
RoomEditor: High-Fidelity Furniture Synthesis with Parameter-Sharing U-Net

Zhenyi Lin
Tianjin University
linzhenyi@tju.edu.cn

Xiaofan Ming
Tianjin University
xiaofanming@tju.edu.cn

Qilong Wang
Tianjin University
qlwang@tju.edu.cn

Dongwei Ren*
Tianjin University
rendw@tju.edu.cn

Wangmeng Zuo
Harbin Institute of Technology
wmzuo@hit.edu.cn

Qinghua Hu
Tianjin University
huqinghua@tju.edu.cn

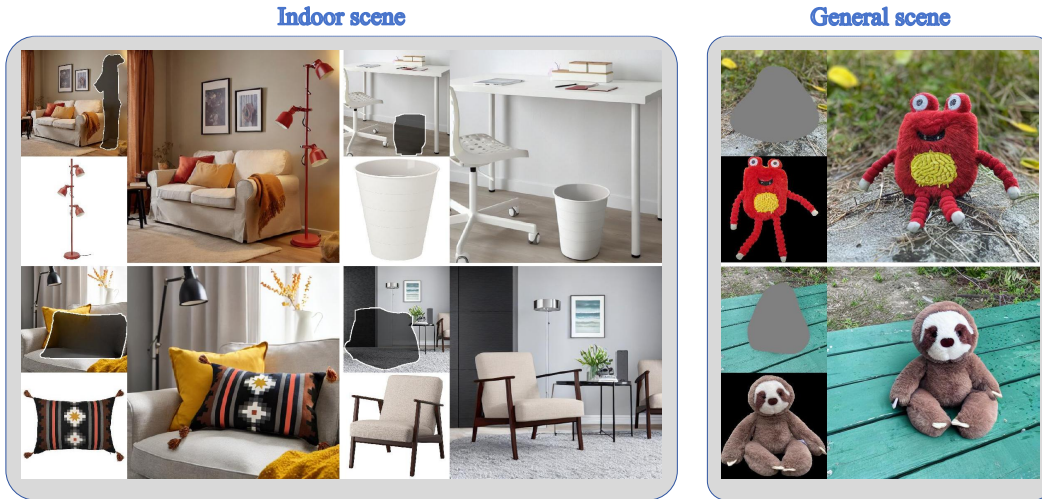


Figure 1: Furniture synthesis with RoomEditor integrates reference objects into environments with geometric coherence and visual fidelity. Moreover, RoomEditor exhibits remarkable generalization capabilities across a wide range of unseen scenes and objects without task-specific fine-tuning

Abstract

Virtual furniture synthesis, a critical task in image composition, aims to seamlessly integrate reference objects into indoor scenes while preserving geometric coherence and visual realism. Despite its significant potential in home design applications, this field remains underexplored due to two major challenges: the absence of publicly available and ready-to-use benchmarks hinders reproducible research, and existing image composition methods fail to meet the stringent fidelity requirements for realistic furniture placement. To address these issues, we introduce RoomBench, a ready-to-use benchmark dataset for virtual furniture synthesis, comprising 7,298 training pairs and 895 testing samples across 27 furniture categories. Then, we propose RoomEditor, a simple yet effective image composition method that employs a parameter-sharing dual U-Net architecture, ensuring better feature consistency by sharing weights between dual branches. Technical analysis reveals that conventional dual-branch architectures generally suffer from inconsistent intermediate features due to independent processing of reference and background images. In contrast, RoomEditor enforces unified feature learning through shared parame-

*Corresponding author

ters, thereby facilitating model optimization for robust geometric alignment and maintaining visual consistency. Experiments show our RoomEditor is superior to state-of-the-arts, while generalizing directly to diverse objects synthesis in unseen scenes without task-specific fine-tuning. Our dataset and code are available at <https://github.com/stonecutter-21/roomeditor>.

1 Introduction

Recent advances in augmented reality and computer vision have revolutionized virtual product visualization for e-commerce applications, particularly in indoor scene synthesis. Within the rapidly expanding home design market (expected to reach around 250 billion dollars in ten years [15]), intelligent image synthesis systems have emerged as critical tools for consumer decision-making. These systems enable in-situ furniture visualization by digitally integrating selected items into user-provided room images as shown in Figure 1, offering significant potential to transform design practices and retail experiences.

Despite progress in generic image composition, indoor furniture synthesis remains underdeveloped due to two key limitations. The first lies in the limited availability of ready-to-use datasets with realistic indoor environments, thereby hindering reproducible research. While Amazon’s dataset [44] demonstrates the value of domain-specific data, its restricted access stifles community efforts. InteriorNet [30], 3D-FUTURE [14], and 3D-FRONT [13] expose detailed 3D models, but constructing reasonable reference-background pairs demands meticulous viewpoint selection and scene setup, revealing challenges in data acquisition. Consequently, the field remains hampered by inadequate benchmarks and persistent data scarcity. To address this issue, we introduce the RoomBench dataset, a ready-to-use, publicly available benchmark tailored for virtual furniture synthesis. Our RoomBench consists of 7,298 training pairs and 895 testing samples across 27 furniture categories, meticulously curated from diverse online sources. Each sample includes high-quality annotations, enabling comprehensive evaluation of geometric coherence and visual realism. This benchmark provides a foundation for future research in indoor furniture synthesis by facilitating both training and evaluation.

The second critical challenge arises from technical limitations in current image composition methods [53, 6, 7], which often fall short of the high visual fidelity required in practical home design applications. Minor geometric misalignments or texture inconsistencies can break user immersion and erode trust. Image encoder-based methods such as PBE [53] and Anydoor [6] use CLIP [38] or DINO [3] to extract reference features and inject them into the U-Net. However, they tend to overlook fine-grained structural details essential for complex furniture integration. Even advanced dual U-Net architectures like MimicBrush [7] struggle to ensure visual consistency and harmony between the inserted object and the surrounding background, often resulting in unnatural compositions with obvious copy-paste artifacts. These limitations result in inconsistent appearances and poses of furniture, thereby undermining practical utility. To address these limitations, we propose RoomEditor, a parameter-sharing dual U-Net that enforces feature consistency via shared weights. We perform an in-depth analysis of existing dual U-Net pipelines and identify feature misalignment between the extraction and inpainting stages, which degrades synthesis quality. By unifying feature extraction and inpainting within the same network, RoomEditor naturally aligns the two stages, enabling complex spatial transformations and seamless object integration. Experiments demonstrate that our method achieves superior objective and subjective results on RoomBench compared to state-of-the-art methods.

Our RoomEditor not only excels in home design but also generalizes effectively to a broader range of scenes and objects. As illustrated in Figure 1, our RoomEditor can be directly applied to general scenes and objects without any task-specific fine-tuning. The superior adaptability of RoomEditor can be attributed to its unified feature learning in parameter-shared U-Net. The contributions of this work are summarized as follows:

- We constructed a ready-to-use open-source benchmark RoomBench specifically for home design, consisting of 7,298 furniture-background training pairs and 895 testing samples across 27 categories with annotations.
- We propose a simple yet effective RoomEditor for high-fidelity furniture synthesis, whose core is a parameter-sharing U-Net. In addition, we provide an in-depth analysis of existing

dual-branch approaches from the perspective of feature consistency, showing the potential advantage of RoomEditor for high-fidelity and visual consistency.

- Extensive experiments validate that RoomEditor achieves superior performance in both quantitative and qualitative evaluations as well as in human perception studies for home design, while demonstrating remarkable generalization across diverse scenes and objects.

2 Related Work

2.1 Furniture Synthesis for Home Design

The most relevant research field to furniture synthesis is virtual try-on [18, 8, 28, 26, 50, 60], which has primarily focused on fashion applications, emphasizing fabric textures, body pose estimation, and occlusion handling. While several virtual try-on methods [26, 50, 60] employ dual U-Net architectures and must model garment-body interactions, their objectives and challenges differ from those in furniture synthesis. In particular, furniture synthesis requires handling diverse and rigid object shapes, whereas clothing in virtual try-on typically exhibits more consistent forms, leading to different integration challenges. Meanwhile, the area of indoor scene synthesis specifically for furniture placement remains relatively underexplored. A few studies [4, 49] have explored interior design through text-to-image generation but provide limited control over individual furniture placement. Diffuse-to-Choose [44] addresses furniture synthesis by incorporating reference features via a U-Net encoder augmented with FiLM [37] layers on a large-scale indoor dataset. However, its training and testing sets are proprietary, limiting reproducibility and further research. To bridge these gaps, this paper introduces RoomBench, a ready-to-use and open-source benchmark for furniture synthesis, along with a method RoomEditor that ensures high-fidelity object preservation and context-aware placement.

