EchoLadder: Progressive AI-Assisted Design of Immersive VR Scenes

Zhuangze Hou
School of Creative Media
City University of Hong Kong
Hong Kong, China
zhuanghou3-c@my.cityu.edu.hk

Jingze Tian
School of Creative Media
City University of Hong Kong
Hong Kong, China
jingztian2-c@my.cityu.edu.hk

Nianlong Li*
Institute of software, Chinese Academy of Sciences
Beijing, China
linianlong16@mails.ucas.ac.cn

Farong Ren
Steinhardt School of Culture, Education, and Human Development
New York University
New York, NY, USA
fr2305@nyu.edu

Can Liu*
School of Creative Media
City University of Hong Kong
Hong Kong, China
canliu@cityu.edu.hk

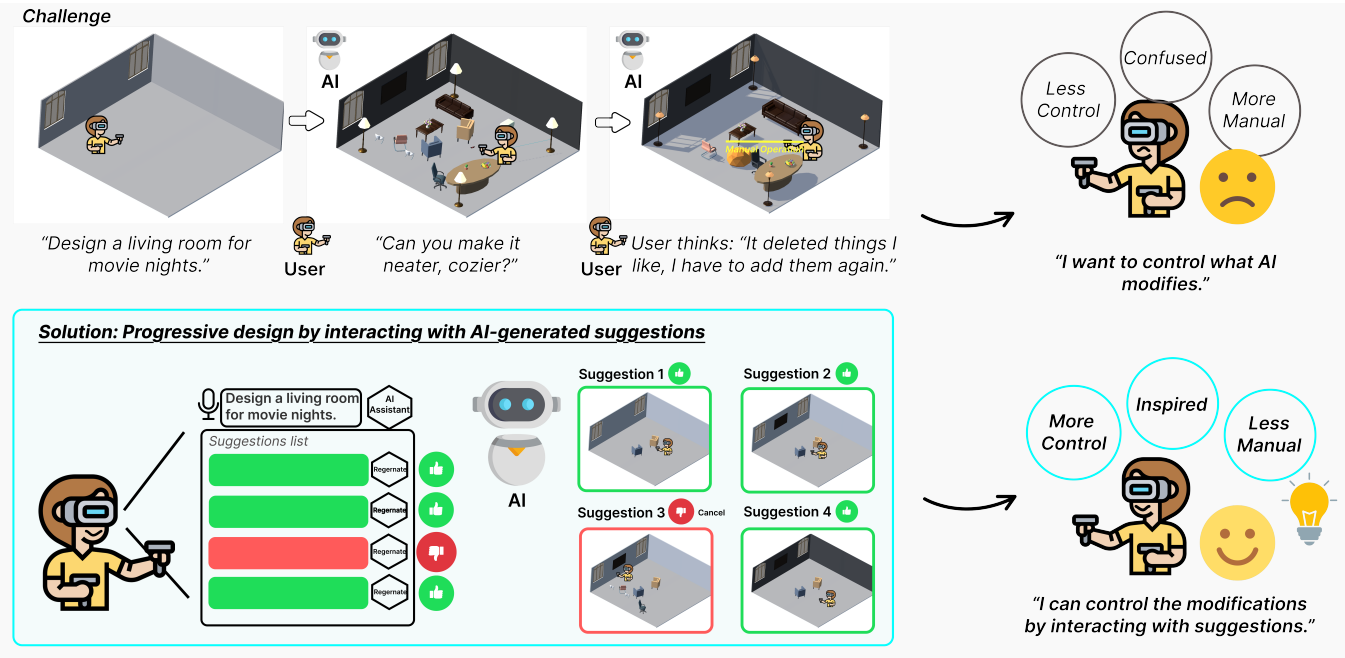


Figure 1: Immersive VR scene authoring with EchoLadder: EchoLadder makes the process of AI scene modification transparent by displaying interactable suggestion modules. Users can better control the AI modification process and modify the scene progressively.

Abstract

Mixed reality platforms allow users to create virtual environments, yet novice users struggle with both ideation and execution in spatial design. While existing AI models can automatically generate scenes based on user prompts, the lack of interactive control limits users' ability to iteratively steer the output. In this paper, we present EchoLadder, a novel human-AI collaboration pipeline that leverages large vision-language model (LVLM) to support interactive scene modification in virtual reality. EchoLadder accepts users' verbal instructions at varied levels of abstraction and spatial specificity, generates concrete design suggestions throughout a progressive

*Co-corresponding authors

UIST '25, Busan, Republic of Korea

ACM ISBN 979-8-4007-2037-6/25/09
https://doi.org/10.1145/3746059.3747659

design process. The suggestions can be automatically applied, regenerated and retracted by users' toggle control. Our ablation study showed effectiveness of our pipeline components. Our user study found that, compared to baseline without showing suggestions, EchoLadder better supports user creativity in spatial design. It also contributes insights on users' progressive design strategies under AI assistance, providing design implications for future systems.

CCS Concepts

• **Human-centered computing** → **Virtual reality**; **Natural language interfaces**; **Empirical studies in HCI**.

Keywords

AIGC, LVLMs, Progressive design, VR space authoring, Spatial design, Multimodal interface, Natural language input

ACM Reference Format:

Zhuangze Hou, Jingze Tian, Nianlong Li, Farong Ren, and Can Liu. 2025. EchoLadder: Progressive AI-Assisted Design of Immersive VR Scenes. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28–October 01, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3746059.3747659

1 Introduction

Recent advancements in generative 3D scenes, such as text-to-3D generation [11, 21, 26] and LLM-based scene design [12, 32], have introduced novel opportunities for AI-assisted VR authoring tools. By combining LLM understanding ability with VR authoring tools, these technologies enable users to craft intended immersive scenes more effectively. However, these automatic full-generation approaches predominantly follow a "black-box" generation model, limiting users to repeatedly re-generating or manually revising it post-hoc.

Some recent works have integrated interactive methods to support users in iteratively building scenes. For instance, VRcopilot [32] assists users in authoring VR layouts by allowing them to draw out areas or place wireframes to guide furniture generation, thereby supporting a scaffolded process. LLMR [6] allows users to use natural language to modify objects in mixed reality scenes by integrating object parameters in the generation pipeline. Such solutions could fundamentally improve the support for iterative content creation. However, while much work is focused on improving AI pipelines for automatic modification, one important question remains: how to support user intervention in that process via effective interface solutions?

To fill this gap, we explore a design concept inspired by the effectiveness of "chain of thought" [23], which is an established approach for improving the quality of AI generation. Previous works on AI-assisted writing identified benefits of exposing the thought process of generation to users. How would this translate to immersive spatial design? This research introduces a novel interface solution to support user intervention in AI-automated scene modification, to improve user agency and foster human-AI co-creation.

In this paper, we introduce a novel system named EchoLadder, based on a Large Vision-Language Model (LVLM), GPT-4o, to enable progressive and interactive scene construction in VR. Different from prior technical solutions [6, 32] executing users' concrete commands, EchoLadder interprets users' abstract instructions, combines scene images and object parameters to generate concrete modification suggestions. The suggestions are displayed as modular interactive widgets for users to selectively apply, undo, or regenerate. Once applied, the system executes a suggestion by modifying the scene. Users could view the direct visual effect and toggle to keep or retract the changes. This approach aims to provide both textual explanation and visual preview of each AI-generated step, while allowing selective and non-linear execution of them.

We evaluated EchoLadder through two studies. An ablation study assessed the impact of removing each input component of the pipeline—*visual input*, *object parameters*, and *AI suggestions*—on the quality of generation results. The finding showed that the full input configuration used by EchoLadder achieved the best performance. A second user study compared EchoLadder with a baseline, which leverages the same pipeline for automatic scene modification but does not display intermediate suggestions to users. Our findings revealed our suggestion-based interface solution could better support user creativity and control, while leading to some distinct differences in design strategies.

Our contributions are threefold:

- A novel interface solution supporting user intervention in iterative AI-automated authoring of VR scenes. It interprets users' natural language requests at any abstraction level and generates interactive suggestions for them to selectively apply.
- An LVLM model-based AI pipeline integrating real-time scene understanding and semantic object retrieval to generate 3D scene modifications responding to users' natural language requests. The effectiveness of each pipeline component is validated by an ablation study.
- Empirical findings from a comparative user study evaluating EchoLadder against an AI-modification baseline, demonstrating the effectiveness of our suggestion-based design, and providing insights into how participants used the system.

2 Related Work

2.1 AI-assisted scene generation

There has been existing research on AI-assisted interior design, both in 2D and 3D. Large models contribute a lot to 2D interior design. For example, an Interactive Interior Design Recommendation System [28] based on reinforcement learning. The interaction with the user taps into the potential preferences of the homeowner and selects the appropriate initial design. Virtual Interior DESign system [14], leveraging cutting-edge technology in generative AI to assist users in generating and editing indoor scene concepts quickly, given user text description and visual guidance. C2Ideas [12], through large models to better automate the generation of interior color design schemes that are more consistent with users' ideas.

Before the bloom of LLMs, traditional AI has been well used for 3D scenes synthesis. For example, the traditional method of using CLIPGraphs [2] can better estimate the position of objects in an indoor scene from a benchmark set of object categories. CompoNeRF [3], which interprets complex text into editable 3D layouts

and supports innovative multi-object composition. And a framework [33] for quickly synthesizing indoor scenes, measuring more reasonable relationships between objects through CSR, and generating various performations simultaneously in seconds. A system [7] for adaptive synthesis of indoor scenes given an empty room and only a few object categories. It exploit a database of 2D floor plans to extract object relations and uses the similar plan references to create the layout of synthesized scenes.

Benefiting from LLMs' strong ability in understanding and reasoning, some recent work on LLM-assisted 3D scene creation improves AI-assisted generation, making it more aligned with users' intentions. For example, Hong et al. proposed 3D-LLMs [11] that take 3D point clouds and their features as input and perform a diverse set of 3D-related tasks. Additionally, Sun et al. developed 3D-GPT [21], which integrates three core agents (the task dispatch agent, the conceptualization agent, and the modeling agent) to conduct an instruction-draven 3D modeling and positions LLM as problem solvers. Moreover, Yang et al. developed a fully automatic system, HOLODECK [26], that can generate diverse 3D scenes with user customized styles.

While bringing benefits, however, these previous works mainly employ end-to-end generation approaches that automatically generate entire scene in one go, instead of allow users to intervene the generation process. Inspired by this, our work bridges the gap by allowing users to interactively iterate and modify the generated scenes, with the system creating suggestions and modifications based on the real-time status of the scenes.

2.2 Immersive creativity support

With the development of VR technology, the demand for creation of virtual scenes has become more and more abundant. A lot of previous work has focused on the immersive creativity support for VR scenes. The traditional Visual Worlds in Miniature metaphor [20] provides a user interface technique for creating 3D scenes from different perspectives. Recently, there has also been a lot of work focused on improving the creative experience of VR immersion, Including creating more novel ways of interacting with VR/AR [16, 31, 34], as well as exploring the topic of collaborative immersive creation [8, 15]. For example, VRGit [29], a new collaborative VCS that visualizes version history as a directed graph composed of 3D miniatures, and enables users to easily navigate versions, create branches, as well as preview and reuse versions directly in VR. FlowMatic [31], an immersive authoring tool that raises the ceiling of expressiveness by allowing programmers to specify reactive behavior. Different from existing creative support in VR scenes, we leverage Vision-LLMs to support the creativity and manipulation of content creation.

2.3 AI-supported progressive scene crafting

The *progressiveness* of AI-supported content creation lies in its iterative refinement process, where users retain authority and guide the AI's execution. This progressive approach has demonstrated significant benefits across multiple HCI studies, such as writing stories with AI suggesting plot, style, and tone to facilitate rapid creative iteration [5, 27]. While the concept of *progressiveness* in AI generation of 3D space has not been widely explored, recent works have made significant contributions by bridging LLMs and 3D content creation. For example, researchers have proposed pipelines to enhance LLMs' understanding of user design intent and generate style-aligned scenes, as seen in HOLODECK [26]. 3D-GPT [21] supports iterative natural language commands to generate and author a 3D scene, eg. changing the color of generated flowers. Meanwhile, HCI research has focused on interactivity between users and LLMs to improve controllability. For instance, Zhang et al. introduced VR-Copilot [32], an LLM-assisted authoring system that improves 3D layout generation by allowing users to scaffold or guide the LLM's output via wireframing. Additionally, Torre et al. [6] presented the Large Language Model for Mixed Reality (LLMR), which supports scene understanding, task planning, self-debugging, and memory management, enabling users to create scene content iteratively by modifying object parameters. The novelty of EchoLadder lies in its focus on supporting user intervention in *automated scene modification* while interpreting abstract user requests. Our AI pipeline integrates real-time full-scene understanding (vision + object-level parameters) and semantic-matching object retrieval to enable automatically adding objects to reasonable positions. While existing works directly execute one authoring process following user commands, EchoLadder uniquely introduces interactive suggestions as intermediate explanation, visualization and control support to bridge automated execution and manual adjustment for each authoring operation.

