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

Pengyu Zeng[1,*], **Maowei Jiang**[1*], **Zihang Wang**[2*], **Jizhizi Li**[3], **Jun Yin**[1], **Shuai Lu**[1†]

[1]Shenzhen International Graduate School, Tsinghua University, China
[2]The University of Chinese Academy of Sciences, China
[3]The University of Sydney, Australia
[†]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 structures 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 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 structure. 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 Luo and Huang [2022], 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 Fan et al. [2023], Zhang et al. [2021], Lazić et al. [2021], Cote et al. [2020], de Almeida et al. [2016].

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 structures at a low cost and with minimal expertise.

[*]These authors contributed equally to this work.
[†]Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

In this paper, we introduce CARD, a cross-modal agent-driven framework that leverages natural language input to generate and edit 3D residential structures. 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 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 Gu et al. [2023]. Nevertheless, several hurdles persist. First, collecting large-scale multimodal data for training, especially for both text and residential structure 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 structures, and continuously learn from interactions. To conduct more precise and standardized Residential structure 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. [2023a], 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. [2023b], Zhu et al. [2023].

2.2 Residential Floor Plan Generation

Recent approaches to residential floor plan generation typically fall into three categories: rule-based methods, GAN-based models, and graph structure-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 diagram structure 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 language drivers are less flexible. In addition, current research on 3D residence focuses on generating

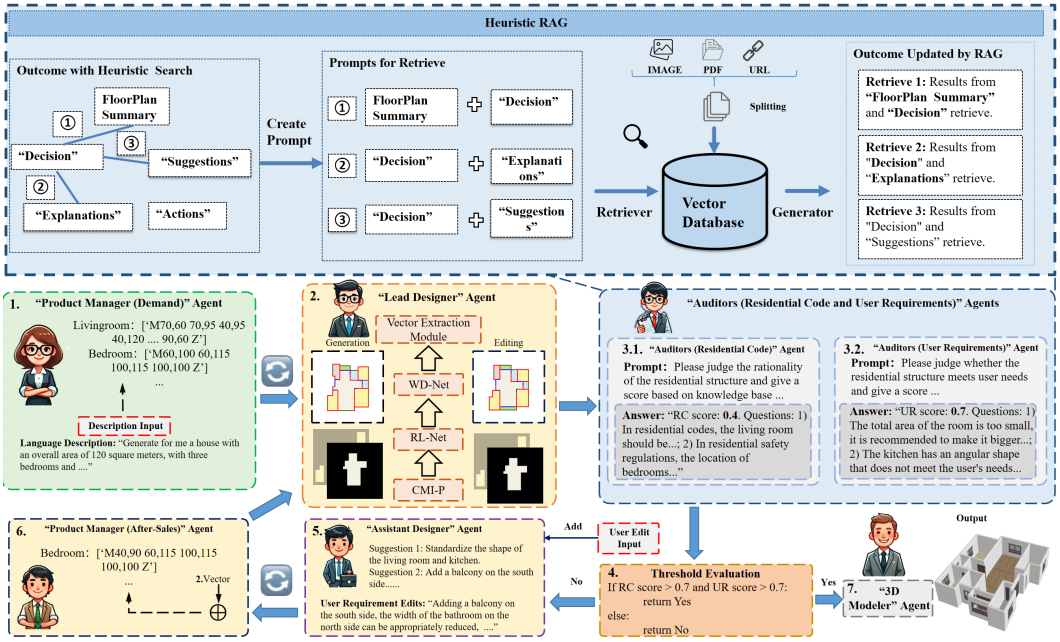

Figure 1: Overview of the CARD Architecture. The language agent is powered by GPT-4, while the image agent leverages our custom-designed Text2FloorEdit model.

3D residence based on template-based language 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 structure editing based on free language is currently in a blank stage.

3 CARD

The CARD framework aims to facilitate the generation and editing of residential structures based on free-text input, enabling non-expert users to navigate the complexities of open-world environments. The framework structure, as illustrated in Figure 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 Requirements), 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 structure 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 structure 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 structure is clearly visualized for the user through a "3D Modeler" agent.

3.2 Text2FloorEdit Model

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. CMI-P enables the model to function without costly multimodal data and effectively represent the geometry and spatial position relationships of residential rooms, by using points to represent residential layouts. It converts point data into image data for generation and back into point data for editing.

