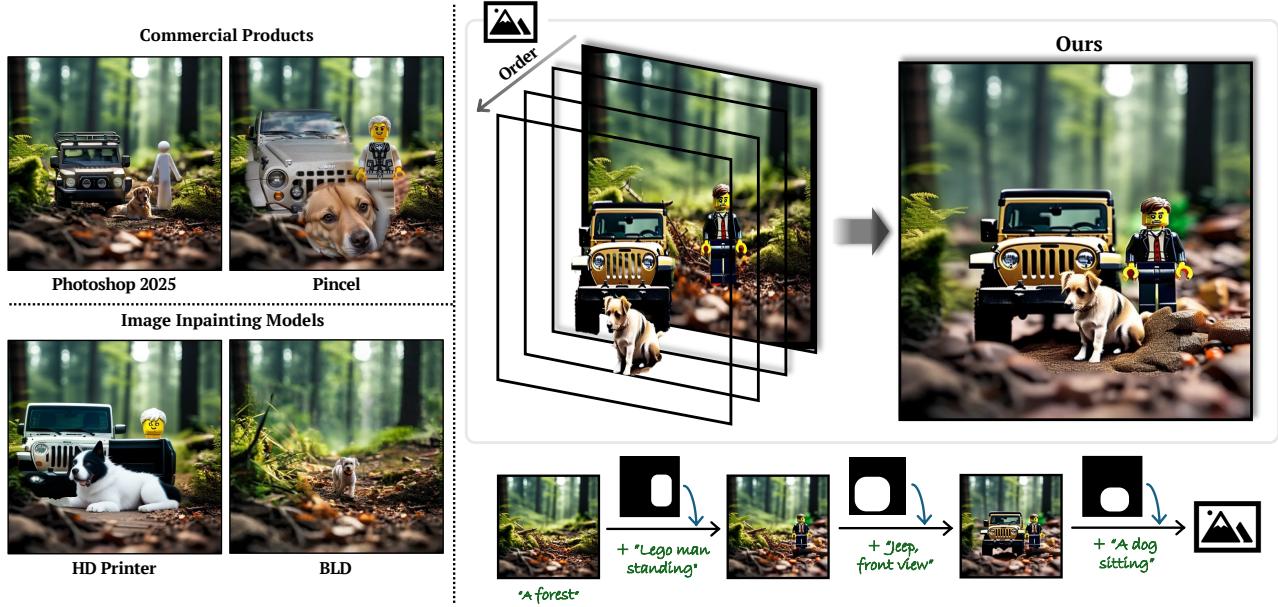


# Improving Editability in Image Generation with Layer-wise Memory

Daneul Kim Jaeah Lee Jaesik Park\*  
 Seoul National University, Republic of Korea  
 {carpedkm, hayanz, jaesik.park}@snu.ac.kr



**Figure 1. Overview.** Our framework enables the interactive generation of images with enhanced control but in a simple manner, by rough mask and prompt, through iterative scene editing. We utilize the background scene generated by our framework to edit in HD Painter [32] or Blended Latent Diffusion (BLD) [3] for comparison and commercial products like Photoshop [1] and Pincel [37].

## Abstract

Most real-world image editing tasks require multiple sequential edits to achieve desired results. Current editing approaches, primarily designed for single-object modifications, struggle with sequential editing: especially with maintaining previous edits along with adapting new objects naturally into the existing content. These limitations significantly hinder complex editing scenarios where multiple objects need to be modified while preserving their contextual relationships. We address this fundamental challenge through two key proposals: enabling rough mask inputs that preserve existing content while naturally integrating new elements and supporting consistent editing across multiple modifications. Our framework achieves this through layer-wise memory, which stores latent representations and

prompt embeddings from previous edits. We propose Background Consistency Guidance that leverages memorized latents to maintain scene coherence and Multi-Query Disentanglement in cross-attention that ensures natural adaptation to existing content. To evaluate our method, we present a new benchmark dataset incorporating semantic alignment metrics and interactive editing scenarios. Through comprehensive experiments, we demonstrate superior performance in iterative image editing tasks with minimal user effort, requiring only rough masks while maintaining high-quality results throughout multiple editing steps.

## 1. Introduction

Recent advances in text-to-image synthesis through powerful diffusion models like Stable Diffusion [20, 42], PixArt [12, 13] and FLUX [11] have transformed visual

\*Corresponding author.

content creation. However, users often demand multiple sequential edits to achieve desired results, iteratively refining and adding elements to their images. Current approaches in inpainting [3, 32] are primarily designed for single-object modifications with limited changes like colors or styles, making them inadequate for complex editing scenarios that involve multiple objects and their interactions.

Image editing approaches, while offering localized modifications, face several limitations. Current methods [3, 16, 32, 48] struggle when modifications need to be applied sequentially, often demanding precise segmentation masks [16] or external modules [39] to maintain background integrity [32]. For example, generating “*A dog sitting*” as shown in Fig. 1 remains challenging, as it requires maintaining previous edits while ensuring the “*dog*” naturally blends with the “*Jeep*” and “*Lego man standing*”—a complex iterative editing scenario that current methods [1, 3, 32, 37] struggle to handle.

Layout-to-image generation offers an alternative approach through various inputs including bounding boxes [28, 47, 58], depth maps [26, 54], and semantic masks [7, 27]. However, these methods require complete regeneration of the entire image for each modification, making iterative editing particularly cumbersome—unable to maintain the surrounding background. While recent work has explored layered representations [25, 41, 55] and instance-based generation [46], they require complex optimization processes or additional training, limiting their practical applicability in iterative editing scenarios.

We address these limitations through two key innovations. First, we enable object placement aligning with the user’s intention using only rough mask inputs while preserving background context. Second, we support consistent iterative editing across multiple modifications. To achieve this, we introduce the concept of *mask order*, which specifies the sequence of object generation during iterative image editing. In Fig. 1, we add “*a lego man*”, “*Jeep*” sequentially with different mask order, but they naturally adapt into the image, side by side. If we add “*A dog sitting*” with overlapping mask, then it means that we aim to put the “*dog*” in front of the “*Jeep*” and “*a lego man*”, making the instance order of “*dog*” in front of “*Jeep*” and “*a lego man*”.

To handle this, we incorporate three technical components: (1) Layer-wise memory for storing editing history, (2) Background Consistency Guidance (BCG) for maintaining unedited regions, and (3) Multi-Query Disentanglement (MQD) in cross-attention for natural object integration. The layer-wise memory stores and manages the latent representations and prompt embeddings from previous editing steps, eliminating redundant computations typical in sequential modifications while maintaining consistency across multiple edits. The BCG not only ensures unedited regions remain stable but also reduces computational over-

head by avoiding repeated forward passes on the original image, enabling reduced editing time, while MQD enables the natural integration of new objects with existing content.

Additionally, we propose Multi-Edit Bench, a comprehensive benchmark for evaluating iterative image editing capabilities. Prior benchmarks either focus on single-turn edits [31, 45] or layout-to-image generation [4, 17, 21], failing to capture the challenges of sequential modifications. Our benchmark introduces layer-wise semantic evaluation metrics to assess both edit quality and cross-modification consistency in multi-step editing scenarios.

To summarize, our key contributions include:

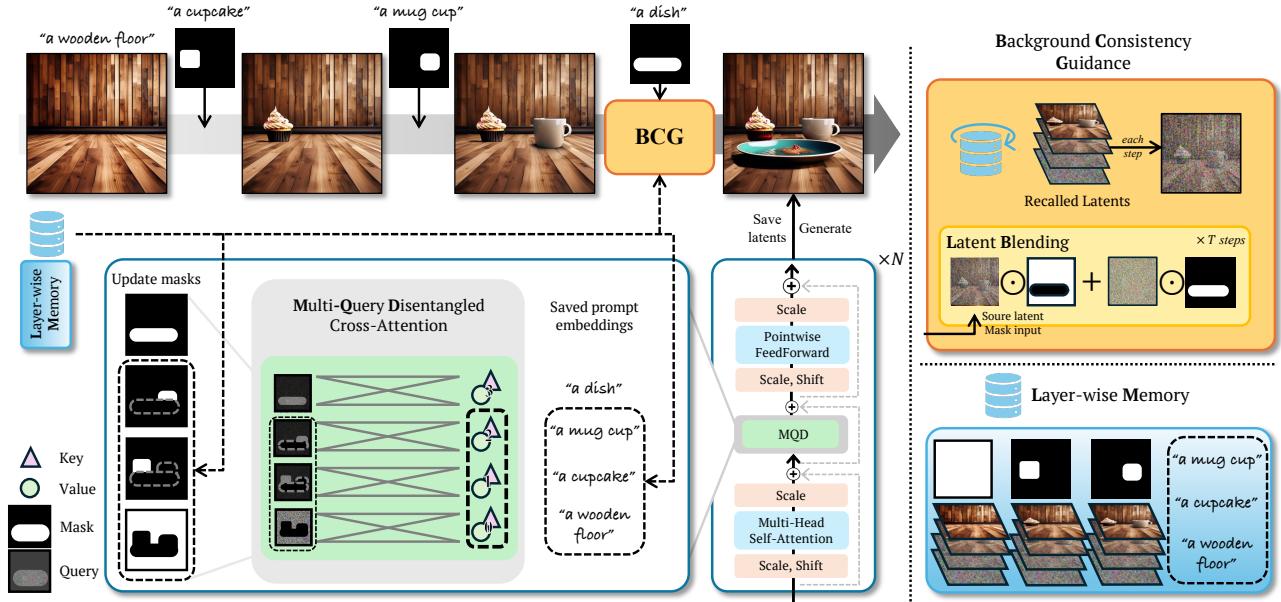
- A framework for interactive mask order-based object placement editing with only rough mask inputs.
- A localized editing mechanism using layer-wise memory that maintains consistency across multiple edits.
- Multi-query disentangled cross attention, allowing natural integration while preserving background context.
- Multi-Edit Bench, a comprehensive benchmark for evaluating the semantic alignment and sequential iterative editing capabilities.

## 2. Related Work

**Image Inpainting.** Image inpainting has evolved from classical patch-based [8, 18] and interpolation-driven [5, 9] methods to powerful deep-learning approaches [49–51] that better handle complex structures. Text-conditioned inpainting [51, 53] further broadened user control, and diffusion-based models [24]—including GLIDE [34], Stable Diffusion [42], and Blended Diffusion [2, 3]—significantly improved realism. Yet, existing works often focus on single-object edits. Recent attempts to relax mask precision [32, 48] or enhance quality [16] remain limited in multi-step scenarios, struggling to maintain consistency across multiple objects or occlusions—hence insufficient for iterative real-world editing tasks.

**Layout-to-Image Generation.** Controllable image synthesis has recently emphasized spatial arrangement via bounding boxes, segmentation masks, or other layout cues [14, 46, 56]. Advances in diffusion-based layout calibration [23] and grounding [43] have improved fidelity, and interactive methods [15, 30, 35] aim to let users refine elements step by step. However, most such pipelines regenerate the entire image for each new edit, losing previously established context and coherence. By contrast, our approach incrementally updates only relevant regions through a memory-assisted process, naturally preserving edited objects.

