HouseTune: Two-Stage Floorplan Generation with LLM Assistance

Ziyang Zong[1], Guanying Chen[1], Zhaohuan Zhan[1], Fengcheng Yu[1], Guang Tan[1]*

[1]Shenzhen Campus of Sun Yat-sen University
zongzy@mail2.sysu.edu.cn, chenguanying@mail.sysu.edu.cn, zhan.z@smbu.edu.cn,
fyu54107@usc.edu, tanguang@mail.sysu.edu.cn

Abstract

This paper proposes a two-stage text-to-floorplan generation framework that combines the reasoning capability of Large Language Models (LLMs) with the generative power of diffusion models. In the first stage, we leverage a Chain-of-Thought (CoT) prompting strategy to guide an LLM in generating an initial layout, Layout-Init, from natural language descriptions, which ensures a user-friendly and intuitive design process. However, Layout-Init may lack precise geometric alignment and fine-grained structural details due to the inherent limitations of LLMs. To address this, in the second stage we propose a Dual-Noise Prior-Preserved Diffusion (DNPP-Diffusion) model to refine Layout-Init into a final floorplan that better adheres to physical constraints and user requirements. By combining LLMs and a dedicated refining model, our approach is able to generate high-quality floorplans without requiring large-scale domain-specific training data. Experimental results demonstrate its advantages in comparison with state of the art methods, and validate its effectiveness in home design applications.

Code — https://github.com/NatalieZZY/HouseTune

Introduction

In architectural design, creating floorplans that meet user requirements remains a challenge. Traditional design methods not only rely on specialized knowledge but also require designers to make iterative adjustments, which makes personalized design difficult. Learning-based models (Murali et al. 2017; Sun et al. 2022; Luo and Huang 2022) have made efforts to improve the accuracy and efficiency of the task. Nevertheless, existing approaches have yet to achieve a level of user-friendliness and accuracy that enables ready adoption by ordinary users.

Existing solutions (Nauata et al. 2021; Shabani, Hosseini, and Furukawa 2023; Zeng et al. 2024) typically treat the floorplan creation task as a conditional generation problem. The conditions are expressed by a bubble diagram, where rooms are represented as nodes (or "bubbles"), and doors as edges specifying the spatial relationships between rooms, as shown in Figure 1(a). Unfortunately, specifying a consistent graph structure can be too demanding for novice users who wish to explore designing. A more user-friendly interface would allow users to express their needs in natural language. For example, the user could simply specify *"I need a house with three bedrooms, a living room, a bathroom, a kitchen, and a balcony adjacent to the living room."* This text-to-floorplan paradigm makes the design process more intuitive for non-expert users.

A solution to this requirement involves training a generative model capable of directly mapping textual descriptions to floorplans, as depicted in Figure 1(b). The Tell2Design method (Leng et al. 2023) addresses this task using a Sequence-to-Sequence approach, where the input text specifies the floorplan boundary and the exact geometry of each room. The input requirement places an even greater burden on users compared to working with bubble diagrams. In addition, the design limits the diversity of the results, which is often crucial for exploratory design.

We introduce HouseTune, a novel two-stage floorplan generation framework. In the first stage, we use a multimodal large language model (LLM) to generate an initial house layout, termed Layout-Init. In the second stage, a diffusion model is designed to refine this initial design into a more precise and reasonable final layout, referred to as Layout-Final. Figure 1(c) shows this two-stage process.

Generating an initial house layout using an LLM is not trivial, as LLMs often fail to satisfy exact numeric and geometric constraints. To address this, we design a Chain-of-Thought (CoT) prompting strategy (Wei et al. 2022; Zhang et al. 2022). This approach ensures the generated layout aligns with the user's core requirements, such as the number of rooms, room types, and approximate arrangements.

The initial layouts produced at this stage typically exhibit structural imperfections in object sizing and alignment, due to the inherent limitations of LLMs in handling intricate geometric details. To address this, we further propose a novel *Dual-Noise Prior-Preserved Diffusion Model* (DNPP-Diffusion). The proposed model achieves three key innovations: (1) integrating the LLM-generated initial layout as a strong prior throughout the noise modeling process; (2) developing a hybrid noise strategy to balance generation diversity and prior preservation; and (3) designing a staged denoising mechanism that preserves the prior. This approach effectively mitigates early-stage mode collapse during training while preserving the Layout-Init prior.

*Corresponding authors.

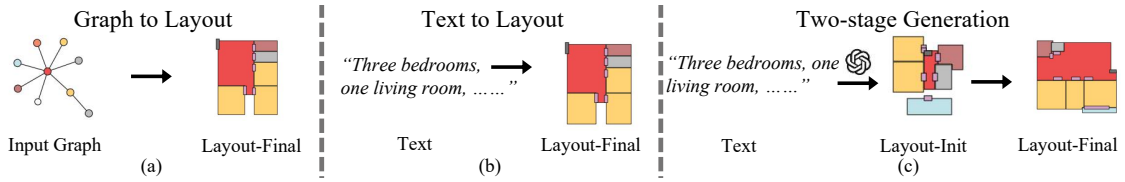

Figure 1: Comparison of different floorplan generation pipelines. (a) Graph-to-floorplan approach (e.g., HouseDiffusion), where rooms are represented as nodes and doors as edges, forming a graph of spatial relationships; (b) Text-to-floorplan approach, which directly maps a textual description to a house layout; (c) Our two-stage pipeline, where an LLM is used to generate an initial layout, Layout-Init, according to the user's textual specification. The initial solution serves as a prior for generating the final layout, Layout-Final, through our Dual-Noise Prior-Preserved diffusion model.

