

---

# Indoor Scene Relighting Agent

---

**Abhilash Shankarampeta, Anandhini Rajendran, Humaira Mohammed, Yueqi Wu**

University of California, San Diego

{ashankarampeta, anrajendran, hfmohammed, yuw251}@ucsd.edu

## Abstract

Lighting in scenes plays a crucial role in many domains, including interior design, AR/VR environment editing, game design, and content creation. Currently, most lighting editing tools require professional knowledge of 3D software and involve complicated interfaces. Other existing methods often rely on pre-defined masks or normal maps, lacking intelligent region selection. In this project, we implement a pipeline that adjusts scene lighting effects based on user input. We first generate a scene using text-to-scene diffusion according to the user’s description, then segment the image. Next, an LLM is used to plan which regions should have their lighting adjusted, and the ScribbleLight model is applied to perform relighting. Finally, we use the CLIP model to evaluate the output. By using an LLM agent, users can generate and edit scene lighting through simple text input, providing a significantly more intuitive and user-friendly interface compared to traditional tools.<sup>1</sup>

## 1 Introduction

Lighting is important in visual representation and digital scenes perception. And lighting techniques are widely used in interior design, VR, AR, game design and all other scene generations. Proper and reasonable lighting can largely enhance realism. Lighting in images can convey mood and also guide user attention in a scene. As programs and softwares are used more and more widely to generate scenes, the demand for intuitive and realistic lighting editing tools is increasing. However, editing lighting remains a complex task for most users without 3D graphics experience.

Many existing lighting editing tools are built into professional 3D software such as Blender or Maya. These tools require expert knowledge and are not user-friendly for non-technical users. Some recent methods allow image-based lighting edits but rely on pre-defined masks or normal maps. For example, ScribbleLight uses normal maps and user-provided scribbles to control relighting effects, but the region to edit still has to be manually selected. These approaches often lack intelligent understanding of scene context or user intent. As a result, users must manually select and adjust regions, which limits flexibility and scalability.

In this project, we developed a user-friendly pipeline that enables lighting adjustments via natural language instructions. Our system allows users to describe the scenes they want in simple text, such as “make the lamp on the left brighter”. Based on the input, we automatically generate a scene, segment it, and use an LLM to select regions to relight. We then apply ScribbleLight to generate new lighting effects accordingly and choose the best as the output. This pipeline significantly eases the process of customizing lighting effects in visual scenes.

Compared to traditional tools, our system doesn’t require users to input masks manually or have 3D modeling skills. The pipeline is fully automatic and can handle diverse scenes and lighting instructions. By combining LLM reasoning with diffusion models and scribble-based relighting, we bridge the gap between users’ intuition and visual editing. This makes lighting editing more accessible for designers, artists, and general users.

---

<sup>1</sup>Code: [https://github.com/anan-123/Indoor\\_scene\\_relighting\\_llm\\_agent](https://github.com/anan-123/Indoor_scene_relighting_llm_agent)

## 2 Related Works

### 2.1 LLM Agents for Image Editing and Scene Generation

Several recent methods utilize large language models (LLMs) to enable image editing or 3D scene generation from natural language instructions. For instance, InstructPix2Pix[2] takes an input image and a text instruction, and uses a diffusion model to perform the desired edit. While intuitive and flexible, the quality of the generated outputs is constrained by the limitations of its training data, leading to inconsistencies in realism.

Another example is LLPlace[9], which proposes a lightweight LLM-based system (fine-tuned LLaMA3) for 3D indoor layout design. It allows users to generate and edit room layouts by specifying the room type and desired objects, without requiring spatial priors or few-shot exemplars. However, the model focuses primarily on object arrangement and layout logic, rather than producing photorealistic scene renderings.

### 2.2 Text-to-scene Generation

Several approaches use diffusion-based models to generate images or scenes from text prompts. For example, Text2Room[10] is a method for generating room-scale textured 3D meshes from textual descriptions. It leverages pre-trained 2D text-to-image diffusion models, such as Stable Diffusion, to synthesize a set of images from different camera viewpoints. These synthesized images are then back-projected and fused into a coherent 3D mesh using volumetric fusion and texture mapping techniques, enabling realistic room-scale reconstructions.

Another representative work is InstructScene[3], which proposes a generative framework for 3D indoor scene synthesis guided by high-level instructions. It combines a semantic graph prior and a layout decoder to ensure geometric plausibility and fine-grained control over object placement. The model interprets natural language instructions and translates them into spatial arrangements, making it suitable for controllable indoor scene generation tasks.