**Evaluation Benchmark.** Most existing editing benchmarks focus on single-turn modifications [31, 45], while layout-centric datasets such as HRS-1k [4] and NSR-1k [21] assess scene quality or prompt alignment in a single-shot manner. Consequently, they overlook the iterative nature of real-world edits where object placements evolve over



**Figure 2. Overview.** (a) The left denotes an illustration of how Multi Query Disentanglement is performed in the cross-attention layer. (b) The upper right figure shows Background Consistency Guidance with recalled latents, conducting latent blending with the saved latents. (c) The right below shows the layer-wise memory, saving the previous editing steps’ latents, masks, and prompt embeddings.

multiple steps. To address this gap, we introduce *Multi-Edit Bench*, a new benchmark that evaluates multi-step editing in terms of semantic accuracy and visual alignment across the entire editing sequence. This multi-step perspective better captures practical editing workflows, where users iteratively refine scenes by adding, removing, or repositioning objects, thereby measuring a method’s capacity to preserve context and maintain coherent compositions over multiple, potentially complex modifications.

### 3. Method

#### 3.1. Overview

We propose a framework for iterative image editing with three key components: Layer-wise memory, background consistency guidance (BCG), and multi-query disentangled cross-attention (MQD). Layer-wise memory stores editing history to enable consistent object placement across sequential edits. BCG leverages this stored information for efficient latent blending while preserving background integrity. MQD ensures the natural object integration through disentangling the attention between queries and latents, handling complex scenarios as shown in Fig. 2 like placing “*a dish*” in front of “*a cupcake*” and “*a mug cup*”.

Our framework’s architecture, built upon transformer-based diffusion models [13], operates through the following systematic workflow: First, we detail the layer-wise memory mechanism (Fig. 2(c)) in Sec. 3.2, which stores editing history. Then, in Sec. 3.3, we explain how BCG leverages

this stored information for efficient latent blending (Fig. 2 (b)). Sec. 3.4 describes MQD’s role in ensuring natural object integration through query disentanglement (Fig. 2 (a)). Finally, in Sec. 3.5, we demonstrate how these components enable advanced editing capabilities, including the removal of occluded elements while preserving foreground integrity.

#### 3.2. Layer-wise Memory

Layer-wise memory enables stable background preservation while ensuring the natural integration of new objects during iterative editing with *mask order*. By storing and managing key information from each editing step according to the specified mask order, the model can reference previous edits when generating new content while maintaining the intended spatial relationships between objects.

Let the set of prompts for generating the background and objects be denoted as  $P_l = \{p_0, p_1, p_2, \dots\}$ , where  $p_0$  is the background generation prompt, and  $p_1, p_2, \dots$  represent the prompts for each object. Similarly, let the set of corresponding masks be denoted as  $M_l = \{m_0, m_1, m_2, \dots\}$ , where each  $m_i$  defines the region of interest (RoI) for the object associated with prompt  $p_i$ . Note that  $m_0$  represents the case with mask of all elements being 1.

The layer-wise memory  $L_l = \{l_0, l_1, l_2, \dots\}$  stores three key elements for each step  $i$ :  $l_i = \{\mathbf{p}_i, \{\mathbf{Z}_i^t\}_{t=1}^T, m_i\}$ , where:  $\mathbf{p}_i$  denotes prompt embedding that guides the generation,  $\{\mathbf{Z}_i^t\}_{t=1}^T$  as denoising latents across timesteps and  $m_i$  as a mask defining the object’s RoI.

For a generation, the initial background latent is cre-

**Table 1. Computational comparison in utilizing Background Consistency Guidance (BCG).** We utilize PixArt- $\alpha$  inpaint pipeline with vanilla latent blending and our custom-implemented pipeline with BCG for comparison. We ran the inpainting task for the single-step editing 5 times for each method. Note that we use PixArt-alpha-XL-1024 pretrained weight for the experiment.

Method	Time (sec)	VRAM
Latent Blending	$4.1218 \pm 0.0521$	<b>16.89GB</b>
BCG	<b>3.8992±0.0142</b>	16.90GB

ated as  $\mathbf{Z}_0 = f_\theta(p_0)$ , where  $f_\theta$  is the pre-trained diffusion model. For each subsequent object  $p_i$ , we initialize its latent independently:  $\mathbf{Z}_i = f_\theta(p_i, m_i)$ . This independence in initialization, combined with the stored history in layer-wise memory, enables flexible generation while maintaining background consistency.

The layer-wise memory serves two key purposes: (1) Context preservation, to store the complete editing history, enabling the model to maintain consistency with previously generated content (2) Localized editing, enabling control over specific regions while preserving the surrounding content through mask-guided generation

By maintaining this structured history of edits, the model can effectively handle complex scenarios where multiple objects need to be integrated coherently. The stored information guides background consistency and natural object integration, as detailed in the following sections.

### 3.3. Background Consistency Guidance

Background Consistency Guidance (BCG) addresses a key challenge in iterative image editing: maintaining the integrity of previous edits while incorporating new elements efficiently. Unlike traditional inpainting approaches, BCG leverages layer-wise memory to ensure stability across multiple sequential modifications.

**Latent Retrieval from Layer-wise Memory:** For each editing step  $i$ , BCG retrieves information from the previous edits stored in layer-wise memory  $L_l$ , enabling seamless integration of new content while preserving prior modifications:

$$l_{i-1} = \{\mathbf{p}_{i-1}, \{\mathbf{Z}_{i-1,t}\}_{t=1}^T, m_{i-1}\} \quad (1)$$

where  $\mathbf{p}_{i-1}$  is the prompt embedding from the prior step,  $\{\mathbf{Z}_{i-1,t}\}_{t=0}^T$  represents the stored latents from each denoising step, and  $m_{i-1}$  is the previous mask. This stored latent provides context for the current edit, preserving the background information without requiring a re-computation on the original image.

**Ensuring Consistency:** When adding a new object, we update only the masked region while preserving the rest:

$$\mathbf{Z}_i = \mathbf{Z}_{i-1} \odot (1 - m_i) + \mathbf{Z}_i \odot m_i \quad (2)$$

This selective update mechanism is crucial for iterative editing since it ensures that each new modification preserves previous edits. The element-wise multiplication  $\odot$  enables control over which regions are updated, maintaining the integrity of the evolving image composition.

### Computational Efficiency for Iterative Editing:

BCG improves the efficiency of iterative image editing by reducing redundant computations in sequential modifications. Traditional inpainting methods [3] that utilize latent blending (LB) require a forward pass on the original image for each edit, becoming costly with multiple edits. While both approaches (*i.e.*, LB, BCG) require denoising costs, BCG avoids repeated forward passes. Let  $\Omega = \text{FLOPs}(T, L, H, W)$  represent the base computational cost for processing an image of size  $H \times W$  through  $T$  denoising steps and  $L$  model layers, and  $C_f$  denote the forward pass cost. The total cost for LB and BCG can be expressed as:

$$\text{Cost}_{\text{LB}} = C_f + \Omega. \quad (3)$$

While for BCG, by skipping the forward pass:

$$\text{Cost}_{\text{BCG}} = \Omega, \quad (4)$$

Assuming  $C_f \approx r\Omega$  with ratio  $r \in (0, 1)$ , we can estimate:

$$\text{Efficiency Gain} = \frac{\text{Cost}_{\text{LB}}}{\text{Cost}_{\text{BCG}}} \approx 1 + r \quad (5)$$

In Tab. 1, we observe about 10% reduction in computational time for single-step editing with BCG compared to LB. This improvement becomes more significant in scenarios requiring multiple sequential modifications, as LB requires multiple forward passes while BCG requires none.

### 3.4. Multi-Query Disentangled Cross-Attention

Multi-query disentangled cross-attention (MQD) ensures the natural integration of objects across different mask orders while preserving their spatial relationships. For example, when generating “a dish” as in Fig. 2, MQD enables the ‘dish’ to blend naturally into the background, despite being in different mask orders, while maintaining the structure and color of the background with proper occlusion of other objects like ‘a mug cup’ and ‘a cupcake’ behind. This is achieved by disentangling attention across multiple queries and leveraging information from layer-wise memory  $L_l$ .

#### 3.4.1. Cross-Attention for Current Object

For each generation step  $i$ , MQD focuses attention on the current mask order’s region of interest (RoI):

$$\mathbf{z}_i^{k,attn} = \text{CrossAttention}(\mathbf{z}_i^{k,t} \odot m_i, p_i) \quad (6)$$

where  $\mathbf{z}_i^{k,t}$  is the latent for the  $k$ -th block at denoising step  $t$ ,  $m_i$  defines the RoI, and  $p_i$  is the prompt embedding. This

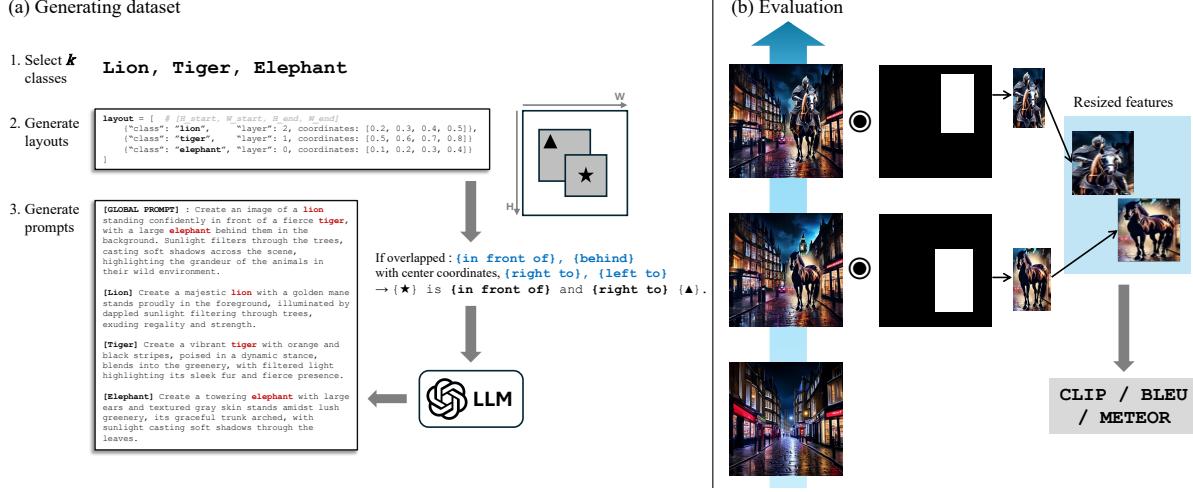


Figure 3. Overview of our proposed Multi-Edit Benchmark for evaluation of iterative editing scenario. (a) explains the dataset generation pipeline through GPT-4 API, and (b) explains the evaluation methodology in visual alignment using CLIP and semantic alignment using LLaVa for single-image and in a layer-wise manner.

ensures focused generation within the current mask order’s region. In the Fig. 2,  $m_i$  is given as a mask corresponding to the  $p_i$  ‘a dish’ with  $\mathbf{z}_i^{k,t}$  corresponding to the latent of the masked region for ‘a dish’.

### 3.4.2. Disentangled Attention for Previous Mask Orders

To adapt the mask orders into the appropriate instance orders, MQD applies attention to non-overlapping regions from previous steps:

$$\mathbf{z}_i^{k,attn} = \bigcup_{j=0}^{i-1} \text{CrossAttention}(\mathbf{z}_i^{k,t} \odot (m_j - \sum_{l=j+1}^i m_l), p_j) \quad (7)$$

This operation ensures that objects from earlier mask orders, (*i.e.*, previous masks for ‘a mug cup’ and ‘a cupcake’ on top of ‘a wooden floor’ behind ‘a dish’ altogether) remain coherent while allowing natural adaptation where needed. The term  $m_j - \sum_{l=j+1}^i m_l$  controls which regions from previous mask orders should influence the current generation, minimally affecting the “mug cup” and “cupcake” even if we put the “dish” in front of them. Please note that for background prompt  $p_0$ : “wooden floor”, we inverse the whole union of masks and do disentangled cross attention with background prompt  $p_0$  as in Fig. 2 (a).

