Improving Editability in Image Generation with Layer-wise Memory

Supplementary Material

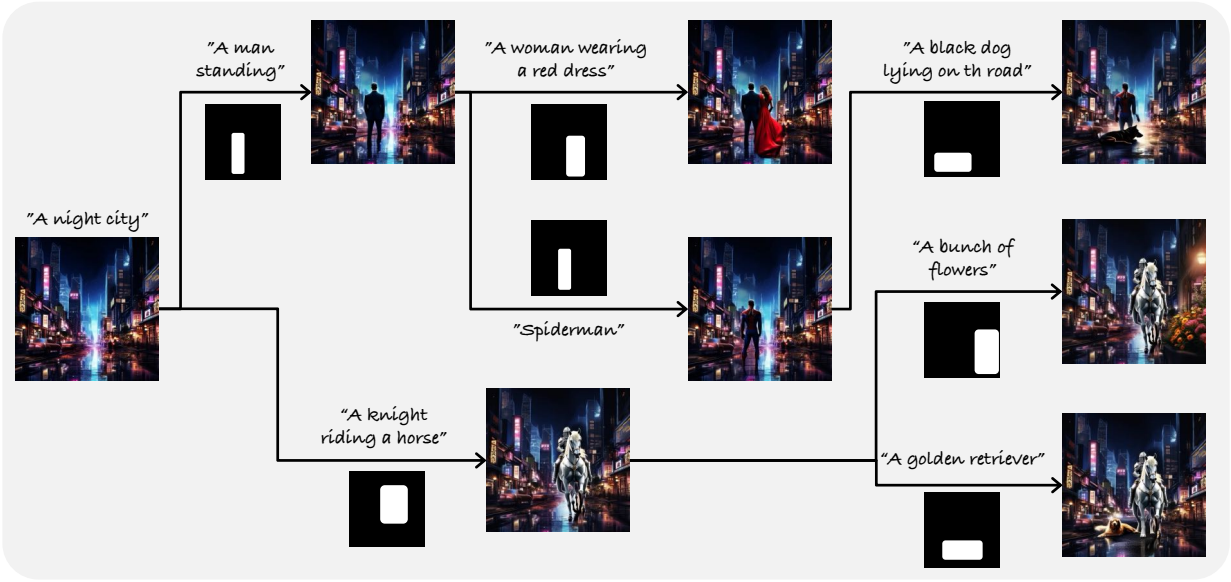


Figure 8. **Overview of interactive image generation under various scenarios.** Our approach can easily generate diverse images by editing in different ways.

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-α 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 $P_l = \{p_0, p_1, \ldots, p_N\}$, Masks $M_l = \{m_0, m_1, m_2, \ldots, m_N\}$, Pre-trained diffusion model f_θ, Diffusion steps T, Number of DiT blocks K

Initialization:

Initialize model parameters θ;

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

      $\mathbf{z}_i^{k,t} = \text{SelfAttention}(\mathbf{z}_i^{k-1,t})$;

      $\mathbf{z}_i^{k,attn} = \text{CrossAttention}(\mathbf{z}_i^{k,t} \odot m_i, p_i)$; # MQD for current object

      for $j = i - 1$ ***to*** 0 **do**

        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

      $\mathbf{z}_i^{k,t} = \text{FeedForward}(\mathbf{z}_\mathbf{i}^{\mathbf{merge}})$

    Update latent $\mathbf{Z}_\mathbf{i}^\mathbf{t} = \mathbf{z}_i^K \odot m_i + \mathbf{Z}_{i-1}^t \odot (1 - m_{i-1})$; # 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.