Compared with state of the art methods, our approach is able to generate higher quality floorplans from free textual specification, without requiring extra domain-specific training data. We validate our approach on the RPLAN dataset, in comparison with state-of-the-art methods. For instance, in comparison with HouseDiffusion, our method achieves the best performance across all metrics, and in particular, with a 28% improvement in diversity. To summarize, this paper makes the following contributions:

- We propose a novel two-stage floorplan generation framework that leverages the initial layouts generated by LLMs as the prior to generate the final floorplans.
- We develop a prompt design based on the Chain-of-Thought technique, successfully guiding the LLM to produce structured and coherent initial house layouts.
- We propose a DNPP-Diffusion framework that integrates Layout-Init priors throughout both noise injection and denoising phases, employing hybrid noise scheduling and iterative prior refinement to maintain structural integrity while enhancing layout diversity and consistency.

Related Work

Floorplan Generation. In the field of building design, generating high-quality floorplans has been an important research direction (Hendrikx et al. 2013; Wu et al. 2018; Hu et al. 2020; Müller et al. 2006; Peng, Yang, and Wonka 2014; Sun et al. 2022). Nauata et al. (Nauata et al. 2020) proposed House-GAN, a method based on Generative Adversarial Networks that achieves end-to-end automated floorplan generation. Nauata et al. (Nauata et al. 2021) further proposed House-GAN++, which improved the original GAN structure by addressing the problems of missing doors. Upadhyay et al. (Upadhyay et al. 2022) expanded on this by considering user inputs in the form of boundaries, room types, and spatial relationships. Hu et al. (Hu et al. 2020) presented Graph2Plan, which retrieves a set of floorplans with their associated layout graphs from a database, allowing users to specify room counts and other layout constraints. Shabani et al. (Shabani, Hosseini, and Furukawa 2023) introduced bubble diagrams as constraints and used diffusion to generate floorplans. Chen et al. (Chen, Deng, and Furukawa 2024) transformed visual sensor data into polygonal shapes with Diffusion Models. Su et al. (Su et al. 2024) developed a bi-directional structure of "corruption and denoise" approach to learn topology graphs. Zeng et al. (Zeng et al. 2024) proposed a multi-conditional two-stage generation model, which allows human designers to intervene and enhance controllability based on the denoising diffusion model. Leng et al. (Leng et al. 2023) introduced the Tell2Design dataset, which paired with natural language descriptions, in support of floorplan generation. HOLODECK (Yang et al. 2024b) explored the use of LLMs to generate floorplans populated with various objects, emphasizing the consistency of the environments rather than dealing with complex floorplans.

Conditional Diffusion. Conditional diffusion models are a subset of diffusion models (Cao et al. 2024; Sohl-Dickstein et al. 2015; Yang et al. 2023) where the generation is conditioned on specific input data, such as labels, images, or text, allowing more control over the output (Zhang, Rao, and Agrawala 2023; Chen, Zhang, and Hinton 2022; Ho et al. 2022). Expanding on the success of diffusion models (Sohl-Dickstein et al. 2015), Ho et al. proposed DDPMs (Song, Meng, and Ermon 2020), introducing constraints within the diffusion process to guide generation, which led to significant significant improvements. Yang et al. (Yang et al. 2024a) improved diffuse image synthesis based on context prediction. Yang et al. (Yang and Mandt 2024) proposed an end-to-end lossy image compression framework using conditional diffusion models to refine the source image. Recently, Khan et al. (Khan, Chen, and Schmid 2025) proposed a method guides the denoising process, which enables seamless generation of compositional objects with coherent backgrounds while permitting refinement of inaccurate priors.

Method

HouseTune uses a two-stage approach to generate house layouts. By breaking the generation pipeline into two stages, HouseTune manages to exploit the power of LLMs in interpreting user demands and generating approximate layouts with their common knowledge and reasoning ability. Further refinement is designed to enhance the layout's quality. Figure 2 depicts the training and testing processes of our method, in which Layout-Init plays a pivotal role.

Layout-Init Generation

Natural language descriptions alone may lack the detailed context necessary for accurate reasoning about specific house layouts (Mann et al. 2020; Min et al. 2022). To address this limitation, we enhance the LLM's reasoning capability