### 3.4.3. Final Latent Update

The final update combines information across all mask orders:

$$\mathbf{z}_i^{merge} = \mathbf{z}_i^{attn} \bigcup_{j=0}^{i-1} \mathbf{z}_j^{attn} \quad (8)$$

After a feedforward layer update and  $K$  transformer blocks, we apply BCG:

$$\mathbf{Z}_i^t = \mathbf{z}_i^{K,t} \odot m_i + \mathbf{Z}_{i-1}^t \odot (1 - m_i) \quad (9)$$

This approach enables objects to adapt naturally to mask orders—for example, placing “*a dog*” in front of “*Jeep*” and “*Lego man*” (Fig. 1), or maintaining ‘a cupcake’ and ‘a mug cup’ despite inserting “*a dish*” in front (Fig. 2)—while preserving scene coherence. MQD balances content preservation with the seamless integration of new elements.

## 3.5. Improving Editability

While BCG and MQD with layer-wise memory enhance multiple editing capabilities in the PixArt- $\alpha$  model, we further extend our framework to support object deletion, particularly for overlapped objects behind foreground elements. We achieve this through a novel application of our MQD and BCG components.

To delete an object added at editing step  $i - 1$ , we leverage the stored latent representations from both step  $i - 2$  and the current step  $i$ :  $\{\mathbf{Z}_{i-2}^t\}_{t=1}^T$  and  $\{\mathbf{Z}_i^t\}$ . Starting from an intermediate denoising step  $\tau$ , we blend these latents as:

$$\mathbf{Z}_{erase}^\tau = m_i \odot \mathbf{Z}_i^\tau + (1 - m_i) \odot \mathbf{Z}_{i-2}^\tau \quad (10)$$

Our deletion process operates in two phases: from step  $\tau$  to  $\frac{\tau}{2}$ , we perform latent blending following Eq. (10) with  $\mathbf{Z}_{i-2}^t$ . After step  $\frac{\tau}{2}$ , we switch to vanilla denoising to ensure a smooth transition. Crucially, we process  $\mathbf{Z}_{erase}$  through MQD without the  $i - 1$  step’s mask or prompt embedding, effectively removing the target object’s influence.

This approach offers two key advantages. First, by leveraging BCG to start from step  $\tau$  rather than  $T$ , we achieve 60% faster editing while preserving background consistency, as we use  $\tau = 8$  out of a total of 20 steps. Second, MQD’s application from step  $\tau$  maintains foreground object integrity while effectively removing background elements, eliminating the need for precise object masks in deletion.

Table 2. **Quantitative results on our proposed benchmark.** Our method shows higher semantic and visual alignment than other baselines, including layout-to-image synthesis and image editing frameworks. Ordering $\dagger$  denotes the adaptation of the framework adequate for iterative image generation.

Type	Method	Resolution	Semantic Alignment		Visual Alignment
			BLEU-2/3/4 $\uparrow$	METEOR $\uparrow$	CLIP <sub>crop</sub> $\uparrow$
2D Layout-to-Image	LayoutGuidance [14]	512 × 512	36.44 / 26.13 / 18.85	0.1361	62.92
	NoiseCollage [43]	512 × 512	55.75 / 42.43 / 32.96	0.1402	64.01
	NoiseCollage + ordering $\dagger$	512 × 512	59.98 / 43.76 / 32.24	0.1464	64.10
3D Layout-to-Image	LooseControl [10]	512 × 512	63.30 / 46.24 / 34.15	0.1373	63.13
Image Editing	BLD [3]	1024 × 1024	55.30 / 40.38 / 29.58	0.1480	62.40
	HD-Painter [32]	1024 × 1024	63.29 / 47.63 / 36.28	0.1484	64.09
	SD3-Inpaint [20]	1024 × 1024	29.90 / 21.64 / 15.78	0.1445	63.98
	Ours	512 × 512	61.19 / 45.04 / 34.06	0.1465	64.28
	Ours	1024 × 1024	<b>64.99 / 47.69 / 36.59</b>	<b>0.1513</b>	<b>64.29</b>

## 4. Multi-Edit Benchmark

Prior benchmarks for image editing [31, 45] are limited to evaluate only single-step editing or limited to instruct-based edit [22], and layout-guided image generation benchmarks are limited to evaluate whether the specific object is generated or not in the bounding box or evaluate only the flat spatial arrangements. [4, 17, 21]. Therefore, we aim to tackle the issue of evaluating the semantic alignment of the generated images by assessing whether the object generated is aligned with the caption that is used for sequential editing. Also, to handle the mask order, we propose to evaluate the generated image in a layer-wise manner aligning with mask orders. Specifically, to evaluate interactive generation and editing, we need to evaluate whether each layer shows the desired object within the given mask.

We generate the dataset by selecting classes from ImageNet-1K and arranging them in layered compositions with varying degrees of occlusion and spatial relationships, ensuring mask order is a key factor in the generation process as in the Fig. 3 (a). We utilize GPT-4 API to assist in selecting object classes (*e.g.*, ‘Lion’, ‘Tiger’, ‘Elephant’) that naturally fit together, creating realistic compositions, and generating template-based captions for both the global scene and individual object layers to integrate spatial relations into prompt and utilize GPT to make prompt richer.

To evaluate the spatial and semantic alignment between generated images and prompts, we crop each object layer using its corresponding mask and evaluate it individually as in Fig. 3 (b). Both cropped and final images are resized to 224x224 for consistency across evaluations. We employ LLaVa [29] to generate captions and compute alignment metrics such as BLEU [36], METEOR [6], ensuring a robust evaluation of both spatial arrangement and semantic accuracy. Note that we average the result from all the layers of each image to evaluate.

Also, to check each layer’s visual alignment of the gen-

erated object, we crop the image for each iteration and calculate the CLIP score following other layout-to-image models [4, 21, 43] as individual editing step is similar to layout-to-image generation. We discuss and provide detailed descriptions of the dataset creation process, evaluation metrics, and the resulting benchmark in the supplement.

## 5. Experiment

### 5.1. Implementation Details

**Baselines.** We evaluate our method against state-of-the-art baselines in both image editing and layout-to-image generation. For image editing, we compare with HD-Painter [32], Blended Latent Diffusion [3] with SD-XL [38], and SD3-ControlNet-Inpaint [20]. As sequential editing shares similarities with layout-to-image generation, we also include train-free layout models: NoiseCollage [43] and Layout-Guidance [14], and 3D-lifted model: LooseControl [10].

Additionally, we adapt NoiseCollage to handle sequential mask inputs (denoted as NoiseCollage + ordering) to explore the potential of converting layout-to-image models for sequential editing tasks. Following our train-free approach using the PixArt- $\alpha$  foundation model [13], all selected baselines operate without additional training.

**Implementation Details.** We leverage PixArt- $\alpha$  [13], the variant of Diffusion Transformer (DiT) for image editing framework. We utilize a  $1024 \times 1024$  pretrained model with DPM-Solver with DPMS guidance of 7.5 and a total denoising step of 20. Also, for a fair comparison with models that generate  $512 \times 512$  resolution [14, 43], we compare quantitatively by generating  $512 \times 512$  images with the  $512 \times 512$  pretrained PixArt- $\alpha$  model.

### 5.2. Quantitative Result

We present the quantitative evaluation results on our proposed Multi-Edit Benchmark in Tab. 2. Our method consistently outperforms both image editing and layout-to-image



Figure 4. Qualitative comparison on the effect of Query Disentanglement (QD).

baselines across key metrics. Among editing baselines, we demonstrate superior performance compared to Blended Latent Diffusion (BLD) [3] with SD-XL, HD-Painter [32], and SD3-ControlNet-Inpaint [20]. Notably, the significant performance gap with SD3-ControlNet-Inpaint can be attributed to the suboptimal property of utilizing ControlNet-Inpainting in multi-step editing.

Our framework shows particular strength in handling sequential editing challenges, demonstrated by consistent improvements across all metrics. This validates our approach’s effectiveness in maintaining background consistency while naturally integrating new objects - a key challenge in sequential editing tasks.

The performance gap grows further relative to 2D layout-to-image baselines, which often struggle with overlapping or cascaded placements. In particular, our method outperforms NoiseCollage by 4–5%p and surpasses Layout-Guidance by over 15%p in BLEU and outperforms in all other metrics. Against the 3D layout-to-image method that leverages pseudo depth maps, LooseControl [10] shows improved BLEU compared to the 2D baselines, but ours outperforms on all metrics. (See supplement for comparison with attribute editing on LooseControl.)

While we adapt NoiseCollage [43] for iterative image editing (+ ordering), NoiseCollage + ordering achieves better visual alignment scores, indicating the potential viability of adapted layout-to-image approaches. However, its limited improvement across semantic alignment metrics suggests that simple adaptations are insufficient for complex sequential editing scenarios. This underscores the need for specialized approaches like ours that can effectively handle the unique challenges of iterative image editing.

### 5.3. Ablation Study

We present an ablation of our key components (Tab. 3), starting from PixArt- $\alpha$  [13]. Background consistency guidance (BCG) reuses latents for faster editing (Tab. 1), improving BLEU and CLIP while preserving METEOR. Next, Query-Disentanglement (QD), inspired by NoiseCollage [43], preserves the existing background rather than fully covering the mask, raising BLEU but slightly reducing CLIP. Direct latent blending instead inflates CLIP by overfilling the mask, lacking context (Fig. 4). Finally, extend-

Table 3. Ablation study of our proposed method. Results are reported on both semantic and visual alignment metrics. We denote vanilla PixArt- $\alpha$  inpainting model as **Baseline**. Also we denote **QD** as Query-Disentanglement without memory, only with the current object and background.

Method	Semantic Align.		Visual Align.
	BLEU-2/3/4↑	METEOR↑	CLIP <sub>c</sub> ↑
Baseline	56.29 / 42.04 / 33.06	<b>0.1586</b>	64.05
+BCG	60.74 / 46.27 / 35.20	0.1585	64.10
+QD	62.68 / 46.42 / 35.03	0.1530	63.99
<b>Ours</b>	<b>64.99 / 47.69 / 36.59</b>	0.1513	<b>64.29</b>

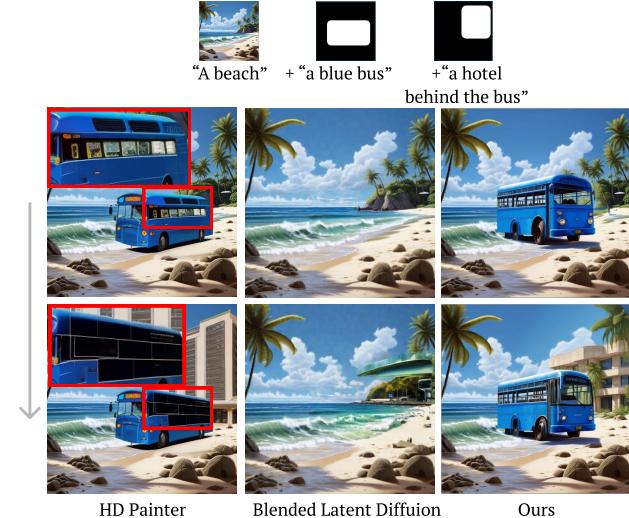


Figure 5. Comparison in image editing capability with latest image editing models. [3, 32] Note that the initial image is generated by our framework, which is equivalent to PixArt- $\alpha$  [13] with no mask input.

ing QD to multi-query disentanglement (MQD) with layer-wise memory integrates new objects into all prior prompts, achieving higher BLEU-2/3/4 and CLIP scores.

