FLOWPLAN: JOINT STRUCTURAL AND GEOMETRIC GENERATION OF POLYGONAL FLOORPLANS VIA FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Floorplan generation is an increasingly prominent challenge at the intersection of computer vision, computer graphics, and architecture. Early rule-based heuristics often fail to generalize across diverse, real-world datasets. Recent generative methods have achieved strong performance, but are predominantly evaluated on relatively regular residential layouts and often rely on predefined room semantics, topology, or cardinality. In this work, we introduce FlowPlan , a "generate all" approach that concurrently produces both discrete room semantics and their corresponding continuous geometry. By conditioning the generation process solely on the building footprint, our method eliminates the need for predefined semantic assumptions. We explore various flow matching formulations for discrete variables and evaluate our approach against established baselines on a real-world dataset. Due to our method's flexibility, a single model can be supplied with additional conditioning at test time and thus compared against baselines with and without semantic generation capabilities, demonstrating performance on par with highly specialized models.

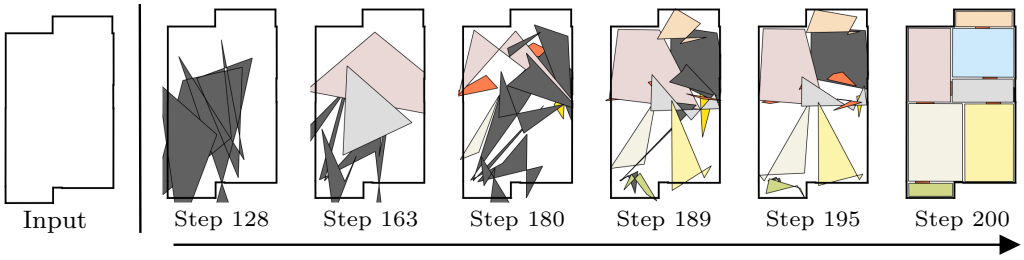


Fig. 1: Given a building footprint, FlowPlan, a unified continuous-discrete flow matching model, generates geometry, structure, and semantics concurrently.

1 INTRODUCTION

Automated floor plan generation is a longstanding problem at the intersection of computer vision, computer graphics, and architecture. Generating floor plans that satisfy given constraints can greatly accelerate the design process and let engineers focus on the important details. Early approaches were rule-based (Peng et al., 2014; Wu et al., 2018; Laignel et al., 2021) utilizing quadratic programming, deformable templates, and genetic optimization. With the advent of deep learning, research shifted to learning-based methods. Early deterministic learning-based works treated the problem as an image prediction task conditioned on the boundary, wall structure, and room graphs (Wu et al., 2019; Hu et al., 2020; Sun et al., 2022). The majority of methods produce deterministic results given a user input, limiting variability. With advances in generative modelling, later works adopted GANs to generate multiple plausible layouts for a given room graph (Nauata et al., 2020; 2021; Luo & Huang, 2022), and later diffusion models became popular for their stable dynamics and higher fidelity (Shabani et al., 2023). However, the majority of methods require a room graph as part of their input signal, essentially requiring the user to set the room structure,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

such as the number of rooms, number of corners per room, and semantics, such as room type, which is a strong assumption. In addition, methods that only require the building footprint (Hong et al., 2024) propose a multi-step approach in which they first independently sample a graph. As shown in Fig. 2, we jointly generate geometry, structure, and semantics. The approach most similar to our work, (Hu et al., 2025), still adopts a two-step approach: it first generates geometry and semantics, then uses a deterministic model for edge prediction, i.e., structure prediction. In contrast, we propose a multimodal flow matching model that generates all modalities concurrently. Multimodal flow matching models have been successfully used in molecular design, which requires generating sequences of both continuous and discrete variables (Campbell et al., 2024; Chen et al., 2025). Drawing inspiration from Shabani et al. (2023), we note that a polygonal floor plan can also be represented as a set of continuous and discrete quantities. Combining these two ideas, we propose a model that generates a sequence of continuous 2D coordinates, discrete connectivity information, and room semantics that are simultaneously decoded into polygonal floor plans.

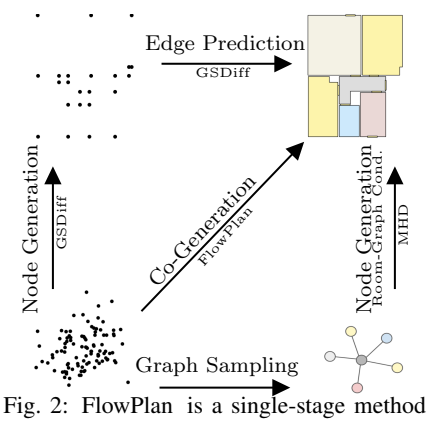


Fig. 2: FlowPlan is a single-stage method that concurrently generates continuous and discrete variables to model polygonal floor plans.