### 2.3 Segmentation for Editable Regions

Segmenting relevant regions as light sources is crucial for editing images under controlled conditions. One notable method is Grounded-SAM[7], which integrates Grounding DINO with the Segment Anything Model (SAM)[11]. Grounding DINO is capable of detecting objects based on arbitrary text prompts, while SAM performs high-quality, promptable segmentation across diverse visual domains. By combining these two models, Grounded-SAM allows users to segment any semantically meaningful region in an image simply by providing a text input, such as “lamp” or “window.” This approach bridges the gap between vision and language, enabling flexible and generalizable region selection, which is especially useful for downstream tasks like relighting or material editing.

### 2.4 Image-based Relighting

Image-based relighting aims to adjust lighting effects in existing images without requiring full 3D reconstruction. ScribbleLight[4] is a lightweight yet effective method that supports local, fine-grained lighting edits through user-drawn scribbles and normal maps. It introduces two key components. One is an Albedo-conditioned diffusion model that preserves the intrinsic color and texture of the scene while applying lighting changes. The other is a ControlNet-based encoder-decoder architecture that integrates normal and scribble inputs to maintain spatial consistency and geometry awareness.

In contrast, methods like PhotoScene[12] and Neural LightRig[5] pursue higher photorealism by using more detailed scene understanding. PhotoScene builds a digital twin of an indoor environment by combining input images with roughly aligned CAD geometry and performs material and lighting transfer based on the reconstructed scene structure. Neural LightRig leverages multi-light priors from 2D diffusion models to generate multiple consistent lighting variants of a scene, which are then used to improve the accuracy of intrinsic property estimation via a G-buffer prediction network based on a U-Net backbone.

Together, these methods represent two ends of the spectrum: ScribbleLight prioritizes flexibility and lightweight user interaction, while PhotoScene and Neural LightRig focus on physical realism and precise control through detailed scene priors.

## 3 Method

### 3.1 Overview

The overview of our method is shown below:



### 3.2 Interior Design Scene Generation

The interior design scene generation component forms the creative heart of our pipeline, transforming simple user descriptions into photorealistic interior spaces through an intelligent combination of language understanding and image synthesis. To understand why this system is necessary, consider the challenge facing any user trying to describe their ideal interior space. While humans naturally think in terms of mood and function ("I want a cozy bedroom"), AI image generation models work best with precise, technical descriptions that specify materials, lighting conditions, and spatial arrangements.

Our system bridges this fundamental gap by acting as an intelligent translator between human intuition and machine precision. Rather than forcing users to become prompt engineering experts, we create a system that understands the design intent and automatically generates the detailed specifications needed for high-quality image synthesis.

#### 3.2.1 Understanding the Prompt Enhancement Challenge

Before diving into our technical approach, it's important to understand why direct text-to-image generation often fails for interior design applications. Consider what happens when someone inputs "cozy living room" into a standard diffusion model. The model might generate a generic space that lacks the specific design choices that create true coziness—the warm color temperature of lighting, the texture contrast between soft fabrics and natural wood, or the careful arrangement of furniture that encourages conversation.

This failure occurs because diffusion models, despite their impressive capabilities, operate as sophisticated pattern-matching systems. They excel when given detailed, structured descriptions that align with their training data, but struggle with abstract concepts like "cozy" or "elegant" that require design expertise to translate into specific visual elements.

Our prompt enhancement system using Qwen2.5-VL-3B [1] addresses this challenge by functioning as a virtual interior design consultant. When a user provides a basic description, the system draws upon its training in both language understanding and visual design principles to expand that description into a comprehensive design specification.

The enhancement process operates through a systematic transformation of user input. When someone provides a basic description such as "modern minimalist bedroom with warm lighting," our system methodically expands this into a comprehensive design specification. Think of this process as similar to how an experienced interior designer might interview a client—taking their general preferences and translating them into specific design decisions.

Our system ensures that every enhanced prompt addresses eight fundamental design elements that professional interior designers always consider. First, we establish the specific design style, moving beyond generic terms like "modern" to precise classifications such as Scandinavian, Industrial, or Mid-Century Modern. This specificity matters because each style carries distinct visual conventions that the diffusion model can recognize and reproduce.

Next, we define a cohesive color palette of three to five specific colors rather than vague descriptions like "warm colors." This approach ensures visual harmony throughout the generated space. We then specify exact furniture pieces and their material compositions, detail both natural and artificial lighting sources, describe flooring and wall treatments alongside fabric choices, include accessories that enhance the space's character, capture the emotional quality or mood of the space, and finally optimize the camera angle and composition for the most effective visual presentation.