### 5.4. Qualitative Result

#### Comparison with Editing Approaches under Interactive Scenario.

In Fig. 5, we demonstrate our method’s effectiveness in iterative editing compared to state-of-the-art single-step editing approaches [2, 32]. Starting with a background of “A beach” generated by vanilla PixArt- $\alpha$  [13], we sequentially place a bus and a hotel. Our method successfully renders the hotel behind the bus, accurately interpreting both the prompt “A beach” and the *mask order* through MQD to achieve the intended spatial arrangement.

In contrast, Blended Latent Diffusion [3] struggles with the initial placement of “bus” in the complex background, while HD-Painter [32] shows inconsistencies in the appearance of “bus” (highlighted in red in Fig. 5). Our method maintains background consistency while naturally integrating new objects into the scene.

**Improved Editability.** Fig. 6 compares our method with



Figure 6. **Improved editability of image through Background Consistency Guidance and Multi-Query Disentangled cross attention.** Through recycling the previous step’s latents, we can remove the object that is behind the foreground object, enabling enhanced editability of the image.

Table 4. **Human preference study with recent literature on image editing on our benchmark.** We utilize HD-Painter [32] and Blended Latent Diffusion with SD-XL [38] to compare under iterative editing scenario and compare the preference on a 5-point Likert scale.

Method	Background Consistency	Natural Adaptation	Text-scene Alignment
HD-Painter [32]	3.71	2.81	3.08
BLD [3]	3.43	2.24	2.04
Ours	<b>4.59</b>	<b>4.28</b>	<b>4.49</b>

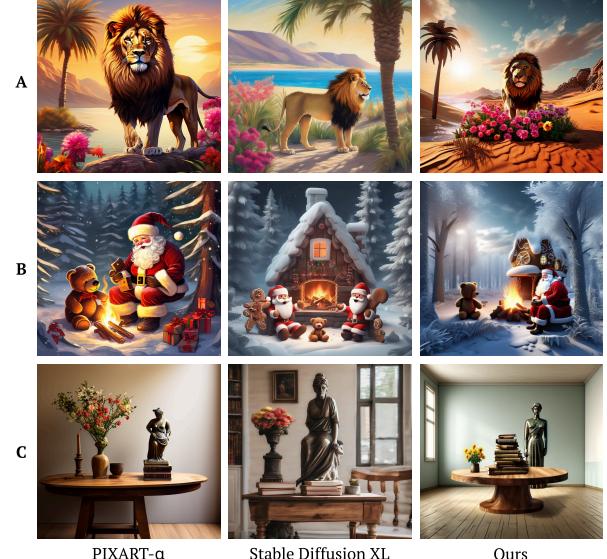
Photoshop 2025 [1]. Placing “*a glass of wine*” or “*a lizard*” is straightforward, but removing “*a glass of wine*” while preserving “*a lizard*” remains challenging for Photoshop’s generative fill, which requires precise brush strokes and often leaves artifacts (red box). Our approach simplifies this by leveraging existing masks and prior-step latents to naturally inpaint the background while applying MQD during partial denoising to preserve the lizard’s structure.

**Comparison with Existing T2I models.** Fig. 7 showcases our method’s ability to handle complex multi-object scenarios that challenge conventional text-to-image models. For instance, in row A, only our framework successfully generates the intended composition of a lion behind a flower. Row B and C further demonstrate the challenge for existing T2I approaches to handle complex spatial arrangements, while illustrating our method’s capability.

## 5.5. Human Preference Study

We conduct a comprehensive human evaluation study with 50 participants to assess the quality of our method compared to the latest editing models [3, 32] on our benchmark. The evaluation focuses on three aspects: background consistency, natural adaptation, and text-scene alignment, rated on a 5-point Likert scale.

Our method outperforms single-step editing approaches [3, 32] across all three categories. These results demonstrate that traditional editing models, while effective



A. A lion stands behind flowers in the desert, positioned next to the water where a palm tree is growing nearby under the bright sun.  
B. Santa Claus and a teddy bear are sitting by a campfire in a snowy forest, with a gingerbread house directly behind them under falling snow.  
C. A wooden table has a flower pot on the left and some stacked books in the center, with a small statue resting behind the table.

Figure 7. **Comparison in interactive scenarios with existing T2I generative models.** Stable Diffusion XL [38] and PixArt- $\alpha$  [13] use text input only.

for single-step modifications, are insufficient for sequential editing. This underscores the necessity of specialized approaches like ours for handling sequential iterative editing scenarios. Additional human evaluation results comparing our method with text-to-image models [13, 38] are provided in the supplementary materials.

## 6. Conclusion

We propose a novel framework for sequential image editing, which poses challenges in maintaining background consistency and seamless integration of new objects into the scenes. Our method includes three key components: *Layer-wise Memory* to preserve previous content, *Background Consistency Guidance* to keep the background stable with faster editing, and *Multi-Query Disentangled Cross-Attention* for natural object adaptation. Experiments on our proposed benchmark dataset show that our approach outperforms state-of-the-art methods in semantic and visual alignment, making it ideal for complex, iterative image editing scenarios. This framework paves the way for future work in improved interactive editing.

**Limitation and Future Work.** Since our approach utilizes image editing, generating multiple objects takes longer, depending on the number of edits. Also, utilizing layer-wise memory requires additional memory costs. We plan to make it more efficient for faster editing in the future.

**Acknowledgement.** This work was supported by IITP grant (RS-2021-II211343: AI Graduate School Program at Seoul National University (5%) and RS-2024-00509257: Global AI Frontier Lab (30%)) and NRF grant (No.RS-2024-00405857 (65%)) funded by the Korea government (MSIT).

## References

- [1] Adobe Inc. Adobe Photoshop. <https://www.adobe.com/products/photoshop.html>, 2024. Version 2025. 1, 2, 8, 17, 20
- [2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022. 2, 7, 16
- [3] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM TOG*, 2023. 1, 2, 4, 6, 7, 8, 11, 13, 17, 20
- [4] Eslam Mohamed Bakr, Pengzhan Sun, Xiaogian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *ICCV*, 2023. 2, 6, 14
- [5] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE TIP*, 2001. 2
- [6] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEValuation@ACL*, 2005. 6
- [7] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In *ICML*, 2023. 2
- [8] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 2009. 2
- [9] Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. Image inpainting. In *SIGGRAPH*, 2000. 2
- [10] Shariq Farooq Bhat, Niloy J. Mitra, and Peter Wonka. Loosecontrol: Lifting controlnet for generalized depth conditioning. In *SIGGRAPH*, 2024. 6, 7, 16
- [11] BlackForestLabs. Flux, 2023. Accessed: 2024-11-12. 1
- [12] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- $\sigma$ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv:2403.04692*, 2024. 1
- [13] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- $\alpha$ : Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024. 1, 3, 6, 7, 8, 13
- [14] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *WACV*, 2024. 2, 6
- [15] Ruidong Chen, Lanjun Wang, Weizhi Nie, Yongdong Zhang, and An-An Liu. Anyscene: Customized image synthesis with composited foreground. In *CVPR*, 2024. 2
- [16] Yifu Chen, Jingwen Chen, Yingwei Pan, Yehao Li, Ting Yao, Zheneng Chen, and Tao Mei. Improving text-guided object inpainting with semantic pre-inpainting. In *ECCV*, 2024. 2
- [17] Jaemin Cho, Linjie Li, Zhengyuan Yang, Zhe Gan, Lijuan Wang, and Mohit Bansal. Diagnostic benchmark and iterative inpainting for layout-guided image generation. In *CVPR Workshop*, 2024. 2, 6
- [18] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE TIP*, 2004. 2
- [19] Abdelrahman Eldesokey and Peter Wonka. Build-a-scene: Interactive 3d layout control for diffusion-based image generation. In *ICLR*, 2025. 16
- [20] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yan-nik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 1, 6, 7, 11
- [21] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. In *NeurIPS*, 2023. 2, 6, 14
- [22] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv:2405.04007*, 2024. 6
- [23] Biao Gong, Siteng Huang, Yutong Feng, Shiwei Zhang, Yuyuan Li, and Yu Liu. Check locate rectify: A training-free layout calibration system for text-to-image generation. In *CVPR*, 2024. 2
- [24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2
- [25] Runhui Huang, Kaixin Cai, Jianhua Han, Xiaodan Liang, Renjing Pei, Guansong Lu, Songcen Xu, Wei Zhang, and Hang Xu. Layerdiff: Exploring text-guided multi-layered composable image synthesis via layer-collaborative diffusion model. In *ECCV*, 2024. 2
- [26] Jonghyun Lee, Hansam Cho, Youngjoon Yoo, Seoung Bum Kim, and Yonghyun Jeong. Compose and conquer: Diffusion-based 3d depth aware composable image synthesis. In *ICLR*, 2024. 2, 17
- [27] Jaerin Lee, Daniel Sungho Jung, Kanggeon Lee, and Kyoung Mu Lee. StreamMultiDiffusion: real-time interactive generation with region-based semantic control. *arXiv:2403.09055*, 2024. 2
- [28] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 2023. 2, 13
- [29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 6

- [30] Zhengyao Lv, Yuxiang Wei, Wangmeng Zuo, and Kwan-Yee K Wong. Place: Adaptive layout-semantic fusion for semantic image synthesis. In *CVPR*, 2024. 2
- [31] Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. I2ebench: A comprehensive benchmark for instruction-based image editing. In *NeurIPS*, 2024. 2, 6, 14
- [32] Hayk Manukyan, Andranik Sargsyan, Barsegh Atanyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Hd-painter: High-resolution and prompt-faithful text-guided image inpainting with diffusion models. In *ICLR*, 2025. 1, 2, 6, 7, 8, 11, 16, 17, 20
- [33] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI*, 2024. 17
- [34] Alex 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. In *ICML*, 2022. 2
- [35] Yotam Nitzan, Zongze Wu, Richard Zhang, Eli Shechtman, Daniel Cohen-Or, Taesung Park, and Michaël Gharbi. Lazy diffusion transformer for interactive image editing. In *ECCV*, 2024. 2
- [36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6
- [37] Pincel. Pincel: Ai image editor, 2024. Accessed: 2024-11-15. 1, 2, 17, 20
- [38] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 6, 8, 11, 13
- [39] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In *ICLR*, 2025. 2
- [40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 14
- [41] Jiawei Ren, Mengmeng Xu, Jui-Chieh Wu, Ziwei Liu, Tao Xiang, and Antoine Toisoul. Move anything with layered scene diffusion. In *CVPR*, 2024. 2
- [42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2
- [43] Takahiro Shirakawa and Seiichi Uchida. Noisecollage: A layout-aware text-to-image diffusion model based on noise cropping and merging. In *CVPR*, 2024. 2, 6, 7
- [44] Jack Urbanek, Florian Bordes, Pietro Astolfi, Mary Williamson, Vasu Sharma, and Adriana Romero-Soriano. A picture is worth more than 77 text tokens: Evaluating clip-style models on dense captions. In *CVPR*, 2024. 15
- [45] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J. Fleet, Radu Soricu, Jason Baldridge, Mohammad Norouzi, Peter Anderson, and William Chan. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *CVPR*, 2023. 2, 6, 13, 14
- [46] Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *CVPR*, 2024. 2
- [47] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wenzian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *ICCV*, 2023. 2
- [48] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *CVPR*, 2023. 2
- [49] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, 2017. 2
- [50] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *CVPR*, 2020.
- [51] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019. 2
- [52] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. In *ECCV*, 2024. 15
- [53] Lisai Zhang, Qingcai Chen, Baotian Hu, and Shuang Jiang. Text-guided neural image inpainting. In *ACM MM*, 2020. 2
- [54] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 11, 17
- [55] Xinyang Zhang, Wentian Zhao, Xin Lu, and Jeff Chien. Text2layer: Layered image generation using latent diffusion model. *arXiv:2307.09781*, 2023. 2
- [56] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, 2019. 2
- [57] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. In *NeurIPS*, 2023. 17
- [58] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *CVPR*, 2023. 2