In summary, our contributions are as follows:

- An assessment of different joint continuous-discrete flow matching formulations.
- FlowPlan, a unified continuous-discrete flow matching model that generates geometry, structure, and semantics concurrently, while only relying on the building footprint.
- We show better performance on a large-scale real-world dataset with improved generation speed.
- We show how to use the flexible "generate all" approach to incorporate additional structural or semantic conditioning at test time.

2 Related Work

Optimization and deterministic synthesis. Early approaches to floorplan synthesis relied on explicit optimization utilizing geometric and architectural constraints. (Peng et al., 2014) optimize deformable templates, (Wu et al., 2018) formulate interior layout generation using mixed-integer quadratic programming, and (Laignel et al., 2021) combine constraint programming with genetic optimization. While these methods provide explicit control over design constraints, their rule-based nature hinders diverse floor plan generation. Learning-based approaches learn spatial priors from data. RPLAN (Wu et al., 2019) proposes a two-stage procedure that firstly places a single room and subsequently generates and refines walls. Graph2Plan (Hu et al., 2020) predicts a raster layout and room boxes conditioned on the building footprint and subsequently refines a valid floorplan. WallPlan (Sun et al., 2022) repeatedly predicts wall graphs and floorplan semantics to iteratively synthesize its geometry and labels. Although their learned predictors avoid explicit optimization at inference, the resulting layout is largely determined by the supplied conditions.

Generative vector floorplans. Generative models enable sampling multiple layouts satisfying the same conditioning signal. House-GAN and House-GAN++ (Nauata et al., 2020; 2021) introduced adversarial generation of graph-constrained residential layouts, while FloorplanGAN (Luo & Huang, 2022) directly targets vector output through a vector generator and a differentiable raster discriminator. However, GANs often struggle with precise vector geometries, yielding jagged boundaries that require post-processing. HouseDiffusion (Shabani et al., 2023) introduced diffusion for vector polygon floorplan generation by employing continuous diffusion followed by denoising discretised coordinates. Its generation, however, is conditioned on a predefined room graph containing room semantics and connectivity, while the number of polygon corners is supplied or sampled prior to generation. Modified HouseDiffusion (MHD) (Van Engelenburg et al., 2024) extends conditioning to building footprints through wall cross-attention and predicts room semantics from a zoning graph using a separate graph network. Cons2Plan (Hong et al., 2024) extends conditional diffusion

to graph, footprint, and joint graph-footprint constraints, while footprint conditioning is optional. GSDiff (Hu et al., 2025) reduces the amount of predefined semantic structure by formulating synthesis as structural graph generation. It first generates semantic wall junctions with diffusion and subsequently predicts wall connectivity using a separate Transformer, from which room polygons are extracted. Recent autoregressive alternatives instead serialize floorplans into discrete representations. CG-Floor (Lian et al., 2026) first predicts size-aware centroids and retrieves codebook polygons during generation. TLC-Plan (Xiong et al., 2026) generalizes the hierarchical discrete codebook to work with only footprint conditioning. FMLM (Shiohara & Yamasaki, 2026) introduced a markup representation that unifies several vector floorplan generation tasks as next-token prediction problems.

Flow Matching. Flow Matching (Liu et al., 2023; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2022) is a simulation free framework to train continuous normalizing flows (Chen et al., 2018). It is considered the current state-of-the-art in generative modeling in continuous spaces across different tasks (Esser et al., 2024; Xiang et al., 2025) due to more stable training dynamics and faster inference compared to popular autoregressive methods. Discrete Flow Matching attempts to apply ideas from Flow Matching to discrete spaces. The literature can be dissected into methods that embed discrete data into a continuous domain and apply mature Flow Matching principles (Chen et al., 2022; Dieleman et al., 2022) or by modeling Continuous Time Markov Chains (Campbell et al., 2024; Austin et al., 2021).

We formulate floorplan synthesis directly as mixed continuous-discrete generation. Continuous corner geometry, discrete structure and semantics are generated jointly from only a vectorized building footprint. Our permutation-equivariant formulation further avoids making the generated geometry dependent on an arbitrary serialization of corner tokens, while reconstructing polygon topology without a separate edge-generation stage.

3 Methodology

FlowPlan is a flow-matching model that generates 2D floorplans, given only a building footprint. We explore different combinations of continuous and discrete flow matching formulations to denoise room corner positions, their connectivity and semantic class. Fig. 4 shows the full model.