3 EchoLadder

In this section, we present the system design of EchoLadder. EchoLadder is a novel AI-assisted VR scene design system that interprets user intent, generates context-aware design suggestions, and facilitates controllable, iterative construction in immersive environments. In EchoLadder, *Labeling Module* leverages LVLM to automatically annotate 3D assets with accurate object-matching information, enabling faster and more precise object retrieval while eliminating the need for manual selection or browsing by the user. *Generative Module* module understands natural language instructions, based on the current scene context (visual input and object parameters) to generate relevant suggestions for scene modification. It automatically handles object selection and editing, allowing users to freely express their ideas in language while the system performs spatial reasoning and executes the corresponding logic. Additionally, rather than executing user instructions directly, the system generates one or more suggestions for each instruction. Users can preview, apply, undo, or regenerate each suggestion individually, combining manual adjustments to iteratively construct the environment and refine their design decisions.

We position EchoLadder within the domian of interior design, a representative and well-established application area for immersive authoring tools [30, 32], such as Home Design 3D VR[1] and IKEA Virtual Interior Designer[2]. However, EchoLadder is not limited to this domain and can be generalized to other spatial design tasks. The prototyping system developed in this paper contains 2156 3D models for interior design from the Unity Asset Store includes a

[1]https://en.homedesign3d.net/vr
[2]https://present.digital/ikea/

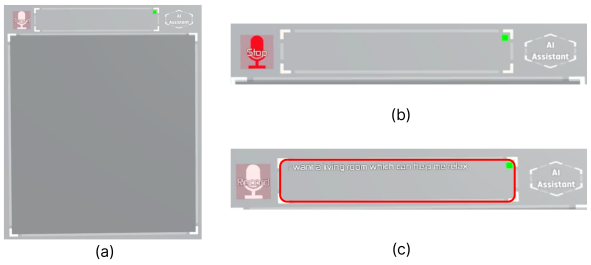
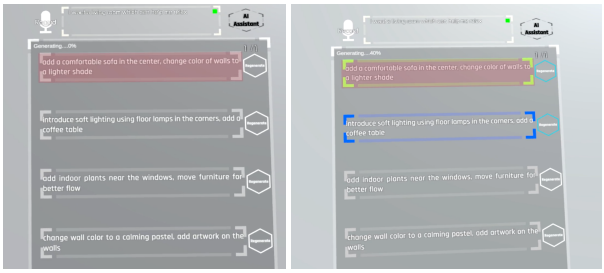


Figure 2: Voice input for user requests: (a) Initial state allows a user to input instruction after pressing the Mic button. (b) Mic button changes to Stop when accepting voice input. (c) Click Mic again to stop recording and check the transcribed user request. Clicking the AI Assistant button will start generating suggestions.



(a) (b)

Figure 3: AI-generated suggestions and their status in the interface. (a) Aystem displays the generated suggestions after interpreting the user request. (b) Three statuses of the suggestions: White (Generation of spatial modification in progress), Blue (Generation completed, pending application), Green (Applied).

wide range of furniture, decorations and textures. The scene modification operations supported by our system are derived from prior literature [29, 32] and existing VR application [1]. These operations encompass common object-level tasks in interior design, including: adding 3D objects (*Add*), modifying object positions (*Move*), rotating (*Rotate*), scaling (*Scale*), changing colors (*Color*), adjusting materials or styles (*Material*), and removing objects (*Destroy*).

3.1 Interaction Design

In this section, we describe interactions of EchoLadder.

3.1.1 ***Instruction Input.*** Considering the absence of physical keyboards in VR and the inefficiency of virtual typing [13, 18], we adopt intuitive voice [19] for user instruction input. As shown in Figure 2, once the interaction interface is activated, users initiate voice recording by selecting the microphone icon using either the controller button or raycasting. The icon turns red to indicate active listening. After issuing a instruction, users select the icon again to stop recording, upon which the system transcribes the voice input into text and displays it on the interface. Based on the transcription accuracy, users can either proceed with the next operation or re-record the instruction.

3.1.2 ***User Interaction with Suggestions.*** Our system uniquely introduces the display of interactive suggestions in order to show AI's reasoning process and selectively intervene in automated scene modification. Once the user confirms the instruction, they can select "AI Assistan" button on the interface. The *Generative Module* then parses the user instruction, object parameters and visual information to generate suggestions, which are individually listed on the interface (Figure 3). Once suggestions are generated, the user is offered three interaction options.

Browsing Suggestions. We provide two ways to help users browse suggestions, text reading and voice reading. Users can either read the textual content displayed in the interface or have the system read a suggestion aloud by clicking the joystick after scrolling down the list (by moving the joystick) to locate it.

Apply and Undo Suggestion. Users can decide whether to apply a suggestion to preview its effect within the scene. Since the system requires time to translate suggestions into executable actions, we use colored borders to indicate the current processing status. As shown in Figure 3(b), a white border (*Processing*) indicates that the AI is generating the corresponding actions in the background. During this phase, users can browse the suggestions but cannot apply them. Once the system completes the analysis and successfully generates the actions, the suggestion border turns blue (*Pending Application*), indicating that it is ready to be applied to the scene. After the user confirms the application and the actions are successfully executed in the VR environment, the suggestion border turns green (*Applied*). The user could view the post-modification effect in the VR environment. In addition, our system supports undoing applied suggestions. When users click on a previously applied suggestion, the system rolls back all modifications associated with that specific suggestion without affecting others. This immediate restoration of the VR scene enhances user control and flexibility.

Regenerate Suggestion. If users are not satisfied with the outcome of a generated suggestion, our system provides a suggestion regeneration function (via the "Regenerate" button next to each suggestion). This allows users to modify the result of a single suggestion without re-entering the original instruction. Upon triggering regeneration, the suggestion border turns white, and the *Generative Module* regenerates the corresponding actions based on the current scene context.

3.1.3 ***Immersive Manual Authoring Interactions.*** The system provides manual operations to support users to customize the design more freely. The manual operations include object modification and object addition:

Object Modification. The system provides the following manual operations for modifying virtual objects:

- Select objects using the ray on the right-hand controller of VR device.
- Manipulate objects by moving them along the handle ray, rotating them, or adjusting their size.

- Modify object properties, including color and material, or delete objects via the interface.

Manual Objects Addition. The manual object addition menu enables users to browse and select object categories for searching 3D models. Selecting a category displays a list of available objects under that classification. Users can then add an object by clicking on its corresponding preview image. It is important to note that the objects available for manual addition differ from those that can be automatically generated by the AI pipeline. This distinction ensures that users retain control over manual customization while leveraging AI-driven automation where necessary.

3.2 Technical Architecture

In this section, we introduce the architecture of EchoLadder in the immersive authoring system. There are two main modules: *Labeling Module* and *Generative Module*.

3.2.1 ***Labeling Module.*** For the system to automatically add objects that match user's requests based on a mixture of considerations such as function and style, we designed a *Labeling Module* to perform multi-dimensional, high-levelsemantic labeling for our 3D models. To the best of our knowledge, there were no publicly available datasets that meet our needs at the time we built the system. To be specific, the *Labeling Module* has two main parts:

Automatic Annotation of 3D Asset. Using LVLMs for image annotation has proven to be an effective approach [24, 25]. This part includes four steps. First, thumbnails of all 3D models in system asset repository are extracted as the dataset for labeling. Second, the LVLM (GPT-4o) is used to iteratively generate open-vocabulary labels for each 3D model. These annotations include the model name, category, and a description summary (e.g., functions, colors), all inferred from the visual content of the thumbnails. Third, the generated annotations are structured in JSON format and stored as text files. The prompts and example of JSON file are available in Appendix A.1. Finally, we manually reviewed the labeled content, with a particular focus on verifying the model description.

Asset Matching. As shown in Figure 4, when *Generative Module* outputs an action such as "*add a comfortable sofa in a neutral color*", it generates a corresponding description based on the user's intent and the specified action. It then passes the object name and description (e.g., "*sofa*") to *Labeling Module*. The *Labeling Module* first identifies the appropriate object category using pre-assigned labels, and then employs a natural language processing algorithm (Sentence-BERT, or S-BERT) to retrieve the most relevant asset from the 3D model library—such as a light gray fabric sofa—that best matches the generated description.

3.2.2 ***Generative Module.*** In this system, the *Generative Module* is responsible for interpreting user instructions, visual input and object parameters, subsequently generating scene modification suggestions along with specific execution actions. The entire process can be decomposed into the following four key steps:

User Intent Recognition and Scene Comprehension. The *Generative Module* first parses the input provided by the user (Appendix A.2). As shown in Figure 4, the input consists of three primary components: user intent (*Instruction*), visual information (*Top view image of scene*), and object parameters (Parameters for each attribute of scene objects) which includes objects' position, rotation, scale, color, and material (Style). After receiving these input, the *Generative Module* analyzes the current scene by combining the visual and object parameters to identify objects, their attributes, and spatial relationships. Finally, through multimodal reasoning, the system determines whether the user's instruction aligns with the existing scene and generates actionable suggestions accordingly.

Generate Reasonable Scene Modification Suggestions. The *Generative Module* formulates suggestions based on the previously analyzed information. As shown in Figure 4 (*Suggestions Generation*), the module infers the most reasonable modifications by considering both the user's instruction and the current scene context—such as adjusting the position, size, color of objects, or adding new elements to the scene. These suggestions are structured in JSON format (Appendix A.2) for downstream processing. The JSON-formatted suggestions are then parsed and converted to interactive buttons within the user interface. Finally, all suggested modifications are visually rendered, allowing users to browse and review them before making a decision.

Translating Suggestions into Executable Actions. At this stage, the *Generative Module* proceeds to the *Action Generation* phase (Figure 4, *Action Generation*), where it transforms the generated suggestions into actions that can be directly applied to the VR scene. The system leverages LVLM to understand the scene, then parse each suggestion into action lists in JSON format (more details in Appendix A.2). Each action represents an executable command generated by the *Generative Module* for a given suggestion, enabling automated scene modification. First, the module iterates through each suggestion, using the visual information, object parameters and suggestions as input, generates a set of concrete actions. These actions include *Move*, *Scale*, *Rotate*, *Color*, *Style*, *Delete*, and *Add* operations. For example, a suggestion like "*add a neutral-colored sofa*" could be translated into actions such as *Add [sofa]* and *Move [sofa] to* (x, y, z). By interpreting both the object parameters and visual information, the system tries to place objects at appropriate positions in reasonable sizes. For actions involving the addition of new objects, the *Labeling Module* is invoked to retrieve a suitable object from the 3D model library and return it to the *Generative Module*. Finally, all generated actions are scructured in JSON format and associated with their corresponding suggestion. When the user applies a suggestion, all linked actions are executed in sequence with in the VR scene.