# Improving Editability in Image Generation with Layer-wise Memory

## Supplementary Material

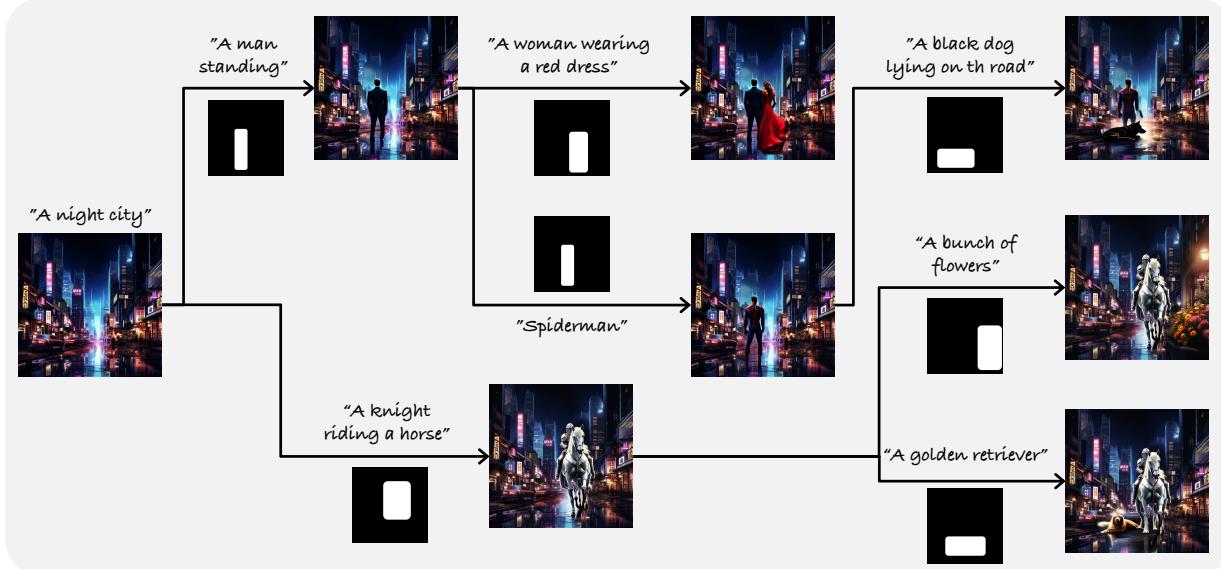


Figure 8. **Overview of interactive image generation under various scenarios.** Our approach can easily generate diverse images by editing in different ways.

## A. Implementation Details

We provide comprehensive implementation details of our framework and baseline methods used for comparison. This section covers the technical specifications of baseline implementations, our interactive editing process, and the detailed algorithmic workflow.

**Implementation Details of Baselines.** We compare our method against three recent image inpainting approaches: Blended Latent Diffusion (BLD) [3], HD-Painter [32] and Stable Diffusion 3 (SD3) [20]. For BLD, we utilize SD-XL [38] as the base model with a DDIM scheduler configured for 50 denoising steps.

For HD-Painter, we enhance the baseline by employing DreamShaper-v8 as the pretrained weight instead of the original SD 1.5 or 2.1, ensuring better output quality. To maintain consistent comparison, we match the resolution with our PixArt- $\alpha$  implementation using HD-Painter’s built-in upscaler. The framework operates with a DDIM scheduler over 50 denoising steps and employs classifier-free guidance of 7.5, adhering to the original configuration.

For SD3, we use ControlNet [54] Inpainting version of SD3. We use a guidance scale of 7.0 with a ControlNet scale of 0.95, with 28 inference steps, which is the original setting.

**Interactive Editing Process.** Our framework enables iterative image editing through a sequence of mask-guided modifications. Our framework processes each edit through three primary components: (1) Layer-wise Memory, (2) Background Consistency Guidance (BCG), and (3) Multi-Query Disentangled Cross-attention (MQD).

The Layer-wise Memory component maintains a comprehensive record of the editing history, storing latent representations, prompt embeddings, and mask information for each modification. This storage system enables retrieval of previous states while ensuring consistency across multiple edits. BCG leverages this stored information to maintain background integrity, implementing selective latent blending based on mask regions while minimizing the computational overhead of repetitive forward passes.

MQD handles the integration of new elements by processing edited regions and background content separately. This separation ensures the natural adaptation of new objects while preserving existing spatial relationships and background details, enabling the natural adaptation of diverse foreground objects into the background as presented in Fig. 8. “A man standing” or “A knight riding a horse” is naturally blended into “A night city”, and when a user adds “A woman wearing a red dress” or “A golden retriever”, a diverse result is achieved, meeting the user’s need.

---

**Algorithm 1:** Layer-wise Memory with Background Consistency Guidance (BCG) and Multi-query Disentangled Cross-Attention (MQD)

---

**Given:** Prompts  $P_i = \{p_0, p_1, \dots, p_N\}$ , Masks  $M_i = \{m_0, m_1, m_2, \dots, m_N\}$ , Pre-trained diffusion model  $f_\theta$ , Diffusion steps  $T$ , Number of DiT blocks  $K$

**Initialization:**

Initialize model parameters  $\theta$ ;  
 Generate background latent  $\mathbf{Z}_0 = f_\theta(p_0)$ ; # Generate background  
 Store  $\mathbf{Z}_0, p_0, m_0$  in memory; # Store initial background

**for**  $i = 1$  **to**  $N$  **do**

- Retrieve  $\mathbf{Z}_{i-1} = \{\mathbf{Z}_{i-1}^t\}_{t=0}^T, p_{i-1}, m_{i-1}$  from memory; # Recall previous latent
- Initialize Latent  $\mathbf{z}_i^{0,T} \sim \mathcal{N}(0, I)$ ;
- for**  $t = T$  **to**  $0$  **do** # Loop over diffusion steps
  - for**  $k = 1$  **to**  $K$  **do** # Perform MQD within each DiT block
    - $\mathbf{z}_i^{k,t} = \text{SelfAttention}(\mathbf{z}_i^{k-1,t})$ ;
    - $\mathbf{z}_i^{k,attn} = \text{CrossAttention}(\mathbf{z}_i^{k,t} \odot m_i, p_i)$ ;
    - for**  $j = i - 1$  **to**  $0$  **do** # MQD for current object
      - Retrieve  $p_j, m_j$  from memory; # Recall previous prompt embedding and mask
      - Update  $\mathbf{z}_i^{k,attn} = \text{CrossAttention}(\mathbf{z}_i^{k,t} \odot (m_j - \sum_{l=j+1}^i m_l), p_j)$ ; # MQD for previous objs
    - Merge  $\mathbf{z}_i^{merge} = \mathbf{z}_i^{attn} + \sum_{j=1}^{i-1} \mathbf{z}_j^{attn}$ ; # Merge attention results
    - $\mathbf{z}_i^{k,t} = \text{FeedForward}(\mathbf{z}_i^{merge})$
  - Update latent  $\mathbf{Z}_i^t = \mathbf{z}_i^K \odot m_i + \mathbf{Z}_{i-1}^t \odot (1 - m_{i-1})$ ; # Apply BCG
  - Store  $\mathbf{Z}_i^t$  in memory after final block for step  $t$ ; # Store final latent for each step
  - Store  $p_i, m_i$  in memory after denoising; # Store prompt embedding and mask

**Final Image Generation:**

Decode final latent  $\mathbf{Z}_N$  into  $\text{Image}_{\text{final}} = \text{Decoder}(\mathbf{Z}_N)$ ; # Decode final latent

**Return**  $\text{Image}_{\text{final}}$ ;

---

**Workflow Details.** Algorithm 1 presents our complete editing pipeline, which operates through four principal stages. The process starts with initialization, where we first generate a background latent  $\mathbf{Z}_0$  from the initial prompt  $p_0$  and store it in layer-wise memory. During iterative editing, we retrieve previous states ( $\mathbf{Z}_{i-1}, p_{i-1}, m_{i-1}$ ) and process them through  $T$  diffusion steps, applying  $K$  DiT blocks with MQD and BCG.

The cross-attention separately processes edited regions and background content, ensuring coherent integration of new elements while preserving existing content. BCG then blends to update the latents with *retrieved latents* and stores results in layer-wise memory, maintaining a complete edit history for future modifications. This proposed pipeline ensures robust background preservation while enabling natural object integration through the coordinated operation of our key components.

Our framework maintains editing to be coherent by leveraging MQD to disentangle cross-attention between edited regions, previously edited content and background, ensuring each modification integrates naturally with the existing scene while preserving intended spatial relationships. This approach enables seamless integration of new elements while maintaining the overall compositional integrity and spatial context of the image.

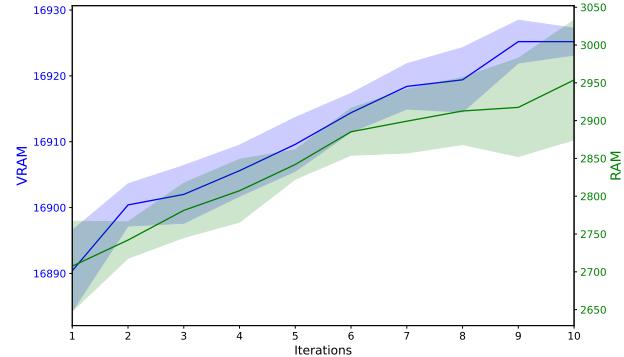


Figure 9. Analysis on computational resources for iterative editing.

## B. Analysis on Computational Overhead

Sequential editing multiple times, as in Figs. 8 and 10, can make the user achieve the intended images. However, multiple editing with layer-wise memory requires additional computational cost, and we analyze computational resource utilization during iterative editing processes. We create a new dataset for this analysis following a similar generation protocol as Multi-Edit Bench and perform 5 independent trials of sequential edits up to 10 iterations, measuring both memory consumption and processing overhead.

Fig. 9 illustrates the resource utilization patterns on a

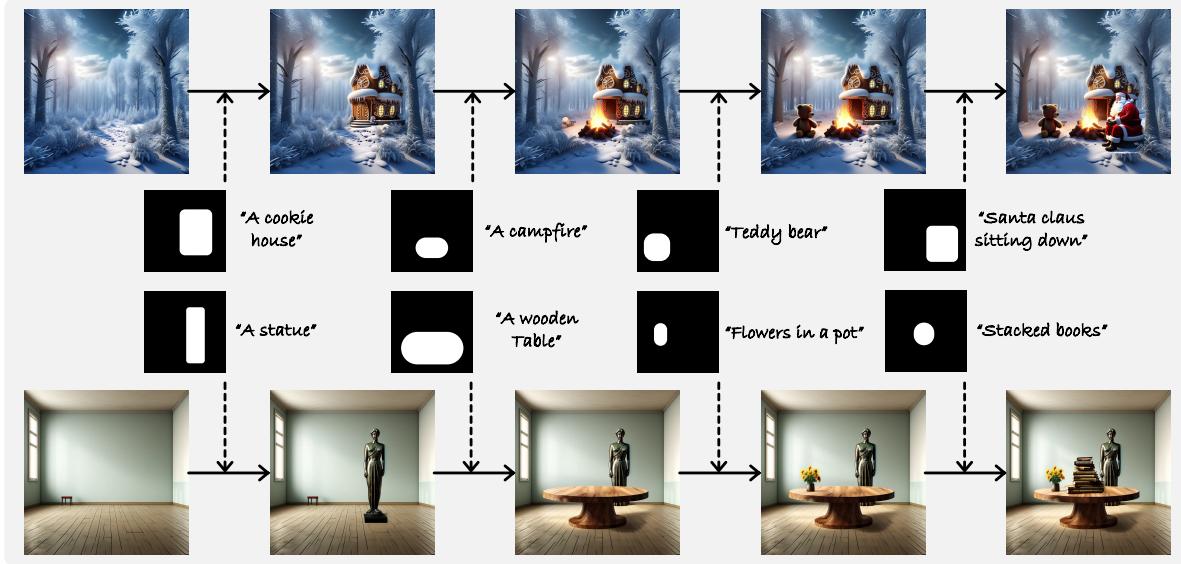


Figure 10. **Extensive multi-editing scenario.** Our framework enables sequential editing of multiple edits, more than just two or three times editing, meeting the user’s need to edit extensively on generated images.