3.1 Preliminaries

Floor Plan Representation. We extend the floor plan representation introduced in HouseDiffusion (Shabani et al., 2023) by relaxing the conditioning and enriching the semantic output. Formally, a floor plan F_j is a collection of n_j corners and their connectivity.

$$F_j = \{C_0, ..., C_{n_j}\}, \;\; C_i = (\mathbf{p}_i, r_i, c_i, rt_i, a_i), \;\; \mathbf{p}_i \in [0,1]^2, \;\; r_i, c_i, rt_i \in \mathbb{Z}_+, \;\; a_i \in \{0,1\} \quad (1)$$

with corner coordinates $\mathbf{p}_i$, the room index r_i, the corner index c_i and the room type rt_i, as illustrated in Fig. 3. As the number of rooms and corners is unknown a priori, we pick reasonable upper bounds N_r and N_c and let the model decide which corners to activate (e.g., a_i). Room types rt_i include labels such as *kitchen*, *living room*, or *bedroom*. Doors have the same parameterization and are treated simply as small rooms. Shabani et al. (2023), in contrast, conditions on a scene graph that includes the number of corners, rooms, and room types, while we attempt to generate them all together. FlowPlan is conditioned only on the building footprint, which we represent through its boundary as a sequence of 9 evenly spaced points along each line segment. Our model processes these points via cross-attention.

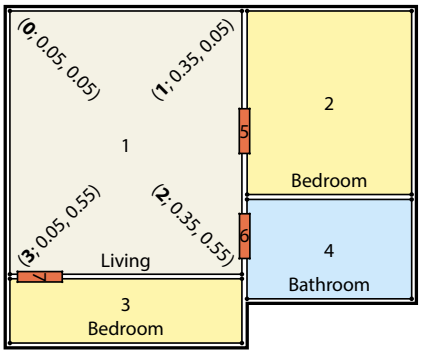


Fig. 3: Illustration of our floor plan representation with tuples representing $(c; p_x, p_y)$ and room IDs in the room center.

Flow Matching. Flow Matching (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2022) learns a time-dependent vector field that transports samples from a simple source distribution p_0 to the data distribution p_{data}. For the commonly used straight-line probability path, we sample

$\epsilon \sim p_0$, $x \sim p_{\text{data}}$, and define

$$\mathbf{z}_t = (1-t)\epsilon + t\mathbf{x}, \qquad t \in [0,1]. \tag{2}$$

The corresponding conditional velocity is constant along the path, $\mathbf{x} - \epsilon$, and can be learned with

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t,\mathbf{x},\epsilon}\left[\|v_\theta(\mathbf{z}_t, t) - (\mathbf{x} - \epsilon)\|_2^2\right]. \tag{3}$$

At inference, we sample $\mathbf{z}_0 \sim p_0$ and integrate $\frac{d\mathbf{z}_t}{dt} = v_\theta(\mathbf{z}_t, t)$ from $t = 0$ to $t = 1$.

Discrete Flow Matching. For discrete variables, an ODE cannot directly move samples between categorical states. Discrete Flow Matching (Campbell et al., 2024; Gat et al., 2024) instead defines a probability path over discrete states, where transition rates of a continuous-time Markov chain replace the continuous velocity field. In practice, these transitions can be obtained from a network $p_\theta(x \mid z_t)$ that predicts the clean data x from an intermediate state z_t.

A particularly simple choice is the *masking path* (Austin et al., 2021). At time t, each discrete variable is either equal to its clean data value x with probability t, or replaced by a special mask token m with probability $1 - t$:

$$p_t(z_t \mid x) = t\,\delta_x(z_t) + (1-t)\,\delta_m(z_t). \tag{4}$$

Thus, the interpolation takes place in probability space. Individual variables do not assume intermediate values, but switch from the mask token to their clean value as t increases. Generation therefore starts from a fully masked sample at $t = 0$ and progressively reveals its values until reaching the data distribution at $t = 1$. The model is trained to predict the clean value from a partially masked state using cross-entropy,

$$\mathcal{L}_{\text{D-CFM}} = \mathbb{E}_{t,x,z_t}\left[-\log p_\theta(x \mid z_t)\right]. \tag{5}$$

In order to sample a discrete sequence, we follow the Euler Method from Lipman et al. (2024) (Equation 6.7). More specifically, given a timestep h the probability of jumping on a new state is $p(jump) = 1 - e^{-hu}$ where u is the accumulated outgoing *rate*. This rule is applied independently on each position of the sequence. However, the predicted rates are not independent, since they come from a single model, and thus the predicted sequence is coherent.

3.2 Flow Matching for Vector Floorplans

Vector floorplans combine continuous geometry with discrete semantic and structural attributes. Moreover, their vector representation is not unique: the same floorplan can be serialized using different corner orders. We therefore consider three model variants that share the same backbone but differ in how discrete variables and corner ordering are represented.

Shared architecture. All variants use a Transformer operating on corner tokens. Coordinates and categorical attributes are embedded independently and summed into a common feature representation. The generation is directly conditioned on the vector representation of the building footprint. In contrast to GSDiff (Hu et al., 2025), which rasterizes the footprint boundary and encodes it with a CNN, we preserve the footprint boundary as a set of line segments. Following the wall augmentation used by HouseDiffusion (Shabani et al., 2023), each segment is sampled at regularly spaced points, whose coordinates are concatenated and projected to the model dimension. A small self-attention encoder exchanges information between segments of the footprint boundary, and the resulting features condition the floorplan Transformer through cross-attention. The variants below differ only in their flow representation and handling of corner order.