3.3 Implementation

The prototype system presented in this paper was developed using Unity (version 2021.3.8f1c1) and integrated with SteamVR 2.8.0, enabling compatibility with both Meta Quest and HTC Vive headsets. The application featured advanced speech recognition and response capabilities through the integration of Whisper (Audio Model) and GPT-4o (Large Vision-Language Model, LVLM), allowing for natural and efficient user interaction. The system was deployed on a Windows 11 desktop equipped with an Intel Core i7-13650HX CPU, 32 GB of RAM, and a NVIDIA GeForce RTX 4060 GPU, which

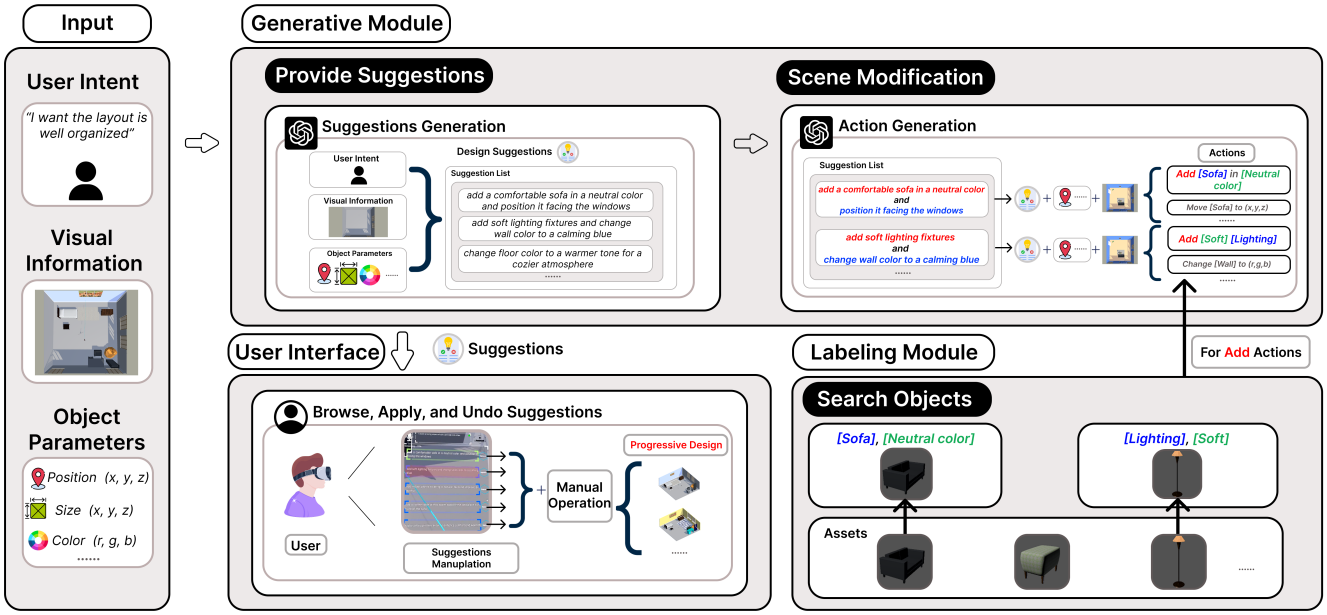


Figure 4: EchoLadder: In each iteration, the user provides a natural language instruction along with visual and object parameters as input. The *Generative Module* then produces design suggestions and corresponding actions. These suggestions are presented in the interface, and *Add* actions are linked to relevant assets by the *Labeling Module*. Users can browses, apply, undo and regernerate suggestions, optionally making manual refinements. This process results in a progressively and iteratively updated scene.

provided robust performance for real-time VR scene generation and interaction. To facilitates information transfer and processing for LVLM, we integrated OpenAI's GPT-4o API. Additionally, a socket-based Python server was established to execute the S-BERT (Sentence-BERT) algorithm, enabling efficient object search and matching during the *Add* operation. In user interactions, users can select objects, manipulate objects, and make UI selections for menus with the ray on right-hand joy stick. Users can open and close menus, select suggestions, and interact with suggestions with the buttons on the left-hand joystick.

4 Study 1 - Pipeline Evaluation

We conducted an ablation study to evaluate the impact of different input components in EchoLadder. Considering that the users' verbal instructions for spatial design can vary at abstraction levels and design goals, we designed two additional independent variables in this study. The study has two primary objectives:

- Evaluate the generation quality of our pipeline components by comparing scene modification quality across four input configurations.
- Examine how our pipeline performs for instructions with different abstraction levels.

4.1 Study Design

To evaluate the effectiveness of the pipeline input information proposed in this paper, we designed this ablation experiment. First, we tested the difference between the final scene results generated by four components conditions containing different information. The four components conditions are:

- Vision + object parameters + Suggestions ($V + OP + S$): Includes scene image information (Vision), scene object parameter information (object parameters) and AI-generated suggestions (Suggestions).
- Vision + Suggestions ($V + S$): Includes scene image information and AI-generated suggestions.
- Vision + Object parameters ($V + OP$): Includes scene image information and scene object parameter information.
- Object parameters + Suggestions($OP + S$): Includes scene object parameter information and AI-generated suggestions.

Then, in order to evaluate the impact of modifying scenarios with instructions, we first design our instruction list. Based on the three interior design requirement dimensions: functional requirements, aesthetic style, and psychological stimulus and meaning [4] and three different levels of natural language abstraction (Low, Medium, High) [17, 22], we designed 9 instructions (3 dimensions × 3 abstraction levels = 9 instructions) as shown in Table 1 and used each instruction to generate with the same initial scenario under four components conditions (36 results, 3 dimensions × 3 abstraction levels × 4 components conditions). Instructions differed between abstraction levels. We counteracted the effects of abstraction level and components conditions. The instructions were carefully crafted to differ across abstraction levels, allowing us to isolate and counterbalance the effects of both abstraction level and components condition on the final scene outcomes.

Design goal	Low Abstraction	Medium Abstraction	High Abstraction
Functional Requirements	*"Add a large screen TV on the wall opposite the couch."*	*"Set up a home theater area for movie nights."*	*"Design a space that brings the cinema experience home."*
Aesthetic Style	*"Change the sofa color to navy blue."*	*"Apply a nautical theme to the living room."*	*"Evoke the tranquility of the ocean in the living space."*
Psychological Stimulation	*"Place a small plant on the coffee table."*	*"Introduce elements of nature to enhance relaxation."*	*"Creating a spatial atmosphere that harmonizes with nature to promote balance and relaxation."*

Table 1: Natural language instructions tested in Study 1, categorized by design goals and levels of abstraction.

Measures	Questions
Relevance	How relevant is the scenario generation/modification to the instruction? (Q1)
Preference	How much do you prefer the generation/modification outcome in this condition? (Q2)
Reasonableness	How reasonable is the scenario generation/modification? (Q3)
Inspiration	How inspiring do you find this generation/modification outcome? (Q4)
Open-ended question	Why do you like and dislike the outcome of each condition? (Q5)

Table 2: Questions for Study1 questionnaire.

4.1.1 ***Task and Procedure.*** The evaluation procedure was designed to enable systematic comparison of scene generation quality across different input conditions. We presented participants with 36 generated scenarios (9 instructions × 4 input configurations) through a standardized slide deck. Each slide displayed an initial scene alongside its modified version generated through one input configuration, with both scenes shown from two distinct viewpoints to provide comprehensive visual context.

To minimize order effects, we employed a Latin square design to counterbalance both abstraction levels and instructions within each level (3 abstraction levels × 3 instructions = 9 sequences). we implemented full randomization of both instruction sequences and input component condition presentations for each participant. Before beginning the evaluation, participants were asked to review the evaluation guidelines and confirm their understanding of the scoring criteria with the experimenter.

During the study, participants proceeded through the slide deck in their randomized order. For each instruction set, they viewed all four component condition variants (labeled A-D) before providing ratings on a questionnaire. This grouped evaluation approach helped participants make relative judgments across conditions while maintaining the context of each design instruction. After assessing all visual materials for a given instruction, participants recorded their 5-point Likert scale ratings for each dimension and provided qualitative feedback through open-ended responses. The complete procedure required approximately 60 minutes per participant.

4.1.2 ***Data Collection.*** Participants evaluated generated scenes through a structured questionnaire (see Table 2) adapted from a recent work [12]. The questionnaire assessed four key dimensions below. Each dimension used a 5-point Likert scale (1 = "Strongly disagre" to 5 = "Strongly agree").

- **Relevance** measured how well the generated scenes matched the instruction intent. Participants evaluated whether elements like object selection and spatial arrangements properly reflected requests.
- **Preference** captured subjective satisfaction with the generated outcome in each scene.
- **Reasonableness** assessed physical plausibility, including proper object scaling, absence of collisions, and realistic material properties.
- **Inspiration** evaluated the novelty and creative potential of each output, determining whether results sparked new design ideas.

Open-ended responses were recorded via audio and later transcribed into text as participants' feedback.

4.1.3 ***Participants.*** We recruited 18 participants (5 female, 13 male) aged 20 to 30 years (*mean* = 25.5, *SD* = 1.98), recruited through university mailing lists. The participants have diverse academic backgrounds including Engineering, Design/Art, Biology, Chemistry and English. Ten participants reported prior experience with immersive space design applications.

4.2 Results

We collected a total of 2754 answers (9 instructions × 4 components conditions × 4 categories questions × 18 participants + 9 instructions × 1 open-ended question × 18 participants). Based on these results, we analyzed the data for different components conditions and abstraction levels.

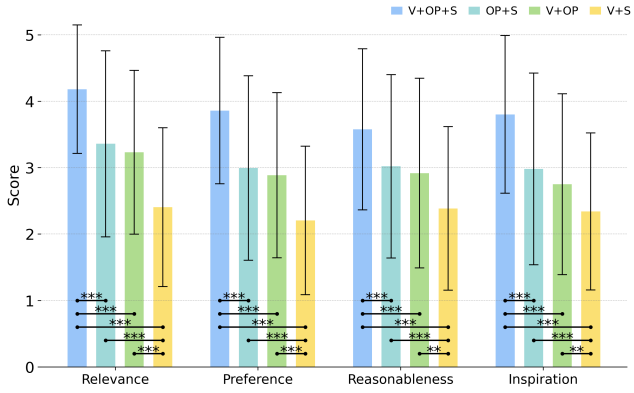


Figure 5: Results of Ablation Study comparing *Input Configurations*. The error bar represents the standard deviation. Statistical significant effects are marked (∗ = p < 0.05, ∗∗ = p < 0.01, ∗ ∗ ∗ = p < 0.001).

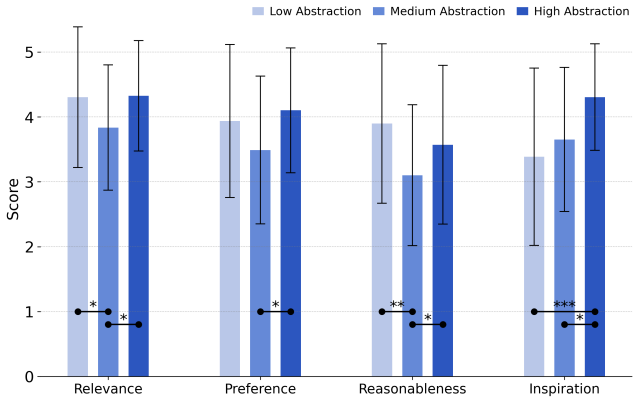


Figure 6: Results of comparing *abstraction levels* (only in $V + OP + S$ - full pipeline condition). The error bar represents the standard deviation. Statistical significant effects are marked (∗ = p < 0.05, ∗∗ = p < 0.01, ∗ ∗ ∗ = p < 0.001).

4.2.1 ***Evaluation of Pipeline Components.*** After confirming non-normal distribution, we employed the non-parametric Friedman test (Table 3) to detect significant effects in the evaluation results. For pairwise comparisons, we used the Wilcoxon signed rank test. We find that the component condition of EchoLadder ($V + OP + S$) shows the best performance in different categories. As shown in Figure 5, statistically significant comparisons are marked with stars. The evaluations of *Input Configuration* are as follow:

Full Pipeline: Vision + SceneInfo + Suggestions($V + OP + S$). Our full pipeline outperformed the other three ablated conditions across all the evaluation categories. As shown in Table 4 and Figure 5, this full input configuration achieved significantly higher scores in relevance, preference, reasonableness, and inspiration than all the other configurations. These results demonstrate that each of the three input components plays an essential role in the system's performance.

From open-ended questions in the questionnaires, we found this input configuration improved the ability of EchoLadder to generate thematically appropriate content with high relevance, such as beach and beach volleyball (Figure 7 (2)) and executing instructions with minimal deviation (Figure 7 (1)). It maintained spatial accuracy, with correctly placed and oriented furniture, and achieved stylistic consistency, rendering elements like nautical wallpapers and ocean-themed decorations that aligned with user-specified atmospheres (Figures 7 (2–3)). Finally, EchoLadder exhibited strong inspirational potential, introducing unexpected yet fitting elements—such as sand or a well—that enriched the scene and exceeded user expectations without sacrificing contextual fidelity (Figure 7 (2)).

Removing Object Parameters ($V + S$). This input configuration performed significantly worse than all the other conditions across all evaluation categories. This condition lacks object parameters, which likely undermines the ability of AI to interpret spatial context and generate coherent or appropriate modifications. Without access to parameters such as position and scale, AI struggles to produce relevant, well-aligned, or inspiring outcomes, despite having visual input and suggestions. The system struggled to identify relevant objects, resulting in poor thematic alignment. As shown in Figure 7 (5)–(6), objects in the scene are not coherent to the ocean theme. Layout errors are more severe, such as furniture overlap in Figure 7 (5)–(6) and illogical spatial arrangements.

Removing Vision ($OP + S$) and removing suggestions ($V + OP$). These two conditions had mediocre performances compared with other conditions. While $OP + S$ yielded slightly higher mean scores than $V + OP$ in all evaluation categories, the differences were not statistically significant. From the open-ended questions, we found that without vision ($OP + S$), it impaired the LVLM's spatial reasoning and color coordination. This was reflected in severe layout issues, including overlapping objects and irrational layouts such as the sofa overlaps with bookshelves in Figure 7 (11). Color perception suffered similarly, for ocean-themed modifications, results were reported as "not blue enough" in Figure 7 (7). On the other hand, the absence of suggestions ($V + OP$) constrained the LLM's creative capacity, producing minimal modifications, such as Figure 7 (8). It also frequently introduced contextually inappropriate objects like a drum kit in a ocean themed room in Figure 7 (9).