To illustrate this transformation process, consider how the simple prompt "cozy living room" might be enhanced to "Scandinavian-inspired living room with cream and sage green palette, natural oak furniture, soft pendant lighting, wool textures, minimalist ceramics, serene atmosphere, wide-angle architectural photography perspective." Notice how each element of our eight-point framework contributes specific visual information that guides the generation process toward a coherent, professional-quality result.

### 3.2.2 FLUX.1-dev Integration: Building an Image Generation Pipeline

Selecting the right diffusion model for interior design generation requires understanding the unique challenges of architectural visualization. Unlike portrait photography or landscape images, interior spaces demand a precise understanding of spatial relationships, lighting physics, and design conventions. After evaluating several state-of-the-art models, we chose FLUX.1-dev [6] with LoRA<sup>2</sup> that have been specifically trained for interior design scenarios for its superior performance in architectural and interior design visualization.

#### Ensuring Consistent Quality Through Dual-Prompting Strategy

Diffusion models operate in a high-dimensional space where small changes in input can lead to dramatically different outputs. To achieve consistent, professional-quality results, we implement what we call a dual-prompting strategy that constrains the model's output space from both positive and negative directions.

The positive constraint comes from our enhanced prompts, which guide the model toward desired characteristics. The negative constraint uses carefully crafted negative prompts that explicitly exclude undesirable characteristics such as "low quality, blurry, distorted, amateur, cartoon, painting, sketch." This bidirectional approach helps ensure that every generation meets professional standards by actively preventing common failure modes.

### 3.2.3 The Challenge of Iterative Design: Moving Beyond One-Shot Generation

Traditional text-to-image systems operate on what we might call a "one-shot" model: you provide a description, receive an image, and if you're not satisfied, you start over with a completely new prompt. This approach fundamentally misunderstands how creative design actually works. Real interior design is an iterative process where designers and clients collaborate, making incremental adjustments until they achieve the desired result.

Our feedback processing system addresses this limitation by taking user feedback and automatically translating that feedback into appropriate prompt modifications.

#### Understanding Design Intent Through Vision and Language Integration

When a user says "add more plants and make it warmer," they're providing feedback that operates on multiple levels simultaneously. On the surface level, they're requesting specific objects (plants) and a general atmospheric change (warmth). But experienced designers understand that "warmer" might refer to color temperature, lighting intensity, material choices, or even spatial arrangement. Our system needs to interpret this feedback in the context of the current design to make appropriate modifications.

This interpretation process begins with Qwen2.5-VL's vision capabilities, analyzing the current generated image. The model examines existing design elements, their spatial relationships, current color palette, lighting conditions, and overall aesthetic coherence. Simultaneously, it processes the user's feedback text, identifying both explicit requests (specific objects or changes) and implicit preferences (mood or style adjustments).

---

<sup>2</sup><https://huggingface.co/SedatAI/Interior-Flux-Lora>

Rather than simply adding new elements to the existing prompt, our system performs contextual reasoning about how the requested changes will interact with existing design elements. For example, if a user requests "more plants" in a minimalist space, the system understands that the plants should align with minimalist principles, perhaps selecting clean-lined planters and architectural plant varieties rather than ornate arrangements that would conflict with the established aesthetic.

Our rewriting algorithm considers four key factors during this process. First, it identifies elements explicitly mentioned in the feedback that require modification, treating these as primary objectives for the new generation. Second, it analyzes successful design elements that should be maintained, ensuring that positive aspects of the current design don't accidentally disappear in the revision process. Third, it derives implicit style preferences from the feedback language, understanding that how someone describes changes often reveals deeper aesthetic preferences. Finally, it considers spatial and aesthetic constraints of the existing layout, ensuring that new elements will integrate harmoniously with the established spatial framework.

This multi-factor analysis allows our system to generate new prompts that feel like natural evolution of the design rather than completely new creations. Users experience a sense of collaborative refinement rather than starting over with each iteration, which more closely mirrors how human designers actually work with clients.

### **3.2.4 Session Management: Supporting the Natural Flow of Creative Work**

Creative design work does not happen in isolation, it unfolds over time through exploration, reflection, and refinement. Understanding this reality, our system implements comprehensive session management that supports the natural rhythms of creative decision-making. Rather than treating each image generation as an independent event, we create a continuous design experience that remembers, learns, and evolves with the user's preferences.

Think of our session management as creating a digital design studio where all previous work remains accessible and informative. Each design session maintains four critical types of information that together create a rich context for ongoing work. The complete prompt evolution history captures not just the final prompts, but the reasoning process that led to each modification. This historical record allows users to understand how their design evolved and potentially revert to earlier directions if needed.

The generated image sequence with metadata provides a visual timeline of the design process, complete with the technical parameters used for each generation. This information proves invaluable when users want to understand why certain generations were more successful than others or when they need to reproduce particular effects in future work.