Table 5. **Human preference study results with recent foundational models [13, 38].** We evaluate our model with SD-XL and PixArt-alpha on various prompts regarding spatial relationships on alignment and overall image quality.

Method	Spatial Alignment	Overall Quality
SD-XL [28]	3.16	3.66
PixArt-alpha [3]	2.76	3.43
Ours	3.57	3.47

single NVIDIA RTX-A6000 GPU. Due to our layer-wise memory architecture, we observe a predictable linear increase in RAM usage from 2,653MB to 2,954MB across 10 iterations, representing an 11% increment over the base memory footprint. This moderate increase is attributed to the storage of latent representations necessary for maintaining edit history and ensuring consistency across modifications. Notably, the VRAM consumption shows remarkable efficiency, increasing marginally from 16,882MB to 16,925MB - a mere 0.2% overhead over the initial usage.

## C. Perceptual Study

We additionally compare complex scenarios with recent Text-to-Image models [13, 38] and compare it in two aspects: (1) spatial alignment with the user’s intention and (2) overall quality. The result is shown in Tab. 5.

Our approach outperforms these latest models by more than 0.4 Likert scale in spatial alignment, showing the capability of synthesizing images while aligning well with a user’s intention through an interactive editing process. Furthermore, through multiple editing processes, we main-

tain the overall quality of the image (*i.e.*, natural blending), achieving competitive results with recent models with a score of 3.47. While this performs slightly lower than SD-XL, it shows improvement over the original PixArt-alpha model it builds upon.

## D. Dataset and Benchmark

In this section, we first showcase our result on other benchmarks for single-turn editing [45]. Afterward, we discuss the limitations of existing datasets and benchmarks, particularly in the context of interactive image generation and sequential image editing. We present details of the benchmark proposed in Sec.4 of the manuscript, which is designed to address these shortcomings by introducing scenarios tailored to evaluate spatial arrangement and semantic alignment in iterative editing tasks.

### D.1. Comparison on EditBench

We evaluate our framework on EditBench [45] to assess its performance on single-turn image editing scenarios. As shown in Tab. 6, our method achieves competitive results on EditBench’s metrics, demonstrating CLIP Text-to-Image scores and R-Precision (Prec.) comparable to state-of-the-art methods like Blended Latent Diffusion (BLD) with SD-XL and HD-Painter.

For CLIP Text-to-Image (T2I) score, ours outperforms all the baselines of Blended Latent Diffusion (BLD) with SD-XL, HD-Painter, and SD3-ControlNet-Inpaint. Also, ours outperforms Imagen-Editor [45] in CLIP T2I score, demonstrating the effectiveness. Especially, SD3-Inpaint showcases competitive results to ours in single-turn edits,

**Table 6. Comparison of latest works on single-turn editing.** We evaluate our model with Blended Latent Diffusion (BLD) with SD-XL and HD-Painter on single inpainting on EditBench. Following EditBench, we evaluate the CLIP Text-to-Image (T2I) score and CLIP R-Precision (Prec.). IM denotes Imagen-Editor proposed in EditBench. [45]

Training	Method	CLIP (T2I)	CLIP R-Prec.
O	IM	31.5	<b>98.6</b>
	BLD	29.84	70.83
	HD-Painter	31.44	87.50
	SD3-Inpaint	31.65	87.92
	Ours	<b>31.69</b>	90.42

but they show lower performance compared to BLD or HD-Painter in multiple edits demonstrated in **Sec. 5** of the main paper. Also, BLD and HD-Painter show lower performance on CLIP-Score in the result of **Sec. 5**. This demonstrates that traditional methods like BLD, HD-Painter, and SD-3-ControlNet-Inpaint are quite effective for single edits. However, they struggle with maintaining consistency across multiple editing steps as they lack mechanisms for preserving editing history and ensuring cross-edit coherence. This highlights a limitation of current benchmarks like EditBench that focus solely on single-turn editing.

## D.2. Limitations of Existing Datasets

Existing datasets and benchmarks in image editing [31, 45] or image synthesis [4, 21] often fail to evaluate the complex tasks involved in interactive image generation adequately. Most notably, they fail to assess how well-generated images align with specific prompts and spatial relationships in the editing or generation process. To summarize, prior works have the following limitations:

- **Lack of Interactive Generation Evaluation:** Current benchmarks do not provide an effective means to evaluate interactive generation scenarios where objects are introduced sequentially into a scene with precise control over spatial arrangements.
- **Lack of Semantic Alignment Evaluation:** Evaluating the semantic alignment between the generated image and the prompt is often reduced to general-purpose metrics such as the CLIP score or mean Average Precision (mAP) from object detection models [40]. These metrics are insufficient to measure how well the generated image aligns with the intended semantics of the prompt, especially in complex, layered scenarios.
- **Inadequacy for Mask Order-aware Arrangement Evaluation:** Existing datasets are not designed to assess spatial relationships and image ordering. They rarely focus on occlusions or specific arrangements of objects in depth-aware compositions, making it difficult to evaluate whether the edited image faithfully captures the intention.

Considering these limitations, a novel benchmark is required to evaluate both interactive generation and editing scenarios while ensuring strong alignment with the input prompts.

## D.3. Details of Proposed Benchmark

We introduce a new benchmark for evaluating sequential image generation and editing interactively, focusing specifically on the limitations mentioned above. This benchmark introduces novel evaluation metrics and scenarios that rigorously test the model’s ability to generate images aligned with spatial constraints (*i.e.*, mask for inpainting) and semantic intent.

**Design.** The proposed benchmark is crafted to assess the performance of models in generating images under an interactive generation scenario with sequential iterative editing. Specifically, this includes the following components:

- **Mask-ordered Prompts and Masks:** Each image generation task involves sequential prompts and corresponding masks that define the region of interest (RoI) for each object. This simulates an interactive generation process where objects are introduced layer by layer.
- **High Occlusion Ratio:** The benchmark is designed to test mask order-aware generation, ensuring that the scenarios involve significant object occlusion, a critical factor in realistic editing.
- **Complex Backgrounds and Detailed Object Arrangements:** Each scene includes a detailed and complex background, as well as intricate arrangements of objects, requiring the model to manage both background consistency and precise object placement effectively.

**Features.** To evaluate both the spatial alignment and overall visual quality of generated images, our benchmark includes the following features:

- **Evaluation of Spatial Alignment:** The benchmark introduces tasks that require the model to add objects in specified spatial arrangements while maintaining spatial relationships after editing.
- **Semantic & Visual-alignment Evaluation Metrics:** We propose several evaluation metrics that measure the alignment quality between the generated image and the intended prompt.

## D.4. Dataset Generation Process

We generate the benchmark through a semi-automated process that ensures diversity in composition while maintaining a high degree of control over spatial relationships and occlusions. The details for each step of the dataset generation process are as follows:

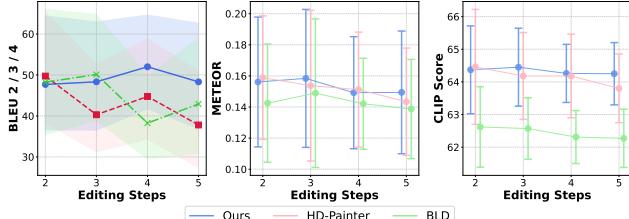


Figure 11. Comparison of BLEU, METEOR, and CLIP score on each step.

**Step 1: Decide on the number of layers (n):** Each image in the dataset consists of  $n$  layers, where  $n$  ranges between 3 and 6, including the background layer.

**Step 2: Select reference class from ImageNet-1K:** One object class is selected as the reference class from ImageNet-1K. This class serves as the anchor for the composition.

**Step 3: Select additional classes via GPT-4:** Using the GPT-4 API,  $n - 1$  additional classes are selected based on their natural compositional compatibility with the reference class. This ensures that the objects in the scene follow a coherent visual and semantic composition.

**Step 4: Generate random layouts (masks) for  $n$  classes:** For each of the  $n$  classes, random layouts are generated with constraints such as “margin from the edges” and the “size of mask”. These constraints ensure the objects are well-distributed without excessive overlap or clutter.

**Step 5: Generate template-based captions:** Based on the center coordinates of each object mask, template-based captions are generated to describe the spatial relationships and contents of the scene. These templates are used to generate global captions for the entire scene and for individual layers regarding the spatial relations.

**Step 6: Generate global and layer-wise captions:** The global caption is generated to describe the entire scene, while individual layer-wise (*i.e.*, editing steps) captions are generated for each object, ensuring that background details are excluded from the layer-wise descriptions with template-based captions through GPT API.

Through this approach, our dataset is designed to rigorously evaluate models’ performance: capabilities in handling interactive generation scenarios, spatial alignments, and semantic accuracy within complex, mask order-aware environments. As a result of this rigorous dataset construction, our benchmark evaluates editing performance across 2 to 5 steps, with distributions of 19% (2-step), 18% (3-step), 26% (4-step), and 37% (5-step), with average occlusion ratio of 18.53% across the layers.

## D.5. Evaluation Details

We evaluate each individual editing step of the edited image by cropping the generated image based on the masks



Figure 12. Qualitative comparison on LooseControl.

Table 7. Quantitative comparison on LooseControl.  $\dagger$  denotes Attribute Editing with cross-frame attention in LooseControl.