4.2.2 ***Effects of Abstraction Levels.*** We analyzed the effects of *Abstraction Level* on the generated results using only the data in the full pipeline condition ($V + OP + S$) (EchoLadder) across levels of abstraction. Same as section 4.2.1, we confirmed non-normal distribution of data, employed Friedman test to detect significant effects. For pairwise comparisons the Wilcoxon signed-rank test was used. While details of the statistical analysis results are in the Appendix A.3, we highlight the main findings here.

Overall we can see in Figure 6, the system performed better for instructions at low and high abstraction level than at medium abstraction for Relevance, Preference and Reasonableness, but not for Inspiration. Instructions at a higher abstraction level seem to generate more inspiring outcome.

Low Abstraction. We can see that low abstraction instructions shows high relevance score and reasonableness score which are

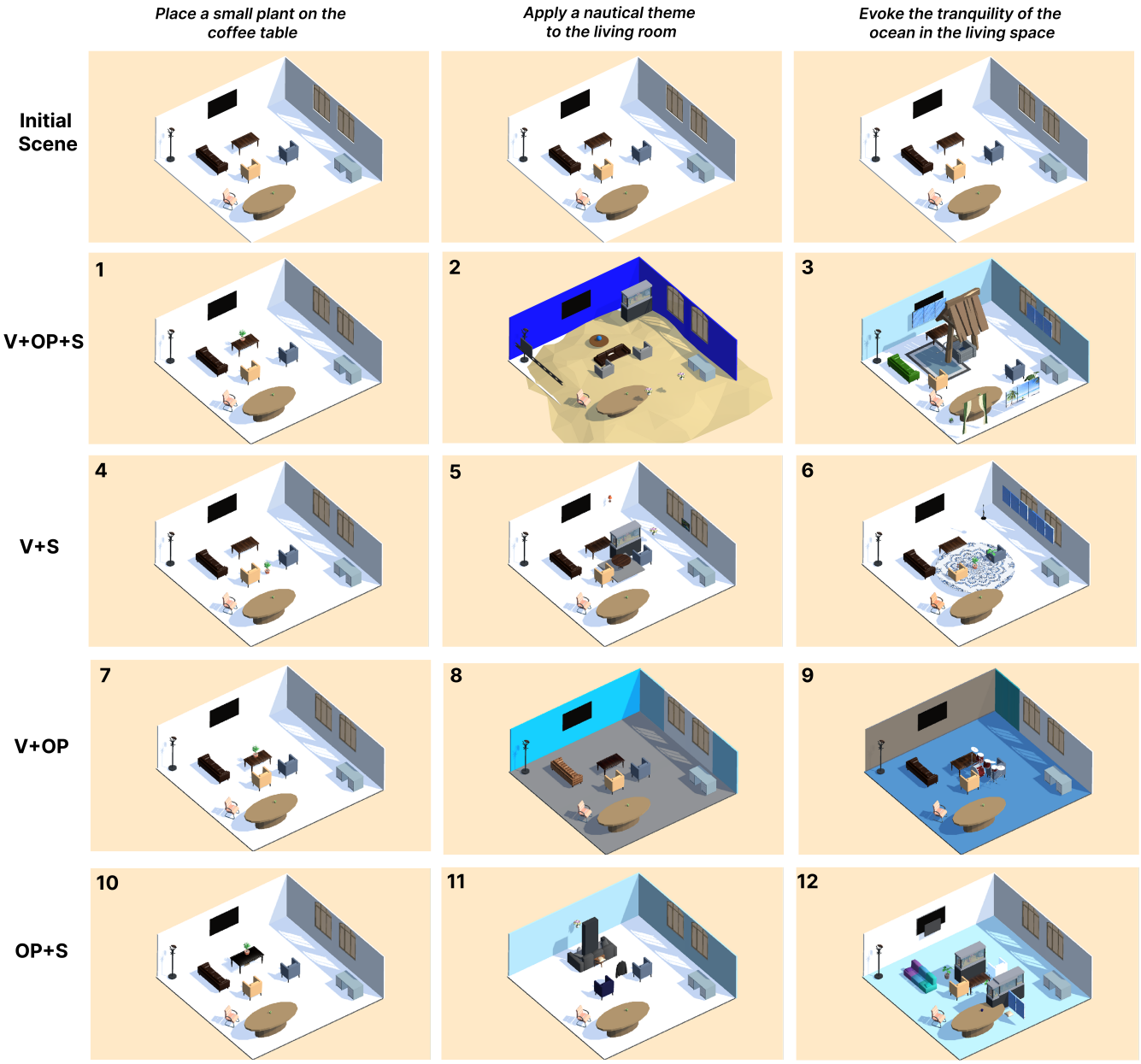


Figure 7: Results of scene modification under four pipeline input configuration conditions. The first row is the starting VR scene for all conditions, each subsequent row is for one condition. The columns from left to write show examples of instructions in low, medium to high abstraction level.

significantly higher than medium abstraction instructions. This might be caused by their explicit nature, yielding precise executions that from open-ended questions answers called "precise" and "directional". However, this specificity came at a cost: while functionally reliable, this condition scored poorly in *Inspiration* which is significantly lower than other abstraction levels, as the predictable outcomes of low abstraction instructions don't need additional operations, which offers little novelty.

Medium Abstraction. Our pipeline performed worse for medium abstraction instructions across metrics. For *Relevance & Reasonableness*, significantly lower than low abstraction and high abstraction (Figure 6). Medium abstraction instructions occupied an awkward middle ground — clearer than high abstraction but vaguer than

low abstraction. In this condition, we find answers of open-ended question were more sensitive to imperfections, such as one extra object will make the answer negative. This probably impacted *Preference*, where this condition was least favored due to perceived lack of creativity and inconsistent execution (Figure 7 (2), the sand is a little strange). Its *Inspiration* score suffered similarly, with outputs deemed neither reliably accurate nor meaningfully novel, failing to deliver on the strengths of either extreme.

High Abstraction. Our pipeline performed strongly across all metrics for high abstraction instructions. For *Relevance & Reasonableness*, the score of high abstraction is significantly higher than medium abstraction instructions in all categories and higher than low abstraction instructions in *Inspiration*. From the answers of open-ended questions, they valued the balance between thematic coherence and interpretive freedom, accepting minor inconsistencies when the overall atmosphere aligned with their vision (e.g., "extra content but the overall atmosphere was good"). In terms of *Preference*, this level was favored for enabling creative diversity, with even imperfect layouts often perceived as intentionally artistic. This condition outperformed both other abstraction levels in *Inspiration*, generating outputs described as "exciting" and "fresh", as the LLM's broad interpretations often surprised and delighted users.

5 Study 2 - User evaluation

The goal of this study is to evaluate the effectiveness of displaying AI-generated interactive suggestions following users' natural language instructions. We created a baseline condition to compare with EchoLadder for this purpose. The baseline is an automated VR scene modification approach based on the same backend pipeline of EchoLadder. It does not provide interactive suggestions, but directly modifies the scene following users' verbal instruction. To keep the final generation quality consistent across conditions, we kept the "suggestions generation" module in the baseline backend although suggestions are not displayed users.

The study aimed to address two core research questions:

RQ1. How does EchoLadder compare with the baseline in terms of user satisfaction, workload and experience?

RQ2. What creative strategies and workflows do participants adopt when using EchoLadder and the baseline, respectively?

5.1 Design

We designed a within-subjects experiment to compare EchoLadder and the baseline.

5.1.1 ***Participants.*** We recruited 12 participants (6 female, 6 male) aged 23-29 (M=25.7, SD=1.65) through university mailing lists and local VR interest groups. Screening ensured all participants had limited professional 3D design experience (<1 year) but were familiar with VR interfaces (10/12 reported regular HMD use). This profile represented our target user base of non-expert designers who might benefit from AI assistance. Participants received $20 compensation for the approximately 2-hour session, including training, tasks, and interviews.

5.1.2 ***Aparatus.*** The experiment used Unity 3D on Meta Quest 2 headsets, with researchers observing via a tethered laptop connection. Sessions were video recorded with participant consent.

5.1.3 ***Task.*** The same as Study 1, we developed three task types that reflected key design goal dimensions: functional, aesthetic and psychological stimulation. The study followed a structured protocol to ensure consistent data collection:

- **Functional Requirements:** Designing a living room/bedroom/study room with a XXX functional requirement.
- **Aesthetic Style:** Designing a living room/bedroom/study room with a XXX aesthetic style.
- **Psychological Stimulus and Meaning:** Designing a living room/bedroom/study room with a XXX psychological stimulus and meaning.

5.1.4 ***Procedure.*** To begin with, participants completed a training session by first watching a system introduction video covering EchoLadder and baseline interfaces as well as manual scene editing controls. They were briefed on the think-aloud protocols and study procedure. They were asked to perform training tasks that familiarized them with suggestion interaction and regeneration features.

In the main session, each participant completed two tasks, each of a different type, using both EchoLadder and baseline in a counterbalanced order. To reduce fatigue, participants were randomly assigned two out of the three available task types. This resulted in a total of four tasks per participant (2 types × 2 conditions). The tasks involved designing rooms with ≥ 10 objects, starting from an identical empty VR space.

After each task, participants completed the NASA-TLX questionnaire [10] to measure perceived mental, physical, and temporal demands. Participants rated both conditions on 7-point Likert scales across four dimensions (preference, inspiration, control, implementation) and participated in a semi-structured interview about their experiences.

5.1.5 ***Data Collection and Analysis.*** We employed a mixed-method approach to capture both behavioral and subjective measures:

Behavioral Data. System logs recorded timestamped interactions including: voice commands, suggestion applications/regenerations, manual edits (additions, deletions, transformations), and undo operations. These were synchronized with think-aloud audio and screen recordings for contextual analysis. We summarized patterns in design and operation strategies by triangulating behavioral logs and think-aloud observation.

Quantitative Measures. The post-task questionnaire assessed *User Satisfaction* using four 7-point Likert scales (1=low, 7=high) for user preference, inspiration, user control, and system execution. NASA-TLX scores measured cognitive load across six subscales (mental, physical, temporal demands; performance; effort; frustration).

Interview data. Two researchers independently coded 25% of the interview transcripts using thematic analysis and achieved

agreement. Discrepancies were resolved through discussion to finalize the themes.

5.2 Results

In this section, we present both quantitative and qualitative findings. The quantitative analysis is based on questionnaire ratings, while the qualitative insights are derived from behavior logs, think-aloud sessions, and interview transcripts. The think-aloud and interview data were analyzed using thematic analysis. Two researchers first reviewed the data from two participants to establish an observational protocol and identify initial themes. Once consensus was reached, one researcher proceeded to analyze the full dataset. We organize our findings based on our research questions in the following.

5.2.1 *User Satisfaction, Experience and Workload (RQ1).*

User Satisfaction, Control and Engagement. Based on user subjective ratings (as shown in Figure 8), EchoLadder achieved significantly higher Inspiration scores than baseline (mean = 5.75 vs. 4, p = 0.032), indicating stronger creativity support. Its average rating also outperformed the baseline on user preference, perceived user control and quality of suggestion execution.

In the interviews, 7 out of 12 participants agreed that EchoLadder fostered a stronger sense of agency in decision-making and content modification due to the flexibility of accepting, rejecting, or adjusting AI-generated suggestions. As P2 noted, EchoLadder felt *"more reassuring"* because it allowes *"make independent decisions and modify every AI-proposed change"*. Specifically, 6 participants appreciated seeing the AI's reasoning process through textual suggestions, which enhanced controllability through procedural visibility. As P9 stated: *"EchoLadder makes me feel like I'm a grading teacher and can see and interact with AI's throught process."* Additionally, 7 participants valued the ability to apply, undo, or regenerate suggestions, granting them operational flexibility and modification authority at each stage. For instance, P11 mentioned: *"Because suggestions can be undone and regenerated multiple times, I can try them one by one. If I don't like them, I can just stop generating"*. P8 added that most of the time LLMs fail to fully understand the intention so the option of selective generation allows them to *"cut losses midway"*. Overall, EchoLadder fosters stronger control and engagement and make user feel like *"I am the master, and it is just a tool"* (P9).