Current design state tracking ensures that the system always understands exactly where the user is in their design journey, maintaining context about the active iteration, applied feedback, and pending refinements. Finally, user feedback patterns and preferences create an implicit profile of the user's aesthetic preferences that can inform future suggestions and improvements.

This persistent state enables several advanced capabilities that transform how users interact with the system. Users can revert to previous design iterations without losing subsequent work, allowing for safe experimentation with new directions. The system begins to learn from within-session feedback patterns, becoming more adept at interpreting the user's design language over the course of the session. Design evolution can be analyzed and optimized, helping users understand which types of modifications tend to produce their preferred results. Most importantly, the final designs include complete provenance information, creating a complete record of the creative process that led to the final result.

### **3.2.5 Implementation Architecture**

Our implementation uses a modular architecture that separates concerns while maintaining efficient communication between components. This approach ensures that individual components can be updated, optimized, or replaced without affecting the entire system.

The Agent System implements the core intelligence of our design assistant, built using the CrewAI<sup>3</sup> framework to provide structured tool usage and clear separation between different AI capabilities. This system orchestrates the complex interactions between prompt enhancement, image generation, feedback processing, and iterative refinement. By using a formal agent framework, we ensure that the system’s decision-making process is transparent, debuggable, and extensible.

Our Tool Framework breaks down complex design tasks into discrete, manageable operations that can be combined flexibly to support various user workflows. The PromptEnhancementTool transforms user descriptions into detailed generation prompts, applying our eight-point design framework systematically to ensure comprehensive coverage of design elements. The ImageGenerationTool manages FLUX.1-dev inference with careful parameter control, ensuring consistent quality while optimizing for available hardware resources. The FeedbackAnalysisTool processes user feedback and generates improved prompts, implementing the sophisticated contextual reasoning we discussed earlier. Finally, the SaveFinalImageTool handles final design persistence with complete metadata, ensuring that finished designs can be archived, shared, and referenced in future work.

The Web Interface, built using Gradio<sup>4</sup>, provides intuitive user interaction while maintaining full access to the system’s advanced capabilities. This interface supports real-time feedback display, design history browsing, and final design export, all while hiding the technical complexity of the underlying AI models from users who simply want to create beautiful interior spaces.

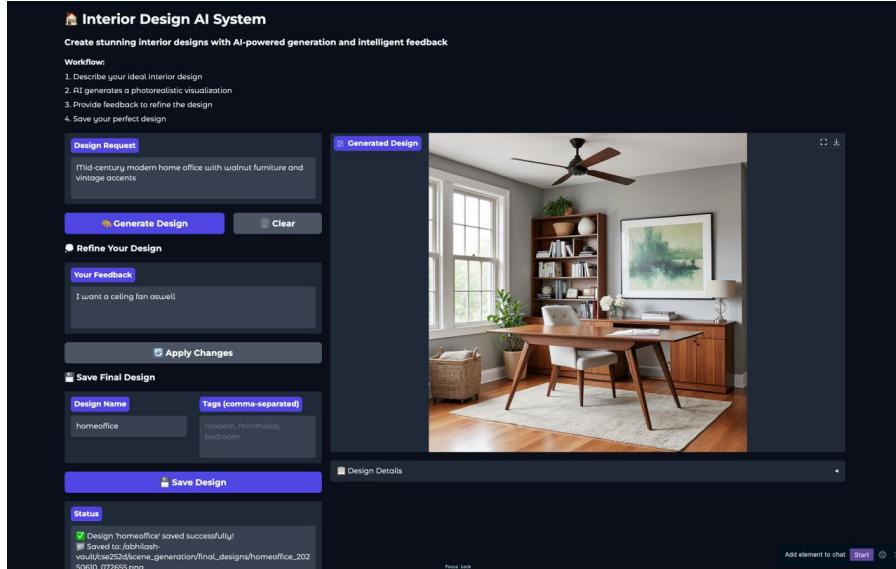


Figure 1: System user interface showing the complete workflow: input area for design descriptions (left-top), feedback input section (left-middle), design management controls (left-bottom), and generated image display (right). The interface provides intuitive access to all system capabilities while maintaining a clean, professional appearance.

### 3.3 Segmentation

We first use Grounding DINO, a state-of-the-art open-set object detector that aligns vision and language, to detect candidate regions in the generated scene. Given the image and a list of textual prompts (e.g., “window”, “bed”, “wall”), Grounding DINO outputs a set of bounding boxes along with confidence scores and class labels for each detected object. Unlike traditional closed-set detectors, it does not require a fixed object vocabulary, allowing us to flexibly detect a wide range of scene elements using natural language. This step gives us localized bounding boxes of potential light sources and furniture and room elements.