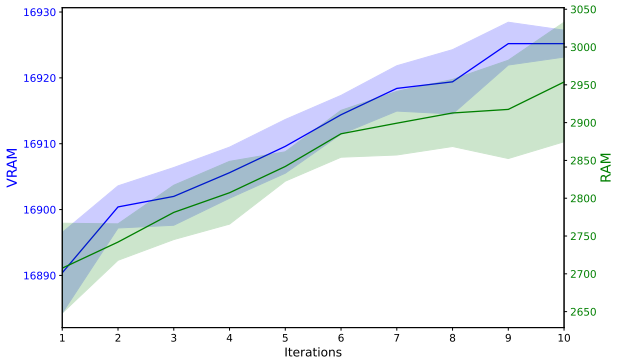


Figure 9. **Analysis on computational resources for iterative editing.**

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

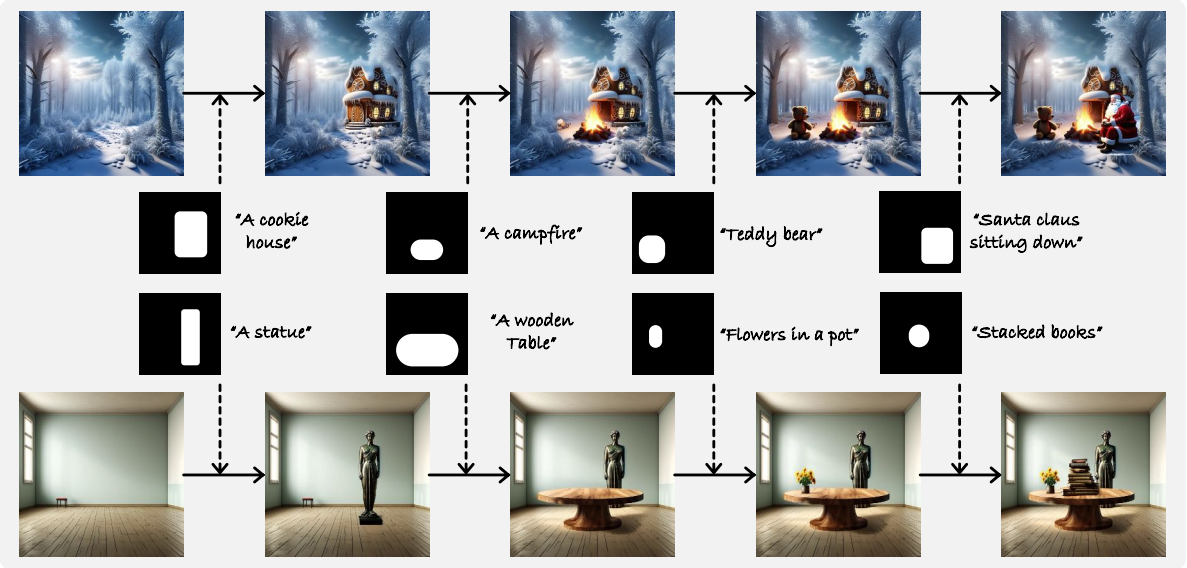


Figure 10. **Extensive multi-editing scenario.** Our framework enables sequential editing of multiple edits, more than just two or three times editing, meeting the user's need to edit extensively on generated images.

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 maintain 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	**98.6**
X	BLD	29.84	70.83
	HD-Painter	31.44	87.50
	SD3-Inpaint	31.65	87.92
	Ours	**31.69**	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:

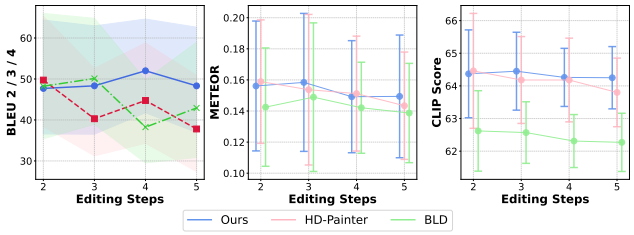


Figure 11. **Comparison of BLEU, METOER, and CLIP score on each step.**

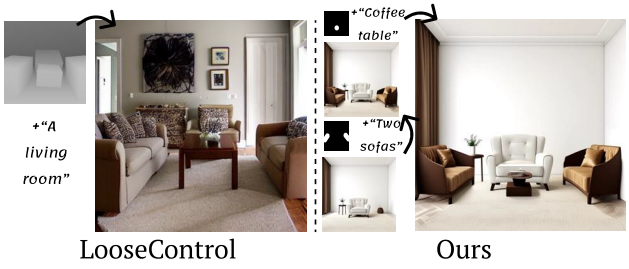


Figure 12. **Qualitative comparison on LooseControl.**

Table 7. **Quantitative comparison on LooseControl.** † denotes Attribute Editing with cross-frame attention in LooseControl.

Method	Semantic Align		Visual Align
	BLEU-2/3/4↑	METEOR↑	$\text{CLIP}_{\text{crop}}$↑
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†	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)	**64.99 / 47.69 / 36.59**	**0.1513**	**64.29**

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.

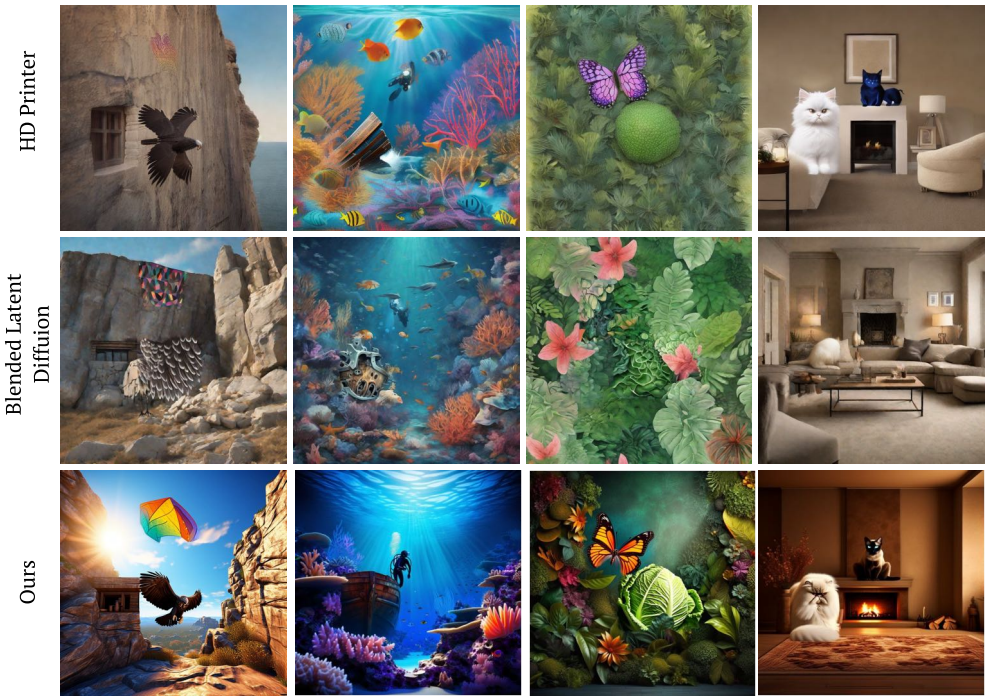
D.5. Evaluation Details

We evaluate each individual editing step of the edited image by cropping the generated image based on the masks 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 224×224 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 512×512 but lags in METEOR and CLIP, while our method surpasses across all metrics at 1024×1024 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].

