CARD: Cross-modal Agent Framework for Generative and Editable Residential Design

Pengyu Zeng[1*], **Jun Yin**[1*], **Miao Zhang**[1], **Yuqin Dai**[1], **Jizhizi Li**[2], **Zhanxiang Jin**[1], and **Shuai Lu**[1†],

[1]Shenzhen International Graduate School, Tsinghua University
[2]The University of Sydney
Correspondence: shuai.lu@sz.tsinghua.edu.cn

Abstract

In recent years, architectural design automation has made significant progress, but the complexity of open-world environments continues to make residential design a challenging task, often requiring experienced architects to perform multiple iterations and human-computer interactions. Therefore, assisting ordinary users in navigating these complex environments to generate and edit residential design is crucial. In this paper, we present the CARD framework, which leverages a system of specialized cross-modal agents to adapt to complex open-world environments. The framework includes a point-based cross-modal information representation (CMI-P) that encodes the geometry and spatial relationships of residential rooms, a cross-modal residential generation model, supported by our customized Text2FloorEdit model, that acts as the lead designer to create standardized floor plans, and an embedded expert knowledge base for evaluating whether the designs meet user requirements and residential codes, providing feedback accordingly. Finally, a 3D rendering module assists users in visualizing and understanding the layout. CARD enables cross-modal residential generation from free-text input, empowering users to adapt to complex environments without requiring specialized expertise.

1 Introduction

With the advancement of modern technology, automated architectural design has garnered significant attention(Zeng et al., 2024; Luo and Huang, 2022; Zeng et al., 2025c), particularly in the realm of residential design, where the demand for efficiency, error reduction (Gao et al., 2021), and cost minimization is high (Gao et al., 2023). As the most common form of architecture, residential floor plan generation has become a focal point in research, attracting considerable interest from both academia and industry (Yin et al., 2025a; Zeng et al., 2025a; Lazić et al., 2021; Yin et al., 2025c; Zeng et al., 2025b).

However, the complexity of open-world environments makes the task of residential design complex, requiring professional expertise (Weber et al., 2022; Fan et al., 2023). Typically, homeowners provide specific requirements, which designers translate into 3D models. This process involves multiple iterations and collaborative revisions, consuming substantial human effort and increasing the complexity for average users to engage freely in the design process (Bo et al., 2022; Omar et al., 2016). Addressing this challenge calls for solutions that assist non-expert users in navigating complex environments for generating and editing residential designs at a low cost and with minimal expertise.

In this paper, we introduce CARD, a cross-modal agent-driven framework that leverages natural language input to generate and edit 3D residential designs. Similar to other LLM-based role-agent systems (Li et al., 2023b; Park et al., 2023), the framework includes multiple agents with specialized roles—including Product Manager (Demand), Lead Designer, Auditors (Residential Code and User Requirements), Assistant Designer, Product Manager (After-Sales), and 3D Modeler—designed to adapt to the complexity of open-world environments. Our system provides a low-threshold, cost-effective solution for editable and flexible residential design.

The framework utilizes natural language text for input and 3D representations for output to ensure usability for ordinary users. Although developing such a language-driven tool presents challenges (Zhang and El-Gohary, 2022), advancements in deep learning, particularly in multimodal modeling, have made this approach feasible (Jiang et al., 2023; Zeng et al., 2022, 2023; Liu et al., 2022). Nevertheless, several hurdles persist. First, collecting

[*]The authors contributed equally to this work.
[†]Shuai Lu is the corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 9304–9319

large-scale multimodal data for training, especially for both text and residential layout editing, is challenging (Rahate et al., 2022). Second, multimodal models are more expensive to train compared to single-mode models (Huang et al., 2021). Furthermore, existing residential design models often rely on rigid, homogeneous input formats, limiting personalization and resulting in inflexible designs. Finally, we introduce a novel modal decomposition mechanism to bridge the gap between text and single-image generative models. This mechanism facilitates cost-efficient cross-modal generation without multimodal datasets by leveraging a new point-based representation of cross-modal information, termed CMI-P, and can couple the geometric shapes and spatial positions of each room in the residential.

CARD combines residential openness design with cross-modal agents, including multiple language agents and one image agent, using a modular approach to process information, generate, evaluate, make decisions, summarizes and edit residential designs, and continuously learn from interactions. To conduct more precise and standardized residential design generation and editing, we embed existing residential specification documents and user needs in the evaluation, and some different agents can exchange information to avoid information bias. In addition, to solve the problem of biased residential vector information generated by agents, a cross-modal housing generation model is designed to normalize the information generated by the agent, thus avoiding the large deviation problem caused by long-term multi-agent interaction. This design provides a good foundation for multi-round interaction of housing design.

To further verify the effectiveness, adaptability to complex environments, and interactive capabilities of our approach, we conducted extensive experiments and evaluated it using comprehensive metrics, as well as a study involving experts and ordinary users. The results show that our approach outperforms others in many aspects and lays a solid foundation for future research.

2 Related works

2.1 Agent Framework

Recent research has focused on enhancing the role-playing and interaction abilities of large language models (LLMs) as agents, improving their capacity to engage with users and act with greater self-awareness (Wang et al., 2023b; Shao et al., 2023; Shanahan et al., 2023; Li et al., 2023a). Other works explore multi-agent interactions, including collaboration in task completion (Li et al., 2023b; Chen et al., 2023; Qian et al., 2023), simulating daily activities (Lin et al., 2023; Park et al., 2023), and facilitating debates (Liang et al., 2023; Du et al., 2023; Chan et al., 2023). Language agents have also been applied in open-world settings, such as text-based games (Côté et al., 2019; Hausknecht et al., 2020) and exploration tasks in Minecraft (Wang et al., 2023a; Zhu et al., 2023).

2.2 Residential Floor Plan Generation

As AI develops (Ma et al., 2025; Zhang et al., 2025; Yin et al., 2025b), approaches to residential floor plan generation have diversified. Recent approaches to residential floor plan generation typically fall into three categories: rule-based methods, GAN-based models, and graph-based techniques. These methods have made significant progress, but there are some limitations, such as the low quality of residential floor plans generated by GAN-based methods (Huang and Zheng, 2018), and the inputs for the type of graph are not conducive to comprehension and editing by the average user (Aalaei et al., 2023; Carta, 2022). Recently, text-based residential image generation models have emerged (Leng et al., 2023). However, these models require the construction of language libraries, while template-based text drivers are less flexible. In addition, current research on 3D residence focuses on generating 3D residence based on template-based text description (Chen et al., 2020), and placing 3D residential objects based on LLM (Feng et al., 2024; Yang et al., 2024). Due to the high cost of constructing a language library for residence generation (Huang et al., 2021), the task of residential design editing based on free-text is currently in a blank stage.