Perceived Work Load and Manual Operations. In terms of perceived workload measured through NASA TLX (Figure 11), we ran a Wilcoxon signed-rank test due to a violation of normality. We found no statistical difference between EchoLadder and the baseline on any of the measures. Qualitative findings revealed some potential causes of mental and physical load. Regarding mental load, while many appreciated EchoLadder's structured approach, some found it taxing: P1 compared it to "*converting word problems into multiple-choice questions*," and others (P2, P3) experienced decision fatigue from evaluating numerous suggestions. P3, a non-designer, felt it was "*more demanding*" and preferred the baseline's ready-made results. However, the baseline's "*all-at-once*" generation (P7, P9) often overwhelmed users when outputs missed their intent, sometimes discouraging iteration—as seen in P4's passive "*Just like it*" acceptance. In terms of physical load, both conditions required

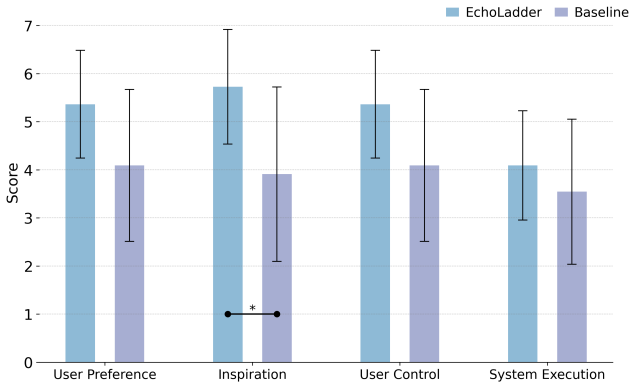


Figure 8: Subjective ratings from participants for EchoLadder vs. Baseline. The error bar represents the standard deviation. Statistically significant effects are marked (∗ = p < 0.05).

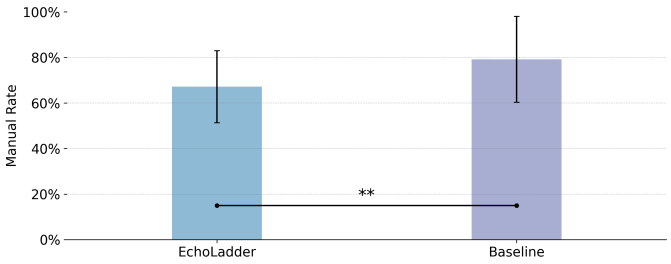


Figure 9: Average percentage of manual operations using EchoLadder and baseline. Statistical significant effects are marked (∗∗ = p < 0.01).

similar effort for fine-tuning, but the baseline demanded more corrective actions, with users frequently deleting misplaced objects (P1, P6, P11). EchoLadder reduced "*sunk effort*" (P9) by allowing early rejection.

To provide an additional perspective on workload, we computed the ratios of manual operations in each task from all participants' interaction logs. We examined the normality of the data and tested for significance by t-test. As illustrated in Figure 9, the manual operation rate of EchoLadder (EchoLadder: $mean = 0.6715, SD = 0.159$, Baseline: $mean = 0.7916, SD = 0.189$, $p = 0.003$) was significantly lower in Possible interpretations include EchoLadder saved more effort of trivial manual operations, and/or participants engaged more with hands-on crafting in the design process.

Effects of Providing Suggestions in EchoLadder. As providing interactive suggestions is the core feature of our system's interface innovation, we dived into analyzing how participants used it and understanding its effects.

We first calculated the *Suggestion Acceptance Rate* from the system log as the ratio of accepted suggestions (applied and not retracted) to the total suggestions provided per task. The overall acceptance rate averaged 70.2% (SD = 0.196), with 7 of 12 participants exceeding the mean, though individual rates varied widely

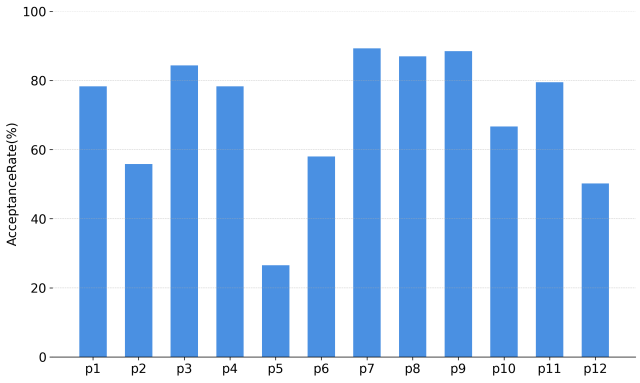


Figure 10: Average suggestions acceptance rate of each participant.

(e.g., 90% for P7 vs. 23% for P5), as shown in Figure 10. This indicates that our system-generated suggestions are generally reasonable and well-used. Three main reasons for rejecting suggestions are identified from interviews: (1) *Content mismatches*, where proposals conflicted with design intent (e.g., P12 dismissing a dining table—*"I don't want this"*, Style Task S1); (2) *Poor execution* of accepted suggestions, later undone due to unsuitable colors (P2/P8) or implausible layouts (*"This implementation is crazy"*, P2); and (3) *Redundancy*, where users ignored excess suggestions after core needs were met (P12 stopping after four proposals: *"Content is sufficient—I'll adjust manually"*).

Interview findings reflected three key benefits of the suggestion-based approach:

(1) *Decision Support & Creativity Stimulation.* Suggestions effectively scaffolded the design process by addressing both ideation bottlenecks and choice overload. For users struggling with initial direction (e.g., P2's uncertainty about color schemes or P12 *"didn't know what to do next"* shown in Figure 12 Task 2), suggestions provided concrete starting points. Conversely, when overwhelmed by options, participants like P4 appreciated how suggestions helped *"narrow thinking directions"*. The sequential nature of suggestions also fostered creative connections—P7's experience typified this, where an initial sofa color suggestion naturally led to complementary furniture adjustments, creating design coherence.

(2) *Spatial Awareness & Control.* Suggestions enhanced 3D scene manipulation through explicit spatial references. As P6 noted, directional cues helped users "accurately locate new elements" in the immersive environment. The modular workflow allowed incremental adjustments that participants found more manageable than holistic generation, e.g.,*"Modifying elements one-by-one gives better control"* (P7). However, this precision came at a cost: P7 and others reported *"higher workload from individual adjustments"*, indicating a trade-off between control and efficiency.

(3) *Design Intent Preservation.* Unlike baseline's monolithic generation, suggestions maintained stronger alignment with user intentions through stepwise refinement. When initial proposals missed the mark (e.g., P8's Chinese lantern suggestion), follow-up recommendations (e.g., matching bookshelf ornaments) enabled course-correction while preserving the overall design vision. This iterative process reduced the *"start-from-scratch"* frustration observed in baseline's workflows.

EchoLadder's *Undo* and *Regenerate* functions proved critical for managing AI-generated content, with the *Undo* feature used by 9 out of 12 participants to efficiently correct mismatches between suggestions and design intent, such as removing unsuitable furniture (P2) or reverting unwanted color changes (P5). Moreover, six participants also employed *Undo* as a diagnostic tool to detect scene changes outside their HMD's field of view, which is a known issue in 3D virtual environments [9]. The *Apply–Undo–Apply* strategy (e.g. Figure 12 P2 Task 1 & P12 Task 3) helped identify subtle modifications, addressing cases where changes were initially imperceptible or too minor to notice (P7,P8). This contrasted with the baseline, where P7 noted difficulty in tracking changes.

5.2.2 ***Creative Strategies and Workflows in the Two Conditions (RQ2).*** Our analysis revealed systematic differences in how participants approached scene design across the two conditions. These manifested in both design iteration activities and operational workflow patterns. Both conditions shared the same core design activities and showed slight differences in their iterative workflows and operational patterns. We also found similarities and differences in their creativity support, as reported below.

Common Design Activities. Participants engaged in three fundamental design activities regardless of the condition. *Global planning* involved high-level conceptualization of space functions and aesthetics, as exemplified by P2's comprehensive vision: *"First, I established this should be a war-themed bedroom with military decor"*. This typically preceded *targeted modifications* of specific aspects—P6's focused adjustment (*"Now make the walls camouflage green"*) being characteristic. Finally, all participants performed *object-level manipulations*, though with differing frequency; P8's precise placement (*"The TV needs 30cm clearance from the couch"*) typified this granular control.

Slightly Different Iterative Workflows. Building upon these design activities, we identified three composite workflow strategies. The most common was *top-down refinement*, where participants like P4 progressed systematically from global concepts to specific implementations as shown in Figure 12 P4 Task 1: *"I first defined a 'relaxing lounge' concept, then selected appropriate furniture styles, and finally adjusted individual pieces"*. This contrasted with the *focused execution* approach favored by P6 and others, who transitioned rapidly to object manipulation after minimal planning (*"I knew it needed a TV, so I placed it first and built around it"*).

Interestingly, the baseline enabled a unique *cyclic refinement* pattern absent in EchoLadder. As P9 described: *"I kept oscillating between adding artworks and tweaking their arrangements—each adjustment inspired new ideas"*. This back-and-forth process was observed in three participants, suggesting a more exploratory nature of using the baseline to design, in contrast to a more top-down approach with EchoLadder.

Different Operational Patterns. The systems elicited different operation patterns. EchoLadder users predominantly adopted either *sequential* (n=8) or *batch* (n=4) processing. Sequential users like P7 emphasized control: *"Applying suggestions one-by-one lets me catch issues early"*. Batch processors like P4 valued efficiency: *"I execute everything first, then clean up—it's faster overall"*. The baseline, by contrast, enabled three distinct modes. The *asynchronous* approach was observed in participants (n=5) like P6 (Figure 12 Task 1) multitasking during generation: *"While the AI worked on walls, I placed furniture"*. Others (n=7) preferred *post-generation review*, with P2 (Figure 12 Task 2 & 3) noting: *"I let the AI finish completely before making any edits"*. In addition, two *manual-centric* participants manually established bases before AI involvement. P9's explanation is: *"I needed to 'anchor' my vision first"*.

Creativity Support. Three types of creativity support emerged in both conditions. First, the system *clarified vague ideas* by materializing abstract concepts. Participants reported sudden clarity when seeing concrete suggestions, with P4 noting: *"The sofa and bookshelf suggestions made me instantly visualize arrangements"*. This effect was particularly strong for users with limited initial vision—P7, who struggled with sports-themed designs, found the treadmill suggestion pivotal, while P4 described how basic items like beds naturally prompted complementary additions (*"A rug underneath came to mind immediately"*). Second, AI proposals *broadened design considerations* by surfacing overlooked elements. P6's experience was typical: *"The system reminded me about lighting and contrast—aspects I'd neglected"*. This expansion occurred both for functional properties (visibility, spatial relationships) and aesthetic dimensions (color coordination, stylistic coherence). Third, users frequently *adopted unexpectedly fitting suggestions* that diverged from their initial plans. P6's incorporation of a chandelier in a princess-themed room exemplified this: *"I hadn't considered how crystal lights would perfect the Barbie aesthetic"*. Such discoveries often produced what P11 described as *"a sudden spark of inspiration"*.

The two conditions also supported creativity differently in some aspects. EchoLadder's textual suggestion lists provided conceptual starting points, while the baseline generation's concrete visuals stimulated more immediate reactions. P9 articulated the difference between the two modes: *"Text is abstract, but objects are tangible. With the baseline, I still need to mentally process what I see — but EchoLadder directly provides the thought process"*. This might explain the higher rating on Inspiration in the survey result (Section 5.5.1). However, this potential came with variability—P7 rejected an unwanted traditional Chinese style scene, but valued the *regenerate* option for managing unpredictability: *"Quick regeneration makes the AI's randomness feel controllable"*.

5.2.3 ***User-suggested Improvements.*** We collected the following user suggestions during their think-aloud and interview answers.

Selective scene iteration. Currently, EchoLadder's AI considers all objects in the scene when generating suggestions, which lacks the flexibility of allowing only specific objects to participate in iterative AI suggestions or targeting individual objects for refinement. On one hand, participants emphasized the need to preserve manually adjusted elements from further AI iterations. For instance, P1 stressed the importance of keeping *"my completed operations unaffected"* and having *"the option to selectively include elements for AI modification"*. Similarly, P1 and P7 expressed frustration when AI suggestions altered their manually optimized objects. On the other hand, P1 requested a *"focus mode"* for modifying individual objects without affecting others, noting: *"I need the ability to change the style of a single object independently without influencing other objects"*.

Iterative prompt crafting. Participants expected the ability to refine prompts iteratively. P1 wanted to *"iterate a prompt by modifying the current one"*, while P3 mentioned the need to *"regenerate a single suggestion if the output is unexpected"*.

Multimodal prompts. P1 highlighted the need for multimodal prompts, stating: *"I'd prefer prompts to include visual sketches alongside text"*.