2.2 Image Composition

Early image composition methods [23, 9, 45, 46, 47, 11, 10] primarily focused on the task of image harmonization, ensuring seamless integration of the foreground with the background. These methods typically relied on manually designed pipelines. In contrast, text-to-image diffusion models [40, 39, 17, 42] have enabled the automatic integration of specific objects into diverse contexts while preserving their identity and attributes. By extracting pseudo-words, methods such as Textual Inversion [16] and DreamBooth [41], along with others [2, 19, 24, 25, 31, 33, 12, 29, 27], rely on fine-tuning with textual prompts but often produce unstable backgrounds. Semantic image composition methods, such as DreamPaint [43], PBE [53], CustomNet [54], and ControlCom [55], which focus on object insertion into predefined scenes, often struggle to preserve fine-grained details in complex scenes when using CLIP [38] image encoder. Recent advancements, including AnyDoor [6], have addressed some of these challenges by leveraging ControlNet [57] and DINO [35] to improve texture fidelity and reference-object consistency. In contrast, MimicBrush [7] employs a dual U-Net architecture [51, 56, 59, 5, 22, 52], which has proven effective in capturing multi-scale reference features, thereby enabling flexible image editing and generating high-fidelity syntheses. Despite these improvements, achieving both high fidelity and seamless integration in home design remains a significant challenge, particularly when the training dataset is insufficient.

3 Benchmark for Furniture Synthesis

In this section, we construct an open-source RoomBench for furniture synthesis, whose procedures for data collection, filtering, and annotation are detailed as follows.

3.1 Data Collection and Filtering

To ensure the diversity in furniture types and scene contexts, we systematically collected about 4,500 furniture items, each of which has several product images and contextual scene backgrounds. Our curation strategy included two key measures. Category Coverage: Hierarchical classification into 27 fine-grained categories, including bedroom essentials (e.g., Bed, Pillow), living room furnishings (e.g., Sofa, Coffee Table), and bathroom fixtures (e.g., Shower, Bathtub). Image Quality Assurance: We require a minimum resolution of 512×512 for both furniture images and background images.

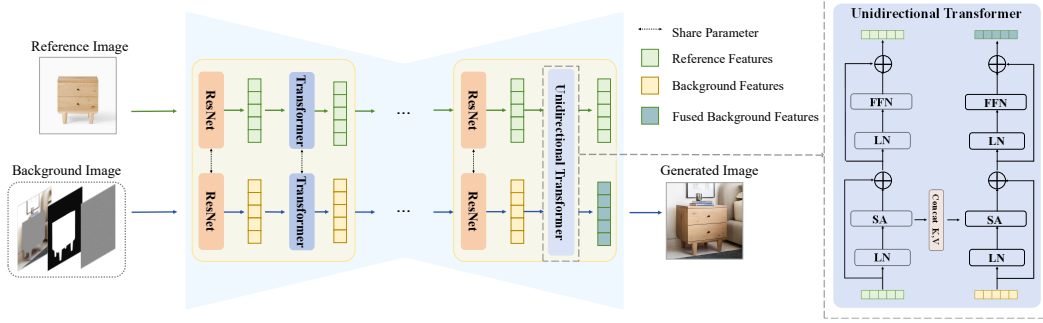


Figure 2: **The architecture of our RoomEditor.** Our method shares parameters between the two U-Nets for unified feature space learning. As shown, reference features propagate independently, while background features interact with reference features through a self-attention module at each layer, ensuring effective feature alignment.

Subsequently, the collected data underwent rigorous filtering. Initially, we manually classified images as either product images or background images. We then leveraged GPT-4o [1] to assist in data filtering. Images were excluded if the furniture was not fully visible in the background image or if there was an inconsistency between the furniture object or color in the reference and background images (i.e., they did not depict the same object), resulting in 5,948 reference furniture images and 4,989 background images. Following this, the dataset was split into training and testing sets. The training set comprises 5,288 reference furniture images and 4,094 background images, with each reference image paired with one or two background images, resulting in a total of 7,298 furniture-background pairs. The testing set contains 660 reference furniture images and 895 background images, resulting in 895 testing samples. For more detailed data-processing procedures and complete dataset information, please refer to Appendix B.

3.2 Data Annotation

For each image pair in RoomBench, background image and reference image are annotated with corresponding furniture masks. In the training set, these masks are hand-labeled precisely to ensure high-quality annotations. This labeling process allows effective data augmentation, improving the model’s generalization capabilities. For the testing set, we included more loosely annotated masks to reflect the variability and potential inaccuracies in user-generated labels. This annotation strategy allows us to evaluate the model’s robustness in real-world scenarios, where annotations are often imperfect or inconsistent. With specific images and annotations, RoomBench is immediately ready-to-use, eliminating the need for additional efforts in data acquisition that are typically required by InteriorNet [30], 3D-FUTURE [14], and 3D-FRONT [13].

4 RoomEditor for Furniture Synthesis

In this section, we first introduce the RoomEditor architecture, whose core is a shared U-Net model for background and reference images with unidirectional information flow. Then, we compare our RoomEditor with existing methods from the perspective of feature consistency, showing the underlying mechanism behind our RoomEditor.

4.1 RoomEditor Architecture

Given a masked background image I_{bg}^M and a masked reference image I_{ref}^M , our goal is to seamlessly integrate the object from I_{ref}^M into the masked region of I_{bg}^M . Different from existing dual U-Net methods (e.g., [7]), our approach utilizes a shared U-Net to synchronously extract reference image features and perform inpainting. Specifically, as shown in Figure 2, we employ an inpainting U-Net $\epsilon_\theta(\cdot)$ as basic model by following previous work [7]. First, we extract the features for I_{ref}^M from the mid- and upper-attention layers of the U-Net. For l -th self-attention layer, the extracted feature of

$\mathbf{I}_{\text{ref}}^M$, denoted as $f_l(\mathbf{I}_{\text{ref}}^M)$, is computed as

$$f_l(\mathbf{I}_{\text{ref}}^M) = \text{softmax} \left(\frac{\mathbf{Q}_{\text{ref}} \cdot \mathbf{K}_{\text{ref}}^\top}{\sqrt{d_k}} \right) \cdot \mathbf{V}_{\text{ref}}, \quad (1)$$

where $\mathbf{Q}_{\text{ref}} = f_{l-1}(\mathbf{I}_{\text{ref}}^M) \mathbf{W}_Q$, $\mathbf{K}_{\text{ref}} = f_{l-1}(\mathbf{I}_{\text{ref}}^M) \mathbf{W}_K$, $\mathbf{V}_{\text{ref}} = f_{l-1}(\mathbf{I}_{\text{ref}}^M) \mathbf{W}_V$, and $\mathbf{W}_{\{Q,K,V\}}$ are the projection matrices in attention layer. $f_{l-1}(\mathbf{I}_{\text{ref}}^M)$ indicates the output of the $(l-1)$ -th layer. It is worth note that our RoomEditor actually takes the image, mask and noise as inputs, while we omit the mask and noise in our formulation for simplicity.

Then, RoomEditor performs a unidirectional interaction to inject features of reference image $f_{l-1}(\mathbf{I}_{\text{ref}}^M)$ into those of masked background image $\mathbf{I}_{\text{bg}}^M$ for inpainting. Particularly, the features of $\mathbf{I}_{\text{bg}}^M$ in l -th layer is computed by a mixture attention with inputs of $f_{l-1}(\mathbf{I}_{\text{bg}}^M)$ and $f_{l-1}(\mathbf{I}_{\text{ref}}^M)$, i.e.,

$$f_l(\mathbf{I}_{\text{bg}}^M) = \text{softmax} \left(\frac{\mathbf{Q}_{\text{bg}} \cdot \mathbf{K}_{[\text{bg}, \text{ref}]}^\top}{\sqrt{d_k}} \right) \cdot \mathbf{V}_{[\text{bg}, \text{ref}]}, \quad (2)$$

where $\mathbf{Q}_{\text{bg}} = f_{l-1}(\mathbf{I}_{\text{bg}}^M) \mathbf{W}_Q$, $\mathbf{K}_{[\text{bg}, \text{ref}]} = \text{cat}(f_{l-1}(\mathbf{I}_{\text{bg}}^M), f_{l-1}(\mathbf{I}_{\text{ref}}^M)) \mathbf{W}_K$, $\mathbf{V}_{[\text{bg}, \text{ref}]} = \text{cat}(f_{l-1}(\mathbf{I}_{\text{bg}}^M), f_{l-1}(\mathbf{I}_{\text{ref}}^M)) \mathbf{W}_V$ and cat indicates the concatenation operation. Besides, existing dual U-Net methods [7, 50] generally integrate reference features by using extra image encoders (e.g., CLIP [38]). Due to similar functions are achieved by a mixture attention in Eqn. (2), our RoomEditor can omit extra image encoders to make our architecture more succinctly.