3 CARD

The CARD framework aims to facilitate the generation and editing of residential designs based on free-text input, enabling non-expert users to navigate the complexities of open-world environments. The framework structure, as illustrated in Fig. 1, involves multiple agents, which simulate the complex environments in residential design. These agents include Product Manager (Demand), Lead Designer, Auditors (Residential Code and User Re-

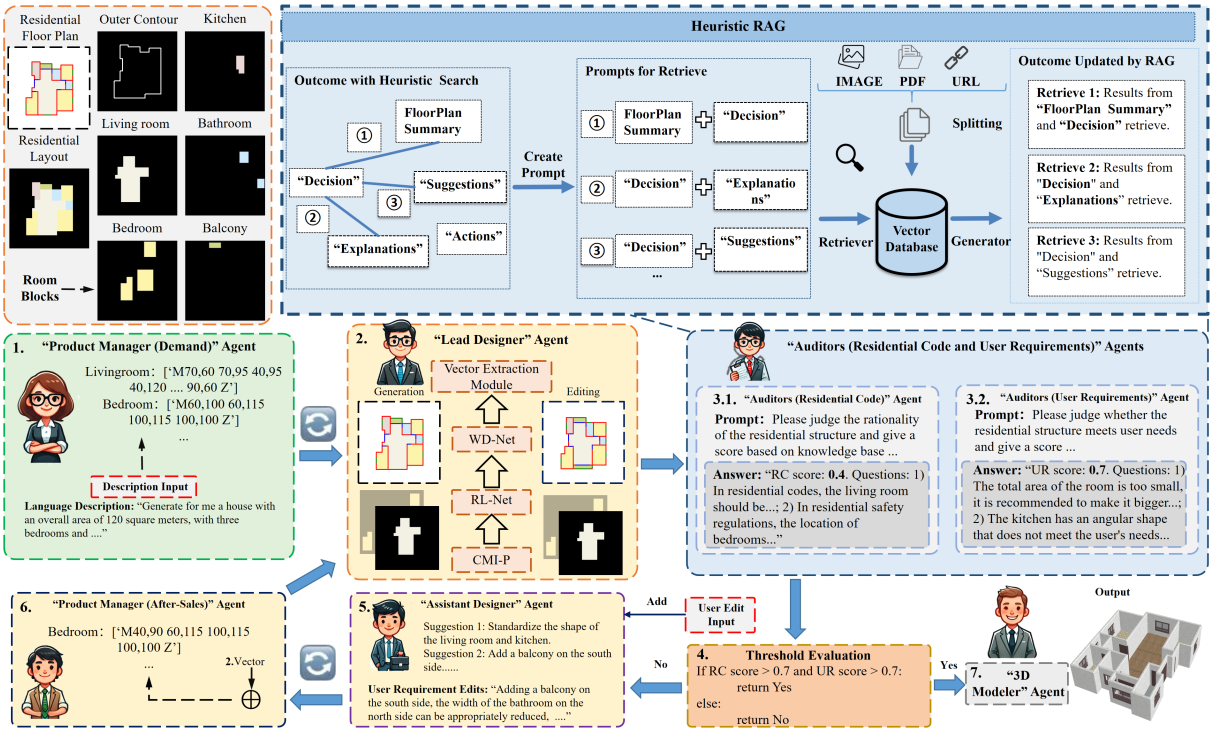

Figure 1: Overview of the CARD Architecture. The language agent is powered by GPT-4, while the "Lead Designer" agent leverages our custom-designed Text2FloorEdit model.

quirements), Assistant Designer, Product Manager (After-Sales), and 3D Modeler agents, each playing distinct roles throughout the design process.

3.1 Overall Framework

The process begins with user input, where users provide a free-text description of their envisioned residential design. This input serves as the foundation for the entire design process. The "Product Manager (Demand)" agent interprets the user's description and abstracts it into structured point-based information. This step ensures that the user's requirements are transformed into a format suitable for further processing. The abstracted point information is then passed to the "Lead Designer" agent, who employs a cross-modal residential generation model, Text2FloorEdit to create a residential floor plan that meets the specified requirements.

The generated residential floor plan is evaluated by two different Auditor agents. One auditor ensures that the design complies with residential codes, while the other auditor verifies that it aligns with the user's needs, scoring the outcome accordingly. If both auditors approve the design, the residential design is finalized and rendered in 3D. If not, the reasons for non-compliance are identified. In case of rejection, the issues found by the auditors, along with any user edits, are forwarded to the "Assistant Designer" agent. This agent summarizes the feedback and outputs a list of modification suggestions, ensuring that both compliance and user requirements are met.

The modification suggestions are then sent to the "Product Manager (After-Sales)" agent, who combines these suggestions with the vector information of the existing structure and outputs a revised set of structured point information. This step ensures that any changes are seamlessly incorporated into the overall design. The modified point information is returned to the "Lead Designer" agent for further adjustments. This iterative process continues until both Auditor agents approve the design. Finally, the design is clearly visualized for the user through a "3D Modeler" agent.

3.2 Initial Generation

Step 1: The "Product Manager (Demand)" agent first interprets the user's natural language input and outputs a set of structured, templated point-based requirements (see Appendix Fig. 5). The information flow of the initial generation phase is illustrated in Fig. 2a).

Step 2: We designed the Text2FloorEdit model to function as the "Lead Designer" agent, tasked with generating a practical and well-laid-out residential floor plan based on these preliminary re-

quirements. The Text2FloorEdit is composed of four main components: 1) CMI-P, 2) The Residential Layout Generation Network (RL-Net), which generates residential layouts to enhance model flexibility, 3) The Window, Door, and Wall Generation Network (WD-Net), which generates comprehensive residential floor plans, reducing model training costs, and 4) the Vector Extraction Module. First, the Information Conversion Module transforms the structured point-based information output by the "Product Manager (Demand)" agent into image-based information. The resulting image is processed by RL-Net to facilitate the generation of the residential layout, and the final residential floor plan is produced through WD-Net. Finally, the vector information is extracted.

CMI-P: Traditional graph-based language parsers struggle to capture room shapes (Aalaei et al., 2023; Carta, 2022). To address this, we developed CMI-P, a cross-modal representation that efficiently conveys input across both modalities, significantly reducing training costs. It converts point data into image data for generation and back into point data for editing. Unlike methods such as LayoutGPT(Feng et al., 2024) and Holodeck(Yang et al., 2024), which only attempted to use text coordinates to represent rectangular boxes, our CMI-P explores the text-based representation of more complex polygonal boxes, which better aligns with real-world residential rooms.

RL-Net: We utilize a diffusion model (Ho et al., 2020) as the foundation of RL-Net. First, we define the model's input formats, setting it to six-channel images representing the outer contour (white boundary), living room (gray block), bedroom (yellow block), bathroom (blue block), balcony (green block), and kitchen (pink block). To enhance the flexibility of the design, we adopt multiple formats for the input, as shown in Fig. 2 b). Specifically, when the right-hand image is zero-valued, it indicates a fixed size input (based on the left image's dimensions); conversely, when the left image is zero-valued, it denotes a random size input, allowing the model to infer dimensions based on context. Additionally, room shapes are represented either as polygons for precise geometry or rectangles for vague approximations. For each instance, one of the five input configurations described above is randomly selected and used as the input to RL-Net. The RL-Net output is a residential layout. To enhance feature relevance (Guo et al., 2022), we designed the Multi-Scale Fusion Redundant Attention (MFDA) module, as shown in Fig. 2b). We chose strip convolution for feature fusion, considering that most residential layouts are horizontal and vertical, in order to reduce training costs.

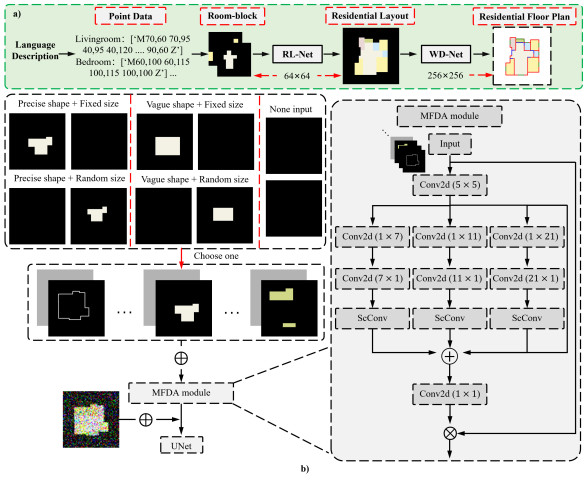

Figure 2: a) Information flow of the initial generation phase. b) 5 input formats and MFDA module.