Continuous Flow Model (CFM). Our first variant represents the complete floorplan in continuous space. Coordinates are modeled directly, while categorical attributes are converted to their binary representation (as opposed to higher-dimensional one-hot encoding) and treated as continuous analog bits following Chen et al. (2023). A floorplan is therefore represented by $N_{\max}$ corner tokens containing geometry and binary-encoded attributes.

We omit explicit corner indices and instead use the order of the generated tokens to define polygon connectivity. Consequently, this representation is a sequence and we provide the Transformer with sinusoidal token positional embeddings. The complete representation is trained with the conditional flow matching objective in eq. (3). At inference, the generated analog bits are thresholded to recover the categorical variables.

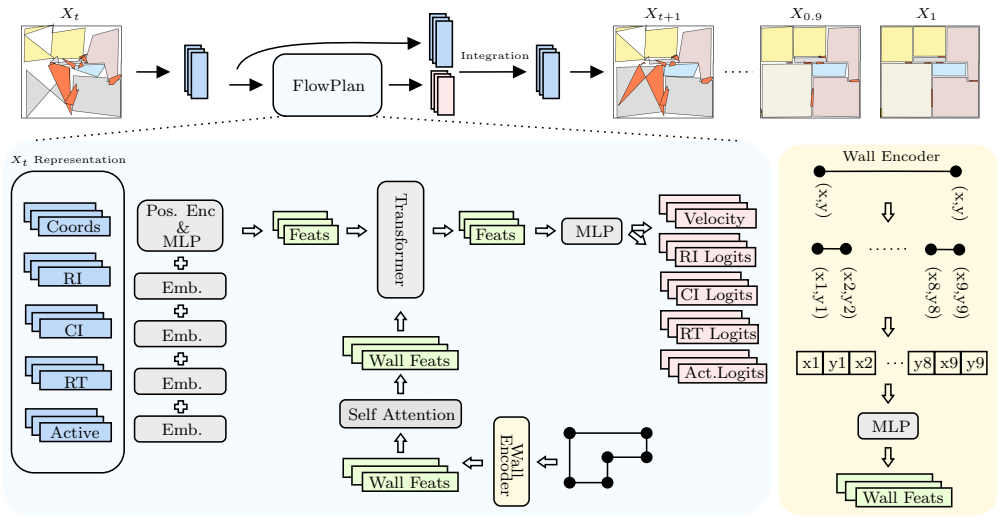


Fig. 4: Illustration of our proposed architecture. As explained in Sec. 3.1, the floorplan is represented as a set of tuples $C_i = (\mathbf{p}_i, r_i, c_i, rt_i, a_i)$, illustrated in blue. The model outputs, indicated in red, are used to perform one integration step. For continuous variables, integration means taking an Euler step; for discrete quantities, it means sampling from the predicted logits $p_\theta(x \mid z_t)$.

Additionally, supervising inactive tokens proved important for stable training; otherwise, unconstrained padded dimensions can introduce arbitrary values into the Transformer.

Discrete Flow Model (DFM). The second variant keeps the same sequence representation but models each variable in its natural domain. Coordinates are generated with continuous flow matching, while categorical attributes are modeled using masked discrete flow matching (Campbell et al., 2024; Gat et al., 2024). Polygon connectivity is still defined by token order and the model therefore retains the positional embeddings of CFM. This variant isolates the effect of native discrete modeling from changes to the floor plan representation.

Corner-Order Equivariant DFM (OE-DFM). The sequence representation introduces an arbitrary dependency on the order in which corners are stored. To remove this dependency, OE-DFM treats the corner tokens as a set by removing token positional embeddings. The Transformer is therefore permutation-equivariant with respect to the corner tokens. To encode polygon connectivity, OE-DFM explicitly predicts the corner index c_i as an additional discrete attribute. We use sinusoidal Fourier features (Mildenhall et al., 2021) to embed the 2D coordinates before projecting them to the model dimension. Connectivity is reconstructed from the predicted room and corner indices after generation. OE-DFM removes the arbitrary serialization order of corner tokens, although the explicit corner indices still admit equivalent cyclic relabelings of the same polygon.

4 Experiments

4.1 Training Details

All variants are trained for 1 million steps on an RTX 4090 for 3 days with a batch size of 64. We use a cosine learning-rate schedule starting from 0.0001 after warmup. Following Esser et al. (2024), we sample timesteps from a Logit-Normal distribution with $\mu = 2$ and $\sigma = 1$, which emphasizes later timesteps. In addition, we found it crucial to supervise all dimensions of inactive tokens. More specifically, coordinates are pushed towards $(-2, -2)$. For categorical variables, we added a "padding" label set to the maximum number of classes. Finally, during inference, we use a non-uniform integration schedule $r_s(t) = \frac{s \cdot t}{1+(s-1)\cdot t}$ with $s = 0.25$ to concentrate more inference steps in the noisy region of the path. We integrate 200 ODE steps on all our models.