Beyond first-person perspective. Participants identified challenges in spatial orientation within virtual environments. P7 noted, *"Operating on objects one by one in a first-person 3D space is fatiguing"*, while P5 remarked: *"I struggle to gauge the scale of the entire scene and would greatly benefit from a god's-eye view to navigate and inspect the space from above"*. These insights suggest a need for improved scene visualization tools to support navigation and decision-making.

6 Discussion

6.1 Summary of findings

Our evaluation of EchoLadder demonstrates its effectiveness in enabling progressive, user-guided VR scene design through AI-generated interactive suggestions. The ablation study confirmed the necessity of integrating visual input, object parameters, and AI suggestions (V+OP+S) to achieve optimal scene generation quality. This configuration outperformed ablated variants in relevance, reasonableness, and inspiration, particularly for high-abstraction instructions, where the system performed well at translating abstract concepts into coherent spatial designs.

The user study revealed that EchoLadder significantly enhanced user creativity and inspiration compared to the baseline generation, with participants leveraging suggestions to iteratively refine their designs while retaining agency. By providing textual suggestions, EchoLadder offers figurative hints that trigger spatial associations and conceptual thinking, introducing unexpected inspiring elements. Key interaction features—undo, regenerate, and selective application of suggestions—empowered users to experiment without fear of irreversible errors, fostering a sense of agency and engagement absent in baseline's workflows. By selecting, reading, and experimenting with suggestions, users gain a more active and deliberate role in shaping the scene.

6.2 Comparison with Prior Work

EchoLadder advances the field of AI-assisted spatial design by addressing a critical gap in existing systems: the lack of intermediate user intervention during scene generation. Unlike end-to-end approaches like HOLODECK or VRCopilot [32], which limit user input to initial prompts or post-generation adjustments, EchoLadder

externalizes the AI's reasoning process through actionable suggestions. This aligns with emerging HCI paradigms that emphasize progressive co-creation outside the spatial design scenarios [5, 27], where users iteratively steer AI outputs rather than passively accepting results. Our findings echo prior work on AI-supported writing tools [27], where intermediate suggestions stimulate ideation, but extend these principles to 3D spatial design by integrating multimodal reasoning and immersive interaction.

6.3 Design Implications

Intermediate Suggestions Enhance Creativity. Exposing AI-generated suggestions as modular, interactable components helps users bridge the gap between abstract ideas and concrete implementations. This approach not only mitigates the "blank canvas" problem but also introduces serendipitous elements that spark new ideas.

Flexible Control Mechanisms. Features like undo and regenerate reduce the cognitive cost of experimentation of ideas, enabling users to explore divergent design paths without friction. Future systems should prioritize such reversible interactions to balance automation with user agency. More exploration of design approaches facilitating quasi-execution could also be promising.

Abstraction-Aware AI Pipelines. EchoLadder's strong performance on high-abstraction instructions suggests that generative AI models are more suitable for supporting tasks with relatively abstract goals. Vague prompts could trigger broader exploratory suggestions, while concrete requests might prioritize precision. While aligning mental models is hard for human-AI collaboration, involving users in the decision making process can facilitate idea buy-in and co-creation. Perhaps for tasks in medium abstraction levels, AI systems could shift the discussion with users between abstraction levels for intent alignment.

6.4 Limitations and Future Work

While EchoLadder demonstrates promise, our studies highlighted areas for improvement. First is a potentially high cognitive load. Participants occasionally experienced decision fatigue when evaluating multiple suggestions. Future iterations could incorporate user intent modeling to prioritize or filter suggestions dynamically. Secondly, users desired finer control over which scene elements are modified by the AI (e.g., protecting manually adjusted objects). Implementing "focus modes" or exclusion zones could address this. In addition, participants also suggested integrating sketches or spatial gestures alongside voice commands to enrich expressiveness. These are all promising features to add in future systems. Moreover, there are other meaningful comparisons to evaluate our proposed system. For instance, comparing it with a similar system like LLM [6], which shares conceptual similarities but differs in technical components, could help assess different implementation choices.

7 Conclusion

This work designed, implemented and tested a novel system that enables progressive spatial design within the immersive VR environment. It differs from existing approaches by focusing on supporting iteration through AI-assisted modification rather than zero to one generation, which is achieved by enabling users to read and interact with the intermediate suggestions of AI automation. Our technical evaluation showed benefits of each of our pipeline component, while our user evaluation revealed benefits of providing this intermediate layer of interaction, including its better creativity support and user control. Our study also found that showing suggestions affected users' creative strategy by leaning more towards a top-down approach with global planning, while a baseline approach appeared more exploratory. Our findings underscore the value of progressive design workflows in immersive environments and provide a foundation for future systems that blend automation with embodied user agency.

Acknowledgments

This project was funded by the National Natural Science Foundation of China - Young Scientists Fund (CityU 62202397), and the Key Project of the Institute of Software Chinese Academy of Sciences (ISCAS-ZD-202401). Special thanks to Ruishan Wu and Chenyue Guo for contributing to early explorations of this work and to Dr Teng Han for fruitful discussions. We also thank our reviewers for putting forward valuable suggestions and all the participants in our user studies.

References

[1] 2023. Home Design 3D VR. https://en.homedesign3d.net/vr. Accessed: 2025-04-10.

[2] Ayush Agrawal, Raghav Arora, Ahana Datta, Snehasis Banerjee, Brojeshwar Bhowmick, Krishna Murthy Jatavallabhula, Mohan Sridharan, and Madhava Krishna. 2023. CLIPGraphs: Multimodal Graph Networks to Infer Object-Room Affinities. arXiv:2306.01540 [cs.RO]

[3] Haotian Bai, Yuanhuiyi Lyu, Lutao Jiang, Sijia Li, Haonan Lu, Xiaodong Lin, and Lin Wang. 2023. CompoNeRF: Text-guided Multi-object Compositional NeRF with Editable 3D Scene Layout. arXiv:2303.13843 [cs.CV]

[4] Francis DK Ching and Corky Binggeli. 2018. *Interior design illustrated.* John Wiley & Sons.

[5] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Visual Sketching of Story Generation with Pretrained Language Models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 172, 4 pages. https://doi.org/10.1145/3491101.3519873

[6] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. Llmr: Real-time prompting of interactive worlds using large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–22.

[7] Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. 2017. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Trans. Graph.* 36, 6, Article 201 (nov 2017), 13 pages. https://doi.org/10.1145/3130800.3130805

[8] Uwe Gruenefeld, Jonas Auda, Florian Mathis, Stefan Schneegass, Mohamed Khamis, Jan Gugenheimer, and Sven Mayer. 2022. VRception: Rapid Prototyping of Cross-Reality Systems in Virtual Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (, New Orleans, LA, USA,) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 611, 15 pages. https://doi.org/10.1145/3491102.3501821

[9] Uwe Gruenefeld, Ilja Koethe, Daniel Lange, Sebastian Weiß, and Wilko Heuten. 2019. Comparing techniques for visualizing moving out-of-view objects in head-mounted virtual reality. In *2019 IEEE conference on virtual reality and 3D user interfaces (VR)*. IEEE, 742–746.

[10] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[11] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 2023. 3D-LLM: Injecting the 3D World into Large Language Models. arXiv:2307.12981 [cs.CV]

[12] Yihan Hou, Manling Yang, Hao Cui, Lei Wang, Jie Xu, and Wei Zeng. 2024. C2Ideas: Supporting Creative Interior Color Design Ideation with Large Language Model. *ArXiv* abs/2401.12586 (2024). https://api.semanticscholar.org/CorpusID:

267095279

[13] Pascal Knierim, Thomas Kosch, Johannes Groschopp, and Albrecht Schmidt. 2020. Opportunities and challenges of text input in portable virtual reality. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*. 1–8.

[14] Minh-Hien Le, Chi-Bien Chu, Khanh-Duy Le, Tam V. Nguyen, Minh-Triet Tran, and Trung-Nghia Le. 2023. VIDES: Virtual Interior Design via Natural Language and Visual Guidance. arXiv:2308.13795 [cs.CV]

[15] Jaewook Lee, Raahul Natarrajan, Sebastian S. Rodriguez, Payod Panda, and Eyal Ofek. 2022. RemoteLab: A VR Remote Study Toolkit. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 51, 9 pages. https://doi.org/10.1145/3526113.3545679

[16] Michael Nebeling, Janet Nebeling, Ao Yu, and Rob Rumble. 2018. ProtoAR: Rapid Physical-Digital Prototyping of Mobile Augmented Reality Applications. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (, Montreal QC, Canada,) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173927

[17] Herbert A Simon. 1979. Information processing models of cognition. *Annual review of psychology* 30, 1 (1979), 363–396.

[18] Florian Spiess, Philipp Weber, and Heiko Schuldt. 2022. Direct interaction word-gesture text input in virtual reality. In *2022 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 140–143.

[19] Alex W Stedmon, Harshada Patel, Sarah C Sharples, and John R Wilson. 2011. Developing speech input for virtual reality applications: A reality based interaction approach. *International journal of human-computer studies* 69, 1-2 (2011), 3–8.

[20] Richard Stoakley, Matthew J. Conway, and Randy Pausch. 1995. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 265–272. https://doi.org/10.1145/223904.223938

[21] Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, and Stephen Gould. 2023. 3D-GPT: Procedural 3D Modeling with Large Language Models. arXiv:2310.12945 [cs.CV]

[22] John Sweller. 2011. Cognitive load theory. In *Psychology of learning and motivation*. Vol. 55. Elsevier, 37–76.

[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[24] Jiaxuan Wu, Xushuo Tang, Zhengyi Yang, Kongzhang Hao, Longbin Lai, and Yongfei Liu. 2025. An Experimental Evaluation of LLM on Image Classification. In *Australasian Database Conference*. Springer, 506–518.

[25] Shuo Yang, Zirui Shang, Yongqi Wang, Derong Deng, Hongwei Chen, Qiyuan Cheng, and Xinxiao Wu. 2024. Data-free Multi-label Image Recognition via LLM-powered Prompt Tuning. *arXiv preprint arXiv:2403.01209* (2024).

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

[27] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*. 841–852.

[28] He Zhang, Ying Sun, Weiyu Guo, Yafei Liu, Haonan Lu, Xiaodong Lin, and Hui Xiong. 2023. Interactive Interior Design Recommendation via Coarse-to-fine Multimodal Reinforcement Learning. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*. ACM. https://doi.org/10.1145/3581783.3612420

[29] Lei Zhang, Ashutosh Agrawal, Steve Oney, and Anhong Guo. 2023. VRGit: A Version Control System for Collaborative Content Creation in Virtual Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (, Hamburg, Germany,) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 36, 14 pages. https://doi.org/10.1145/3544548.3581136

[30] Lei Zhang, Ashutosh Agrawal, Steve Oney, and Anhong Guo. 2023. Vrgit: A version control system for collaborative content creation in virtual reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.

[31] Lei Zhang and Steve Oney. 2020. FlowMatic: An Immersive Authoring Tool for Creating Interactive Scenes in Virtual Reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 342–353. https://doi.org/10.1145/3379337.3415824

[32] Lei Zhang, Jin Pan, Jacob Gettig, Steve Oney, and Anhong Guo. 2024. VRCopilot: Authoring 3D Layouts with Generative AI Models in VR. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–13.