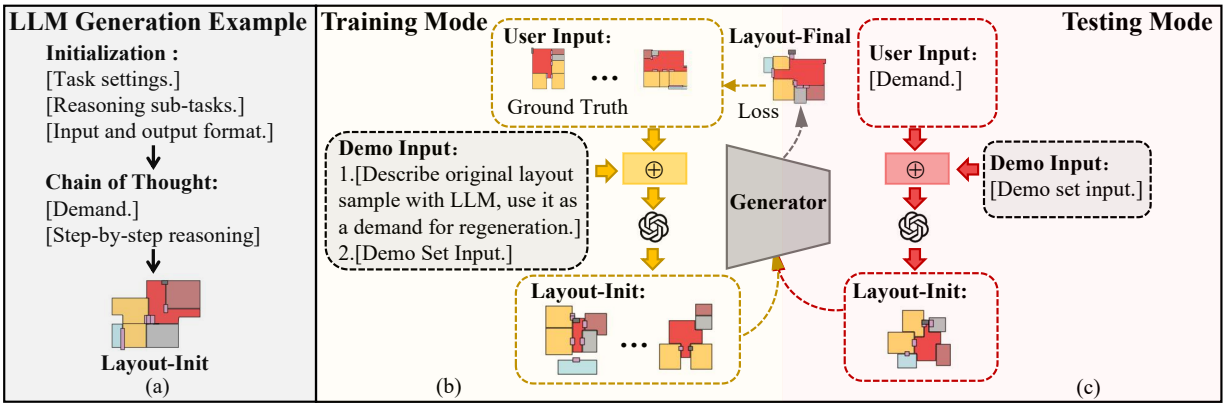

Figure 2: Training and testing processes of our method. (a) An example of LLM generating a Layout-Init according to user demands. (b) Given a house layout sample, we use the LLM to describe it. The textual description is used to mimic the user's demands. Using multiple examples as in (a) as demos, we ask the LLM to generate a Layout-Init for each sample. These initial layouts serve as priors for the generator, which outputs Layouts-Final. (c) Given a textual description from the user, we again use the demos like (a) to obtain a Layout-Init, which goes through the diffusion model to generate Layout-Final.

by allowing it to reference multiple human-designed layout demos, following the Chain of Thought (CoT) approach. By demonstrating each step explicitly, including room selection, placement, and sizing decisions, CoT helps the LLM break down complex layout generation tasks into manageable, step-by-step inferences.

The demo set E for the prompt consists of pairs of language description e_d and corresponding Layout-Init e_i:

$$E = \{(e_i, e_d)_1, (e_i, e_d)_2, ..., (e_i, e_d)_n\}. \quad (1)$$

This demo set essentially encompasses diverse examples with varying house layout designs. Figure 2(a) shows the reasoning process of an LLM generation example.

Figures 2 (b), (c) illustrate the training and testing pipelines of our approach. In the testing mode, we input user prompts expressed in natural language and the demo set E into the LLM to generate the initial layout. The output of the LLM is structured in JSON format.

The training process requires establishing a mapping from ground-truth layouts to Layout-Init, which are further used as priors for generating Layout-Final. Given a house layout sample from the dataset, we first use the LLM to give a description of the layout, which mimics the user's demands. The description is then fed to the LLM, which generates Layout-Init using the CoT-based prompt.

Figure 3 shows a sample of prompt design. The prompt used in our method consists of three parts: Initialization, Chain of Thought (CoT), and Layout-Init generation.

Initialization. Initialization introduces task settings, input and output format specifications, and reasoning subtasks. Specifically, the task utilizes the role-prompting technique (John 2023) to instruct the LLM to play the role of a designer: we present specific tasks to the LLM, prompting it to explain, step by step, the reasoning behind the design of each house location. An example prompt illustrating this setup is provided in see Figure 3 (a).

Chain of thought. In each separate reasoning, the LLM needs to determine the current layout, then randomly select the next room to be generated, and infer the information for the next room based on the existing rooms. The emphasis is on explaining the rationale behind these decisions. Figure 3 (a) presents an example that shows how to guide the model through each step of the inference process, including room selection, sizing, and positioning within the layout.

Layout-Init generation. In this step, demos are input into the LLM to prompt the generation of the house layout. The user provides natural language input, instructs the LLM to perform a step-by-step reasoning process. The final output is structured in the JSON format.

Dual-Noise Prior-Preserved Diffusion

We propose to incorporate the Layout-Init prior into the denoising. Figure 4 presents the specific training method.

Unlike conventional conditional diffusion models, which apply conditions only during the noise decoding stage, our framework processes two parallel inputs in both the noise introduction and decoding phases: ground truth layout x_0^g and Layout-Init prior x_0^p. We synchronously corrupt both inputs with Gaussian noise ϵ_g and ϵ_p at training time. By embedding Layout-Init as a prior from the outset and injecting Gaussian noise into it, the model receives explicit generation guidance during the early noise accumulation phase. This co-noising strategy ensures generation flexibility and prevents early model rigidification caused by priors.

Specifically, we employ a hybrid generation strategy: the initial input combines prior noise at timestep t_s and pure noise at timestep T. However, inconsistencies in noise levels between these different phases during denoising can lead to inaccurate noise predictions, potentially damaging the prior structure. To address this, we implement a phased hybrid denoising strategy in which, during the initial T to t_s timesteps, the prior undergoes synchronized denoising but is reverted to its original state x_0^p after each denoising operation. This protocol ensures noise-level synchronization between the prior and ground truth at timestep t_s, while keeping the prior in