4.2 DATASETS

We train and evaluate FlowPlan on MagicPlan (Hossieni et al., 2023). We choose MagicPlan for two reasons. RPLAN (Wu et al., 2019) is limited to Manhattan-type rooms, while MagicPlan contains real-world noisy floorplans with arbitrary room geometry. Further, compared to the real-world dataset of Van Engelenburg et al. (2024), MagicPlan is 12 times larger and contains varying wall gap widths between room polygons. Instead of using the rendered images provided by Hossieni et al. (2023), we preprocess the raw data and convert it to the format used by Shabani et al. (2023). After preprocessing, we obtain 67k floorplans and use a 0.75/0.15/0.15 train/val/test split.

4.3 METRICS

We report results on *FID* (Heusel et al., 2017) and IoU on the test set. *FID* compares two image distributions and empirically agrees with human perception. We compute *FID* between sets of ground-truth and generated images. The result is affected by both the generated geometry and the generated room semantics due to room coloring. Following Stein et al. (2023), we also calculate *FID* using DINOv2 (Oquab et al., 2023) features, reported as *FID-DINOv2*. To evaluate coverage and how well the generated samples follow the conditioning, we calculate the binary IoU between the covered ground-truth area and the generated sample conditioned on the building footprint. To evaluate room layout geometry, we compute *FID* on images without room coloring and refer to it as *FID-Geom*.

4.4 BASELINES

GSDiff For the task of footprint-conditioned generation, we compare against the recent state-of-the-art approach GSDiff (Hu et al., 2025). GSDiff assumes zero-thickness walls between rooms, leading to floor plans that can be modeled using a simple connected graph. This leads to shared corners and edges of adjacent rooms, resulting in simpler geometry. In principle, GSDiff can also be applied to walls with non-zero thickness; however, preliminary experiments show that the method struggles to produce complete polygons. We attribute this to the difficulty of correctly connecting corners that are close but belong to adjacent rooms. To be fairer, we construct heuristic pre- and post-processing methods. Before training, we convert dataset samples to simple connected graphs assuming zero-width walls and shared corners and edges between adjacent rooms. After generation, we convert back to room polygons using the average wall thickness measured from the training data. Fig. 5 illustrates this heuristic.

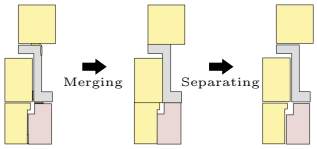


Fig. 5: Visual results from the pre/post-processing pipeline.

MHD As an additional task and to showcase the flexibility of our method, we compare against HouseDiffusion (HD) (Shabani et al., 2023) which generates coordinates based on structure and semantics. Because this model conditions only on the room graph, we use the modified version of HouseDiffusion (MHD) (Van Engelenburg et al., 2024) that adds a footprint-conditioning module. The room graph carries much more information than the building footprint we use. The room graph includes the number of rooms, their types, the respective number of corners and room adjacency. To level the playing field, we apply Repaint (Lugmayr et al., 2022) during inference for our method. Repaint supplies the model with the ground truth trajectory by interpolating the initial noise sample with the ground truth data instead of using the model's prediction. However, even via inpainting, the information of room adjacency is not passed, leaving us with a slight disadvantage. To counteract this, we remove the attention layer in MHD that models room adjacency and retrain the model.

4.5 RESULTS

4.5.1 QUANTITATIVE RESULTS

In Tab. 1 we present the quantitative comparison between GSDiff (Hu et al., 2025) and our method, conditioned only on the building footprint. Our model outperforms the baseline on all metrics, especially on FID (1.9×) and inference time (3.6×). Based on the average number of rooms predicted, we are closer to the ground truth and note that our method synthesizes more complex layouts, while GSDiff prefers simpler layouts (see Fig. 8). We further show that a smaller model can increase

Tab. 1: **Footprint-conditioned results** on MagicPlan without door modeling, reported using FID and IoU. **Best** is bold and second best is underlined.