For optimizing our RoomEditor, we first construct masked background as $\mathbf{I}_{\text{bg}}^M = \mathbf{M}_{\text{bg}} \odot \mathbf{I}_{\text{gt}}$ by giving ground-truth image $\mathbf{I}_{\text{gt}}$ and a mask $\mathbf{M}_{\text{bg}}$, and then sample random noise ϵ and a timestep t to generate the noisy image $\mathbf{I}_{\text{gt}}^t$. Finally, we minimize the following loss function, i.e.,

$$\mathcal{L} = \mathbb{E}_{t, \mathbf{I}_{\text{gt}}, \epsilon} \left\| \epsilon_\theta(\mathbf{I}_{\text{gt}}^t, \mathbf{I}_{\text{bg}}^M, c, t) - \epsilon \right\|_2^2, \quad (3)$$

where c denotes the features extracted from $\mathbf{I}_{\text{ref}}^M$.

4.2 Discussion on Merit of RoomEditor

In this subsection, we compare with previous works [53, 6, 7] from the perspective of feature consistency, and show the potential advantage of our RoomEditor architecture in the context of high-fidelity furniture synthesis task.

4.2.1 Task Description

Since high-fidelity furniture synthesis aims to prominently place reference furniture $\mathbf{I}_{\text{ref}}$ in the background image $\mathbf{I}_{\text{bg}}$, it can be formulated as an ideal copy-paste problem. Specifically, the target image $\mathbf{I}_{\text{gt}}$ can be represented as a composition of a masked background image $\mathbf{M}_{\text{bg}} \odot \mathbf{I}_{\text{bg}}$ and a masked reference image $\mathbf{M}_{\text{ref}} \odot \mathbf{I}_{\text{ref}}$ with pose transformation $\mathcal{R}$, i.e., $\mathbf{I}_{\text{gt}} = \mathbf{M}_{\text{bg}} \odot \mathbf{I}_{\text{bg}} + \mathcal{R}(\mathbf{M}_{\text{ref}} \odot \mathbf{I}_{\text{ref}})$, where $\odot$ indicates element-wise multiplication. Therefore, the key issue is learning the precise transformation $\mathcal{R}$, given a pair of background and reference images $\{\mathbf{I}_{\text{bg}}, \mathbf{I}_{\text{ref}}\}$. As $\mathcal{R}$ depends on the spatial and visual relationship between $\mathbf{I}_{\text{bg}}$ and $\mathbf{I}_{\text{ref}}$, including factors such as mask geometry and background illumination, the final objective can be formulated as

$$\mathbf{I}_{\text{gt}} = \mathbf{M}_{\text{bg}} \odot \mathbf{I}_{\text{bg}} + \mathcal{R}(\mathbf{M}_{\text{ref}} \odot \mathbf{I}_{\text{ref}} \mid \mathbf{M}_{\text{bg}} \odot \mathbf{I}_{\text{bg}}). \quad (4)$$

To learn transformation $\mathcal{R}$, diffusion model $\epsilon_\theta(\cdot)$ is generally employed to achieve this implicitly by training on abundant pairs of background and reference images $\{\mathbf{I}_{\text{bg}}, \mathbf{I}_{\text{ref}}\}$ with masks $\{\mathbf{M}_{\text{bg}}, \mathbf{M}_{\text{ref}}\}$.

Based on the empirical observation that inpainting models implicitly preserve unmasked regions during training, we treat predictions in these areas as ground truth. Accordingly, we consider two complementary cases: (1) masking the object region in $\mathbf{I}_{\text{gt}}$ to obtain $\mathbf{I}_{\text{bg}}^M$, where the background prediction remains unchanged; and (2) masking the background region to retain only the object (denoted as $\mathcal{R}(\mathbf{I}_{\text{ref}}^M \mid \mathbf{I}_{\text{bg}}^M)$), where the object prediction remains unchanged. Combining these two masked predictions yields an approximation of the target image:

$$\begin{aligned} \epsilon_\theta(\mathbf{I}_{\text{gt}}) &= \epsilon_\theta(\mathbf{M}_{\text{bg}} \odot \mathbf{I}_{\text{bg}} + \overline{\mathbf{M}_{\text{bg}}} \odot \mathcal{R}(\mathbf{I}_{\text{ref}}^M \mid \mathbf{I}_{\text{bg}}^M)) \\ &\approx \mathbf{M}_{\text{bg}} \odot \epsilon_\theta(\mathbf{I}_{\text{bg}}^M) + \overline{\mathbf{M}_{\text{bg}}} \odot \epsilon_\theta(\mathcal{R}(\mathbf{I}_{\text{ref}}^M \mid \mathbf{I}_{\text{bg}}^M)), \end{aligned} \quad (5)$$

where $\mathbf{I}_{\text{bg}}^{\text{M}}$ and $\mathbf{I}_{\text{ref}}^{\text{M}}$ represent $\mathbf{M}_{\text{bg}} \odot \mathbf{I}_{\text{bg}}$ and $\mathbf{M}_{\text{ref}} \odot \mathbf{I}_{\text{ref}}$, respectively, and $\overline{\mathbf{M}_{\text{bg}}} = \mathbf{1} - \mathbf{M}_{\text{bg}}$ denotes the complement of the mask $\mathbf{M}_{\text{bg}}$.

Similarly, since the inpainting model’s output in non-masked regions remains essentially unchanged regardless of input conditions, its impact at the feature level is marginal. Thus, for the l -th layer, we obtain

$$\begin{aligned} f_l(\mathbf{I}_{\text{gt}}) &= f_l(\mathbf{M}_{\text{bg}} \odot \mathbf{I}_{\text{bg}} + \overline{\mathbf{M}_{\text{bg}}} \odot \mathcal{R}(\mathbf{I}_{\text{ref}}^{\text{M}} | \mathbf{I}_{\text{bg}}^{\text{M}})) \\ &\approx \mathbf{M}_{\text{bg}} \odot f_l(\mathbf{I}_{\text{bg}}^{\text{M}}) + \overline{\mathbf{M}_{\text{bg}}} \odot f_l(\mathcal{R}(\mathbf{I}_{\text{ref}}^{\text{M}} | \mathbf{I}_{\text{bg}}^{\text{M}})), \end{aligned} \quad (6)$$

Furthermore, the diffusion model implicitly learns $\mathcal{R}$ by progressively transforming and fusing features at each layer, rather than by directly transforming the original image. Thus, Eqn. (6) can be rewritten as follows in practice:

$$f_l(\mathbf{I}_{\text{gt}}) \approx \mathbf{M}_{\text{bg}} \odot f_l(\mathbf{I}_{\text{bg}}^{\text{M}}) + \overline{\mathbf{M}_{\text{bg}}} \odot \mathcal{R}_l(f_l(\mathbf{I}_{\text{ref}}^{\text{M}}) | f_l(\mathbf{I}_{\text{bg}}^{\text{M}})). \quad (7)$$

where $\mathcal{R}_l$ denotes the transformation in feature space. Given that the first term is solely related to the background, the differences among various methods [53, 6, 7] are primarily concentrated on the second term. According to Eqn. (6) and Eqn. (5), better feature consistency between ground-truth $f_l(\mathbf{I}_{\text{gt}})$ and practice one (i.e., the right part of Eqn. (6)) will help to optimize the inpainting model in Eqn. (5). The experimental verification of Eqn. (5) and Eqn. (6) is presented in subsection C.1.

4.2.2 Comparison with Previous Works

From the perspective of feature consistency, we compare our RoomEditor with several previous works [53, 6, 7]. Here we consider the l -th interaction layer and show how different methods approximate $f_l(\mathbf{I}_{\text{gt}})$. Encoder-based methods [53, 6] take advantage of a frozen image encoder $E(\cdot)$ and a trainable linear projector $L(\cdot)$ to extract reference features, formulated as $E_L(\mathbf{I}_{\text{bg}}^{\text{M}}) = L(E(\mathbf{I}_{\text{bg}}^{\text{M}}))$. Then, cross-attention is applied to inject the reference features $L(E(\mathbf{I}_{\text{ref}}^{\text{M}})) \in \mathbb{R}^{T' \times d}$ into the background ones:

$$\widehat{f_l(\mathbf{I}_{\text{gt}})} = \text{Cross-Attn}(f_{l-1}(\mathbf{I}_{\text{bg}}^{\text{M}}), E_L(\mathbf{I}_{\text{ref}}^{\text{M}})), \quad (8)$$

where $\text{Cross-Attn}(\cdot, \cdot)$ denotes cross-attention.