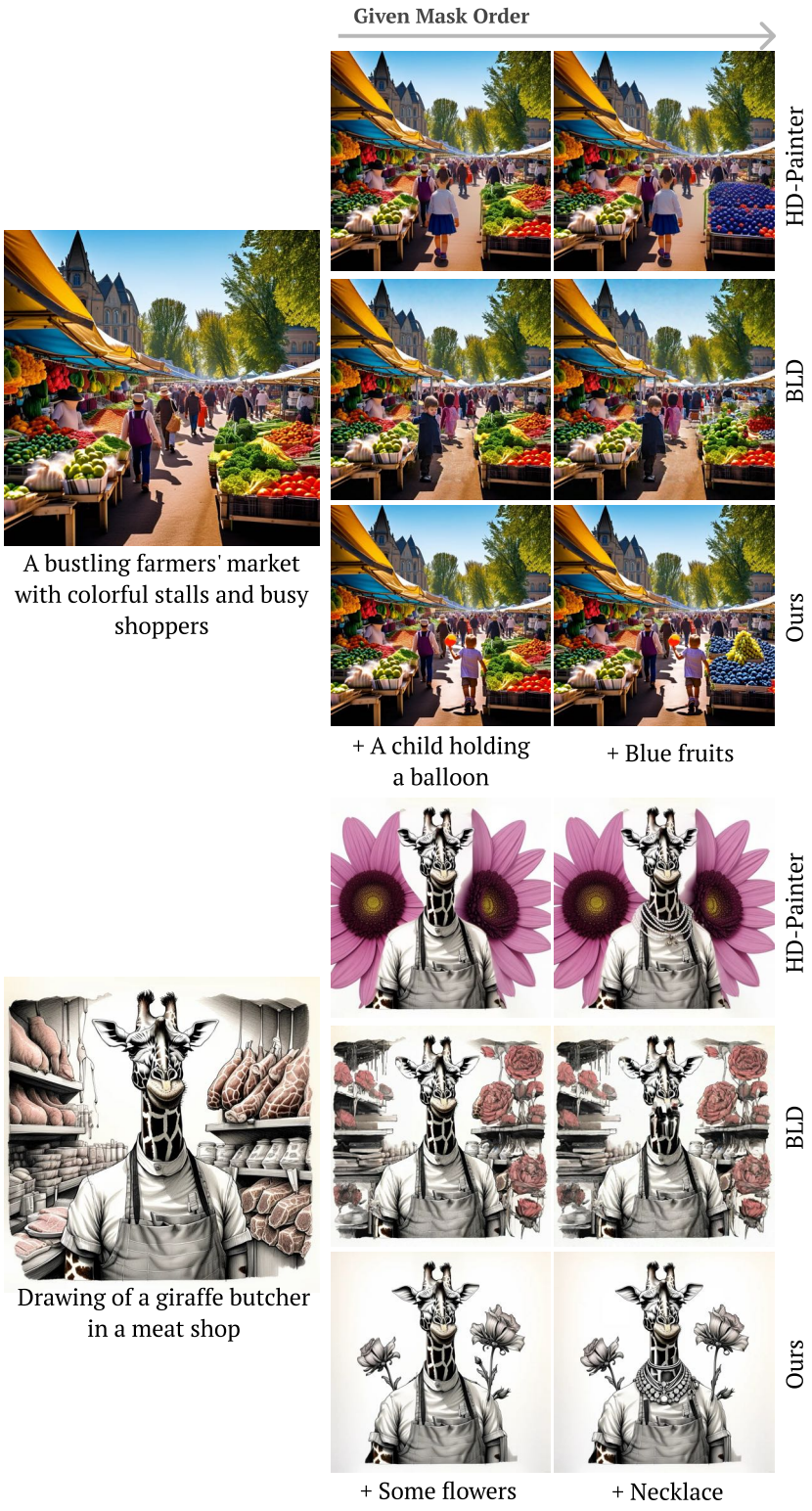


Figure 14. **Comparison with other latest editing approaches.** The approaches in the first two rows for each example show results with baseline editing approaches. The background image is generated by our framework.