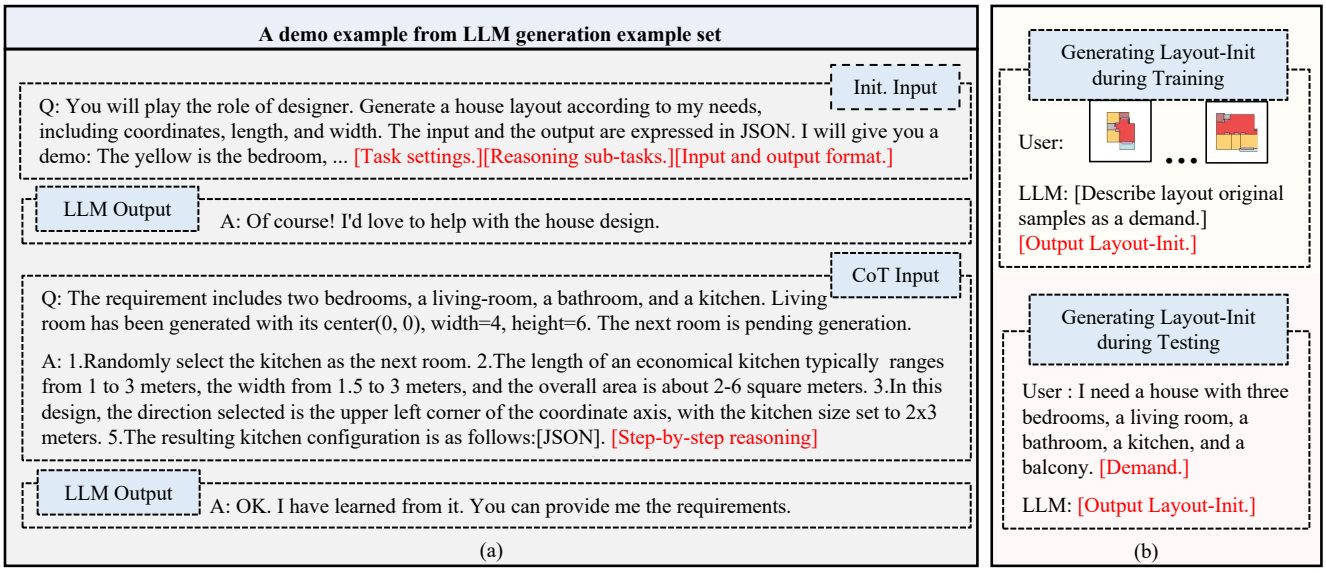

Figure 3: CoT-based prompting enables generation of Layout-Init and its function in training and testing. (a) An example showing how to interact with the LLM to obtain Layout-Init. The Initialization section defines the LLM's role as a house designer, and standardizes output format; the Chain of Thought section directs the LLM to create a house layout step by step, ensuring reasonable room placement and sizing. (b) Invoking the Layout-Init generation during training and testing.

its original state throughout the early denoising phase of ground truth data. Consequently, the persistent undenoised prior provides explicit guidance during the initial generation phase, while the progressive denoising of the prior in later stages enhances output flexibility and diversity.

The Diffusion Model Building upon the general framework of diffusion models for related tasks (Shabani, Hosseini, and Furukawa 2023; Zeng et al. 2024), our work aims to address data source issues while enhancing the reasoning capabilities for generation and prior guidance.

Diffusion models progressively refine Gaussian noise x_T^g over T, with a prior Layout-Init x_T^p. The forward process takes an initial sample x_0^g and a prior Layout-Init x_0^p, generating a noisy sample x_t^g by sampling Gaussian noise $\epsilon_g \sim N(0,1)$ for the data sample and a noisy sample x_t^p at time step t by sampling Gaussian noise $\epsilon_p \sim N(0,1)$ for the prior:

$$x_t = \sqrt{\alpha_t} * (x_0^g + x_0^p) + \sqrt{1-\alpha_t} * (\epsilon_g + \epsilon_p). \quad (2)$$

α_t represents the noise schedule, controlling noise intensity changes from 1 to 0 at each time step t.

The reverse denoising process starts with a fully noise-added sample x_T^g and gradually removes noise to produce a sample that matches the target data distribution x_0^g. This reverse process is typically implemented as an iterative process, where each time step depends on the output of the previous step. Our method incorporates a prior x_0^p for the generation. Rather than starting from pure Gaussian noise at the final timestep $t_s = T$, we initiate the algorithm with the prior x_0^p and use a noised prior at some intermediate step $t_s < T$, providing a stronger starting point for generation.

The reverse process updates the state of x_t^g each step to remove noise and gradually guide the generated image toward the structure of the prior x_t^p:

$$x_{t-1} = \begin{cases} \sqrt{\alpha_{t-1}} * (\hat{x}_0^g + \hat{x}_0^p) + \sqrt{1-\alpha_{t-1}} * \epsilon_g & t > t_s \\ \sqrt{\alpha_{t-1}} * (\hat{x}_0^g + \hat{x}_0^{pn}) + \sqrt{1-\alpha_{t-1}} * (\epsilon_g + \epsilon_p) & t <= t_s \end{cases} \quad (3)$$

α_{t-1} is the noise level parameter at time step $t-1$. ϵ is the residual noise predicted by the model.

By iterating the update formula from time step T to 1, the reverse process progressively removes noise, bringing x_t^g closer to a clean sample x_0^g. At each time step, the prior x_t^p continues to guide the generation.

Experiments

Datasets and implementation details

Datasets. We use the public dataset RPLAN (Wu et al. 2019) for experiments. RPLAN represents the largest dataset for floorplan with over 80K images. We divide RPLAN into five groups (5, 6, 7, 8 rooms) according to the number of rooms for cross-validation. Ablation experiments were conducted on the 6-room task. To create paired groups for training, we extracted 10K initial solutions from the LLM using the prompting method. GPT-4o was used as the LLM in ours.