Dual U-Net methods (e.g., [7]) employ a frozen U-Net $g_{l-1}(\cdot)$ to extract attention features at $(l-1)$ -th layer. These features are injected via mixture attention (which can be approximated as a blend of self- and cross-attention), expressed as

$$\widehat{f_l(\mathbf{I}_{\text{gt}})} \approx \underbrace{\text{Self-Attn}(f_{l-1}(\mathbf{I}_{\text{bg}}^{\text{M}}))}_{f_l(\mathbf{I}_{\text{bg}}^{\text{M}})} + \underbrace{\text{Cross-Attn}(f_{l-1}(\mathbf{I}_{\text{bg}}^{\text{M}}), g_{l-1}(\mathbf{I}_{\text{ref}}^{\text{M}}))}_{\mathbf{A}_{[\text{bg}, \text{ref}]}(g_{l-1}(\mathbf{I}_{\text{ref}}^{\text{M}})) \cdot \mathbf{W}_V}, \quad (9)$$

where $\text{Self-Attn}(\cdot, \cdot)$ denotes self-attention and $\mathbf{A}_{[\text{bg}, \text{ref}]}$ denotes attention score matrix. Compared to Eqn. (7), these methods approximate $\mathcal{R}_l(f_l(\mathbf{I}_{\text{ref}}^{\text{M}}) | f_l(\mathbf{I}_{\text{bg}}^{\text{M}}))$ through the operation $\mathbf{A}_{[\text{bg}, \text{ref}]}(g_{l-1}(\mathbf{I}_{\text{ref}}^{\text{M}})) \cdot \mathbf{W}_V$, while $\mathbf{M}_{\text{bg}}$ and $\overline{\mathbf{M}_{\text{bg}}}$ are implicitly learned through the attention mechanism.

In contrast, our RoomEditor uses a single U-Net $f_{l-1}(\cdot)$ to extract reference features which are integrated through mixture attention:

$$\widehat{f_l(\mathbf{I}_{\text{gt}})} \approx \underbrace{\text{Self-Attn}(f_{l-1}(\mathbf{I}_{\text{bg}}^{\text{M}}))}_{f_l(\mathbf{I}_{\text{bg}}^{\text{M}})} + \underbrace{\text{Cross-Attn}(f_{l-1}(\mathbf{I}_{\text{bg}}^{\text{M}}), f_{l-1}(\mathbf{I}_{\text{ref}}^{\text{M}}))}_{\mathbf{A}_{[\text{bg}, \text{ref}]}(f_{l-1}(\mathbf{I}_{\text{ref}}^{\text{M}})) \cdot \mathbf{W}_V}. \quad (10)$$

Compared to Eqn. (7), our method approximates $\mathcal{R}_l(f_l(\mathbf{I}_{\text{ref}}^{\text{M}}) | f_l(\mathbf{I}_{\text{bg}}^{\text{M}}))$ using $\mathbf{A}_{[\text{bg}, \text{ref}]}(f_{l-1}(\mathbf{I}_{\text{ref}}^{\text{M}})) \cdot \mathbf{W}_V$. Under this formulation, we assume $\mathcal{R}_l$ corresponds to a cross-attention operation, which applies a complex transformation to the reference features based on their similarity to the background features. It is worth note that the optimal $g_l(\cdot)$ may be not $f_l(\cdot)$, but $f_l(\cdot)$ is a natural choice for offering the promising performance. Empirical comparisons in terms of loss visualizations and image quality indicate that the shared U-Net outperforms both the frozen U-Net and the unshared one.

Empirical Validation. To further investigate the effect of feature consistency, we compare three different methods on our RoomBench, i.e., frozen reference U-Net as in Mimicbrush [7], trainable reference U-Net for reference images, and our RoomEditor with shared U-Net for background and reference images. Specifically, we train all models on our RoomBench under the same exactly setting,

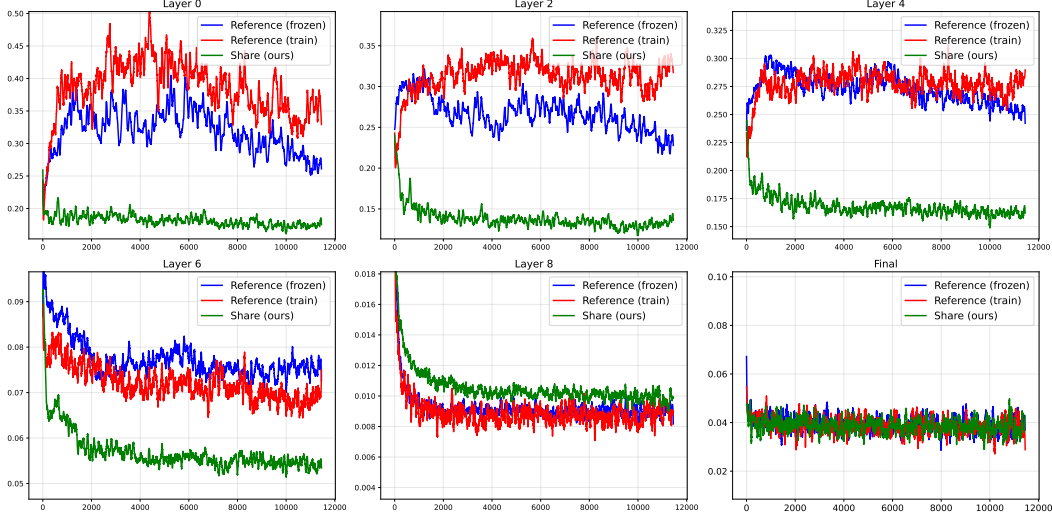


Figure 3: ℓ_2 error values in different layers by three methods: our RoomEditor (green line), and two non-shared dual U-Net methods with frozen (blue line) and trainable (red line) reference U-Net.

and compute ℓ_2 loss between features from two separate inputs ($\widehat{f_l(I_{gt})}$) and those of the ground-truth images, which are outputted by the intermediate layers of inpainting U-Net ($f_l(I_{gt})$). Figure 3 shows the results of different methods along with various layers and training epochs, where we can observe (1) our RoomEditor with shared U-Net (green line) is consistently low in ℓ_2 loss across all intermediate layers. Particularly, RoomEditor is significantly lower than other two methods at the early stages in terms of ℓ_2 loss. (2) The frozen (blue line) and trainable (red line) reference U-Net only approximate the features of ground-truth images at the last stage under the strong constraint of training loss. (3) The frozen reference U-Net has lower ℓ_2 loss than trainable one in the early stages, indicating unshared U-Net is more difficult to optimize, due to large parameter space and limited training data. The above comparisons show that our RoomEditor, using a shared U-Net, achieves better feature consistency, leading to better performance (refer to experiments in Table 4).

5 Experiments

5.1 Experiment Setup

Implementation Details. Our RoomEditor is initialized with the Stable Diffusion 1.5 inpainting model [40] and trained on the RoomBench dataset for 20k steps with a batch size of 32 across four NVIDIA A6000 GPUs. We use the AdamW optimizer [32] with a constant learning rate of 1×10^{-5} , and set the input resolution to 512×512 . To enhance robustness against imperfect mask inputs and domain variations, we apply augmentation techniques to both masks and images. The augmented masks are shown in Figure 6.

Objective Evaluation Metrics. To compare with other methods, we use SSIM [48] and PSNR [21] for assessing reconstruction quality, FID [20] and LPIPS [58] for assessing perceptual realism, CLIP-score [38] and DINO-score [35] for assessing semantic consistency.

Human Evaluation Metrics. To complement objective metrics, we conducted a user study following [7], evaluating results in fidelity (retaining reference identity and details), harmony (seamless integration with the background), and quality (overall visual appeal and detail).

5.2 Evaluation on RoomBench

Our RoomEditor is compared with PBE [53], Anydoor [6] and Mimicbrush [7]. These competing methods are evaluated under three settings: the released models trained on their respective large datasets, the fine-tuned models (denoted with $\star$), whose released model weights are further fine-tuned

Method	Train Data [‡]	RoomBench	FID↓	SSIM↑	PSNR↑	LPIPS↓	CLIP↑	DINO↑
PBE [53]	1.4M	✗	28.35	0.766	19.96	0.139	84.63	75.72
AnyDoor [6]	500K	✗	28.03	0.767	18.85	0.137	87.74	76.91
MimicBrush [7]	400K	✗	22.50	0.784	19.41	0.111	88.21	79.11
PBE*	1.4M	✓	26.85	0.769	20.26	0.136	86.46	79.22
AnyDoor*	500K	✓	26.62	0.772	19.13	0.135	88.73	79.37
MimicBrush*	400K	✓	<u>19.04</u>	<u>0.791</u>	<u>20.44</u>	<u>0.098</u>	90.87	85.65
MimicBrush [†]	0	✓	21.12	0.785	20.39	0.106	89.48	83.26
RoomEditor (Ours)	0	✓	18.42	0.793	21.15	0.094	<u>90.51</u>	<u>85.47</u>

Table 1: **Quantitative comparison on RoomBench.** PBE, AnyDoor, and MimicBrush are evaluated under three settings: (i) released models on large datasets, (ii) fine-tuned on RoomBench (marked with $\star$), and (iii) re-trained on RoomBench (marked with $\dagger$). $\ddagger$: data volume after filtering via SD [40].