Model Name	FID↓	FID-Dinov2↓	FID-Geom.↓	IoU ↑	#Rooms (G.T.: 5.56)	Infer. Time (ms)
GSDiff (Hu et al., 2025) (137m)	12.49	138.20	27.82	0.884	4.50	909.54
OE-DFM-Small (33m) (ours)	12.15	182.88	40.79	0.923	**6.02**	**99.87**
OE-DFM (144m) (ours)	**6.75**	**94.11**	**26.86**	**0.936**	6.06	249.68

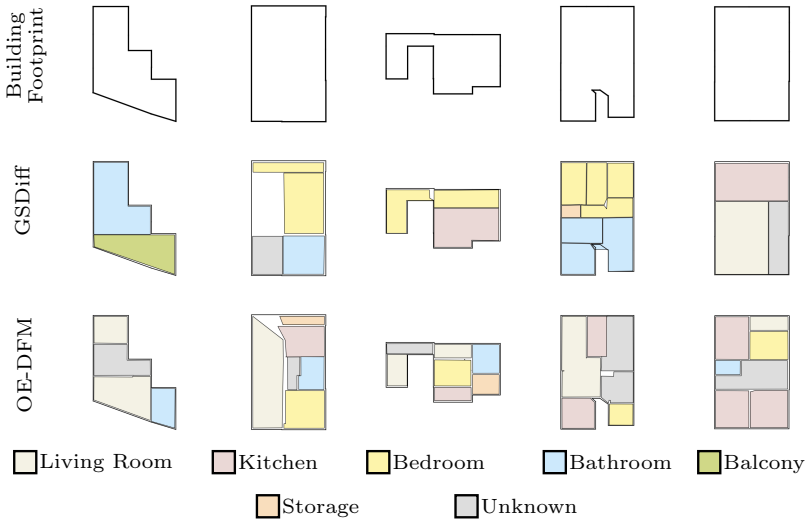


Fig. 6: Generated samples conditioned only on the building footprint from our best model and the baseline.

throughput by 9.1× while remaining competitive with the baseline. To demonstrate the versatility of our method, we show how to incorporate additional conditioning signals without the need for retraining via ground-truth inpainting. The results are reported in Tab. 2. We show competitive performance even though our model was never trained for this setting.

Tab. 2: **Footprint- and room-graph-conditioned results** via inpainting on MagicPlan.

Model Name	FID↓	FID-Dinov2↓	FID-Geom.↓	IoU ↑
MHD (Van Engelenburg et al., 2024)(Baseline)	**3.99**	**68.94**	**11.42**	0.944
MHD w/o Adj. (Baseline)	4.30	74.75	12.40	0.943
DFM (ours)	4.93	122.94	14.65	**0.960**
OE-DFM-Small (ours)	9.88	280.27	22.41	0.925
OE-DFM (ours)	6.18	193.54	17.75	0.932

4.5.2 Qualitative Results

In Fig. 6 we present the qualitative comparison using five random building footprints from the test set between our best model *OE-DFM* and GSDiff (Hu et al., 2025). The modeling assumption of GSDiff does not include door polygons, thus we also remove doors from our generated results. For the second task, generated samples conditioned on the same five random building footprints and further conditioned on the room graph via test-time inpainting are shown in Fig. 7.



Fig. 8: Diversity of the generated results.

4.5.3 Observations

As shown in Fig. 6, both models generate plausible room arrangements. GSDiff tends to produce less crowded floorplans, consistent with its lower average number of rooms in Tab. 1. Its perfectly

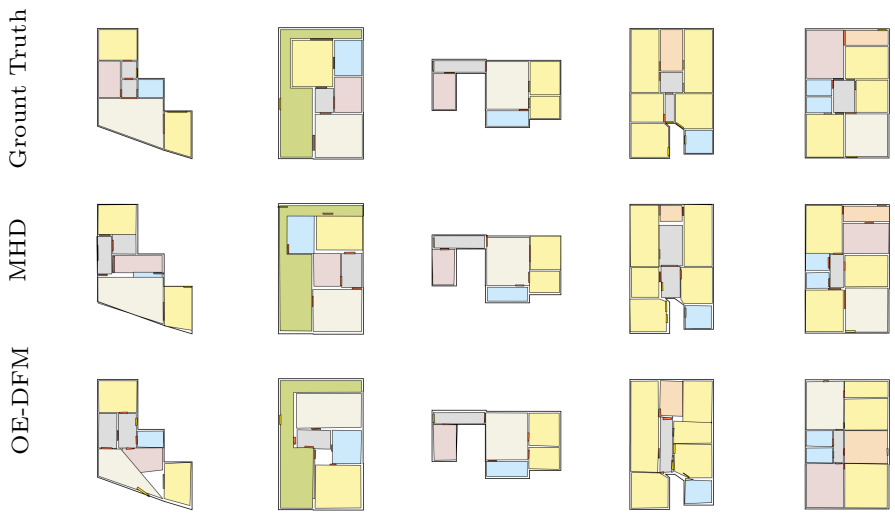


Fig. 7: Generated samples with test-time room graph conditioning v.s. native MHD.

parallel walls are imposed by post-processing when shared edges are converted into separate room polygons, whereas our model must learn this geometric regularity directly from the data.