WD-Net: To generate a comprehensive residential floor plan, WD-Net refines the output from RL-Net by adding doors, windows, and walls in their appropriate positions. Specifically, WD-Net takes the residential layout as input and produces a residential floor plan with doors, windows, and walls. Two primary challenges arise: 1) RL-Net is trained with low-resolution images (e.g., 64x64) to minimize training costs, but rendering these outputs in 3D poses difficulties. 2) Using higher-resolution images (e.g., 256x256) from datasets like RPLAN (Wu et al., 2019) improves edge recognition but significantly increases training costs. To balance these challenges, we introduce WD-Net with a resolution fine-tuning strategy. This process tackles two key subtasks: 1) capturing the spatial distribution of doors, windows, and walls, and 2) refining the edges of the floor plans.

We first pre-train the diffusion model using 64×64 resolution images to capture positional relationships efficiently. Afterward, we fine-tune this pre-trained model with 256×256 images to capture detailed edge features. This strategy significantly reduces the training time and overall costs while achieving higher-resolution outputs for more accurate 3D renderings.

Vector Extraction Module: This module extracts vectorized data from the generated residential floor plan. This is crucial for preserving the geometric details of the generated residential floor plan for

subsequent iterative editing. Image segmentation techniques are first used to detect different room types, which are then classified according to predefined room categories. Their boundary points are extracted and reconstructed into vector data. We also use the Douglas-Peucker algorithm (Douglas and Peucker, 1973) to ensure the accuracy of the extracted vector information.

3.3 Auditors Agent

Auditor (Residential Code): Auditor (Residential Code): We use a Retrieval-Augmented Generation (RAG) method (Lewis et al., 2020) to allow the auditor to retrieve relevant regulatory information from a pre-built database (e.g., building codes, safety regulations) and generate context-aware evaluations. This approach incorporates key standards such as the General Code for Fire Protection in Buildings (GB 55037-2022), the Code for Residential Buildings (GB 50368—2005), and the China Architectural Design Data Set, Volume 2, Residential (Third Edition) to ensure the design aligns with the latest architectural and safety regulations. **Step 1:** Query Construction: Once the initial residential floor plan is generated, the auditor formulates queries based on key design aspects that require verification (e.g., room types, functionality, location). These queries are sent to the RAG model for targeted information retrieval. **Step 2:** Retrieval from the Code Database: The RAG model retrieves relevant data from a residential building code database, including fire protection guidelines from GB 55037-2022 and residential standards from GB 50368—2005. The China Architectural Design Data Set serves as an additional resource to enhance design accuracy. **Step 3:** Contextual Validation: The retrieved codes are cross-referenced with the design elements. The auditor checks compliance by comparing parameters like room dimensions and spacing with the regulations, ensuring the design meets the required standards, such as fire protection and room size limits. **Step 4:** Decision: The results of the comparison determine whether the design complies with residential codes. If any violations are found, feedback is provided, highlighting necessary modifications. The design is refined iteratively until it fully meets the relevant standards.

Auditor (User Requirements): This process follows a similar RAG-based approach. Using the RAG model, the auditor retrieves information from a User Requirements Database, which is built from the user's initial input, interactive inputs during the editing process, and personalized case studies from previous users. The retrieved requirements and preferences are then compared with the generated design. This comparison can be expressed as a similarity or distance function, see Appendix B. In the final decision phase, judgments are made based on the similarity score. If the design is flagged as non-compliant, feedback is generated for further editing and revision.

3.4 3D Modeler Agent

To visualize residential floor plans, we developed a 3D residential renderer system. This renderer transforms a residential floor plan into a 3D residential design, allowing users to visualize the details and overall ideas of the floor plan from a spatial perspective. Details in Appendix C. To ensure a uniform visual experience, a virtual camera is placed above a specific corner of each rendered 3D residential design model. Besides, the viewing angle can be manually adjusted, allowing users to rotate and view the model from different directions. This part is an engineering development effort, and the code will be open-sourced for community use.

4 Experiment

4.1 Experimental Settings

Datasets: The "Lead Designer" Agent is a model designed by ourselves. We used the residential floor plan dataset generated by the RPLAN toolbox (Wu et al., 2019). To reduce the training overhead, we down-sampled the original 256×256 images to 64×64 resolution. The datasets were divided into training, and test sets containing 70126, and 11000 plan images, respectively. Given the limited number of residential design categories, we selected 300 - 1000 samples from the common 20 categories for the test set based on the category distribution.

Implementation Details: Our editable residential design generation task based on free-text has no comparable methodology. Therefore, simplified versions of our proposed network were compared.

We first compared the models based on different input conditions: 1) Non-text input: We chose HouseDiffusion(Shabani et al., 2023), House-GAN++(Nauata et al., 2021), Graph2Plan(Hu et al., 2020), Building Floor Plan (BFP)(Wan et al., 2022), and CycleGAN(Li, 2023) models, where we chose FID(Heusel et al., 2017), PSNR(Huynh-

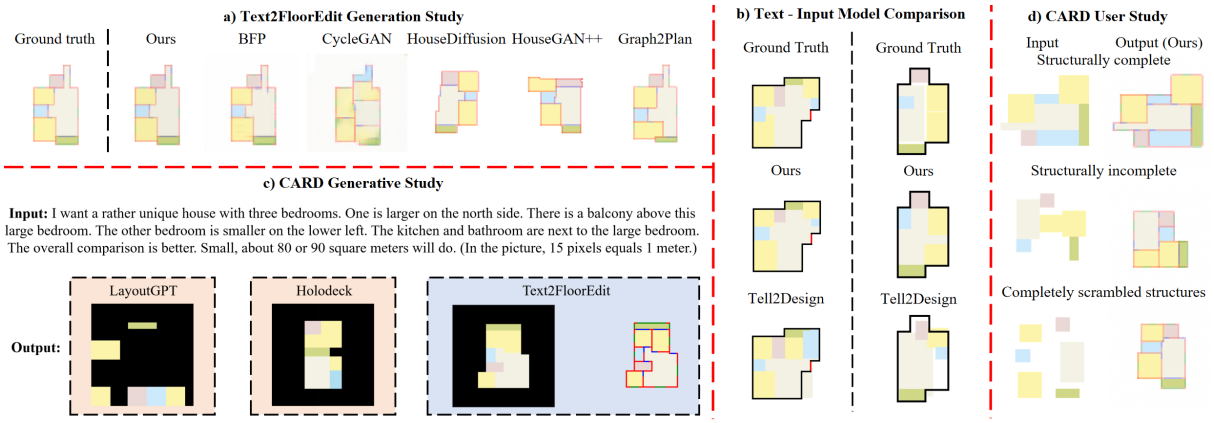

Figure 3: Qualitative analysis results of the model on multiple studies.