Method	Fidelity Best (%)	Fidelity Rank↓	Harmony Best (%)	Harmony Rank↓	Quality Best (%)	Quality Rank↓
PBE* [53]	5.1	3.36	18.0	2.85	4.5	3.34
AnyDoor* [6]	8.9	2.96	7.6	3.16	3.5	3.20
MimicBrush* [7]	29.4	2.11	30.5	2.16	38.3	1.80
RoomEditor (Ours)	56.6	1.57	43.9	1.82	53.6	1.66

Table 2: **User study results.** In each trial, annotators were presented with four images (one from each method) and asked to rank them with respect to fidelity, harmony, and overall quality. “Best (%)” denotes the percentage of cases where a method was ranked first (i.e., perceived as the best under the given criterion), while “Rank↓” indicates the average ranking (lower is better) across all trials.

on our RoomBench, and the re-trained models (denoted with $\dagger$), whose model parameters are trained by keeping the same configuration as our RoomEditor.

5.2.1 Quantitative Comparison

As shown in Table 1, the released models of PBE, AnyDoor, and MimicBrush exhibit limited performance for furniture synthesis despite being trained on large-scale datasets. Notably, the large-scale datasets used for training AnyDoor and MimicBrush primarily consist of video data. The number of training image pairs is estimated to be approximately four times the number of videos, significantly exceeding the scale of our RoomBench dataset. This highlights the necessity of establishing a specific dataset tailored for home design. Fine-tuning on our RoomBench significantly improves the performance of AnyDoor* and MimicBrush* across all metrics. In particular, MimicBrush* achieves the highest CLIP-score and DINO-score.

From a technical standpoint, our RoomEditor, which is trained solely on RoomBench, outperforms competing methods in most quantitative metrics. Specifically, the state-of-the-art MimicBrush[†] is notably inferior to both MimicBrush* and our RoomEditor, demonstrating that our proposed parameter-sharing U-Net architecture is a more effective solution for image composition, especially when large-scale pre-training datasets are not available.

5.2.2 User Study

Following the protocol in [7], we conduct a user study to evaluate the perceptual quality of generated images. We let 25 annotators (14 undergraduate students, 7 parents of some of the students, and 4 university faculty or staff members, all informed of the evaluation criteria) rank one hundred randomly selected generation results of different methods based on our benchmark (introduced in section 3) from three aspects: fidelity, harmony, and overall quality. As shown in Table 2, our RoomEditor outperforms competing methods across all aspects. Notably, while MimicBrush* achieves superior results on some objective metrics, it lags behind our approach in real user assessments.

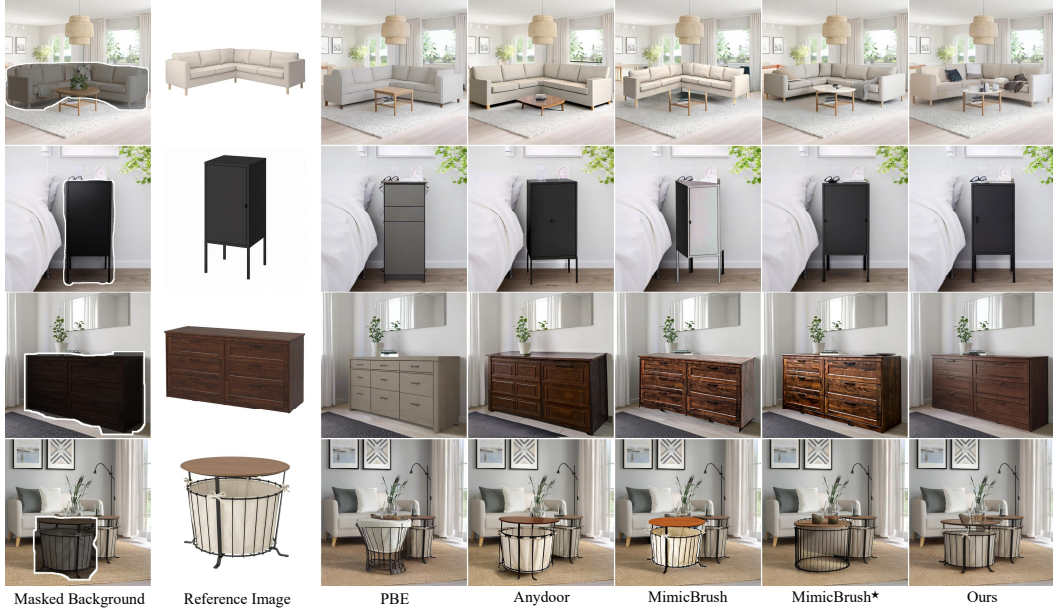


Figure 4: **Visual results on RoomBench.** PBE and AnyDoor achieve good background harmony but lack object consistency. MimicBrush preserves fidelity but suffers from a copy-paste effect. In contrast, our method ensures both high fidelity and strong background harmony, resulting in realistic and coherent furniture synthesis.

Method	FID↓	SSIM↑	PSNR↑	LPIPS↓	CLIP↑	DINO↑
PBE [53]	108.57	0.580	15.98	0.340	80.98	69.98
AnyDoor [6]	86.07	0.567	14.51	0.372	86.41	82.67
MimicBrush [7]	77.16	0.585	15.35	0.361	87.66	83.70
RoomEditor (Ours)	68.78	0.594	16.43	0.304	89.62	85.04

Table 3: **Quantitative evaluation on the DreamBooth dataset.** We compare our method with existing approaches. Despite being trained on only on RoomBench, our method achieves the best results, showcasing strong generalization capability.

5.2.3 Qualitative Comparison

As illustrated in Figure 4, PBE [53] fails to keep object identity, leading to significant deviation from the reference. AnyDoor [6] better preserves identity but faces issues of improper scaling, color mismatches, and misplacement. MimicBrush [7] maintains object structure but suffers from artifacts like copy-paste effects and distortion. In contrast, MimicBrush* and our RoomEditor achieve seamless integration, realism, and correct orientation. Given the large-scale dataset used for pre-training MimicBrush*, our RoomEditor is more suitable for furniture synthesis.

5.3 Generalization to Diverse Scenes and Objects

We evaluate the generalization performance of our model using the DreamBooth dataset [41], which contains 30 object categories, including backpacks, animals, sunglasses, and cartoon characters. Each sample is annotated with object masks, consistent with the RoomBench dataset. In this experiment, the RoomEditor model is neither re-trained nor fine-tuned on external datasets. We compare it with publicly available models from [53, 6, 7]. As shown in Table 3, our method significantly outperforms existing approaches trained on large-scale datasets. These results demonstrate the strong generalization ability of RoomEditor in synthesizing realistic object placements in unseen environments, indicating its robustness for real-world applications. Qualitative results in Figure 12 further show that RoomEditor naturally integrates diverse objects and scenes.

Method	All / Train Params (M)	FID↓	SSIM↑	PSNR↑	LPIPS↓	CLIP↑	DINO↑
RoomEditor (Ours)	943 / 815	18.42	0.793	21.15	0.094	90.51	85.47
RoomEditor+CLIP	1803 / 860	18.38	0.792	21.15	0.094	90.45	85.55
Dual U-Net (Frozen g)+CLIP	2432 / 860	21.12	0.785	20.39	0.105	89.48	83.26
Dual U-Net (Trainable g)+CLIP	2432 / 1717	19.69	0.787	20.56	0.099	90.09	84.36

Table 4: **Ablation Study.** We assess the impact of incorporating CLIP and varying dual U-Net configurations. Adding CLIP improves semantic alignment. Making the reference branch g trainable improves fidelity but increases parameters significantly.

5.4 Ablation Study

Since our RoomEditor is highly concise, we conduct ablation study to evaluate its integration with key techniques (e.g., CLIP image encoder and non-shared reference UNet g) in existing methods as shown in Table 4. First, we examine the integration of CLIP image encoder, which is commonly adopted to facilitate feature extraction from reference images as in MimicBrush, resulting in RoomEditor+CLIP. Although incorporating CLIP leads to a slight improvement in certain metrics, it introduces a substantial increase in the number of parameters. Next, we analyze the effects of non-shared dual U-Net configurations, where the reference U-Net g is either frozen or trainable. The results show that without our parameter-sharing strategy, both variants exhibit significant performance degradation, particularly when the reference U-Net is frozen. We note that the variant with frozen g is exactly the same as MimicBrush[†]. These findings verify the effectiveness of our parameter-sharing U-Net design for unified feature learning.