<sup>3</sup><https://github.com/crewAIInc/crewAI>

<sup>4</sup><https://www.gradio.app>

We then use these bounding boxes as inputs to Meta’s Segment Anything Model (SAM)[11] to obtain accurate pixel-level masks for the detected objects. SAM is a general-purpose segmentation model trained on over a billion masks and is capable of segmenting any object given a bounding box. Its strong generalization ability allows it to produce fine-grained object masks even on novel categories and complex indoor scenes.

After segmentation, we analyze each detected region to compute a set of lighting-related features, including brightness, color temperature (warm, cool, or neutral), contrast, and the dominant color. These measurements provide rich contextual cues for understanding the lighting characteristics of each object and are crucial for generating effective relighting suggestions in the next step.

### 3.4 RelightZone selection

To generate region-specific relighting suggestions, we begin by extracting a global scene description using BLIP (Bootstrapped Language-Image Pretraining)[8], a powerful vision-language model designed to generate natural language captions from images. BLIP is pre-trained on large-scale image-text pairs and excels at capturing both the objects and their relationships within a scene. By producing a coherent textual summary of the room, including its layout, function, and overall ambiance, BLIP provides our downstream language model with contextual grounding beyond raw pixel data. This step ensures that the lighting recommendations are not just object-aware but also scene-aware.

We then combine this high-level description with fine-grained segmentations obtained earlier using SAM and the region-level lighting features we compute for each segmented region. Together, these form a structured and interpretable representation of the visual environment, capturing both the semantic content and their physical appearance. This is essential for LLM to determine how lighting adjustments will affect the scene.

We format these inputs into a structured prompt and feed it into GPT-4o-mini, which serves as a lighting design expert. The model is instructed to consider both direct lighting and indirect lighting effects and generate JSON-formatted recommendations. For each region, the model specifies whether the lighting should be turned on, turned off, or left unchanged, accompanied by a natural language explanation. This reasoning-driven process enables context-aware and interpretable relighting decisions, ensuring that lighting changes are semantically meaningful and stylistically coherent with the rest of the scene.

### 3.5 Image Relighting

To apply the LLM’s lighting recommendations, we generate a scribble mask that encodes region-specific relighting instructions. For the chosen region, if the action is "on", the region is marked in white; if "off", it’s marked in black; otherwise, it’s filled with a neutral gray. The result is a single-channel control mask where pixel intensity directly corresponds to lighting actions. Once the scribble mask is generated, we feed both the original image and the mask into ScribbleLight, a state-of-the-art image-based relighting model. ScribbleLight is built on top of Stable Diffusion with a ControlNet encoder-decoder architecture, which enables fine-grained geometry-preserving lighting control. It is conditioned on both the scribble mask and a normal map, and incorporates an Albedo-preserving mechanism to retain the original color and texture of the scene. The model interprets the spatial lighting cues from the mask and relights the image accordingly, producing a new image that reflects the desired changes in illumination while maintaining photorealism and semantic coherence. This step translates high-level, natural language lighting requests into physically plausible and visually consistent lighting edits, without requiring users to manually select regions or understand low-level image editing concepts.

### 3.6 CLIP Evaluation

To quantitatively evaluate the lighting quality of the relit image, we use a CLIP-based classifier. We define a set of lighting-related textual prompts (e.g., “good lighting”, “dark scene”, “balanced illumination”, “harsh shadows”) and compute their similarity with the relit image using CLIP model (ViT-B/32). The image is encoded, compared against the prompts, and the model returns a softmax probability distribution indicating which lighting condition best matches the image. This lightweight scoring mechanism provides a fast and interpretable way to assess the overall lighting impression.

## 4 Results

### 4.1 Scene Generation

This section presents visual examples demonstrating our interior design generation system’s capabilities across different use cases. The examples showcase both the quality of initial generations and the system’s ability to iteratively refine designs based on user feedback.

#### 4.1.1 Text-to-Scene Generation Results

Our system transforms simple user descriptions into detailed, photorealistic interior designs. The following examples demonstrate how the prompt enhancement system creates coherent, professionally-styled spaces from basic requests (see figs. 2 and 3).