Thu and Ghanbari, 2008), and SSIM(Wang et al., 2004) metrics. 2) Template-text input: We chose Tell2Design(Leng et al., 2023) model, where we chose the FID, Micro IoU(Leng et al., 2023), and Macro IoU metrics(Leng et al., 2023). 3) Free-text input: We choose LayoutGPT(Feng et al., 2024) and Holodeck(Yang et al., 2024), where we chose FID, PSNR, and SSIM metrics. Since the Graph Edit Distance(GED)(Abu-Aisheh et al., 2015) metric is primarily used to assess the consistency of model-generated results based on graph input, and our model is based on room blocks or text input, we did not use the GED metric. Specifically, for our model, the inputs in the cases of non-text input and text input are six-channel room blocks and text input respectively. Additionally, to evaluate the adaptability of our architecture to complex environments, we provided a multi-round iteration example. Finally, we used experts and ordinary users to comprehensively evaluate the effectiveness of the model generation. We invited experts to compare and evaluate actual images with model-generated images. To ensure fairness, all the trainable models mentioned above are trained and tested on RPLAN. We unify the room colors in the generated results of each model and do not change the residential layout. We trained the model on a single NVIDIA A100 GPU with a batch size of 128.

4.2 Model Comparison Study

Non-text input: The first step is a qualitative comparison. As shown in Fig 3a), these results reveal that our model perfectly reproduces the semantic information and functional layout of the actual images. Housediffusion, House-GAN++ and Graph2Plan generate better structures using graph information, but are unable to specify room sizes and shapes and lack some flexibility. Moreover, the

Table 1: Test results of our model and various baseline models on the test set of the RPLAN dataset.

Method	FID ↓	PSNR ↑	SSIM ↑
CycleGAN	134.79	50.4	0.72
BFP	71.39	56.3	0.95
Graph2Plan	32.45	62.8	0.95
HouseGAN++	34.06	61.2	**0.99**
HouseDiffusion	28.72	61.7	**0.99**
Ours	**8.41**	**86.1**	**0.99**

BFP and CycleGAN models exhibit generation instability problems. In addition, we performed quantitative comparisons, as shown in Table 1. Compared with the Baseline model, our model performs optimally in FID, PSNR, and SSIM indicators.

Template-text input: To evaluate our text-based residential layout generation architecture, we compared it with state-of-the-art text-generated residential layout models, as shown in Fig. 3b). While other models are trained on multimodal datasets consisting of image-text pairs, our model is trained under zero-shot conditions without using any paired image-text data for supervision. Our model generates residential floor plans that do not exceed the overall contour boundary because of the format input of the outer contour. It also captures the spatial topological relationships and area characteristics of each room. In addition, in Table 2 quantitative results show that the FID, Micro IoU, and Macro IoU metrics of our model are 30.5%, 15.6%, and 6.0% higher, respectively, than those of the second-best model. The above results show that our model can perform flexible cross-modal generation with zero explicit multimodal training data, demonstrating its ability to generalize across modalities without paired examples.

Table 2: Quantitative results from the template-text input.

Method	FID ↓	Micro IoU ↑	Macro IoU ↑
Tell2Design	11.01	0.77	0.67
Our	**7.65**	**0.89**	**0.71**

Table 3: Quantitative results from the free-text input.

Method	FID ↓	PSNR ↑	SSIM ↑
LayoutGPT	30.8	58.4	0.95
Holodeck	21.2	60.9	**0.99**
Ours	**11.4**	**64.7**	**0.99**

Free-text input: To evaluate the superiority of our CARD architecture in generating residential designs, we compared our model with the LLM-based LayoutGPT and Holodeck models, which also utilize free-text descriptions. The qualitative results are shown in Fig. 3c). The experimental results indicate that although LayoutGPT and Holodeck primarily focus on the selection and placement of objects within the residence, their performance in generating residential layout is not as effective as our model. In addition, the quantitative results presented in Table 3 demonstrate that the residential designs generated by our model, based on free-text descriptions, have distinct advantages in meeting both residential specifications and user needs.

4.3 CARD Adaptability Study

To evaluate the adaptability of the CARD framework to complex open-world environments, we assessed the model from three aspects: generation accuracy, generation diversity, and iterative editing.

Generation Accuracy: We evaluated generation accuracy from two perspectives: text description and residential code compliance, as shown in Fig. 4a). Our model precisely captured free-text descriptions and generated reasonable residential designs. Even with strict requirements like "the balcony is located on the south side adjacent to the living room, with a depth greater than 1.5 meters," it produced accurate designs. Additionally, we set a similarity threshold of 0.7 for residential specifications, and the outputs exceeded this benchmark, highlighting the model's reliability.

Generation Diversity: To assess the impact of vague text descriptions, we conducted a generation diversity experiment, with qualitative results shown in Fig. 4b). For quantitative evaluation, we generated 1000 images for three case("One/Two/Three bedroom, one living room, one kitchen, one toilet, one balcony"). The results had 105, 192, and 161 residential layout types, distinguished by different graph features. Notably, only LayoutGPT and Holodeck can accept such free-text input, but these models are unable to generate doors and windows, making it impossible to calculate the number of residential layout types based on room connectivity features. Therefore, here we only perform quantitative evaluation for our model. The results show that when the text description was imprecise, our model generated various residential designs that adhered to the description, offering diverse outcomes for users to choose from. This demonstrates that even with imprecise free-text input, our model maintained a high level of generation quality. Notably, at each iteration, the "Lead Designer" agent can generate diverse residential floor plans, providing users with a variety of options to choose from.

Table 4: Human expert test results for our model vs. real images.

	Generation	Real images	All
Precision	54.4%	45.6%	100
Truth	50%	50%	100

User Iterative Editing Example: If the generated results do not fully meet the user's needs, our model allows for editing of the residential design in multiple ways. As shown in Fig. 4c), users were able to make precise edits to the given residential designs. Additionally, the model could adjust the residential design based on recommendations from the "Residential Code Auditor" agent, such as "making the structure more rectangular overall." Furthermore, users could manually modify the inputs and outputs of RL-Net to make precise edits to the residential design. Users can iteratively edit the residential design until it satisfies their requirements. Finally, we tested agent efficiency by setting thresholds at 0.7, 0.8, and 0.9, and evaluated 100 groups. The system averaged 1.8, 3.1, and 4.9 iterations for output, with each iteration taking 70.6 seconds on average. Appendix E for more examples.

4.4 CARD User Study

Expert and User Evaluation: Given the similarity of many residential floor plans in the RPLAN dataset, there may be some feature leakage. Therefore, we invited experts and ordinary users to create 100 residential description manually (includ-