6 Conclusion

This work advanced furniture synthesis by addressing two key challenges: absence of ready-to-use benchmarks and feature space divergence in dual-branch architectures. Specifically, we collect and release RoomBench, a ready-to-use benchmark for furniture synthesis, while presenting RoomEditor, a parameter-sharing dual U-Net architecture that ensures robust feature alignment. By training our RoomEditor on the collected RoomBench, it achieves state-of-the-art performance for furniture synthesis in terms of geometric coherence and visual fidelity and demonstrates strong generalization ability to diverse scenes. We hope that our work can encourage further research in furniture synthesis.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (62576241 , 62276186, 62172127,U23B2049 and U22B2035).

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18208–18218, 2022.
- [3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [4] Junming Chen, Zichun Shao, and Bin Hu. Generating interior design from text: A new diffusion model-based method for efficient creative design. *Buildings*, 13(7):1861, 2023.
- [5] Mengting Chen, Xi Chen, Zhonghua Zhai, Chen Ju, Xuewen Hong, Jinsong Lan, and Shuai Xiao. Wear-any-way: Manipulable virtual try-on via sparse correspondence alignment. *arXiv:2403.12965*, 2024.

- [6] Xi Chen, Lianghai Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6593–6602, 2024.
- [7] Xi Chen, Yutong Feng, Mengting Chen, Yiyang Wang, Shilong Zhang, Yu Liu, Yujun Shen, and Hengshuang Zhao. Zero-shot image editing with reference imitation. *Advances in Neural Information Processing Systems*, 37:84010–84032, 2025.
- [8] Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14131–14140, 2021.
- [9] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. Color harmonization. In *ACM SIGGRAPH 2006 Papers*, pages 624–630. 2006.
- [10] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8394–8403, 2020.
- [11] Xiaodong Cun and Chi-Man Pun. Improving the harmony of the composite image by spatial-separated attention module. *IEEE Transactions on Image Processing*, 29:4759–4771, 2020.
- [12] Ziyi Dong, Pengxu Wei, and Liang Lin. Dreamartist: Towards controllable one-shot text-to-image generation via positive-negative prompt-tuning. *arXiv preprint arXiv:2211.11337*, 2022.
- [13] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.
- [14] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129:3313–3337, 2021.
- [15] Market Research Future. Interior design market research report information by end use (residential and commercial), by type of decoration (renovation and new), and by region (north america, europe, asia-pacific, and rest of the world) – market forecast till 2034, 2025.
- [16] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion.
- [17] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10696–10706, 2022.
- [18] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7543–7552, 2018.
- [19] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control.
- [20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [21] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.
- [22] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *CVPR*, 2024.
- [23] Jiaya Jia, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Drag-and-drop pasting. *ACM Transactions on graphics (TOG)*, 25(3):631–637, 2006.
- [24] Bahjat Kavar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6007–6017, 2023.

- [25] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2426–2435, 2022.
- [26] Jeongho Kim, Guojung Gu, Minh Park, Sunghyun Park, and Jaegul Choo. Stableviton: Learning semantic correspondence with latent diffusion model for virtual try-on. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8176–8185, 2024.
- [27] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1931–1941, 2023.
- [28] Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. Tryongan: Body-aware try-on via layered interpolation. *ACM Transactions on Graphics (TOG)*, 40(4):1–10, 2021.
- [29] Dongxu Li, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *Advances in Neural Information Processing Systems*, 36:30146–30166, 2023.
- [30] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. *arXiv preprint arXiv:1809.00716*, 2018.
- [31] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 289–299, 2023.
- [32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization.
- [33] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. pages 16784–16804, 2022.
- [34] OpenAI. Introducing 4o image generation, 2025.
- [35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, pages 1–31, 2024.
- [36] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- [37] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.
- [38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.
- [40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [41] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dream-booth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.
- [42] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

- [43] Mehmet Saygin Seyfioglu, Karim Bouyarmane, Suren Kumar, Amir Tavanaei, and Ismail B Tutar. Dream-paint: Few-shot inpainting of e-commerce items for virtual try-on without 3d modeling. *arXiv preprint arXiv:2305.01257*, 2023.
- [44] Mehmet Saygin Seyfioglu, Karim Bouyarmane, Suren Kumar, Amir Tavanaei, and Ismail B Tutar. Diffuse to choose: Enriching image conditioned inpainting in latent diffusion models for virtual try-all. *arXiv preprint arXiv:2401.13795*, 2024.
- [45] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. *ACM Transactions on Graphics (TOG)*, 29(4):1–10, 2010.
- [46] Michael W Tao, Micah K Johnson, and Sylvain Paris. Error-tolerant image compositing. *International journal of computer vision*, 103:178–189, 2013.
- [47] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3789–3797, 2017.
- [48] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [49] Zhaowei Wang, Ying Hao, Hao Wei, Qing Xiao, Lulu Chen, Yulong Li, Yue Yang, and Tianyi Li. Roomdiffusion: A specialized diffusion model in the interior design industry. *arXiv preprint arXiv:2409.03198*, 2024.
- [50] Yuhao Xu, Tao Gu, Weifeng Chen, and Chengcai Chen. Ootdiffusion: Outfitting fusion based latent diffusion for controllable virtual try-on. *arXiv preprint arXiv:2403.01779*, 2024.
- [51] Zhengze Xu, Mengting Chen, Zhao Wang, XING Linyu, Zhonghua Zhai, Nong Sang, Jinsong Lan, Shuai Xiao, and Changxin Gao. Tunnel try-on: Excavating spatial-temporal tunnels for high-quality virtual try-on in videos.
- [52] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. In *CVPR*, 2024.
- [53] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023.
- [54] Ziyang Yuan, Mingdeng Cao, Xintao Wang, Zhongang Qi, Chun Yuan, and Ying Shan. Customnet: Zero-shot object customization with variable-viewpoints in text-to-image diffusion models. *arXiv preprint arXiv:2310.19784*, 2023.
- [55] Bo Zhang, Yuxuan Duan, Jun Lan, Yan Hong, Huijia Zhu, Weiqiang Wang, and Li Niu. Controlcom: Controllable image composition using diffusion model. *arXiv preprint arXiv:2308.10040*, 2023.
- [56] Lvmin Zhang. Reference-only controlnet. <https://github.com/Mikubill/sd-webui-controlnet/discussions/1236>, 2023.
- [57] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.
- [58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [59] Shilong Zhang, Lianghua Huang, Xi Chen, Yifei Zhang, Zhi-Fan Wu, Yutong Feng, Wei Wang, Yujun Shen, Yu Liu, and Ping Luo. Flashface: Human image personalization with high-fidelity identity preservation. *arXiv:2403.17008*, 2024.
- [60] Xujie Zhang, Ente Lin, Xiu Li, Yuxuan Luo, Michael Kampffmeyer, Xin Dong, and Xiaodan Liang. Mmtryon: Multi-modal multi-reference control for high-quality fashion generation. *arXiv preprint arXiv:2405.00448*, 2024.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: In the abstract, we present a summary of the overall content and our contributions.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We include the discussion of limitations in the appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: see section 4

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Detailed information and hyperparameters are provided to facilitate reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: we make the dataset and code available at <https://anonymous.4open.science/status/RoomEditor-77B7>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: see subsection 5.1

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Due to our limited computational and human resources, we don't report error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: see subsection 5.1

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Yes, we follow the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the potential positive and negative societal impacts of our work in the conclusion section.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our contribution does not include new datasets or pre-trained models that pose a risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Code and model that we derive from earlier work is properly licensed and referenced.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We provide detailed illustration of new data benchmark and visualized examples.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[Yes\]](#)

Justification: We employ annotators for our research and provide them with lawful compensation.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: Do not apply.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Implementation Details.

Hyperparameters. In our experiments, the inpainting U-Net is initialized with the Stable Diffusion 1.5 [40] inpainting model, which takes a 9-channel input, while the reference U-Net is initialized with the standard Stable Diffusion 1.5 [40] model, which takes a 4-channel input. For the CLIP [38] model, we employ CLIP-H as the image encoder, following [7]. During both training and inference, images are processed to a resolution of 512×512 by first padding them to a square aspect ratio and then resizing them accordingly. To compute the CLIP-score and DINO-score, we use CLIP ViT-B/32 [38] and DINO ViT-S/16 [3], following the protocols of [41, 53, 6, 7]. For the user study, we recruited 20 annotators to evaluate and rank different methods based on fidelity, harmony, and overall quality.

Data Augmentation To enhance robustness against imperfect masks and domain variations, we apply both image and mask augmentation, as shown in Table 5.