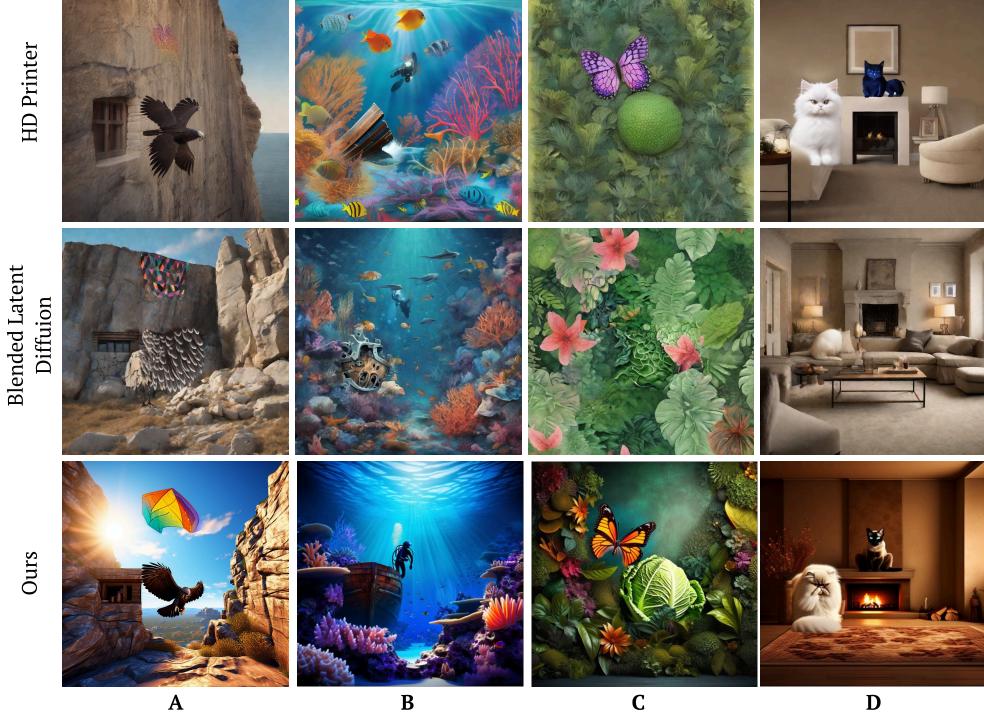
Method	Semantic Align		Visual Align
	BLEU-2/3/4↑	METEOR↑	CLIP <sub>crop</sub> ↑
LayoutGuidance	36.44 / 26.13 / 18.85	0.1361	62.92
NoiseCollage	55.75 / 42.43 / 32.96	0.1402	64.01
LooseControl	63.30 / 46.24 / 34.15	0.1373	63.13
LooseControl + Edit $\dagger$	58.74 / 45.00 / 34.76	0.1359	62.32
Ours (512 × 512)	61.19 / 45.04 / 34.06	0.1465	64.28
Ours (1024 × 1024)	<b>64.99 / 47.69 / 36.59</b>	<b>0.1513</b>	<b>64.29</b>

provided for each step. This method allows for fine-grained evaluation of how well each individual object was added following its corresponding prompt and spatial arrangement.

**Cropped Image Evaluation** All cropped images from the individual editing step’s evaluation are resized to a resolution of  $224 \times 224$  for evaluation. This uniform resolution ensures that variations in image size do not introduce inconsistencies in the evaluation results. The evaluation metrics used on the cropped images include the following metrics:

- CLIP Score:** We measure the similarity between each cropped image and its corresponding prompt. Since CLIP’s text encoder input is limited to 77 tokens and our prompt exceeds its length, CLIP score’s expressiveness can be constrained [44, 52]. Hence, we adopt the template “An image of {CLASS} in {BACKGROUND}” to describe the local cropped region within the token limit.
- LLaVa-based Alignment with BLEU and METEOR** For each cropped editing step’s image, LLaVa generates captions based on the bounding box of each object. The alignment between these captions and the intended prompt is measured using BLEU and METEOR scores, ensuring the model accurately captures the intended semantic information for all editing steps.

By evaluating each individual editing step, we ensure a comprehensive assessment of the model’s ability to edit holistically in a spatially aligned and semantically accurate manner.



- A. A large bird flies between cliffs with a kite overhead, and a cliff dwelling built into the rock.
- B. A scuba diver swims near a sunken boat on the ocean floor, surrounded by coral reefs and illuminated by sunlight streaming down from the surface.
- C. A butterfly hovers next to a large cabbage, surrounded by lush green foliage and flowers.
- D. A white cat sits on the floor in front of a fireplace, while a black cat on the mantel, in a cozy room.

Figure 13. **Comparison with other latest editing approaches [2, 32] with Multi-Edit Bench Dataset.** The approaches in the first two rows show results with baseline editing approaches. The background image is generated by our framework.

## D.6. Effect of Editing Steps

We conduct additional experiments on editing steps and present the results in Fig. 11 with BLEU, METEOR, and CLIP scores. Our method maintains stable performance as steps increase, whereas BLD and HD-Painter exhibit a continuous decline in CLIP and METEOR after three steps, along with consistently lower BLEU. Overall, our method remains steady across all metrics as editing steps increase.

## D.7. Comparison with 3D-lifted Work

We further compare our method with 3D-lifted approaches [10, 19]. Since Build-A-Scene [19] is unavailable, we evaluate against LooseControl [10] and its 3D-Editing approach in Tab. 7 and Fig. 12.

For fair evaluation, we lift 2D boxes to 3D using pseudo-depth maps and project them back for appropriate mask usage. LooseControl outperforms in BLEU at  $512 \times 512$  but lags in METEOR and CLIP, while our method surpasses across all metrics at  $1024 \times 1024$  resolution.

Additionally, we compare with LooseControl’s attribute editing. In multi-step editing, LooseControl consistently underperforms across all metrics compared to our method.

## E. Qualitative Results

We provide extensive qualitative results demonstrating our framework’s versatility in handling various image editing scenarios. Fig. 8 demonstrates the capability of interactive image generation under diverse scenarios. We also provide Fig. 10 to demonstrate the effectiveness of image synthesis under extensive multi-editing scenarios.

As we tackle the challenge of multiple editing, we showcase the qualitative comparison on our proposed Multi-Edit Bench in Sec. E.1. In addition, we present more qualitative result on advanced editing (*i.e.*, deleting the object behind the generated object in overlapped scenario.) under Sec. E.2. Furthermore, we show the application of our method in depth-order aware generation in Sec. E.3 by comparing with depth-aware approaches, denoting our model’s possibility in order-aware generation empowered by Background Consistency Guidance (BCG) and Multi-Query Disentangled cross-attention (MQD), maintaining the overlapped object’s shape and context even we add an additional object with high occlusion ratio.

### E.1. Comparison on Multi-Edit Bench

We present the result on Multi-Edit Benchmark dataset. Note that we utilized prompts inside our generated dataset. Due to the lack of space, we omit the prompt as a short sentence inside the qualitative result.

**Qualitative Comparison.** In Fig. 13, we present results on our Multi-Edit Bench, compared to other baselines of Blended Latent Diffusion (BLD) [3] and HD-Painter [32]. HD-Painter shows a quality image, but as seen in column A, ‘kite’ is not apparent in the image compared to the naturally blended kite in ours. For BLD, they fail to add objects in most examples, showing degraded image quality. In contrast, ours show images that align with the given masks in the dataset.

**Comparison under Real-world Interaction Scenarios.** As we proposed Multi-Edit Bench to evaluate sequential editing scenarios, we additionally compare with arbitrary cases, as this work focuses on interactive generation. We gave arbitrary prompts and masks to look out for more interactive editing scenarios. We designed prompts and masks arbitrarily but used the same prompts and masks for all the baseline models. We sampled 5 times and used the best-appearing sample for the qualitative comparison. We showcase the comparison in Fig. 14 and Fig. 15.

### E.2. Comparison on Improved Editing

We present a qualitative comparison under an improved editing scenario in Fig. 16. To achieve improved editability, we utilize our method to delete the object behind the foreground object (*i.e.*, previous mask order). Other methods, including commercial products [1, 37] and baselines [3, 32] show artifacts when removing the previously ordered object in the examples in Fig. 16. However, ours removes the previous object without any artifact, re-gaining the previous background through layer-wise memory. Also, we maintain the foreground object’s identity through MQD, describing our method’s efficacy.

### E.3. Comparison with Depth-aware Approaches

We additionally compare our method with depth-aware models in Fig. 17, as we can also generate order-aware images. ControlNet [54], T2I-Adapter [33], or Uni-ControlNet [57] show artifacts, but ours show results following the user’s intention, like the model which is trained from scratch [26].

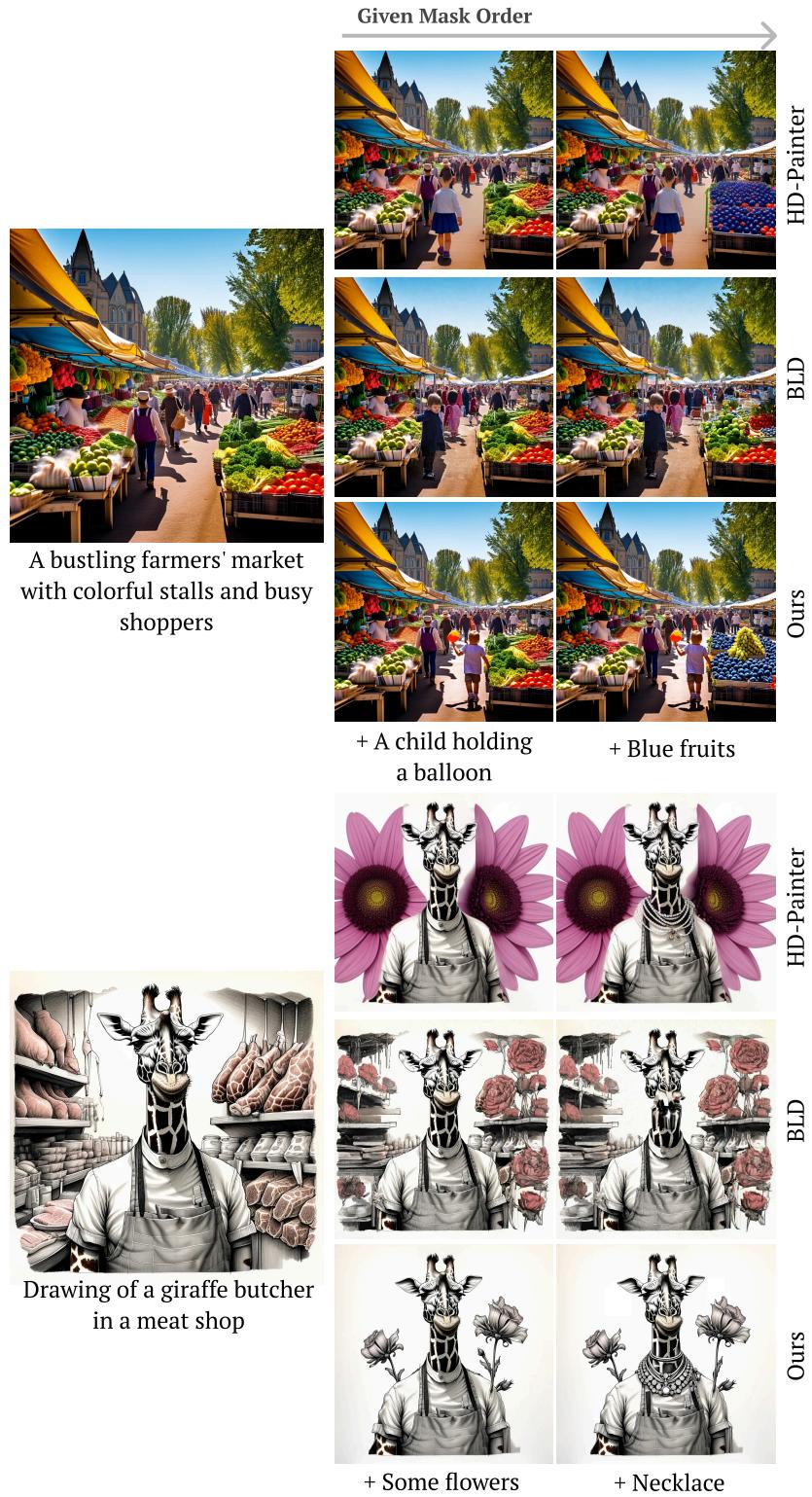


Figure 14. **Comparison with other latest editing approaches.** The approaches in the first two rows for each example show results with baseline editing approaches. The background image is generated by our framework.

