance (in MIoU) might stem from the different losses. Clearly, the loss of UN (cross-entropy at pixel level) is closely aligned with evaluating MIoU. However, in the case of MHD, the loss and evaluation are not necessarily as closely aligned. The objective is similar to other diffusion models: at each iteration, you randomly select a time-step t and learn a mapping for the reverse noise, which is parameterized by a neural network as $e_\theta\left(C_{l,m}, t\right)$. Hence, the neural network $e_\theta\left(\bullet, \bullet\right)$ learns to effectively denoise corner points *for a given time step*. This is not necessarily the same as learning a mapping from input (structure and graph) to a *fixed* output (floor plan layout), which could for a part explain the discrepancy in performance.

6.1 Ablation Studies

UN. From Table 2, the effects of adding the GCN and/or pre-processing are significant and increase MIoU. The impact of the GCN is most significant for smaller building complexes. The MIoU scores for larger floor plans are comparable across the three methods, which suggests that the GCN struggles with larger graphs, emphasizing the need for graph models that can cope better with both small- as well as large graphs. The examples in Fig. 4 show that the addition of the pre-processing tends to improve the placement of the areas within the interior of the building.

MHD. Observed from the generated examples in Fig. 4, adding WCA leads to floor plans that follow the building structure reasonably well. This is also shown by the increase in MIoU scores between the models with and without WCA. However, the addition of WCA leads to degraded graph compatibility, likely because when conditioning on the building structure, the model has to learn to place rooms along the existing structure, instead of only placing rooms relative to each other. The model that uses the minimum rotated rectangle (MRR) approximation performs better than the model with full polygons (POL), both on MIoU and compatibility, which we attribute to the following potential causes. First, some rooms in MSD have many corners and are likely more complicated to learn. Second, the number of corners for POL is sampled independently of the building structure, which can lead to room polygons having too few or too many corners to fit the building structure.

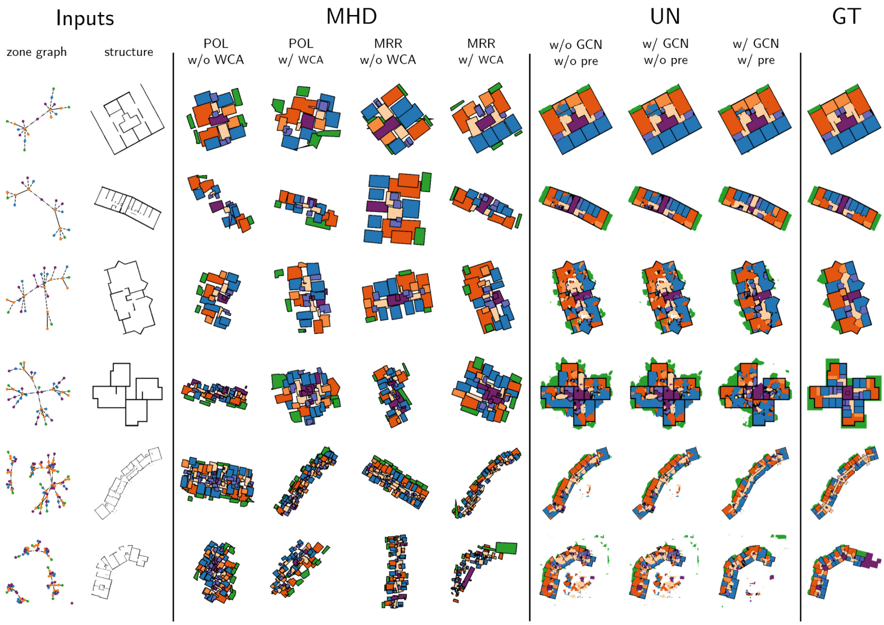

Fig. 4. Example generations of MHD and UN. Columns 1 and 2 show the inputs: the zone graph and building structure respectively. Columns 3–6 show the floor plans generated by the MHD variants. Columns 7–9 show the floor plans generated by the UN variants. Column 10 shows the ground truth.

Overall, the floor plans often look infeasible. We could, however, train MHD on RPLAN successfully (see suppl. materials); hence, we believe that the poor results do *not* come from improper training. Instead, we attribute the poor results to the more complex benchmark we set: more complex graphs; more irregularly shaped rooms; unit connectivity; no axis alignment; etc.

7 Conclusion

We developed MSD – a large-scale floor plan dataset of building complexes. In contrast to the other floor plan datasets, MSD contains more complex floor plans. To test the generalizability and scaling of the current state-of-the-art floor plan generation method to MSD, we developed two baseline models. The baseline models were highly inspired by previous works and only altered where needed. Our experiments show that the generation of more complex, hence more realistic, floor plans cannot yet be properly addressed by strategies that are currently most promising in floor plan generation. To address real-world floor plan design, our benchmark asks for even smarter methods in the future.

Acknowledgment. We would like to thank all architects and students who participated in our user study (see suppl. mat.).