Type	Augmentation	Description and Parameters
Image	Horizontal Flipping	Applied with a probability of 50%.
	Rotation	Applied with a probability of 50%, up to 30° .
	Scaling	Applied with 30% probability, range $\pm 20\%$.
	Cropping	Minimum retained ratio: 0.75.
Mask	Perturbation	25% probability: Random dilation and erosion for variation.
	Blurring	25% probability: Boundary smoothing via blurring and thresholding.
	BBox	25% probability: Mask replaced with its bounding box.
	No Augmentation	25% probability: Original fine-grained mask retained.

Table 5: Image and Mask Augmentation Strategies

B RoomBench Dataset

In this section, we introduce the construction and characteristics of our RoomBench dataset. As summarized in Table 6, RoomBench covers a broad range of furniture categories with diverse styles and high-quality images, enabling realistic simulation of real-world indoor scenes.

To construct the dataset, we first collected raw images from the internet. The data filtering process, illustrated in Figure 5, was conducted with the assistance of GPT-4o [1], which helped to efficiently filter and organize high-quality image-text pairs. For the training set, we performed precise, pixel-level mask annotation, which supports various mask augmentations—examples of which are shown in Figure 6. In contrast, the testing set was annotated using coarse masks to better reflect real-world conditions, where user-provided annotations are often noisy or imprecise, as depicted in Figure 7. This annotation strategy allows us to evaluate model robustness under imperfect labeling scenarios.

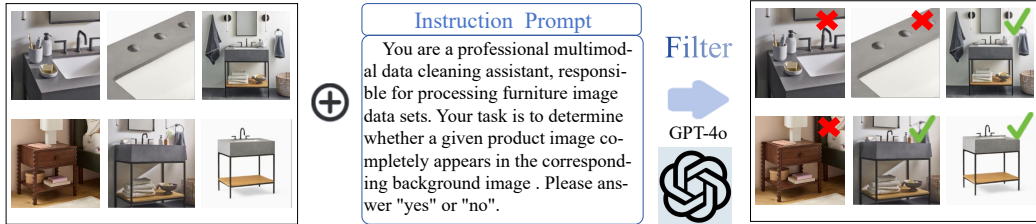


Figure 5: Diagram of data filtering, where images first are labeled as furniture and background images. Then, GPT-4o with specific prompt is used to filter out unmatched and incomplete images.

Categories	Training Set	Testing Set
Sofa	1294	61
Lamp	1280	231
Coffee Table	805	24
Wardrobe	681	13
Armchair	563	61
Bed	460	9
Pillow	440	73
Nightstand	319	36
Desk and Chair	158	118
Other	1298	269
Total	7298	895

Table 6: Categories and distribution in our RoomBench.



Figure 6: Different forms of data augmentation for precisely annotated mask in training set.



Figure 7: In RoomBench, precise masks are annotated for the training samples and can be augmented accordingly. Coarse masks are provided for the testing samples to simulate user interactions.

C Additional Experiments

C.1 Additional Verification Experiments for Eqn. (5) and Eqn. (6)

From subsubsection 4.2.1, we derived Eqn. (5) and Eqn. (6) based on the empirical properties of inpainting models and assumptions about the feature space. To more rigorously justify the validity of these assumptions and properties, we conducted the following experiments.

Training Loss (Avg.)	Region Merging Loss (Avg.)
3.9×10^{-2}	4.3×10^{-3}

Table 7: Quantitative comparison between the average training loss and the region merging loss.

Layer	d0	d1	d2	d3	d4	d5	m0	u0	u1	u2	u3	u4	u5	u6	u7	u8
cos_sim	0.962	0.955	0.942	0.931	0.922	0.917	0.912	0.904	0.906	0.913	0.921	0.925	0.927	0.937	0.941	0.945

Table 8: Cosine similarity across feature layers. Here, d, m, and u correspond to the *downsampling*, *middle*, and *upsampling* blocks of the network, respectively.

For Eqn. (5), the core assumption is that the inpainting diffusion model can preserve the unmasked regions of an image with high fidelity. Therefore, our goal is to demonstrate that the expression

$$\mathbf{M}_{\text{bg}} \odot \epsilon_{\theta}(\mathbf{I}_{\text{bg}}^{\text{M}}) + \overline{\mathbf{M}_{\text{bg}}} \odot \epsilon_{\theta}(\mathcal{R}(\mathbf{I}_{\text{ref}}^{\text{M}} | \mathbf{I}_{\text{bg}}^{\text{M}}))$$

can effectively approximate the ground truth (GT) denoising prediction $\epsilon_{\theta}(\mathbf{I}_{\text{gt}})$. To verify this assumption, we conducted experiments using a pretrained diffusion inpainting model on our full test set. During the experiments, we added random noise to the input images and sampled random timesteps t . The model was then provided with partial content from the full image (either the background or the object regions). We extracted the predicted noise components from the corresponding regions and computed the L_2 loss as the evaluation metric:

$$\|\epsilon_{\theta}(\mathbf{I}_{\text{gt}}) - (\mathbf{M}_{\text{bg}} \odot \epsilon_{\theta}(\mathbf{I}_{\text{bg}}^{\text{M}}) + \overline{\mathbf{M}_{\text{bg}}} \odot \epsilon_{\theta}(\mathcal{R}(\mathbf{I}_{\text{ref}}^{\text{M}} | \mathbf{I}_{\text{bg}}^{\text{M}})))\|_2.$$

The results summarized in Table 7 show that the merged result—obtained by feeding the model with separated regions and combining their corresponding outputs—yields a significantly lower loss than the average training loss (i.e., the average denoising loss during model training, visualized in Figure 3), thereby supporting the validity of the assumption in Eqn. (5).

For Eqn. (6), the underlying assumption is that a property analogous to that in Eqn. (5) also holds at the feature level. To evaluate this hypothesis from a similarity perspective, we adopt *cosine similarity* as the evaluation metric. Specifically, we compute the cosine similarity between the feature representations extracted from different attention layers on the validation set, defined as

$$\text{cos_sim}\left(f_l(\mathbf{I}_{\text{gt}}), \mathbf{M}_{\text{bg}} \odot f_l(\mathbf{I}_{\text{bg}}^{\text{M}}) + \overline{\mathbf{M}_{\text{bg}}} \odot f_l(\mathcal{R}(\mathbf{I}_{\text{ref}}^{\text{M}} | \mathbf{I}_{\text{bg}}^{\text{M}}))\right),$$

where $f_l(\cdot)$ denotes the feature map extracted from the l -th layer of the model. The results in Table 8 show that cosine similarities remain consistently above 0.9 across most layers, supporting the validity of the assumption in Eqn. (6) and indicating that the feature-level composition effectively preserves the representational coherence of the full image.

C.2 Generalization to Unseen Datasets: 3D-FUTURE Results

To evaluate the generalization ability of our method on unseen datasets, we conducted cross-dataset experiments using the 3D-FUTURE [14] dataset, which differs from our RoomBench in visual domain and object composition.

We selected 1,020 samples covering 34 furniture categories (randomly 30 samples per category). As the dataset does not provide paired reference images, we generated pseudo-pairs by horizontally flipping reference objects and applying Gaussian-blurred masks to prevent trivial copy-paste solutions.

As shown in Table 9, despite being trained exclusively on RoomBench, our model achieves comparable or superior performance to MimicBrush [7] on most metrics, demonstrating strong generalization to unseen furniture data and novel datasets without fine-tuning. The visualized results are shown Figure 8.

C.3 Comparison with GPT-4o

To further benchmark our method against a powerful commercial system, we include the closed-source GPT-4o [34] in our evaluation. As a state-of-the-art multimodal model capable of handling multi-

Method	FID↓	SSIM↑	PSNR↑	LPIPS↓	CLIP↑	DINO↑
MimicBrush [7]	14.20	0.6595	20.64	0.272	80.10	64.61
RoomEditor (Ours)	13.91	0.6581	20.93	0.260	80.28	66.97

Table 9: Cross-dataset evaluation on 3D-FUTURE [14]. Our model outperforms MimicBrush across multiple metrics, demonstrating strong generalization to unseen data distributions.



Figure 8: Additional visualization results on the 3D-FUTURE [14] dataset.

Model	FID (↓)	SSIM (↑)	PSNR (↑)	LPIPS (↓)	CLIP (↑)	DINO (↑)
MimicBrush [7]	56.44	0.780	19.40	0.113	88.12	79.49
GPT-4o [34]	72.12	0.502	13.84	0.290	87.80	83.30
RoomEditor (Ours)	44.78	0.792	21.05	0.094	90.42	85.12

Table 10: Quantitative comparison with GPT-4o [34] on a 200-sample subset of RoomBench.

image inputs, GPT-4o exhibits strong general-purpose image editing and composition capabilities, making it a valuable reference for assessing high-level visual reasoning and semantic understanding.