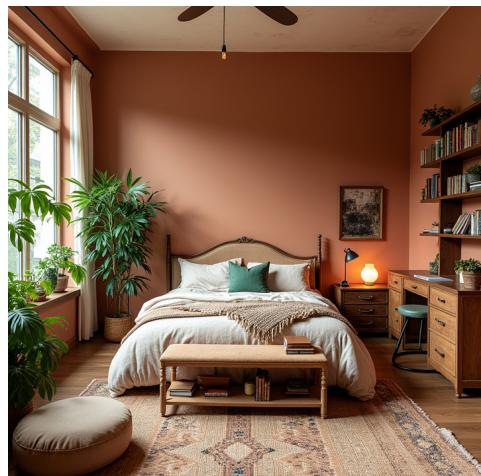
(a) Minimalist Japanese-inspired bathroom with clean lines and zen atmosphere



(b) Luxury Art Deco dining room with bold geometric patterns and gold accents



(c) Industrial loft kitchen with exposed brick and stainless steel appliances



(d) Cozy bohemian bedroom with lots of plants

Figure 2: Diverse interior design styles generated from simple user descriptions, demonstrating system versatility across different room types and aesthetic preferences.

### User Input

*"Coastal cottage style bedroom with soft blues and whites, nautical elements"*

### Enhanced Prompt

*"Design a coastal cottage bedroom with a soft blue and white color palette. Include nautical elements such as starfish and rope accents. Key furniture pieces include a wooden bed with a white headboard and a side table adorned with seashell decor. Use natural wood flooring and light-colored walls to create a serene atmosphere. Incorporate coastal art and fabric accents with beach-inspired prints."*



Figure 3: Generated interior design from enhanced prompt showing successful translation of user intent into detailed visual specification.

#### 4.1.2 Iterative Refinement: Before and After Feedback

The following examples demonstrate our system's ability to interpret user feedback and make targeted improvements while preserving successful design elements. Each example (fig. 4) shows the original generation, user feedback, and the refined result.

#### 4.2 Results for Relighting

The result of running our agentic pipeline on the input image shown in Figure. 5.

### Original Generation

*Prompt: "Design a mid-century modern home office featuring walnut furniture and vintage accents. The color palette consists of deep walnut, crisp white, soft gray, warm oak, and a touch of vintage green. Key furniture pieces include a sleek walnut desk, vintage chair, and a mid-century bookshelf. Lighting is a mix of natural light from large windows and soft artificial LED task lights. The floor is covered in polished oak."*



(a) Initial generation

### After Feedback

*Revised: "Design a mid-century modern home office with a sleek walnut desk, vintage chair, and mid-century bookshelf. Include a ceiling fan for added comfort. Use a color palette of deep walnut, crisp white, soft gray, warm oak, and vintage green. Add decorative elements like a vintage green abstract painting and a woven basket. Ensure the space has ample natural light from large windows and soft artificial LED task lights. The floor is covered in polished oak."*



(b) After refinement

### User Feedback

*"I want a ceiling fan"*

Figure 4: Iterative refinement example: Successfully incorporating user feedback by adding a ceiling fan.

To check if the lighting improved after processing the images, we used CLIP to compare how well each image matched descriptions like “*good lighting*,” “*balanced illumination*,” “*harsh shadows*,” and “*dark scene*.”

Here are the scores before and after processing:

Description	Before	After
Good lighting	0.19	0.22
Dark scene	0.009	0.020
Balanced illumination	0.67	0.69
Harsh shadows	0.13	0.07

Table 1: CLIP similarity scores for lighting-related prompts before and after processing.

From these results, we can see that the output image has better lighting overall. The scores for *good lighting* and *balanced illumination* went up, meaning the image looks more evenly lit. At the same time, the score for *harsh shadows* went down, which suggests fewer strong or distracting shadows.

Another example is given in the figure6 : The relighting of the chandelier is very faint hence unable to the see the difference better. This is due to limited training of the scribbleLight model.



Figure 5: Pipeline: Image from scene generation → Relighted Image



Figure 6: Pipeline: Image from scene generation → Relighted Image

## 5 Limitations

While our interior design generation system demonstrates significant capabilities in creating and iteratively refining and relighting photorealistic interior spaces, several important limitations constrain its current functionality and deployment scenarios.

**Image Inpainting Capabilities:** Our current implementation lacks image inpainting functionality, which would allow users to selectively modify specific regions of generated images while preserving surrounding areas. This limitation stems primarily from computational resource constraints and the additional model complexity required for precise spatial editing. Without inpainting, the system must regenerate entire images even for localized changes, which can inadvertently alter successful design elements that should remain unchanged.

**Lack of 3D Spatial Understanding:** Our system generates 2D images without an underlying 3D scene representation, limiting its ability to maintain spatial consistency across different viewing angles or to understand physical constraints like room dimensions, furniture placement feasibility, or architectural limitations. Users cannot examine designs from multiple perspectives or verify that proposed layouts would function in real physical spaces.

**Loss of quality in relighting:** We were unable to train ScribbleLight due to GPU and time constraints, so we used a pre-trained model instead. This resulted in images with poor texture quality.