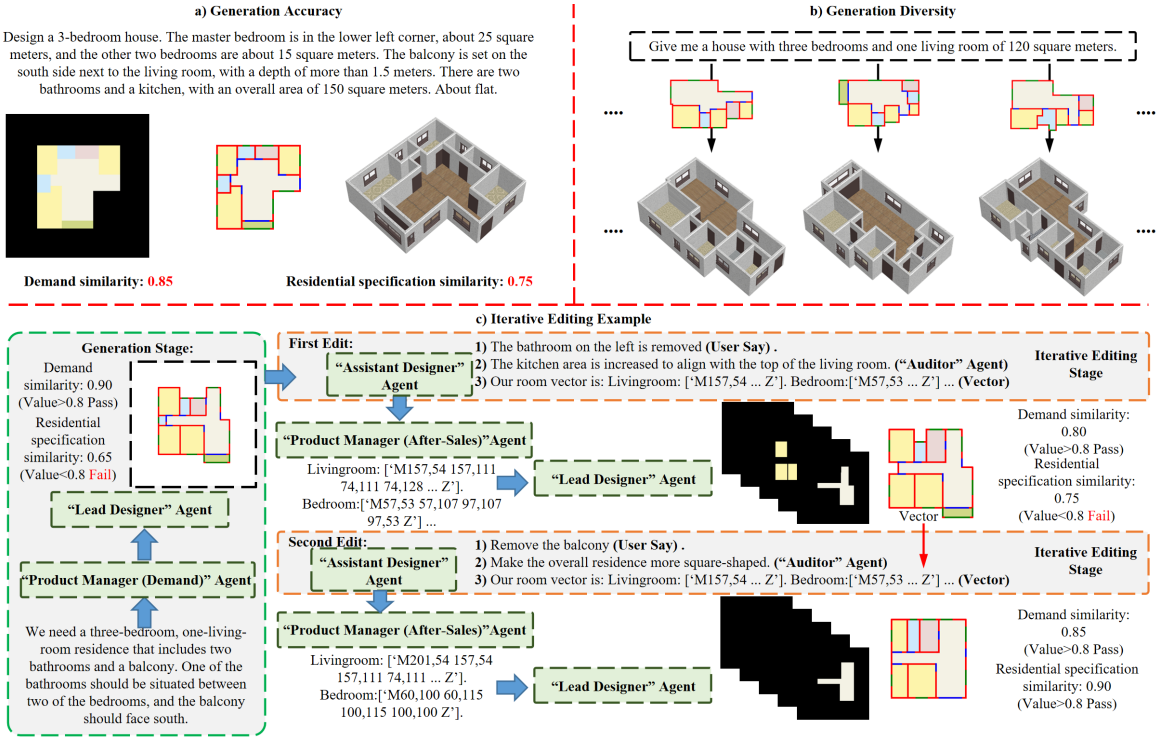

Figure 4: Results of a study on the model's adaptability in a complex open-world environment.

Table 5: Consistency and rationality test.

Method	Consistency↑	Rationality↑
CARD	**87.8**	**81.4**
Tell2Design	77.6	72.3
Holodeck	62.3	56.8
LayoutGPT	31.8	12.4

ing free-text input and manually constructed images) and generate residential floor plans using our model, which were then mixed for the experts to judge. The results are shown in Table 4. The accuracy of our model generation is 54.4%, which higher than the 45.6% accuracy of the actual images. Therefore, the residential floor plan generated by the CARD framework is very accurate and similar to real residential floor plans, proving correctness and authenticity of the generated residential floor plans. We present the generated results for the User tests in Fig. 3d), which shows that our model creates a complete residential floor plan when the user specifies different types of inputs. Moreover, the model ensures the high stability and quality of the generated image while generating a complete residential floor plan. These results demonstrate the high flexibility of our model for various inputs.

Consistency and Rationality: We used expert scoring to assess. A total of 37 experts and users were invited, and each person conducted 10 tests. The final result is the average value calculated from these tests. Experts determine the input conditions for each model based on their design requirements, and the models generate floor plans based on these inputs. Consistency are rated by the experts who provide the input, while a separate group of experts rated layout rationality. The scoring criteria are as follows: 60=passing, 70=medium, 80=good, 90=excellent. As shown in Table 5, the consistency score reached 88, indicating excellent alignment with user requirements, while the rationality score was 81, demonstrating good adherence to practical and logical design principles. The results indicate that our model outperforms existing models in terms of consistency and rationality in generation.

5 Conclusion

In this work, we proposed the CARD framework, a language-based agent system designed for the generation and editing of residential floor plans in complex open-world environments. CARD enables non-expert users to engage in residential design without the need for specialized knowledge. This work advances the integration of cross-modal agents and language-driven design tools in the architectural domain, and demonstrates the potential of LLM-based agent frameworks in domains lacking paired image-text data, such as architecture.

6 Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 52578029).

7 Limitations

We identified and summarized several failure cases:

1. **Infinite Loop at High Thresholds:** We observed that when the threshold is set above 0.96, the CARD system often becomes trapped in an infinite generation loop. We attribute this issue to the limitations of the LLM in analyzing residential layouts and its tendency toward hallucination.

2. **Abstract User Requirements:** When user inputs are overly abstract, the consistency score tends to be significantly lower. For example, the input "I desire a living space that transcends the boundaries of conventional design. It should be a nebulous realm where functionality melds with the abstract, where light dances in unpredictable patterns, and where each nook whispers a cryptic tale." yields a consistency score of only 71.4. We believe this is due to a dimensional mismatch—where the semantic requirements imply a 3D spatial conception that exceeds the representational capacity of our 2D layout generation model. Additionally, LLMs often struggle to interpret such abstract or poetic descriptions.

3. **Unrealistic or Unconventional Room Configurations:** In cases where the user requests an excessive number of rooms (e.g., ten bedrooms) or illogical configurations (e.g., three bathrooms without any bedrooms), the Product Manager (Demand) agent is generally able to parse the input and generate point-based specifications accordingly. However, the Text2FloorEdit model, trained on the RPLAN dataset, lacks exposure to such atypical cases and thus defaults to generating floor plans that conform to conventional residential design standards.

4. **Missing Physical Simulation Metrics:** In addition, the current version of the CARD system does not account for physical simulation metrics such as building energy consumption, daylighting performance, or construction costs. These metrics could be readily incorporated into the existing CARD framework by integrating them into the "Auditors" agent module. Therefore, we believe that CARD can serve not only as a standalone generative system but also as an extensible foundational framework for integrated, performance-aware architectural design.

8 Potential Risks

The CARD framework brings promising capabilities for accessible residential design, but also entails risks related to misuse, bias, and privacy.

1. **Misinformation and Misuse:** As CARD enables natural language-driven residential design generation, there is a risk that unqualified users could misuse the system to produce building layouts that appear valid but violate essential architectural, structural, or safety standards if the system is used without proper oversight. To mitigate this, CARD incorporates code auditors and embedded regulation checks; however, ensuring compliance across diverse global building codes remains a challenge and warrants ongoing refinement.

2. **Bias in Training Data and Design Norms:** The training data and evaluation criteria used in the framework may reflect cultural or socioeconomic biases, favoring particular housing styles, spatial configurations, or regional norms. This could lead to overrepresentation of certain types of residences while underrepresenting others, potentially excluding marginalized groups or non-mainstream living arrangements. Future work could explore the inclusion of more diverse datasets and adaptive feedback mechanisms.

3. **Privacy Concerns and Data Security:** If future versions of CARD are deployed in a cloud-based service, user-generated prompts and design preferences may contain sensitive personal or proprietary information. Adequate encryption, access controls, and responsible data handling policies must be adopted to protect user privacy.

9 Ethical Concerns

We do not anticipate immediate ethical or societal impacts arising from our work. However, as an engineering application based on LLMs for text interpretation, we recognize that CARD could be

affected by various types of hallucinations inherent in LLMs. We therefore urge researchers and practitioners to use our proposed framework mindfully, particularly when deploying such LLM-based agents in real-world applications.

References

Mohammadreza Aalaei, Melika Saadi, Morteza Rahbar, and Ahmad Ekhlassi. 2023. Architectural layout generation using a graph-constrained conditional generative adversarial network (gan). *Automation in Construction*, 155:105053.

Zeina Abu-Aisheh, Romain Raveaux, Jean-Yves Ramel, and Patrick Martineau. 2015. An exact graph edit distance algorithm for solving pattern recognition problems. In *4th International Conference on Pattern Recognition Applications and Methods 2015*.

Wang Bo, Chen Mengjia, and 1 others. 2022. Reconstruction design of existing residential buildings based on 3d simulation method. *Discrete Dynamics in Nature and Society*, 2022.