For a fair and efficient comparison, we evaluate GPT-4o [34], MimicBrush [7], and our method on a randomly sampled subset of 200 images from the RoomBench dataset. This subset size ensures sufficient scene diversity while keeping closed-source inference costs tractable. A comprehensive set of quantitative metrics is used to capture complementary aspects of visual fidelity and semantic consistency, including FID, SSIM, PSNR, LPIPS, CLIP, and DINO. Among these, CLIP and DINO are particularly informative for assessing object-level fidelity—one of the key challenges in compositional image editing.

The quantitative results are summarized in Table 10. Metrics that emphasize fine-grained spatial and structural consistency, such as FID and LPIPS, show GPT-4o performing notably worse than both MimicBrush and our method. In contrast, its CLIP and DINO scores are comparable to or even higher than those of MimicBrush, reflecting strong global semantic coherence and visually appealing compositions.

While GPT-4o [34] produces visually coherent and semantically meaningful results, it shows two consistent weaknesses. First, **background preservation**: GPT-4o often modifies the surrounding background, introducing shifts in color or texture that reduce spatial consistency and thus worsen FID and LPIPS. Second, **object fidelity**: for certain objects, the generated results exhibit deviations from the reference in geometry or texture, which contributes to lower SSIM and PSNR scores. The visualized results of the comparison are shown in Figure 9.



Figure 9: Our RoomEditor compares its visual results with those of Mimicbrush [7] and GPT-4o [34] on RoomBench.

Method	FID↓	SSIM↑	PSNR↑	LPIPS↓	CLIP↑	DINO↑
MimicBrush [7]	24.99	0.761	18.42	0.135	87.06	75.13
RoomEditor (Ours)	21.16	0.766	19.83	0.120	88.90	81.46

Table 11: Evaluation of robustness to mask quality using bounding boxes as masks. Our method maintains high performance even without accurate shape priors.

C.4 Robustness to Mask Quality

To further assess the robustness of our model to imperfect masks, we evaluated it using coarse bounding boxes as input masks, thereby removing any explicit shape prior and simulating practical scenarios where precise segmentation is unavailable.

As described in Appendix B, our training strategy already introduces imperfect masks through random perturbations and simplified bounding boxes, encouraging the model to learn shape-consistent reconstruction even under imprecise guidance. During evaluation, we adopt a similar setup by replacing object masks with axis-aligned bounding boxes covering target regions.

As summarized in Table 11, despite lacking detailed shape cues, our method outperforms MimicBrush [7] on most metrics, demonstrating strong robustness to mask variations and high-quality synthesis even with coarse or noisy masks—highlighting its practicality for real-world editing scenarios.

Light Intensity	0.6	0.8	1.0 (base)	1.2	1.4
MimicBrush [7]	70.58	71.16	71.92	70.77	69.97
RoomEditor (Ours)	73.80	73.96	74.77	73.48	72.76

Table 12: Performance under different lighting intensities. Higher values indicate better performance.

Model	Params (M)	GPU Mem (GB)	Speed (samples/min)
PBE [53]	<u>1310</u>	<u>3.9</u>	<u>14.5</u>
AnyDoor [6]	2451	7.3	14
MimicBrush [7]	2432	6.9	10.7
RoomEditor (Ours)	943	3.2	15.8

Table 13: Comparison of computational efficiency across different methods.

C.5 Robustness to Lighting Changes

To further evaluate the robustness of our method under varying illumination conditions, we conducted additional experiments with systematically adjusted lighting intensities. Since the original test set does not explicitly account for lighting variation, we simulated different illumination conditions by scaling the brightness of background images. Specifically, brightness scaling factors of 0.6 and 0.8 (darker conditions), as well as 1.2 and 1.4 (brighter conditions), were applied to the entire test set.

We compared our method with MimicBrush [7] under these modified lighting conditions, and report the averaged results across all metrics (rescaled to a 0–100 range, with FID and LPIPS inverted for consistency). As shown in Table 12, although both methods slightly degrade under extreme lighting, our approach consistently outperforms [7] across all illumination levels, demonstrating strong robustness and generalizability to lighting variations without explicit modeling.

C.6 Computational Efficiency Analysis

To provide a fair comparison of computational efficiency across different methods, we report the model size, memory usage, and inference speed on a unified experimental setup. All experiments were conducted on a single NVIDIA A6000 GPU in FP16 precision, using single-image inference with 50 denoising steps.

Table 13 summarizes model efficiency in terms of parameters (Params), GPU memory usage (Mem), and inference throughput (Speed). Our method strikes a favorable balance between performance and efficiency—using significantly fewer parameters and less memory than [7] while achieving about $1.5\times$ higher inference speed—demonstrating its suitability for practical deployment.

D Additional Visualization Results

In this section, we present additional qualitative results. Figure 10 showcases our results on the RoomBench dataset, while Figure 11 illustrates our performance on the DreamBooth dataset. As observed, our method consistently achieves high fidelity and seamless harmony, demonstrating its effectiveness across both indoor environments and general scenarios. Finally, as shown in Figure 12, qualitative results demonstrate that RoomEditor effectively integrates diverse objects into complex scenes, further validating the utility of RoomBench for real-world applications.

E Limitations and Future Work

Our method has two primary limitations, which we plan to address in future work.

First, the presence of uncommon furniture items (such as clothes racks and microwaves) results in a long-tail distribution in our collected dataset. Consequently, our model may struggle to capture complex or rare furniture shapes, as the large intra-category variation in geometry makes it difficult for the dataset to fully represent all possible shape patterns. This occasionally leads to distorted



Figure 10: Additional visualization results on the RoomBench dataset, demonstrating the high fidelity and seamless integration of furniture into indoor environments.



Figure 11: Additional visualization results on the DreamBooth dataset, showcasing the performance of our method in various general scenarios with high fidelity and consistent harmony.

or inconsistent generations, as illustrated in Figure 13. The issue stems from both the insufficient training samples for such items and the challenge of learning robust shape priors under limited data diversity.

Second, our current approach does not explicitly model complex visual phenomena such as object shadows and varying illumination conditions. Since the model edits only the masked region, shadows or lighting effects outside the mask cannot be synthesized. This limitation is inherent to the masked-editing paradigm and is also shared by prior works such as PBE [53], AnyDoor [6], and MimicBrush [7]. These factors may significantly affect visual realism and remain challenging for both our method and existing approaches.

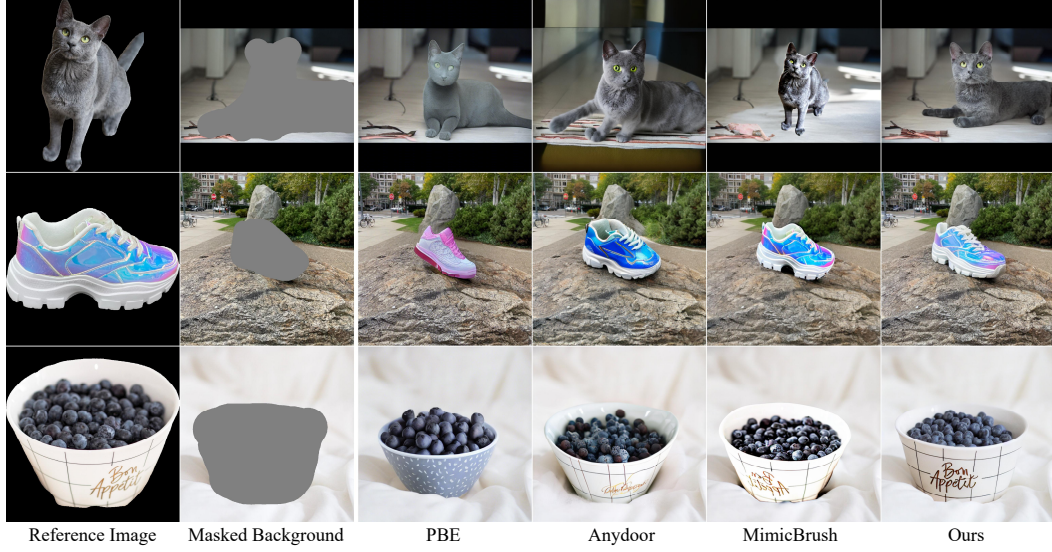


Figure 12: Visual comparison on DreamBooth dataset [41].



Figure 13: Failure cases illustrating distortions and inconsistencies when handling less common furniture items.

Additionally, given the structural differences between our model and recent DiT-based [36] diffusion architectures, adapting our formulation and analytical framework to more advanced DiT diffusion models represents a promising direction for future research.

We leave these aspects for future exploration.