These limitations represent opportunities for future development and highlight the current system’s position as a powerful design ideation tool rather than a complete professional design solution. Addressing these constraints would require significant additional development in areas such as understanding of the 3D scene, integration of professional workflows, and better computational infrastructure.

## 6 Conclusion

In this report, we introduced an agentic pipeline designed to perform relighting for indoor scenes using natural language input. Unlike traditional methods that rely on 3D tools or require significant human effort for relighting, this pipeline eliminates the need for manual intervention. By integrating scene generation, segmentation, a RelightZone selector, and scribble-based relighting, we developed an indoor scene agent capable of interpreting simple text descriptions to adjust lighting automatically without requiring users to manually select or paint the regions to be edited.

In future work, we can extend this pipeline to include material and texture understanding of objects, having applications in the interior design area. Real-time feedback and boundary-aware tools can be incorporated into this pipeline to do finer control of the lighting. We can also incorporate multimodal input, like sketch or voice instructions. In conclusion, this work demonstrates the potential of AI tools to build agentic systems with applications in AR/VR, game development, and other areas of digital content creation.

## 7 Contributions

The contributions of each team member to this project are as follows:

- **Abhilash Shankarampete**: Text-to-scene generation pipeline, agentic system setup, and user interface development
- **Anandhini Rajendran**: Segment Anything Model (SAM) integration, RelightZone selection implementation, and CLIP evaluation system
- **Humaira Mohammed**: Integration of relighting pipeline with SAM and RelightZone selection components
- **Yueqi Wu**: ScribbleLight model setup and agentic system architecture

All team members contributed equally to the writing and preparation of this report.

## References

- [1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-v1 technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022.
- [3] Atilim Celen, Arda Firat, Azarakhsh Arslan, Nicola Pezzotti, Tevfik Metin Sezgin, and Evrim Gavves. Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. *arXiv preprint arXiv:2402.04717*, 2024.
- [4] Youngseok Choi, Soonmin Kwon, Jaejun Park, and Jae-Pil Heo. Scribblelight: Single image indoor relighting with scribbles. *arXiv preprint arXiv:2411.17696*, 2024.
- [5] Zexin He, Yifan Wang, Xiaoyang Jin, Yinghao Xu, Jingyi Yu, and Dong Yu. Neural lightrig: Intrinsic estimation by multi-light diffusion priors. *arXiv preprint arXiv:2412.09593*, 2024.
- [6] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024.
- [7] Jiarui Li, Xinyu Jiang, Yanan Zhang, Kaiyu Huang, Yifan Zhao, Yutong Bai, Xiao Liu, Peng Gao, Hongyang Li, Yu Qiao, et al. Grounded-segment-anything: From language to parts. *arXiv preprint arXiv:2401.14159*, 2024.
- [8] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

- [9] Yasheng Li, Jiayi Hou, Xudong Wen, Yixuan He, Sihui Liu, and Chi-Wing Fu. Llplace: 3d indoor scene layout generation and editing via large language model. *arXiv preprint arXiv:2406.03866*, 2024.
- [10] Fabian Manhardt, Jiacheng Li, Danhang Xu, Angela Dai, and Daniel Cremers. Text2room: Text-driven generation of 3d scenes using 2d diffusion. *arXiv preprint arXiv:2303.11989*, 2023.
- [11] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.
- [12] Yifan Yeh, Weikai Xie, Jingtian Xu, Lingxi Xie, Xin Wang, Zhangyang Wang, and Dahua Lin. Photoscene: Photorealistic material and lighting transfer for indoor scenes. *arXiv preprint arXiv:2207.00757*, 2022.

## A System Prompts and Examples

### A.1 Prompt Enhancement System

The prompt enhancement system transforms basic user descriptions into comprehensive design specifications. This transformation requires the language model to think like a professional interior designer, considering multiple design dimensions simultaneously.

#### Prompt Enhancement Instruction for Qwen2.5-VL

```
You are an expert interior designer. Enhance this prompt for AI image
generation:
Original request: {original_prompt}
Style preferences: {style_preferences if style_preferences else "Not
specified"}
Create a detailed prompt that includes:
1. Specific design style (e.g., Modern, Scandinavian, Industrial)
2. Color palette (3-5 specific colors)
3. Key furniture pieces and materials
4. Lighting description (natural and artificial)
5. Textures and materials (flooring, walls, fabrics)
6. Decorative elements and accessories
7. Mood and atmosphere
8. Camera angle and composition
Keep the prompt under 50 words but make it rich in visual details. Return
ONLY the enhanced prompt, nothing else.
```

This prompt instructs the language model to systematically expand user input across eight critical design dimensions. The constraint to keep output under 50 words forces the model to be precise and selective, choosing only the most impactful descriptive elements. The instruction to return only the enhanced prompt ensures clean integration with the image generation pipeline.