RL-Net: We utilize a diffusion model as the foundation of RL-Net. First, we define the model's input formats, setting it to five images representing the living room, bedroom, bathroom, balcony, and kitchen. To enhance the flexibility of the design, we adopt multiple formats for the input. For instance, we utilize bounding boxes for each room and its contour, a binary representation where zero-values indicate vacant room information, and input for size adjustment. To further enhance the relevance of information and reduce the design cost of various residential layouts, we drop some model input formats. That is, $x_t' = x_t \oplus A$, where $A \in \mathbb{R}^{P \times D}$ denotes the input formats after dropout and $\oplus$ indicates image splicing. $A_p = \mathcal{Z}\left(\mathcal{D}\left(\alpha_i^p, \beta_i^p, \gamma_i^p\right)\right) \in \mathbb{R}^D$, where $\alpha_i^p, \beta_i^p, \gamma_i^p$ means that p-type room inputs are respectively zero values, original values, or bounding-box values (the room shape is transformed into a bounding box). We consider two cases for room size ($D = 2$): variable or non-variable. Function $\mathcal{D}(\cdot)$ randomly deletes multiple conditions, and only one condition is selected. In $\mathcal{Z}(\cdot)$, if the state set by $\mathcal{D}(\cdot)$ belongs to the room size variable, then $A_p = \text{Zero} \oplus \mathcal{D}\left(\alpha_i^p, \beta_i^p, \gamma_i^p\right)$; otherwise, $A_p = \mathcal{D}\left(\alpha_i^p, \beta_i^p, \gamma_i^p\right) \oplus Zero$, where $Zero$ is a zero-valued image.

To enhance feature relevance Guo et al. [2022], we designed the Multi-Scale Fusion De-redundant Attention (MFDA) module inspired by multi-scale feature fusion. Initially, we extract features from the input formats using a 5×5 convolutional layer. Given the straight-line nature of room contours, we apply 1×n and n×1 convolutions for low-cost feature extraction, followed by de-redundancy via ScConv. Features are fused at different scales (n=7,11,21), summed, and convolved by 1×1 to model inter-channel relationships. The resulting features are used as attention weights to reweight the MFDA input. The representation is shown in Eq. 1.

$$\begin{aligned} Att &= \mathrm{Conv}_{1\times 1}\left(\sum_{i=0}^{3} \mathrm{Scale}_i(\mathrm{ScConv}(\mathrm{FConv}(F)))\right), \\ \text{Out} &= Att \otimes F. \end{aligned} \tag{1}$$

Where F represents the input features. Att and Out represent the attention map and the output. $\otimes$ is the element-by-element matrix multiplication operation. $\mathrm{FConv}(\cdot)$ is fusion convolution. $\mathrm{ScConv}(\cdot)$ is the de-redundancy convolution Li et al. [2023c]. Scale_i represents multi-scale braking, where $i \in \{0, 1, 2, 3\}$. We set the kernel size for different scales to 7, 11, and 21. We chose strip convolution for feature fusion, considering that most residential layouts are horizontal and vertical, in order to reduce training costs.

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. 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 structure 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 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 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 structure 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, as represented by Eq. 2. 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.

$$\mathcal{S}(D, U) = \sum_{i=1}^{n} \omega_i \cdot \mathrm{sim}\left(d_i, u_i\right) \tag{2}$$

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.

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 structure, allowing users to visualize the

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

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, validation, and test sets containing 70126, 500, and 500 plan images, respectively.

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

To evaluate the residential floor plan generation, we compared our model with the HouseDiffusionShabani et al. [2023], House-GAN++Nauata et al. [2021], Graph2PlanHu et al. [2020], Building Floor Plan (BFP)Wan et al. [2022], and CycleGANLi [2023] models. For evaluation, we chose the FIDHeusel et al. [2017], PSNRHuynh-Thu and Ghanbari [2008], and SSIMWang et al. [2004] metrics. To evaluate the multimodal residential layout generation, we compared our model with the Tell2DesignLeng et al. [2023] and ImagenSaharia et al. [2022] models (based on T5Raffel et al. [2020]), where we chose the FID, Micro IoU, and Macro IoU metricsEveringham et al. [2010]. To evaluate the generative capabilities of our CARD architecture, we performed a visual comparison with LayoutGPT and Holodeck. To evaluate the adaptability of our architecture to complex environments, we provided a multi-round iteration example for reference. In addition, we used human 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. We trained the model on a single NVIDIA A100 GPU with a batch size of 128.