Metrics. Following prior work (Nauata et al. 2020), we employ Realism, FID, and Compatibility as evaluation metrics. Realism is an estimate based on a user survey, considering the user's subjective experience. Diversity is measured by the FID score, defined as the Fréchet Inception Distance(Heusel et al. 2017) between the two Gaussian distributions. The Compatibility score(Abu-Aisheh et al. 2015) is used to measure the graph editing distance between the generated layout and the ground truth. Macro IoU and Micro IoU are also introduced for comparison with the Text-to-Layout method Tell2Design (Leng et al. 2023).

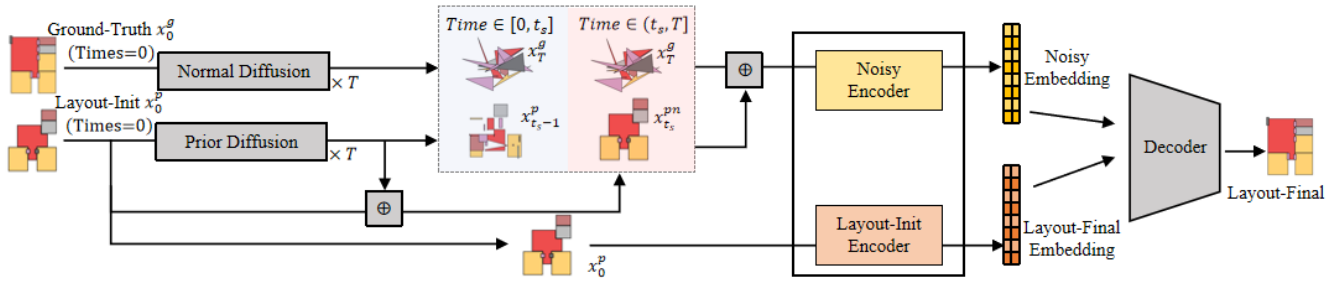

Figure 4: DNPP-Diffusion network for refining Layout-Init. The forward process takes the ground-truth house layout x_0^g and the Layout-Init x_0^p and adds Gaussian noises ϵ_g and ϵ_p to create a noisy house layout x_T. The reverse process takes a noisy house layout at time t and the Layout-Init as the priors with hybrid denoising strategy. Two encoders are used to encode and obtain the latent representations for x_t and x^p.

Method	Realism ↑	Diversity ↓				Compatibility ↓				Mic IoU ↑	Mac IoU ↑
Task	6	5	6	7	8	5	6	7	8	-	-
House-GAN(2020)	-0.95	37.5	41.0	32.9	66.4	2.5	2.4	3.2	5.3	-	-
House-GAN++(2021)	-0.52	30.4	37.6	27.3	32.9	1.9	2.2	2.4	3.9	-	-
HouseDiffusion(2023)	-0.19	11.2	10.3	10.4	9.5	1.5	1.2	1.7	2.5	-	-
PuzzleFusion(2024)	-	10.55				0.97				-	-
Tell2Design (2023)	-1.03	42.74				-				42.93%	38.48%
Tell2Design* (2023)	-	-				-				9.13%	6.06%
Ours	**-0.03**	**8.6**	**7.5**	**8.1**	**9.0**	**0.24**	**0.25**	**0.28**	**0.32**	33.15%	32.83%

Table 1: Generation performance of different methods.

Experimental results

Table 1 presents the main results, where we reproduce the reported performance of existing methods for fair comparison. As shown in the table, our approach consistently outperforms prior methods across all evaluation metrics. Specifically, compared to HouseDiffusion, our method improves diversity by 28% and compatibility by 79%, demonstrating its effectiveness in generating flexible and spatially efficient layouts. Tell2Design is evaluated using the IoU metric, with 54.34% Micro IoU and 53.30% Macro IoU. IoU results exhibit lower values than Tell2Design due to weaker supervision, but our method achieves much higher diversity (+79.88%) and user preference. Table 1 shows that under a comparable setup Tell2Design* (*Training on artificial instructions only*), HouseTune outperforms Tell2Design by +71.85% (Micro IoU) and +88.61% (Macro IoU). Tell2Design employs an autoregressive Seq2Seq framework, in which generation is strictly dependent on sequence order. This results in high IoU scores but significantly limits its diversity (see Fig.5, as it demands a large amount of labeled data to capture layout variations. In contrast, our non-autoregressive approach inherently mitigates these limitations, leading to a 79.88% improvement in diversity compared to Tell2Design. Furthermore, Tell2Design relies on a costly training process involving artificial pre-training followed by fine-tuning on human annotations. In contrast, our LLM-driven method eliminates the need for manual annotation while still achieving a higher user absolute preference score (-0.03 v.s. -1.03). For realism evaluation, we follow the same procedure as House-GAN++ to ensure comparability. User surveys indicate that 65% of participants perceive generated layouts as comparable to the ground truth.

Method	Diversity ↓	Compatibility ↓
1-stage: Text-to-Layout	109.5	10.6
2-stage: HouseTune	**7.5**	**0.25**

Table 2: Comparison between one-stage and two-stage methods.