4.5.4 Failure Cases

We identify two failure cases of our model, shown in Fig. 9. In the first case (left), we observe a misplaced corner in an otherwise well-formed room. This occurs when two corners of the same room sample the same corner index during one step. It is partly caused by the steep rescaling function at later steps, which increases the timestep h and, in turn, the probability of jumping from the current state, as described in Sec. 4.1. In the second case (right), we observe uncovered spaces. This typically occurs for complex building footprints that may be underrepresented in the dataset.

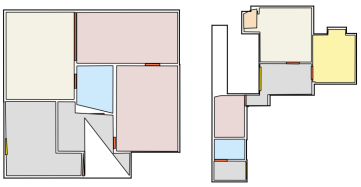


Fig. 9: Two distinctive failure cases.

4.6 Ablating Flow Formulations

We evaluate the design choices from Sec. 3.2 for discrete floor plan generation. While naive DFM yields no improvement, its strong inpainting performance (Table 3 v.s. Table 2) indicates that continuous modeling succeeds when given accurate discrete variables. Motivated by this, we introduce *2S-DFM*, which uses independent sampling for continuous and discrete timesteps as in Campbell et al. (2024) to decouple modality exploration. *2S-DFM* substantially improves the sequence-based formulation but ultimately scales poorly, demonstrating reduced adherence to the building footprint, indicated by decreased IoU. Furthermore, applying separate samplers does not benefit the order-equivariant model (OE-DFM). OE-DFM performs equally well with and without inpainting, suggesting its discrete variables are already modeled correctly. Ultimately, the scaled corner order-equivariant formulation achieves the best quantitative, in Table 3, and qualitative, in Figure 10, results.

5 Limitations.

Flow matching requires a significant amount of data, making small-scale real-world datasets like MSD (Van Engelenburg et al., 2024) challenging. Future work needs to investigate more data-

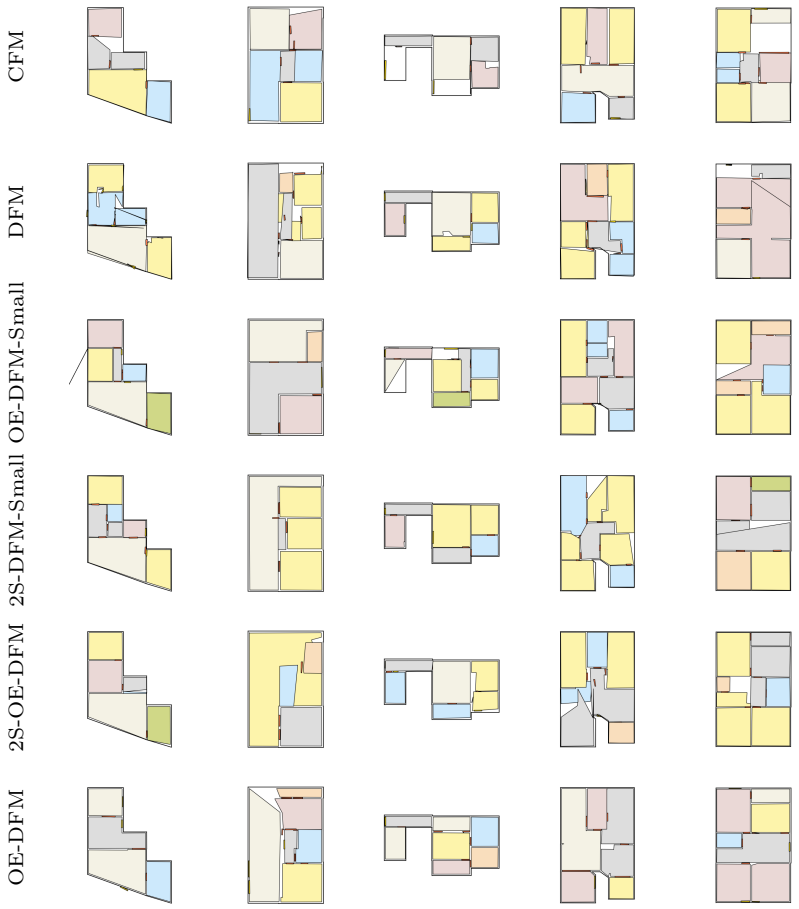


Fig. 10: Generated samples conditioned on the building footprint. Comparison between our model variants.

Tab. 3: Comparison of different flow formulations proposed in Sec. 3.2 on MagicPlan (with doors).

Params.	Model Name	FID↓	FID-Dinov2↓	FID-Geom.↓	IoU ↑
33 M	CFM	16.19	423.88	30.21	0.933
	DFM	24.48	413.56	35.03	0.939
	OE-DFM-Small	9.52	187.03	22.18	0.927
	2S-DFM-Small	8.26	155.35	17.83	**0.947**
	2S-OE-DFM	18.37	353.85	33.89	0.902
144 M	2S-DFM	12.40	406.42	32.20	0.864
	OE-DFM	**6.67**	**101.37**	**15.33**	0.940

efficient or transfer learning strategies. A further line of work could investigate unconditional generation altogether.