4.2 Residential Floor Plan Generation Study

To evaluate our designed "Lead Designer" Agent, namely the Text2FloorEdit model, we asked each model to generate 500-floor plans and compared the results qualitatively and quantitatively, as shown in Fig 2a). The first step is a qualitative comparison. These results reveal that our model perfectly reproduces the semantic information and functional structure 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 BFP and CycleGAN models exhibit generation instability problems. In addition, we performed quantitative comparisons, as shown in Table 1a). Compared with the Baseline model, our model performs optimally in FID, PSNR, and SSIM indicators.

4.3 Multimodal Residential Layout Generation Study

To evaluate our free-language-based residential structure generation architecture, To evaluate this approach, we compared it with state-of-the-art text-generated residential layout models, as shown in Fig. 2b). Where our model performs free text generation under zero multimodal dataset conditions, the other models are trained to generate on a multimodal dataset of template text. 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 1b) 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 multimodal datasets while generating high quality.

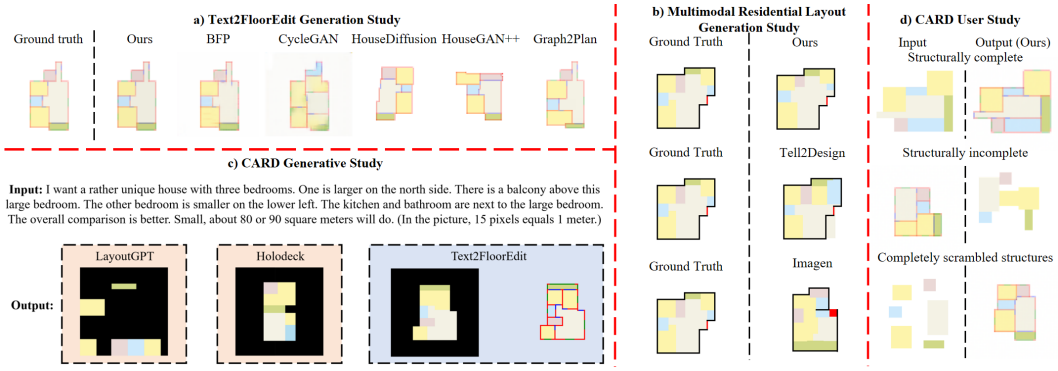

Figure 2: Qualitative analysis results of the model on multiple studies.

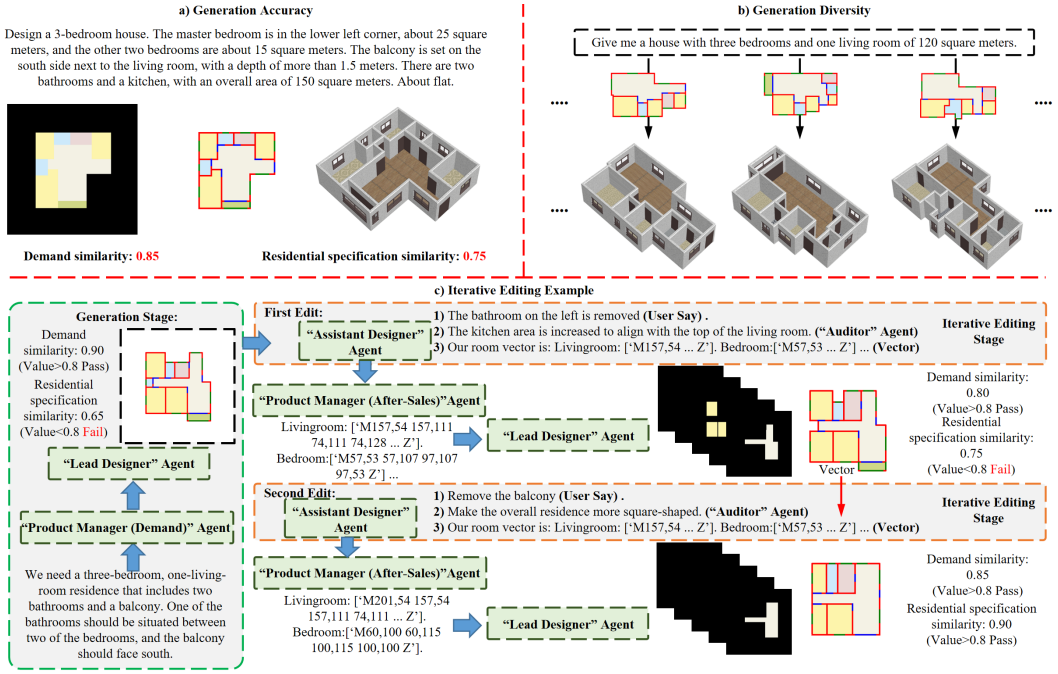

Figure 3: Results of a study on the model's adaptability in a complex open-world environment.