Figure 5 presents a visual comparison of layouts generated by HouseTune, Tell2Design, and HouseDiffusion. The layout descriptions for HouseTune, Tell2Design's annotations, and HouseDiffusion's graphs are all derived from the reference samples shown in Fig.5(a). Due to the limitation of the plain Seq2Seq model, Tell2Design often produces inaccurate room counts, resulting in unrealistic layouts (see Fig.5(B)). In contrast, HouseTune and HouseDiffusion explicitly enforce these constraints, producing more structured and diverse layouts. HouseDiffusion, despite generally producing reasonable layouts, often exhibits misaligned and misplaced objects. This issue stems from its conditioning strategy, which applies constraints only during the denois-

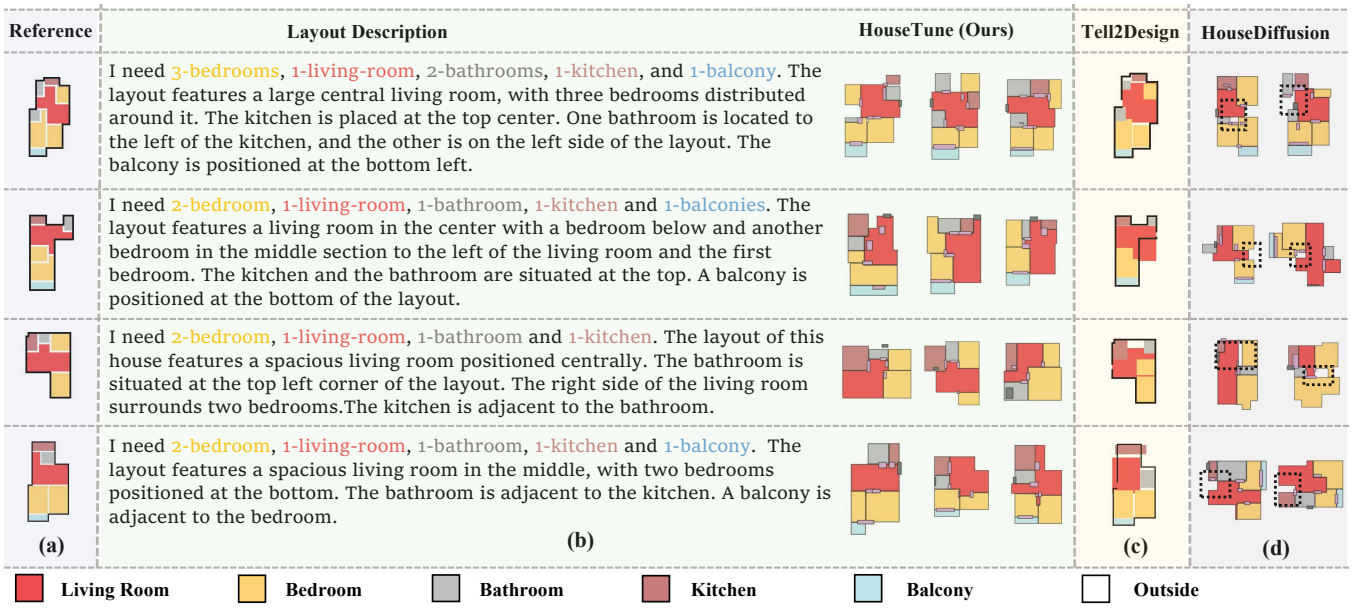

Figure 5: Generation samples from Tell2Design, HouseDiffusion and HouseTune. The results of HouseTune align well with user requirements in terms of room count and type, with reasonable and diverse room layout. The results produced by Tell2Design exhibit high similarity to the Reference layouts, showing limited diversity. Also, the number of generated houses deviates from the Reference, indicating inconsistencies in quantity. HouseDiffusion performs reasonably well; yet, in some cases, it generates gaps or holes within the house, as indicated by the dashed-rectangles, along with misaligned object placement, which makes the layout unrealistic.

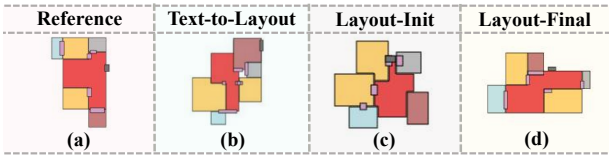

Figure 6: Effect of two stages in house layouts generation. (a) reference; (b) the one-stage method that maps text directly to layout; (c) LLM only, and (d) our method.

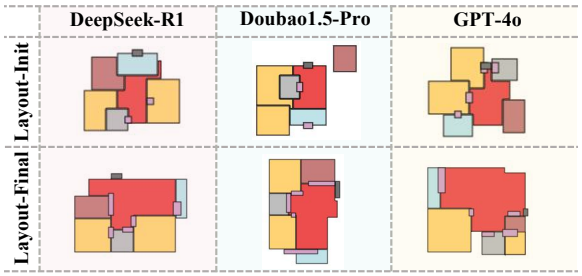

Figure 7: Generation results with different LLMs.

ing phase, making it less effective in maintaining spatial consistency throughout the generation process (see our ablation study for further analysis). HouseTune refines an initial LLM-generated layout using a diffusion model with a prior incorporated in both the noise addition and denoising stages.