Silvio Carta. 2022. *Machine learning and the city: applications in architecture and urban design*. John Wiley & Sons.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Dake Chen, Hanbin Wang, Yunhao Huo, Yuzhao Li, and Haoyang Zhang. 2023. Gamegpt: Multi-agent collaborative framework for game development. *arXiv preprint arXiv:2310.08067*.

Qi Chen, Qi Wu, Rui Tang, Yuhan Wang, Shuai Wang, and Mingkui Tan. 2020. Intelligent home 3d: Automatic 3d-house design from linguistic descriptions only. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12625–12634.

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, and 1 others. 2019. Textworld: A learning environment for text-based games. In *Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7*, pages 41–75. Springer.

David H Douglas and Thomas K Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338.

Zesen Fan, Jiepeng Liu, Lufeng Wang, Guozhong Cheng, Minqing Liao, Pengkun Liu, and Y Frank Chen. 2023. Automated layout of modular high-rise residential buildings based on genetic algorithm. *Automation in Construction*, 152:104943.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2024. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36.

Wen Gao, Shuai Lu, Xuanming Zhang, Qiushi He, Weixin Huang, and Borong Lin. 2023. Impact of 3d modeling behavior patterns on the creativity of sustainable building design through process mining. *Automation in Construction*, 150:104804.

Wen Gao, Chenglin Wu, Weixin Huang, Borong Lin, and Xia Su. 2021. A data structure for studying 3d modeling design behavior based on event logs. *Automation in Construction*, 132:103967.

Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. 2022. Segnext: Rethinking convolutional attention design for semantic segmentation. *Advances in Neural Information Processing Systems*, 35:1140–1156.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2020. Interactive fiction games: A colossal adventure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7903–7910.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Ruizhen Hu, Zeyu Huang, Yuhan Tang, Oliver Van Kaick, Hao Zhang, and Hui Huang. 2020. Graph2plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics (TOG)*, 39(4):118–1.

Weixin Huang and Hao Zheng. 2018. Architectural drawings recognition and generation through machine learning. In *Proceedings of the 38th annual*

conference of the association for computer aided design in architecture, Mexico City, Mexico, pages 18–20.

Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. 2021. What makes multi-modal learning better than single (provably). *Advances in Neural Information Processing Systems*, 34:10944–10956.

Quan Huynh-Thu and Mohammed Ghanbari. 2008. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

Maowei Jiang, Pengyu Zeng, Kai Wang, Huan Liu, Wenbo Chen, and Haoran Liu. 2023. Fecam: Frequency enhanced channel attention mechanism for time series forecasting. *Advanced Engineering Informatics*, 58:102158.

Marko Lazić, Ana Perišić, and Branko Perišić. 2021. Residential buildings complex boundaries generation based on spatial grid system. *Applied Sciences*, 12(1):165.

Sicong Leng, Yang Zhou, Mohammed Haroon Dupty, Wee Sun Lee, Sam Joyce, and Wei Lu. 2023. Tell2design: A dataset for language-guided floor plan generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14680–14697.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi Mi, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, and 1 others. 2023a. Chatharuhi: Reviving anime character in reality via large language model. *arXiv preprint arXiv:2308.09597*.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023b. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Yuqian Li. 2023. Research on architectural generation design of specific architect's sketch based on image-to-image translation. *Hybrid Intelligence*, page 314.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Jiaju Lin, Haoran Zhao, Aochi Zhang, Yiting Wu, Huqiuyue Ping, and Qin Chen. 2023. Agentsims: An open-source sandbox for large language model evaluation. *arXiv preprint arXiv:2308.04026*.

Pengjie Liu, Fucheng Pan, Xiaofeng Zhou, Shuai Li, Pengyu Zeng, Shurui Liu, and Liang Jin. 2022. Dsapaml: A parallel automated machine learning system via dual-stacked autoencoder. *Neural Computing and Applications*, 34(15):12985–13006.

Ziniu Luo and Weixin Huang. 2022. Floorplangan: Vector residential floorplan adversarial generation. *Automation in Construction*, 142:104470.

Xintong Ma, Tiancheng Zeng, Miao Zhang, Pengyu Zeng, Borong Lin, and Shuai Lu. 2025. Street microclimate prediction based on transformer model and street view image in high-density urban areas. *Building and Environment*, 269:112490.

Nelson Nauata, Sepidehsadat Hosseini, Kai-Hung Chang, Hang Chu, Chin-Yi Cheng, and Yasutaka Furukawa. 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*, pages 13632–13641.

Osama Omar, Rania El Messeidy, and Maged Youssef. 2016. Impact of 3d simulation modeling on architectural design education. *Architecture and Planning Journal (APJ)*, 23(2):6.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 6.

Anil Rahate, Rahee Walambe, Sheela Ramanna, and Ketan Kotecha. 2022. Multimodal co-learning: Challenges, applications with datasets, recent advances and future directions. *Information Fusion*, 81:203–239.

Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. 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*, pages 5466–5475.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature*, 623(7987):493–498.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*.

Da Wan, Xiaoyu Zhao, Wanmei Lu, Pengbo Li, Xinyu Shi, and Hiroatsu Fukuda. 2022. A deep learning approach toward energy-effective residential building floor plan generation. *Sustainability*, 14(13):8074.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Man Zhang, and 1 others. 2023b. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv preprint arXiv:2310.00746*.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

Ramon Elias Weber, Caitlin Mueller, and Christoph Reinhart. 2022. Automated floorplan generation in architectural design: A review of methods and applications. *Automation in Construction*, 140:104385.

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

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, and 1 others. 2024. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16227–16237.

Jun Yin, Wen Gao, Jizhizi Li, Pengjian Xu, Chenglin Wu, Borong Lin, and Shuai Lu. 2025a. Archidiff: Interactive design of 3d architectural forms generated from a single image. *Computers in Industry*, 168:104275.

Jun Yin, Yangfan He, Miao Zhang, Pengyu Zeng, Tianyi Wang, Shuai Lu, and Xueqian Wang. 2025b. Promptlnet: Region-adaptive aesthetic enhancement via prompt guidance in low-light enhancement net. *arXiv preprint arXiv:2503.08276*.

Jun Yin, Pengyu Zeng, Haoyuan Sun, Yuqin Dai, Han Zheng, Miao Zhang, Yachao Zhang, and Shuai Lu. 2025c. Floorplan-llama: Aligning architects' feedback and domain knowledge in architectural floor plan generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6640–6662.

Pengyu Zeng, Wen Gao, Jizhizi Li, Jun Yin, Jiling Chen, and Shuai Lu. 2025a. Automated residential layout generation and editing using natural language and images. *Automation in Construction*, 174:106133.

Pengyu Zeng, Wen Gao, Jun Yin, Pengjian Xu, and Shuai Lu. 2024. Residential floor plans: Multi-conditional automatic generation using diffusion models. *Automation in Construction*, 162:105374.

Pengyu Zeng, Guoliang Hu, Xiaofeng Zhou, Shuai Li, and Pengjie Liu. 2023. Seformer: a long sequence time-series forecasting model based on binary position encoding and information transfer regularization. *Applied Intelligence*, 53(12):15747–15771.

Pengyu Zeng, Guoliang Hu, Xiaofeng Zhou, Shuai Li, Pengjie Liu, and Shurui Liu. 2022. Muformer: A long sequence time-series forecasting model based on modified multi-head attention. *Knowledge-Based Systems*, 254:109584.