### A.2 Feedback Analysis and Prompt Rewriting

When users provide feedback on generated images, our system must understand both what they want to change and what they want to preserve. This requires sophisticated reasoning about design intent and visual analysis of the current image.

### Feedback Analysis Prompt for Vision-Language Model

```
You are an expert interior designer analyzing user feedback on a generated design.  
Current prompt: {current_prompt}  
User feedback: {user_feedback}  
Previous attempts: {len(previous_prompts)}  
Looking at the generated image and considering the feedback:  
1. Identify what the user wants to change  
2. Determine what elements to keep  
3. Understand their style preferences from the feedback  
4. Try to incorporate the feedback at the start improved prompt  
5. Maintain the successful elements from the current design  
Create an improved prompt that:  
- Addresses ALL the user's feedback points  
- Maintains successful elements from the current design  
- Adds specific details to realize the user's vision  
- Stays under 50 words  
Return ONLY the new improved prompt, nothing else.
```

This prompt guides the model through a systematic analysis process that mirrors how human designers process client feedback. The five-step reasoning framework ensures comprehensive consideration of both change requirements and preservation needs. The emphasis on addressing "ALL" feedback points while maintaining successful elements creates the balanced evolution that users expect in iterative design.

### A.3 Quality Control Prompts

Our system uses carefully crafted positive and negative prompts to ensure consistent, professional-quality output from the diffusion model.

#### Positive Prompt Template

```
Professional interior design photograph: {enhanced_user_prompt}. High quality.
```

#### Negative Prompt Template

```
low quality, blurry, distorted, amateur, ugly, deformed, cartoon, painting, sketch, {additional_negative_elements}
```

The positive prompt template frames every generation as a "professional interior design photograph," which guides FLUX.1-dev toward realistic, high-quality architectural photography aesthetics. The negative prompt systematically excludes common failure modes we observed during system development, from technical issues like blurriness to stylistic problems like cartoon-like rendering.

### A.4 Example Transformations

These examples demonstrate how our prompt enhancement system transforms natural user descriptions into detailed generation prompts.

#### Example 1: Basic Living Room Request

**User Input:** "Modern minimalist living room with warm lighting"

**Enhanced Output:** "Scandinavian modern living room, cream and warm oak palette, low-profile linen sofa, pendant brass lighting, natural wood floors, wool textures, minimal ceramics, cozy intimate atmosphere, wide architectural angle"

#### Example 2: Bedroom with Style Preferences

**User Input:** "Cozy bohemian bedroom with lots of plants"

**Enhanced Output:** "Bohemian bedroom, terracotta and sage palette, macrame wall hanging, rattan furniture, warm Edison bulb string lights, jute rugs, abundant hanging plants, dreamy relaxed mood, intimate corner perspective"

#### Example 3: Kitchen with Specific Style

**User Input:** "Industrial loft kitchen with exposed brick"

**Enhanced Output:** "Industrial loft kitchen, charcoal and copper palette, exposed brick walls, stainless steel appliances, pendant Edison bulbs, concrete counters, metal bar stools, urban sophisticated atmosphere, wide island view"

### A.5 Feedback Processing Examples

These examples show how our system interprets user feedback and modifies prompts while preserving successful design elements.

#### Feedback Example 1: Adding Elements

**Original Prompt:** "Modern minimalist living room, white and gray palette, clean lines, natural lighting"

**User Feedback:** "Add more plants and warmer colors"

**Revised Prompt:** "Modern minimalist living room, warm beige and sage green palette, clean lines, abundant potted plants, natural lighting, fiddle leaf figs, ceramic planters, fresh organic atmosphere"

#### Feedback Example 2: Atmospheric Changes

**Original Prompt:** "Scandinavian bedroom, light wood and white palette, platform bed, minimal decor"

**User Feedback:** "Make it cozier and more intimate"

**Revised Prompt:** "Scandinavian bedroom, warm oak and cream palette, low platform bed, soft wool throws, warm table lamps, intimate reading nook, cozy hygge atmosphere, soft evening lighting"

#### Feedback Example 3: Style Adjustments

**Original Prompt:** "Contemporary kitchen, stainless steel appliances, marble counters, bright lighting"

**User Feedback:** "Too cold, add some rustic elements"

**Revised Prompt:** "Contemporary farmhouse kitchen, stainless steel with reclaimed wood accents, marble and butcher block counters, brass fixtures, warm pendant lighting, rustic shelving, welcoming family atmosphere"