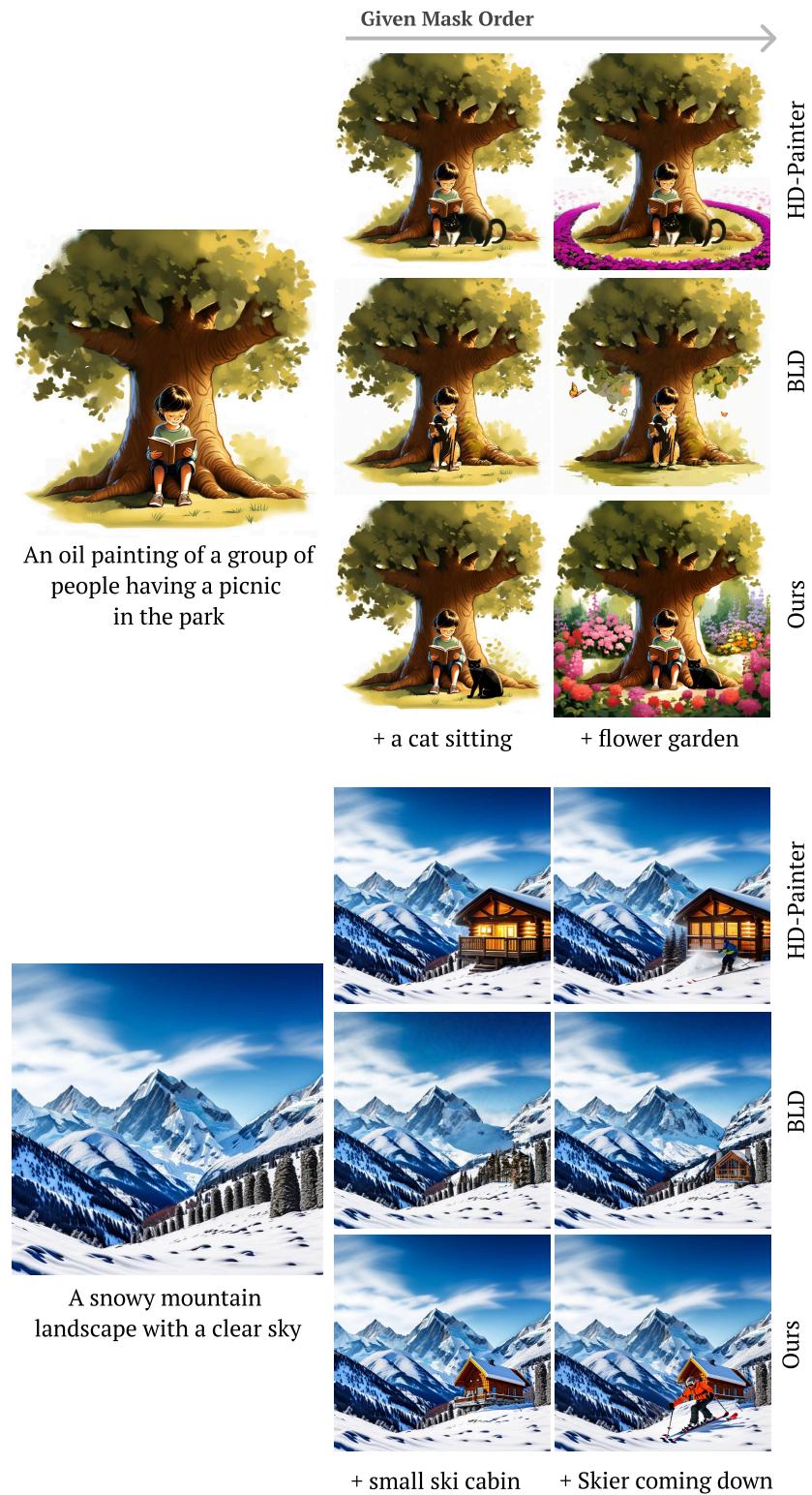


Figure 15. **Comparison with other latest editing approaches.** The approaches in the first two rows for each example show results with baseline editing approaches. The background image is generated by our framework.

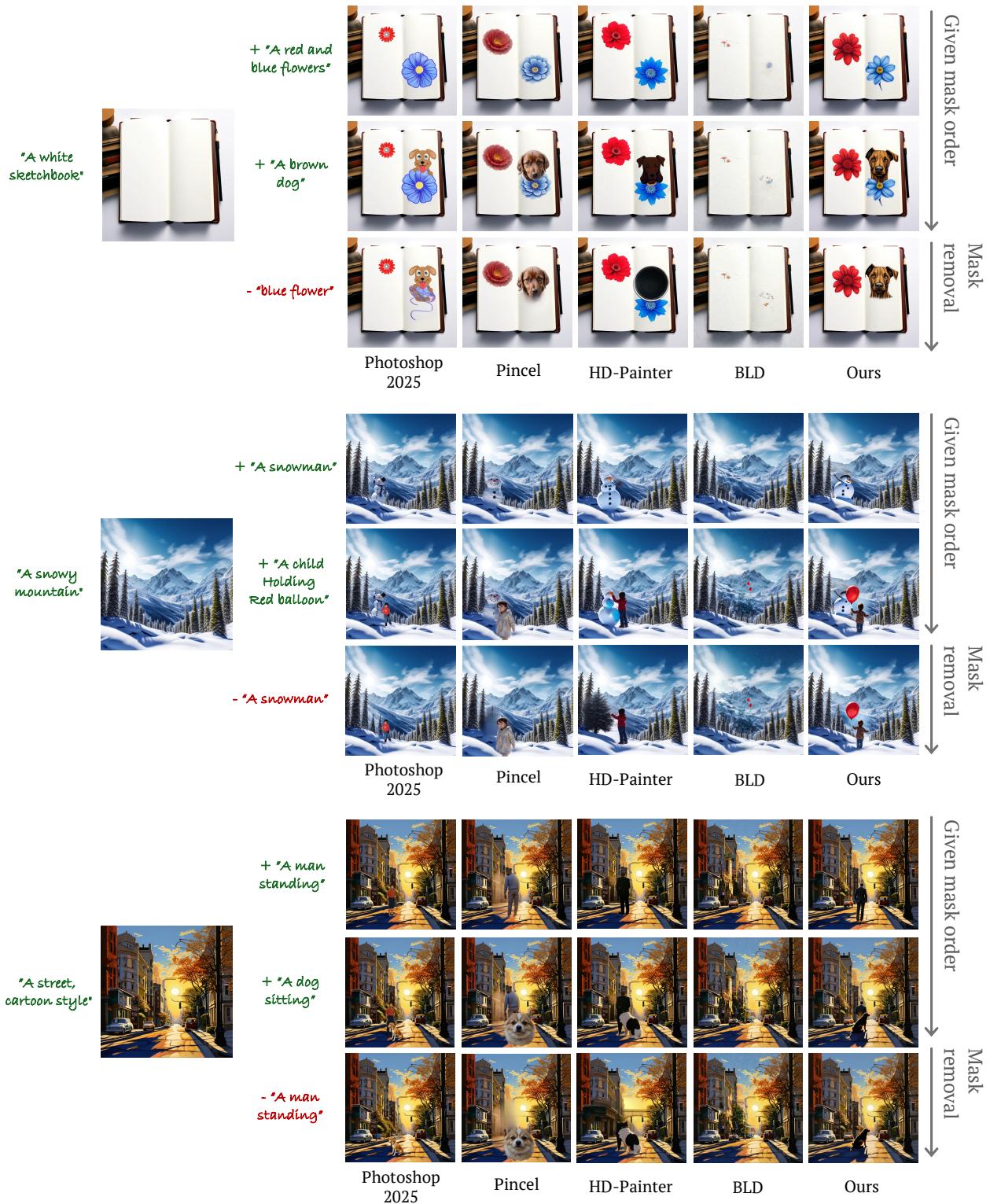


Figure 16. **Comparison under improved editing scenario.** Ours maintain the background well compared to other commercial products [1, 37] or baselines [3, 32].

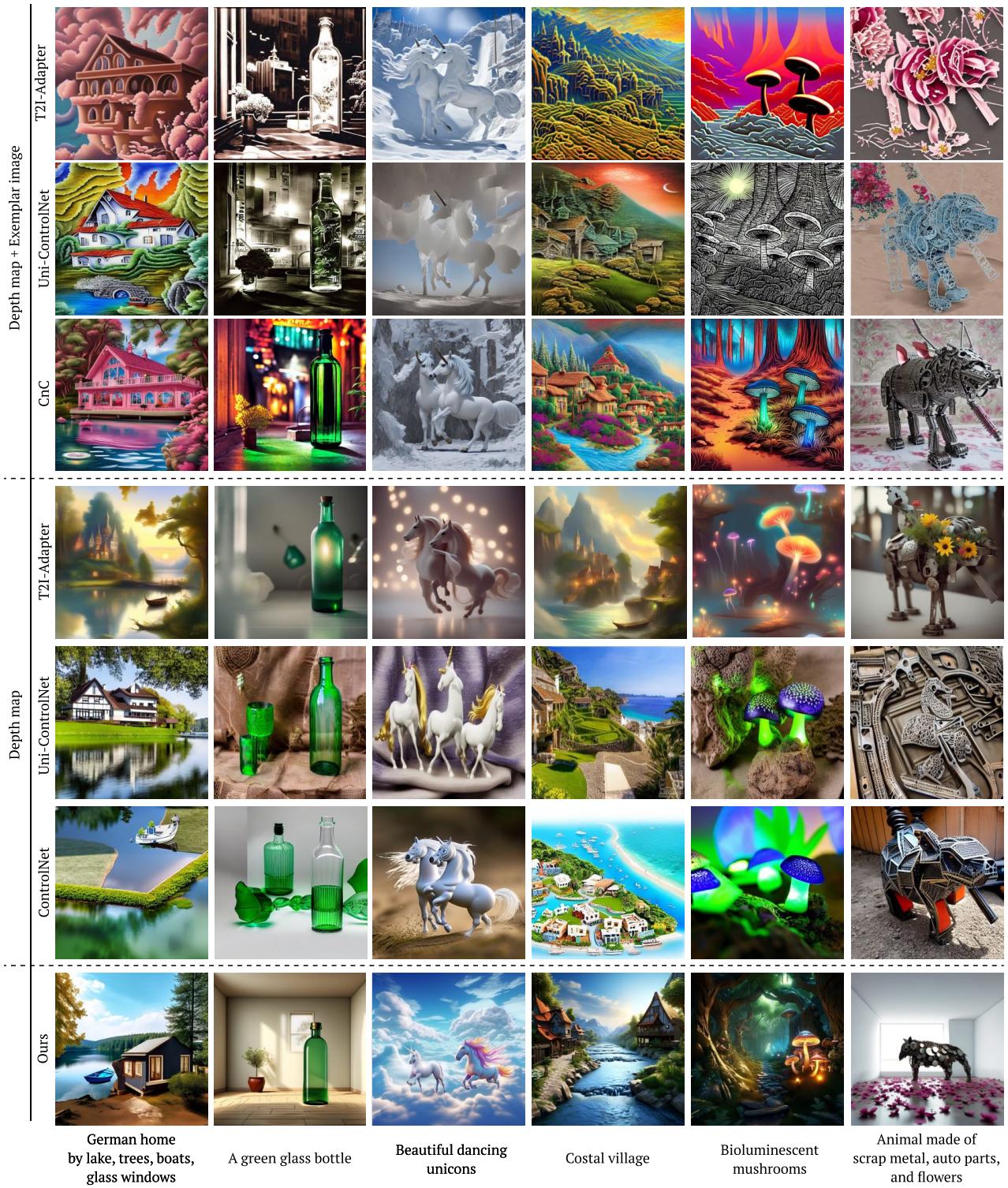


Figure 17. **Comparison with depth-aware text-to-image approaches.** The approaches in the first three rows utilize a depth map, exemplar image, and text prompt. The approaches in the next three rows get a depth map and text prompt. Our approach rivals the baseline approaches without using depth maps or exemplar images.