Pengyu Zeng, Jun Yin, Yan Gao, Jizhizi Li, Zhanxiang Jin, and Shuai Lu. 2025b. Comprehensive and dedicated metrics for evaluating ai-generated residential floor plans. *Buildings*, 15(10):1674.

Pengyu Zeng, Jun Yin, Miao Zhang, Jizhizi Li, Yachao Zhang, and Shuai Lu. 2025c. Unified residential floor plan generation with multimodal inputs. *Automation in Construction*, 178:106408.

Miao Zhang, Jun Yin, Pengyu Zeng, Yiqing Shen, Shuai Lu, and Xueqian Wang. 2025. Tscnet: A text-driven semantic-level controllable framework for customized low-light image enhancement. *Neurocomputing*, 625:129509.

Ruichuan Zhang and Nora El-Gohary. 2022. Natural language generation and deep learning for intelligent building codes. *Advanced Engineering Informatics*, 52:101557.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, and 1 others. 2023. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*.

A Experimental Settings

Comparison Methods: We generated residential floor plans and compared them with those obtained using baseline approaches. Our editable residential design generation task based on free-text has no comparable methodology. Therefore, simplified versions of our proposed network were compared. We compared both residential floor plans and multimodal residential layout generation.

To evaluate the residential floor plan generation, we compared our model with the Graph2Plan(Hu

et al., 2020), Building Floor Plan (BFP)(Wan et al., 2022), and CycleGAN(Li, 2023) models. Graph2Plan is popular because of its compelling features and advantages. This model uses a GNN to process and learn the spatial topological relationship of residential floor functions to generate more stable residential floor plans. BFP is a residential floor plan generation model based on the Pix2Pix network(Isola et al., 2017), and we reproduced it according to the published description using the conventional Pix2Pix network. The CycleGAN model, by contrast, is realized by adversarial training with unpaired image transformations.

To evaluate the multimodal residential layout generation, we compared our model with the Tell2Design(Leng et al., 2023) models. Tell2Design is an optimal multimodal generation model for generating residential layout maps from text. It is based on the Sequence2Sequence model, which generates residential layout diagrams using linguistic guidance.

Evaluation Metrics: To compare generated residential floor plans, we used the FID(Heusel et al., 2017), PSNR(Huynh-Thu and Ghanbari, 2008), and SSIM(Wang et al., 2004) metrics. FID measures the similarity between generated images that calculates the statistical distance between the distributions of the actual and generated images. PSNR measures image reconstruction quality that evaluates the fidelity of an image by calculating the mean squared error between the original and reconstructed images. SSIM evaluates structural similarity, which objectively assesses the realism of the image and its fidelity.

In the multimodal residential layout generation comparison task, we used the FID, Micro IoU, and Macro IoU metrics(Everingham et al., 2010). Micro IoU evaluates the degree of overlap between the predicted and actual labels by calculating the ratio of the intersection and concatenation of all categories. In contrast, Macro IoU calculates the IoU of each type and averages it. Values closer to 1indicate a more complete overlap between the predicted and actual labels.

User Test: In addition, we used human experts and ordinary users to comprehensively evaluate the effectiveness of the model generation. For the remainder of this paper, we will refer to them as users. We invited users to compare and evaluate actual images with model-generated images. Specifically, we mixed the model-generated images with actual images and asked the users to select the authentic images. The final average score given by several users was used to measure the effectiveness of the model generation. Because the random correctness rate was 0.5, the results generated by the model may confuse the users. Details of the user's evaluation setup are in the following.

Many evaluation metrics have been proposed to evaluate the effectiveness of models comprehensively and accurately. However, these metrics cannot perfectly replace human subjective evaluations. Therefore, this study adopted the user evaluation method used in the Turing test to comprehensively evaluate the effectiveness of the model. This involved users who identify real FPs among a mixed collection of real plans and plans generated by our model. Specifically, 20 users were invited to participate in the user review. The users included registered architects, graduate architecture students, and ordinary users, who were first requested to provide five sets of input so that our model could generate floor plans, where each set of inputs had to be in the form of a bounding box to avoid the bias caused by human inputs. The dataset consisted of 200 FPs comprising 100 real and 100 generated samples. For each experiment, we randomly selected ten samples, which included five real and five generated samples. We subsequently combined the generated and real images and presented them in a shuffled order to the users. The users had to identify the five real images in each set. Each user performed this task for five sets, i.e., in total reviewing 50 plans and selecting 25. (We remove the plans developed by the users themselves for each user during the assessment phase to avoid the users recognizing that the plans were developed by themselves). Since our task involved choosing five samples from a pool of ten, the randomized correctness rate was 0.5, i.e., meaning that the model generated results that confused the users. Lower precision and recall values indicate greater difficulty in distinguishing between the generated and real images. Hence, we utilized precision evaluations provided by multiple users to assess the effectiveness of the generated images. Based on this, the strengths and weaknesses of the generated models were evaluated more comprehensively. This subjective evaluation method allows the generation results to be intuitively measured from human perspectives and provides a powerful means to assess model performance.

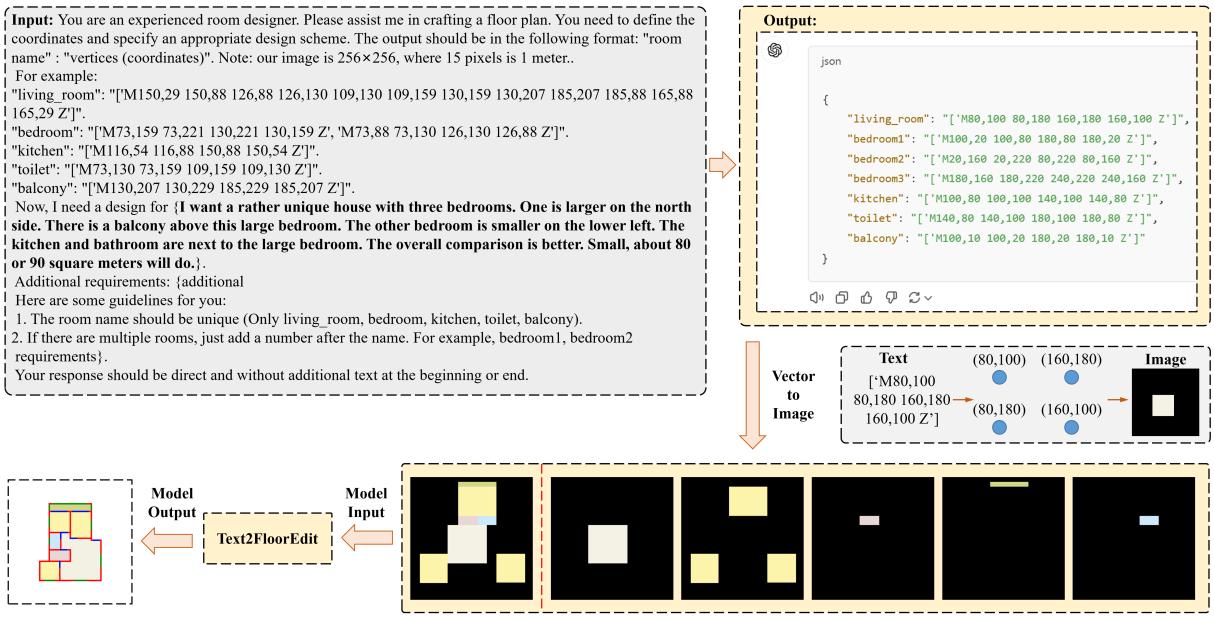

Figure 5: Generate residential floor plans through free text.