[33] Song-Hai Zhang, Shao-Kui Zhang, Wei-Yu Xie, Cheng-Yang Luo, Yong-Liang Yang, and Hongbo Fu. 2022. Fast 3D Indoor Scene Synthesis by Learning Spatial Relation Priors of Objects. *IEEE Transactions on Visualization and Computer Graphics* 28, 9 (Sep. 2022), 3082–3092. https://doi.org/10.1109/TVCG.2021.3050143

[34] Zhengzhe Zhu, Ziyi Liu, Tianyi Wang, Youyou Zhang, Xun Qian, Pashin Farsak Raja, Ana Villanueva, and Karthik Ramani. 2022. MechARspace: An Authoring System Enabling Bidirectional Binding of Augmented Reality with Toys in Real-time. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 50, 16 pages. https://doi.org/10.1145/3526113.3545668

A Appendix

A.1 Details of Labeling Module

Prompt:

System Prompt: Assume you're assisting users in automating picture labeling, You will receive a base64 code of a image. Based on all this data, generate the information of data as JSON format. The format should like:
{
"name":"object_name",
"description":"object_description",
"category":"object_category"
}.
Here, I will offer you the object_name, you should use it to generate the JSON. Description should only include the function, color, material, aesthetics and psychology of this object in the image, please use at most three simple sentences to finish the description, try to keep description very concise. Category is the category in reality of the object in the image. Categories such as "3D model", "3D shape" and so on are not be allowed. Do not generate extra string or information when you generate JSON.
User Prompt: object_name: Armchair1_C1
image: *the base64 code of model image.*

JSON Format Object Annotation:

{
"name": "Armchair1_C1",
"description": "This is a contemporary style armchair with a sleek black color finish, likely made of a material such as leather or synthetic upholstery. Its design is intended for comfortable seating with a modern aesthetic, potentially contributing to a sophisticated and minimalistic ambiance in a living space.",
"category": "Chair"
}

A.2 Details of Generative Module

Scene Understanding:

System Prompt: I will give you a list of objects in json format, includes the names, coordinate points, rotation vectors, sizes of the objects, and hexadecimal color codes of objects in the 3D scene, also I will provide you the top view picture of the 3D scene, please understand this scene, please understand this scene.
User Prompt: Object list: *JSON format objects' parameters.*
Top View Image: *the base64 code of top view image of scene.*

Suggestions Generation:

System Prompt: As a VR scene designer, you are presented with a detailed information of a 3D space scene. Your task is to interpret abstract user instructions for modifying this VR scene. Based on the scene's current layout, objects' attributes, and user commands, propose several creative and feasible suggestions for adjustments. These suggestions may involve repositioning furniture, altering object colors, adjusting sizes, or introducing new elements to enhance the space's functionality and aesthetic appeal. Ensure your proposals are clear, specific, and aligned with the user's desires, providing a blend of practicality and innovative design. Please provide modification suggestions and solutions with JSON format. For example, if you provide some suggestions, the result is:
{
"suggestions":[
{
"suggestion":"add something and move something, change color"
},
{
"suggestion":"add something and change color, also, change style"
},
{
"suggestion":"change color, destroy something"
},
....
{
"suggestion":"move something, rotate something"
}]
}
Each suggestions item can only include the suggestion, DO NOT include any other characters. Avoid extraneous text or characters outside the specified JSON format. The return format only includes JSON content, start with the first { of json.
User Prompt: User Instruction : *User Instruction*
Object list: *JSON format objects' parameters.*
Top View Image: *the base64 code of top view image of scene.*

Suggestions JSON format:
{
"suggestions":[
{
"suggestion":"add a large screen on Wall_N for a cinema effect and install surround sound speakers around the room"
},
{
"suggestion":"change the wall color to dark gray or black for an immersive cinema feel"
},
{
"suggestion":"rearrange the room by adding comfortable recliner chairs in front of the screen"
},
....
{
"suggestion":"adjust the ceiling height to accommodate a projector or large screen installation"
}]
}

Actions Generation:

System Prompt: Translate design suggestions into specific VR 3D space modifications based on JSON scene parameters. Output must strictly adhere to the JSON format below, detailing implementation steps for Add, Move, Rotate, Scale, Color, Style, and Destroy actions. You must remember DO NOT include other redundant text in the generated content, the return format only includes JSON content, start with the first "{" of JSON:

```
{
"steps": [{
"action": "Specify_Action_Name",
"action_command": "Action_Name {Object_Name} to [Modification_Value]",
"selected_obj": "Object_Name",
"key": "Modification_Value"
},
...
{
"action": "Specify_Action_Name",
"action_command": "Action_Name {Object_Name} to [Modification_Value]",
"selected_obj": "Object_Name",
"key": "Modification_Value"
}]
}
```

Notes: For Add Command: Set 'action_name' to "Add", use the format "Add {Object} to [(Position)]", and provide "key" with Vector3 position in (0,0,0) format as "Modification_Value". For Move Command: Use "Move {Object_Name} to [(New_Position)]" format. For Rotate Command: Use "Rotate {Object} [(Angle)]" format, specifying Vector3 angle in (0,0,0) format in "key". Make sure the back of objects facing the nearest wall. For Scale Command: Use "Scale {TV} [1.2] times", should specify scaling extent as an integer in "key". For Color Command: Use "Color {Table} to red[(255, 0, 0)]", color require RGB Vector3 in (0,0,0) format for Modification_Value. For Style Command, Use "Change {Table} to [Wood]", "key" is the material type as a string, including Basket, Black_Plastic, Brick, Bronze_Metal, Copper_metal, Dark_Oak, Flow_Water, Flower_Pattern, Glass, Glass_Dark, Golden_metal_material, Grass, Leaf_Pattern, Leather, Marble, Rustic_Wood, Shiny_Metal. For Destroy Command: "Destroy {Cup}", need "selected_obj", action command and key. If the object you want to manipulate does not exist in the scene, you will need to "Add" this object before you manipulate it. Do not forget {} and () Avoid extraneous text or characters outside the specified JSON format, the return format only includes json content, start with the first "{" of JSON"

User Prompt: Suggestion : *Suggestion*
Object list: *JSON format objects' parameters.*
Top View Image: *the base64 code of top view image of scene.*

Actions JSON format:

```
{
"steps": [
{
"action": "Add",
"action_command": "Add Movie_Poster to [(-3.80, 1.00, 0.05)]",
"selected_obj": "Movie_Poster",
"key": "(-3.80, 1.00, 0.05)"
},
...
{
"action": "Move",
"action_command": "Move Movie_Poster to [(-1.00, 1.00, -3.95)]",
"selected_obj": "Movie_Poster",
"key": "(-1.00, 1.00, -3.95)"
}]
}
```

For "Add" action, EchoLadder sends LLM the object name and categories list from our 3D model asset. LLM selects appropriate category and description for the object to be added based on context. EchoLadder searches for the object that best matches the description generated by the LLM among the responding category and adds it to the scene. The specific prompt is as follows:

System Prompt:
I will offer you a name of object, a list of categories, you should provide me with the perfect category that best fit the object and the description about the object, description should include the function, material, aesthetics and psychology of this object, please use at most three simple sentences to finish the description, try to keep description very concise.you give me categories you chosen and description as this JSON format:

```
{
"Category1":"Category1",
"Description":"description"
}
```

User Prompt:
The object is : object_name.
Categories include: category_list.

A.3 Statistical Data of Ablation Study

Category	Friedman Test			
	W	df	p	$\chi^2(3)$
Relevance	0.315	3	<0.001	148.53
Preference	0.295	3	<0.001	138.96
Reasonableness	0.128	3	<0.001	60.38
Inspiration	0.242	3	<0.001	113.91

Table 3: Friedman Test of scene modification with different components conditions.

	Relevance	Preference	Reasonableness	Inspiration
EchoLadder - V+OP+S	*mean* = 4.18, *SD* = 0.966	*mean* = 3.86, *SD* = 1.100	*mean* = 3.59, *SD* = 1.214	*mean* = 3.80, *SD* = 1.185
OP+S	*mean* = 3.37, *SD* = 1.402	*mean* = 3.01, *SD* = 1.394	*mean* = 2.99, *SD* = 1.441	*mean* = 2.99, *SD* = 1.441
V+OP	*mean* = 3.24, *SD* = 1.237	*mean* = 2.89, *SD* = 1.240	*mean* = 2.76, *SD* = 1.361	*mean* = 2.76, *SD* = 1.361
V+S	*mean* = 2.40, *SD* = 1.192	*mean* = 2.21, *SD* = 1.155	*mean* = 2.39, *SD* = 1.433	*mean* = 2.35, *SD* = 1.187

Table 4: The mean score and *SD* of each *Input Configuration* in different categories.

	Relevance	Preference	Reasonableness	Inspiration
Low	*mean* = 4.32, *SD* = 1.08	*mean* = 3.94, *SD* = 1.17	*mean* = 3.92, *SD* = 1.23	*mean* = 3.40, *SD* = 1.36
Medium	*mean* = 3.86, *SD* = 0.97	*mean* = 3.52, *SD* = 1.16	*mean* = 3.12, *SD* = 1.08	*mean* = 3.68, *SD* = 1.11
High	*mean* = 4.34, *SD* = 0.85	*mean* = 4.12, *SD* = 0.96	*mean* = 3.60, *SD* = 1.23	*mean* = 4.32, *SD* = 0.82

Table 5: The mean score and *SD* of different abstraction levels in different categories, in this table Low, Medium, and High are Low Abstraction, Medium Abstraction and High Abstraction.

Category	Abstraction Level	Friedman Test			
		W	*df*	*p*	$\chi^2(2)$
Relevance	L×M×H	0.099	2	0.007	9.94
Preference	L×M×H	0.094	2	0.009	9.373
Reasonableness	L×M×H	0.109	2	0.004	10.88
Inspiration	L×M×H	0.165	2	<0.001	16.513
Category	Abstraction Level	Wilcoxon signed-rank tests			
		W	*Z*	*p*	*r*
Relevance	L×M	180	2.800	0.015	0.396
	L×H	175	-0.026	1.0	0.004
	M×H	145	-2.568	0.031	0.363
Preference	L×M	217	2.103	0.106	0.297
	L×H	241	-0.724	0.468	0.102
	M×H	165	-2.702	0.021	0.382
Reasonableness	L×M	165	3.362	0.002	0.475
	L×H	297	1.327	0.554	0.188
	M×H	265	1.994	0.046	0.282
Inspiration	L×M	319	-1.100	0.814	0.177
	L×H	120	-3.550	<0.001	0.545
	M×H	107	2.450	0.043	0.453

Table 6: Statistical data of scene modification with different abstraction levels. In this table, L, M, H are Low, Medium, and High Abstraction.

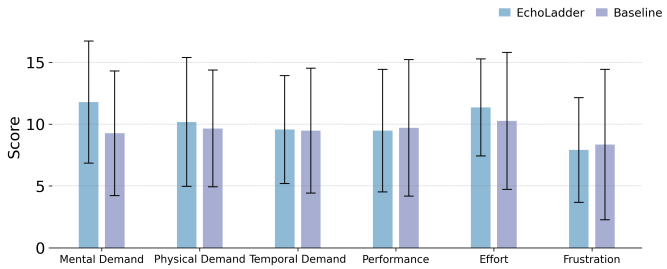
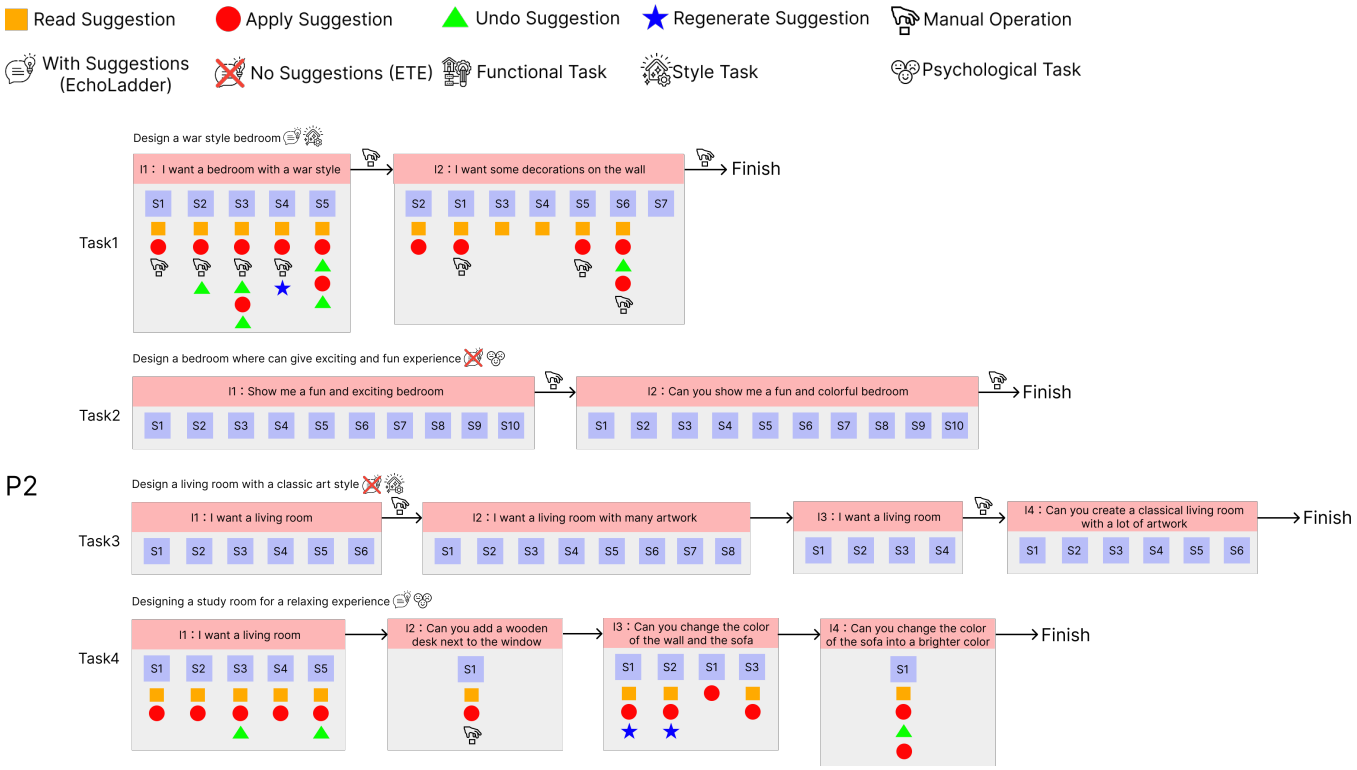
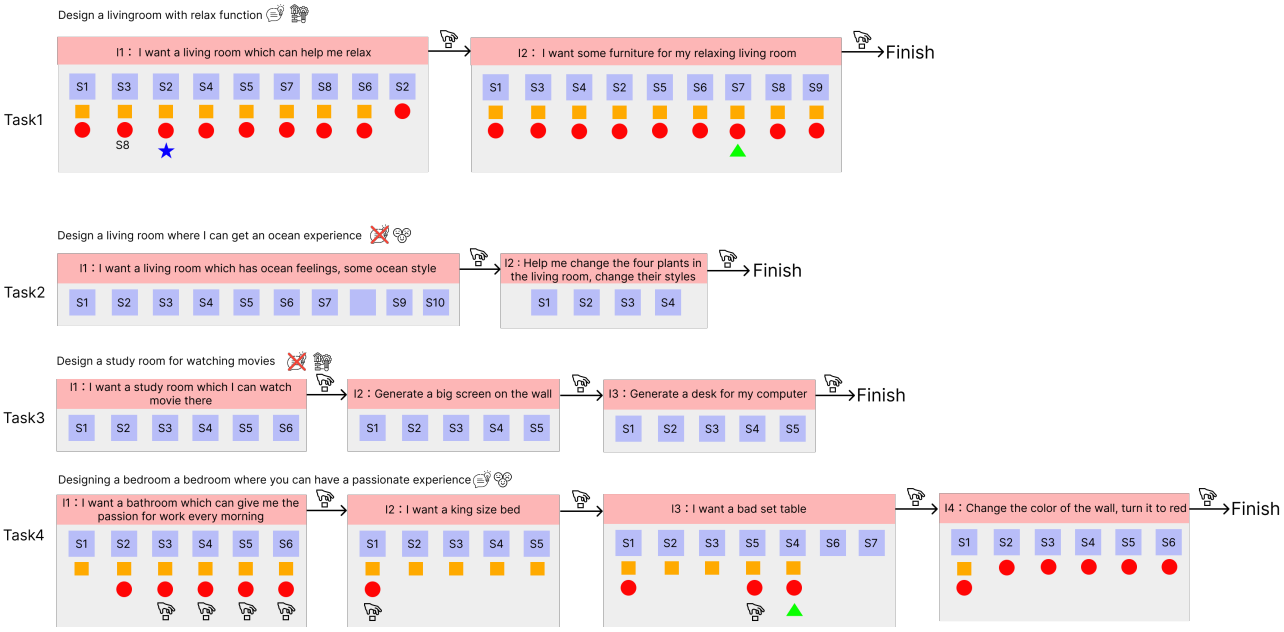