Ablation study

One-stage v.s. two-stage Methods. We compare the one-stage Text-to-Layout approach using the text from LLM with our two-stage method. Figure 6 presents the comparison results under identical generation conditions. The one-stage method frequently produces overlapping rooms and fails to maintain a reasonable spatial distribution. This limitation arises because textual descriptions alone do not provide sufficient geometric constraints that enable the LLM to generate well-structured layouts. Additionally, Table 2 conduct a quantitative evaluation of both the one-stage and two-stage methods. Diversity indicates that the distribution of Text-to-Layout method result differs significantly from RPLAN. This is due to a mismatch between the text complexity and task, requiring extensive data support. Comparing the Layout-Init and Layout-Final, we can observe that former produces many overlapping rooms, as shown in Figure 6. These limitations suggest while the LLM-generated initial layout can capture the general structure of a house layout, it cannot enforce fine-grained spatial constraints.

Robustness and effectiveness of the prompts. Our method maintains consistent quality across both commercial and open-source LLMs, namely DeepSeek-R1 and Doubao-1.5Pro. As shown in Fig.7 and Table.3a, the generated JSON representations preserve structural integrity across different models, enabling seamless downstream diffusion refinement without additional fine-tuning. This shows that our prompting strategy exhibits low dependency on proprietary mod-

LLMs	Diversity ↓	IoU ↑
DeepSeek-R1	94.3	34.3%
DouBao-1.5Pro	95.6	30.2%
GPT-4o	95.1	33.8%

(a) Comparative analysis of different LLMs.

	Description	Diversity ↓	IoU ↑
P1	Task definition prompt	98.6	12.36%
P2	Common sense prompt	92.3	14.57%
P3	Think step by step	86.5	20.69%
P4	Predefined CoT prompt	75.2	21.41%

(b) Comparison of different prompting strategies.

Table 3: Prompt ablation study.

	Mic IoU ↑	Mac IoU ↑
Conditional Diffusion	21.84%	17.75%
DNPP-Diffusion	34.15%	32.83%

Table 4: DNPP-Diffusion v.s. Conditional Diffusion.

Method	Mic IoU ↑	Mac IoU ↑
Fixed	28.03	25.06
Weak to Strong	15.94	13.96
Strong to Weak	34.15	32.83

Table 5: Comparison of different Prior Preserving strategies.

Forward	Reverse	Diversity ↓	Mac IoU ↑
✓		8.57	9.95%
	✓	14.17	18.97%
✓	✓	7.47	21.84%

(a) Effect of forward v.s. reverse conditioning.

Rate	Mac IoU ↑	Mic IoU ↑
1e-1	31.37%	28.46%
1e-2	30.12%	26.59%
1e-3	28.67%	25.43%

(b) Effect of different conditions on the generated results.

Table 6: Conditional diffusion ablation study.

els. Furthermore, Table 3b presents the ablation for different prompt designs. **P1** employs only the task definition prompt to generate semantic representations. **P2** incorporates a common-sense prompt, improving reasoning accuracy. **P3** adopts a basic Chain-of-Thought (CoT) prompt ("Let's think step by step" (Kojima et al. 2022)), further enhancing performance. **P4** introduces explicit reasoning chains, leading to additional improvements over P3. These results indicate that our tailored prompting strategy effectively enhances the quality of the Layout-init generation.

DNPP-Diffusion v.s. Conditional Diffusion. We analyze the impact of DNPP diffusion and conditional diffusion methods by comparing the Mic-IoU and Mac-IoU. As shown in Table 4, incorporating Layout-Init as the prior during the entire generation phase leads to better performance compared to applying conditional information only in the denoising stage. This is because the early accumulation of noise enables the model to establish prior-related features earlier, while the hybrid prior protection strategy prevents the prior from being corrupted by noise prematurely.

Comparison of different prior preserving strategies. Table 5 compares the impact of different prior preserving strategies. For *Fixed*, we fix the strength of prior preserving, meaning that the prior and denoising prior are used in a fixed ratio throughout the denoising process. With this setup, the performance of IoU is mediocre. For *Weak to Strong* setting, the strength of the prior preserving varies with the time step, being weak in the early stage and becoming stronger later on. This approach denoises the prior too early, destroying the integrity of the early prior and resulting in limited constraints on the generation by the prior. Conversely, for *Strong to Weak*, the prior preserving is strong early on but weak later, providing strong guidance during the early denoising stages.

Conditional diffusion ablation study. Our approach integrates conditional constraints in both the forward and reverse processes. To evaluate the effectiveness of this design, we analyze the impact of each constraint by independently varying their proportions. As shown in Table 6a, incorporating conditional information during the noise addition phase leads to better performance compared to applying it only during the denoising phase. This is because early integration of conditions allows the model to establish condition-related features at an earlier stage. A single-stage conditioning approach may result in weaker constraints, leading to errors in room sizes and spatial relationships. The best performance is achieved when conditions are applied in both phases. Furthermore, we examined the effect of different conditional participation ratios on generation quality, as presented in Table 6b. As the conditional ratio increases, model performance gradually declines, suggesting that an excessive amount of conditional information causes the model to diverge from the true data distribution.

Conclusion

This paper proposes a two-stage text-to-floorplan generation model, enabling user-friendly house layout generation. This work guides the reasoning of the LLM through Chain-of-Thought prompting to generate an initial layout based on user requirements. The initial layout is then refined using a diffusion model to produce the final house layout. Experimental results show that our method achieves state-of-the-art performance. Given the potential of LLMs, we expect that our solution be extended to the generation of other complex architectural designs such as shopping malls and office buildings. This will be a subject of our future study.