6 Conclusion

We presented FlowPlan, a unified flow matching method for continuous and discrete variables that jointly generates floor plan geometry, structure, and semantics. Our method conditions solely on the building footprint and does not rely on rich scene graphs. We achieve state-of-the-art performance on footprint-conditioned floor plan generation. Furthermore, additional conditioning signals can be supplied via inpainting at test time without retraining. Under joint footprint and room-graph conditioning, FlowPlan performs competitively despite never being trained with room-graph inputs.

7 AI USAGE DISCLOSURE

In this work, we used AI to implement methods and reformat datasets. We did not use AI to generate synthetic datasets, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, support qualitative and thematic data analysis, or interpret results. Moreover, the use of AI to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, or assist in the writing of proofs was not applicable in our work. We also used AI to summarize and refine the vocabulary of manually written passages. All AI-generated content was verified by team members, and we take responsibility for the final content of this work.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022. 3

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021. 3, 4

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=kQwSbv0BR4. 2, 3, 4, 5, 8

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 3

Ruizhe Chen, Dongyu Xue, Xiangxin Zhou, Zaixiang Zheng, Xiangxiang Zeng, and Quanquan Gu. An all-atom generative model for designing protein complexes. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9516–9545. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/chen25bz.html. 2

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022. 3

Ting Chen, Ruixiang ZHANG, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3itjR9QxFw. 4

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022. 3

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 3, 5

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, volume 37 of *NIPS '24*, pp. 133345–133385, Red Hook, NY, USA, December 2024. Curran Associates Inc. ISBN 979-8-3313-1438-5. 4, 5

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 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*, pp. 3248–3256, July 2024. URL https://openreview.net/forum?id=xnpEpIqjUU. 2

Sepidehsadat Sepid Hossieni, Mohammad Amin Shabani, Saghar Irandoust, and Yasutaka Furukawa. Puzzlefusion: Unleashing the power of diffusion models for spatial puzzle solving. *Advances in Neural Information Processing Systems*, 36:9574–9597, 2023. 6

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, 2020. 1, 2

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*, volume 39, pp. 17323–17332, 2025. 2, 3, 4, 6, 7

Graziella Laignel, Nicolas Pozin, Xavier Geffrier, Loukas Delevaux, Florian Brun, and Bastien Dolla. Floor plan generation through a mixed constraint programming-genetic optimization approach. *Automation in Construction*, 123:103491, 2021. 1, 2

Hongjin Lian, Jian Ma, Hongjie Chen, Jia Li, Ruizhen Hu, Yu-Kun Lai, and Kun Li. Cg-floor: Centroid-guided diffusion for large-scale floorplan generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. 3

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t. 3

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024. 4

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z. 3

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11451–11461. IEEE, 2022. 6

Ziniu Luo and Weixin Huang. FloorplanGAN: Vector residential floorplan adversarial generation. *Automation in Construction*, 142:104470, October 2022. ISSN 0926-5805. doi: 10.1016/j.autcon.2022.104470. URL https://www.sciencedirect.com/science/article/pii/S0926580522003430. 1, 2

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106, 2021. 5

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*, pp. 162–177. Springer, 2020. 1, 2

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 *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 13627–13636. IEEE, 2021. 1, 2

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6

Chi-Han Peng, Yong-Liang Yang, and Peter Wonka. Computing layouts with deformable templates. *Association for Computing Machinery*, 33(4), July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601164. URL https://doi.org/10.1145/2601097.2601164. 1, 2

Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5466–5475. Ieee, 2023. 1, 2, 3, 4, 6

Kaede Shiohara and Toshihiko Yamasaki. Unified Vector Floorplan Generation via Markup Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39262–39271, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Shiohara_Unified_Vector_Floorplan_Generation_via_Markup_Representation_CVPR_2026_paper.html?utm_source=chatgpt.com. 3

George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36:3732–3784, 2023. 6

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. 1, 2

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*, pp. 60–75. Springer, 2024. 2, 6, 7, 8

Wenming Wu, Lubin Fan, Ligang Liu, and Peter Wonka. Miqp-based layout design for building interiors. *Computer Graphics Forum*, 37(2):511–521, 2018. doi: https://doi.org/10.1111/cgf.13380. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13380. 1, 2

Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. Data-driven interior plan generation for residential buildings. *Association for Computing Machinery*, 38(6), November 2019. ISSN 0730-0301. doi: 10.1145/3355089.3356556. URL https://doi.org/10.1145/3355089.3356556. 1, 2, 6

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025. 3

Biao Xiong, Zhen Peng, Ping Wang, Qiegen Liu, and Xian Zhong. TLC-Plan: A Two-Level Codebook Based Network for End-to-End Vector Floorplan Generation, February 2026. URL http://arxiv.org/abs/2602.07100. arXiv:2602.07100 [cs.CV]. 3