4.4 CARD Generative Study

To evaluate the superiority of our CARD architecture in generating residential structures, we compared our model with the LLM-based LayoutGPT and Holodeck models, which also utilize free-text descriptions. The results are shown in Fig. 2c). 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 structures is not as effective as our model. In addition, the results show that the residential structures generated by our model have certain advantages in both residential specifications and user needs.

4.5 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: language description and residential code compliance, with results shown in Fig. 3a). Our model accurately captured information from free-text descriptions and generated reasonable residential structures based on that

Table 1: Quantitative analysis results of the model in multiple studies.

(a) CMRG model test results.

Method	FID ↓	PSNR ↑	SSIM ↑
CycleGAN	134.79	70.98	0.72
BFP	71.39	79.43	0.95
Graph2Plan	32.45	79.57	0.95
HouseGAN++	34.06	84.81	**0.99**
HouseDiffusion	28.72	84.28	**0.99**
Ours	**8.36**	**86.25**	**0.99**
Change in ratio	70.9%	1.7%	0.0%

(b) Multimodal generation test results.

Method	FID ↓	Micro IoU ↑	Macro IoU ↑
Imagen	17.02	0.56	0.20
Tell2Design	11.01	0.77	0.67
Our	**7.65**	**0.89**	**0.71**

(c) User test results.

	Model generation	Real images	All
Precision (average)	54.4%	45.6%	100
Truth	50%	50%	100
Change in ratio	+4.4%	-4.4%	0%

information. 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," the model successfully generated an accurate residential structure. Additionally, in this experiment, we set a residential code compliance threshold of 0.7, and the residential structure outputs achieved a code compliance score higher than the set threshold.

Generation Diversity: To assess the impact of vague language descriptions, we conducted a generation diversity experiment, with results shown in Fig. 3b). When the language description was imprecise, our model generated various residential structures 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.

User Iterative Editing Example: If the generated results do not fully meet the user's needs, our model allows for editing of the residential structure in multiple ways. As shown in Fig. 3c), users were able to make precise edits to the given residential structures. Additionally, the model could adjust the residential structure 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 structure. Users can iteratively edit the residential structure until it fully satisfies their requirements.

4.6 CARD User Study

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 (including free language input and manually constructed images) and generate complete residential floor plans using our model, which were then mixed together for the experts to judge. The results are shown in Table 1c). The accuracy of our model generation is 54.4%, which is 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 the correctness and authenticity of the generated residential structures. We present the generated results for the User tests in Fig. 2d), which shows that our model creates a complete residential floor plan when the user specifies different types of structures. 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 types of inputs.

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. The framework incorporates multiple specialized agents that work collaboratively across the entire process—from requirement understanding and design generation to regulatory compliance checking and 3D visualization—enabling non-expert users to engage in residential design without the need for specialized knowledge. Our experimental results demonstrate that the proposed method outperforms existing

baseline models in generating residential floor plans and exhibits strong adaptability to complex environments through multi-round iterative generation. Additionally, evaluations by both human experts and lay users further validate the accuracy and compliance of the generated outputs. Looking forward, we plan to integrate more architectural regulations and personalized user requirements into the generation process to enhance the practicality and applicability of the model.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 52078294); Shenzhen Science and Technology Innovation Committee (WDZC20231129201240001).

References

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

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

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

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

Jingyu Zhang, Nianxiong Liu, and Shanshan Wang. Generative design and performance optimization of residential buildings based on parametric algorithm. *Energy and Buildings*, 244:111033, 2021.

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

Melissa Cote, Alireza Rezvanifar, and Alexandra Branzan Albu. Automatic generation of electrical plan documents from architectural data. In *Proceedings of the ACM Symposium on Document Engineering 2020*, pages 1–4, 2020.

Ana de Almeida, Bruno Taborda, Filipe Santos, Krystian Kwiecinski, and Sara Eloy. A genetic algorithm application for automatic layout design of modular residential homes. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 002774–002778. IEEE, 2016.

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

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

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

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

Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*, 2023.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 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, 2019.

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

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

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. 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*, 2023.

Weixin Huang and Hao Zheng. 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, 2018.

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

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

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

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

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

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. 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, 2024.

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

Jiafeng Li, Ying Wen, and Lianghua He. Scconv: spatial and channel reconstruction convolution for feature redundancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6153–6162, 2023c.

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

Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5466–5475, 2023.

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

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

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

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

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

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

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

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