Acknowledgments

This work was supported in part by Shenzhen Basic Research Fund under grant JCYJ20241202130025030.

References

Abu-Aisheh, Z.; Raveaux, R.; Ramel, J.-Y.; and Martineau, P. 2015. An exact graph edit distance algorithm for solving pattern recognition problems. In *4th International Conference on Pattern Recognition Applications and Methods 2015*.

Cao, H.; Tan, C.; Gao, Z.; Xu, Y.; Chen, G.; Heng, P.-A.; and Li, S. Z. 2024. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*.

Chen, J.; Deng, R.; and Furukawa, Y. 2024. Polydiffuse: Polygonal shape reconstruction via guided set diffusion models. *Advances in Neural Information Processing Systems*, 36.

Chen, T.; Zhang, R.; and Hinton, G. 2022. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*.

Hendrikx, M.; Meijer, S.; Van Der Velden, J.; and Iosup, A. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1): 1–22.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022. Video diffusion models. *Advances in Neural Information Processing Systems*, 35: 8633–8646.

Hossieni, S. S.; Shabani, M. A.; Irandoust, S.; and Furukawa, Y. 2024. PuzzleFusion: unleashing the power of diffusion models for spatial puzzle solving. *Advances in Neural Information Processing Systems*, 36.

Hu, R.; Huang, Z.; Tang, Y.; Van Kaick, O.; Zhang, H.; and Huang, H. 2020. Graph2plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics (TOG)*, 39(4): 118–1.

John, I. 2023. The art of asking chatgpt for high-quality answers. *Nzunda Technologies Ltd*.

Khan, Z.; Chen, S.; and Schmid, C. 2025. ComposeAnything: Composite Object Priors for Text-to-Image Generation. *arXiv preprint arXiv:2505.24086*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Leng, S.; Zhou, Y.; Dupty, M. H.; Lee, W. S.; Joyce, S. C.; and Lu, W. 2023. Tell2design: A dataset for language-guided floor plan generation. *arXiv preprint arXiv:2311.15941*.

Luo, Z.; and Huang, W. 2022. FloorplanGAN: Vector residential floorplan adversarial generation. *Automation in Construction*, 142: 104470.

Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1.

Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; and Zettlemoyer, L. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Müller, P.; Wonka, P.; Haegler, S.; Ulmer, A.; and Van Gool, L. 2006. Procedural modeling of buildings. In *ACM SIGGRAPH 2006 Papers*, 614–623.

Murali, S.; Speciale, P.; Oswald, M. R.; and Pollefeys, M. 2017. Indoor Scan2BIM: Building information models of house interiors. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6126–6133. IEEE.

Nauata, N.; Chang, K.-H.; Cheng, C.-Y.; Mori, G.; and Furukawa, Y. 2020. House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 162–177. Springer.

Nauata, N.; Hosseini, S.; Chang, K.-H.; Chu, H.; Cheng, C.-Y.; and Furukawa, Y. 2021. 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*, 13632–13641.

Peng, C.-H.; Yang, Y.-L.; and Wonka, P. 2014. Computing layouts with deformable templates. *ACM Transactions on Graphics (TOG)*, 33(4): 1–11.

Shabani, M. A.; Hosseini, S.; and Furukawa, Y. 2023. 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*, 5466–5475.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Su, P.; Lu, W.; Chen, J.; and Hong, S. 2024. Floor plan graph learning for generative design of residential buildings: a discrete denoising diffusion model. *Building Research & Information*, 52(6): 627–643.

Sun, J.; Wu, W.; Liu, L.; Min, W.; Zhang, G.; and Zheng, L. 2022. Wallplan: synthesizing floorplans by learning to generate wall graphs. *ACM Transactions on Graphics (TOG)*, 41(4): 1–14.

Upadhyay, A.; Dubey, A.; Arora, V.; Kuriakose, S. M.; and Agarawal, S. 2022. Flnet: graph constrained floor layout generation. In *2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 1–6. IEEE.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wu, W.; Fu, X.-M.; Tang, R.; Wang, Y.; Qi, Y.-H.; and Liu, L. 2019. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG)*, 38(6): 1–12.

Wu, Y.; Wu, Y.; Gkioxari, G.; and Tian, Y. 2018. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*.

Yang, L.; Liu, J.; Hong, S.; Zhang, Z.; Huang, Z.; Cai, Z.; Zhang, W.; and Cui, B. 2024a. Improving diffusion-based image synthesis with context prediction. *Advances in Neural Information Processing Systems*, 36.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2023. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4): 1–39.

Yang, R.; and Mandt, S. 2024. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36.

Yang, Y.; Sun, F.-Y.; Weihs, L.; VanderBilt, E.; Herrasti, A.; Han, W.; Wu, J.; Haber, N.; Krishna, R.; Liu, L.; et al. 2024b. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16227–16237.

Zeng, P.; Gao, W.; Yin, J.; Xu, P.; and Lu, S. 2024. Residential floor plans: Multi-conditional automatic generation using diffusion models. *Automation in Construction*, 162: 105374.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhang, Z.; Zhang, A.; Li, M.; and Smola, A. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.