References

1. Dodge, S., Xu, J., Stenger, B.: Parsing floor plan images. In: International Conference on Machine Vision Applications (2017)
2. van Engelenburg, C., Khademi, S., van Gemert, J.: SSIG: a visually-guided graph edit distance for floor plan similarity. In: International Conference on Computer Vision Workshop (2023)
3. Goyal, S., Mistry, V., Chattopadhyay, C., Bhatnagar, G.: BRIDGE: building plan repository for image description generation, and evaluation. In: International Conference on Document Analysis and Recognition (2019)
4. de las Heras, L.P., Terrades, O.R., Robles, S., Sánchez, G.: CVC-FP and SGT: a new database for structural floor plan analysis and its groundtruthing tool. Int. J. Doc. Anal. Recogn. **18** (2015)
5. Hu, R., Huang, Z., Tang, Y., Van Kaick, O., Zhang, H., Huang, H.: Graph2Plan: learning floorplan generation from layout graphs. ACM Trans. Graph. **39** (2020)
6. Kalervo, A., Ylioinas, J., Häikiö, M., Karhu, A., Kannala, J.: CubiCasa5K: a dataset and an improved multi-task model for floorplan image analysis. In: Felsberg, M., Forssén, P.-E., Sintorn, I.-M., Unger, J. (eds.) SCIA 2019. LNCS, vol. 11482, pp. 28–40. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20205-7_3
7. Khodabakhshi, K., Khaghani, S., Garmaroodi, A.A.: A procedural approach for configuration of residential activities based on users' needs and architectural guidelines. Nexus Netw. J. (2022)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representation (2017)
9. Kirillov, A., et al.: Segment anything (2023). http://arxiv.org/abs/2304.02643
10. Liu, C., Wu, J., Kohli, P., Furukawa, Y.: Raster-to-vector: revisiting floorplan transformation. In: International Conference on Computer Vision (2017)
11. Ltd., L.C.: LIFULL home's - a large-scale database of residential floor plans (2015). https://www.nii.ac.jp/dsc/idr/en/lifull/
12. Lu, Z., et al.: Data-driven floor plan understanding in rural residential buildings via deep recognition. Inf. Sci. **567** (2021)
13. Luo, Z., Huang, W.: FloorplanGAN: vector residential floorplan adversarial generation. Autom. Constr. **142** (2022)
14. Mostafavi, F., Tahsildoost, M., Zomorodian, Z.: Energy efficiency and carbon emission in high-rise buildings: a review (2005-2020). Build. Environ. **206** (2021)
15. Müller, P., Wonka, P., Haegler, S., Ulmer, A., Van Gool, L.: Procedural modeling of buildings. ACM Trans. Graph. **25** (2006)
16. 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: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 162–177. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_10
17. Nauata, N., Hosseini, S., Chang, K.H., Chu, H., Cheng, C.Y., Furukawa, Y.: House-GAN++: generative adversarial layout refinement network towards intelligent computational agent for professional architects. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)
18. Peng, C.H., Yang, Y.L., Wonka, P.: Computing layouts with deformable templates. TOG **33** (2014)

19. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
20. Sanfeliu, A., Fu, K.S.: A distance measure between attributed relational graphs for pattern recognition. IEEE Trans. Syst. Man Cybern. **13** (1983)
21. Shabani, M.A., Hosseini, S., Furukawa, Y.: HouseDiffusion: vector floorplan generation via a diffusion model with discrete and continuous denoising. In: IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada (2023)
22. Sharma, D., Gupta, N., Chattopadhyay, C., Mehta, S.: DANIEL: a deep architecture for automatic analysis and retrieval of building floor plans. In: International Conference on Document Analysis and Recognition, vol. 01 (2017)
23. Standfest, M., et al.: Swiss dwellings: a large dataset of apartment models including aggregated geolocation-based simulation results covering viewshed, natural light, traffic noise, centrality and geometric analysis (2022). https://zenodo.org/records/7788422
24. Sun, J., Wu, W., Liu, L., Min, W., Zhang, G., Zheng, L.: WallPlan: synthesizing floorplans by learning to generate wall graphs. ACM Trans. Graph. **41** (2022)
25. Surikov, I.Y., Nakhatovich, M.A., Belyaev, S.Y., Savchuk, D.A.: Floor plan recognition and vectorization using combination UNet, faster-RCNN, statistical component analysis and Ramer-Douglas-Peucker. In: Chaubey, N., Parikh, S., Amin, K. (eds.) COMS2 2020. CCIS, vol. 1235, pp. 16–28. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-6648-6_2
26. Swaileh, W., Kotzinos, D., Ghosh, S., Jordan, M., Vu, N.-S., Qian, Y.: Versailles-FP dataset: wall detection in ancient floor plans. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12821, pp. 34–49. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86549-8_3
27. Tang, H., Shao, L., Sebe, N., Gool, L.V.: Graph transformer GANs with graph masked modeling for architectural layout generation. IEEE Trans. Pattern Anal. Mach. Intell. (2024)
28. Upadhyay, A., Dubey, A., Arora, V., Kuriakose, S.M., Agarawal, S.: FLNet: graph constrained floor layout generation. In: International Conference on Multimedia and Expo Workshops (2022)
29. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representation (2018)
30. Wu, W., Fu, X.M., Tang, R., Wang, Y., Qi, Y.H., Liu, L.: Data-driven interior plan generation for residential buildings. ACM Trans. Graph. **38** (2019)
31. Yin, Y., Hutchcroft, W., Khosravan, N., Boyadzhiev, I., Fu, Y., Kang, S.B.: Generating topological structure of floorplans from room attributes. In: ACM International Conference on Multimedia (2022)
32. Zhang, F., Nauata, N., Furukawa, Y.: Conv-MPN: convolutional message passing neural network for structured outdoor architecture reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)