Figure 11: NASA TLX results for EchoLadder and baseline (Full score is 21).

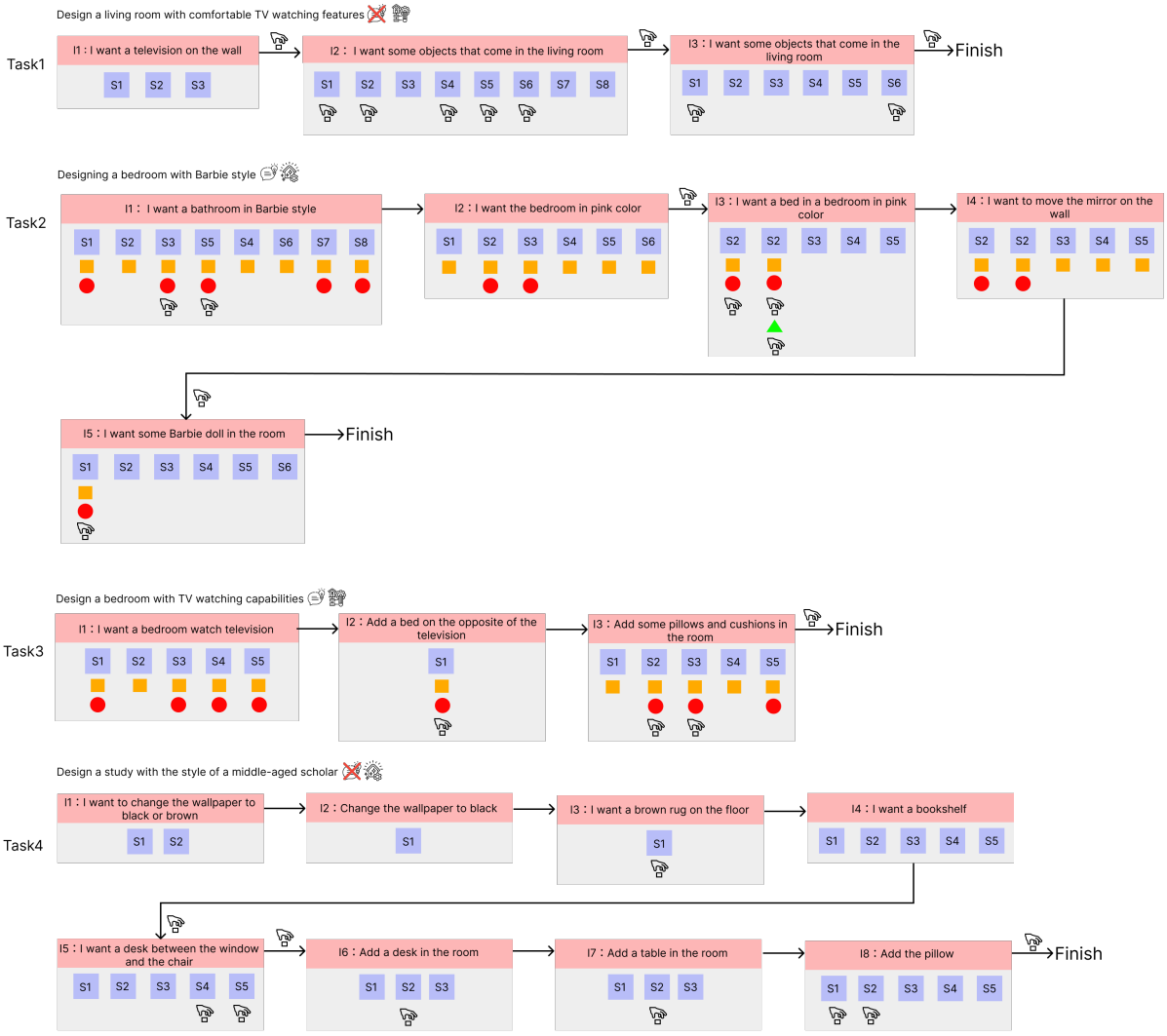
A.4 Participants' Workflow



P4


Design a livingroom with relax function
I1： I want a living room which can help me relax
I2： I want some furniture for my relaxing living room
Finish
Task1
Design a living room where I can get an ocean experience
I1：I want a living room which has ocean feelings, some ocean style
I2：Help me change the four plants in the living room, change their styles
Finish
Task2
Design a study room for watching movies
I1：I want a study room which I can watch movie there
I2：Generate a big screen on the wall
I3：Generate a desk for my computer
Finish
Task3
Designing a bedroom a bedroom where you can have a passionate experience
I1：I want a bathroom which can give me the passion for work every morning
I2：I want a king size bed
I3：I want a bad set table
I4：Change the color of the wall, turn it to red
Finish
Task4

P6


Design a living room with comfortable TV watching features
Task1
I1 : I want a television on the wall
I2 : I want some objects that come in the living room
I3 : I want some objects that come in the living room
Finish
Designing a bedroom with Barbie style
Task2
I1 : I want a bathroom in Barbie style
I2 : I want the bedroom in pink color
I3 : I want a bed in a bedroom in pink color
I4 : I want to move the mirror on the wall
I5 : I want some Barbie doll in the room
Finish
Design a bedroom with TV watching capabilities
Task3
I1 : I want a bedroom watch television
I2 : Add a bed on the opposite of the television
I3 : Add some pillows and cushions in the room
Finish
Design a study with the style of a middle-aged scholar
Task4
I1 : I want to change the wallpaper to black or brown
I2 : Change the wallpaper to black
I3 : I want a brown rug on the floor
I4 : I want a bookshelf
I5 : I want a desk between the window and the chair
I6 : Add a desk in the room
I7 : Add a table in the room
I8 : Add the pillow
Finish

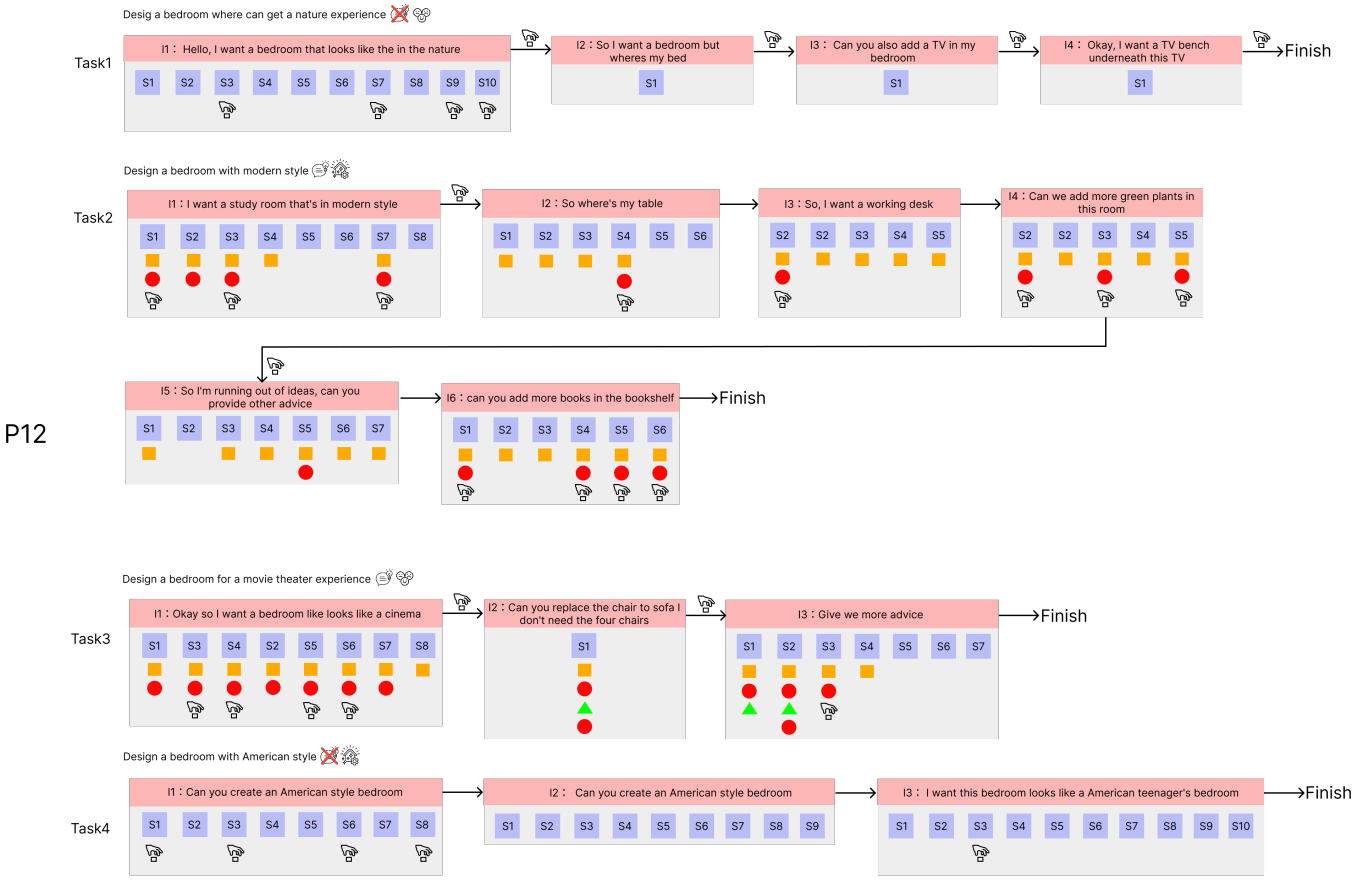


Figure 12: Example participants' workflows (P2, P4, P6, P12).