B RAG

This process follows a similar RAG-based approach. Using the RAG model, the auditor retrieves information from a User Requirements Database, which is built from the user's initial input, interactive inputs during the editing process, and personalized case studies from previous users. The retrieved requirements and preferences are then compared with the generated design. This comparison can be expressed as a similarity or distance function, as represented by Eq. 1. In the final decision phase, judgments are made based on the similarity score $\mathcal{S}(D, U)$. If the design is flagged as non-compliant, feedback is generated for further editing and revision.

$$\mathcal{S}(D, U) = \sum_{i=1}^{n} \omega_i \cdot \mathrm{sim}\left(d_i, u_i\right) \qquad (1)$$

where: D represents the design features, and U represents user requirements. ω_i is a weight that reflects the importance of requirement i. $\mathrm{sim}\left(d_i, u_i\right)$ is a similarity function that measures how closely design feature d_i matches user requirement u_i.

C 3D Modeler Agent

To visualize residential floor plans, we developed a 3D residential renderer system. This renderer transforms a residential design plan into a 3D residential design, allowing users to visualize the details and overall ideas of the floor plan from a spatial perspective.

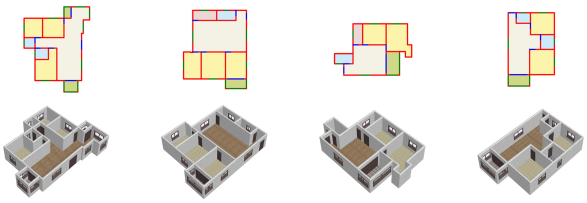

Figure 6: 3D model generation and rendering to transform 2D residential design drawings into 3D residential designs.

This 3D rendering system receives the 256×256 resolution complete residential planes output from RL-Net as input, and the output is a 3D representation of the residential planes, i.e., 3D residential deisgns. The rendering result is shown in Fig. 6. In this system, we first carefully separate the semantic blocks in the floor plan, i.e., separate the individual color blocks and then use them to generate walls. The default heights of walls, doors, and windows are 2.85 m, 2 m, and 1.2 m, respectively (these values can be adjusted). To enhance the realism and personalization of the rendered model, we designed a material-rendering system to customize it. Users can choose from several preset materials and adjust their color, transparency, and other attributes.

To ensure a uniform visual experience, a virtual camera is placed above a specific corner of each rendered 3D residential design model. Besides, the viewing angle can be manually adjusted, allowing users to rotate and view the model from different directions.

Table 6: Comparison of the model's results under the same input conditions.

Functional partition	BGR Range
walls	[0, 0, 245] ∼[10, 10, 255]
doors	[245, 0, 0] ∼[255, 10, 10]
windows	[0, 128, 0] ∼[10, 138, 10]
Living room	[225, 235, 240] ∼[235, 245, 250]
bedroom	[165, 240, 245] ∼[175, 250, 255]
kitchen	[210, 210, 230] ∼[220, 220, 240]
toilet	[245, 227, 200] ∼[255, 237, 210]
balcony	[130, 210, 202] ∼[140, 220, 212]

D Parameterize the Generated Floor Plans

In this system, the OpenCV library in Python is utilized to extract semantic information from residential floor plans. The process begins with color normalization, where specific BGR value ranges are defined for various elements, such as rooms, doors, and windows. These BGR values are detailed in Table 6. Using these predefined color ranges, the system identifies and isolates the corresponding semantic blocks within the floor plan.

The first step in the process is color normalization, where pixel values for each element (e.g., room, door, window) are mapped to specific ranges. The OpenCV inRange() function is used to create a mask for each element based on these predefined BGR ranges, enabling the extraction of regions corresponding to different parts of the floor plan.

Once the color information is extracted, the next step involves vectorization. The system converts the color-processed floor plans into vector representations, providing a more structured layout. This is achieved through contour detection and the extraction of paths from the floor plan using the findContours() function. These contours are subsequently stored as SVG paths, representing the floor plan elements in a vectorized format. An example is shown in Fig. 7.

This process facilitates accurate feature extraction and conversion of floor plan images into a structured vector format, enabling downstream tasks such as floor plan generation and optimization.

E CARD 3D Generation and editing Studies

We expanded the generation diversity experiment results, as shown in Fig. 8.

Editing Precision: When the user wants to make individualized edits to the generated 3D residential design that are difficult to describe in words, the user can make precise editing changes to the 3D residential design by manually adjusting the inputs and outputs of RL-Net.

The above results show that our model is robust to a variety of free text conditions on both the generation and editing tasks. Meanwhile, the experiment verifies that the model can manually and flexibly edit and modify the residential design at all task steps and generate modified high-quality residential diagrams.

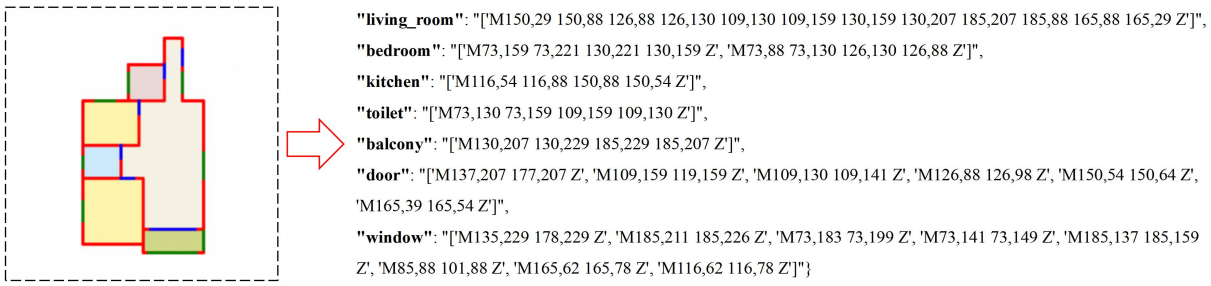

Figure 7: Example of parameterized results.

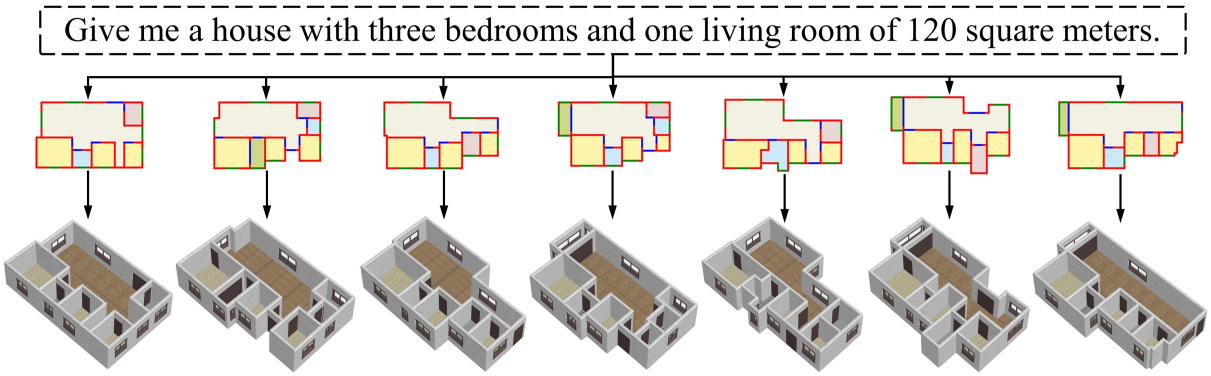

Figure 8: Diversity of the generated results of our model, which generates multiple 3D residential designs based on the same linguistic description.