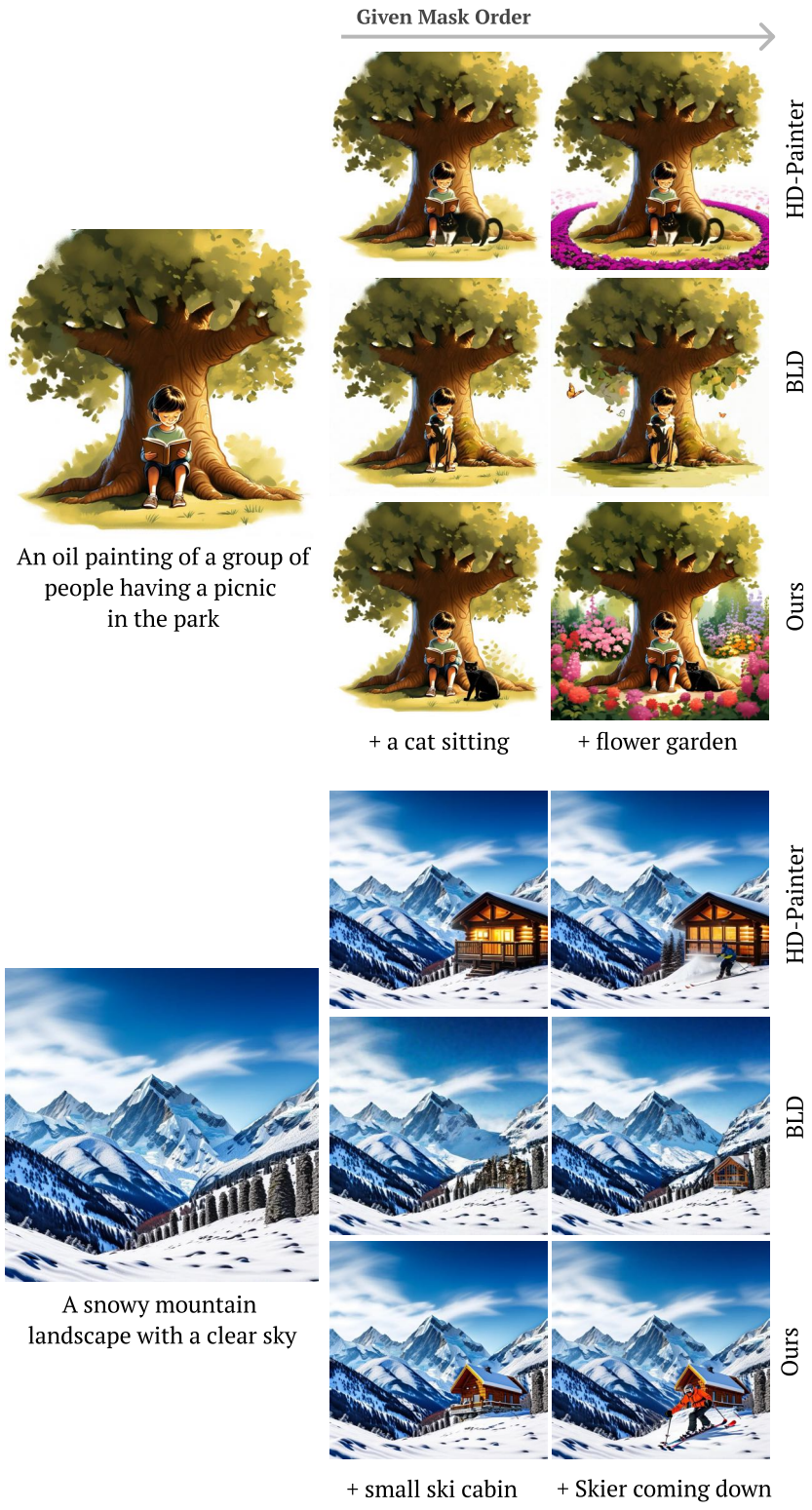


Figure 15. **Comparison with other latest editing approaches.**The approaches in the first two rows for each example show results with baseline editing approaches. The background image is generated by our framework.

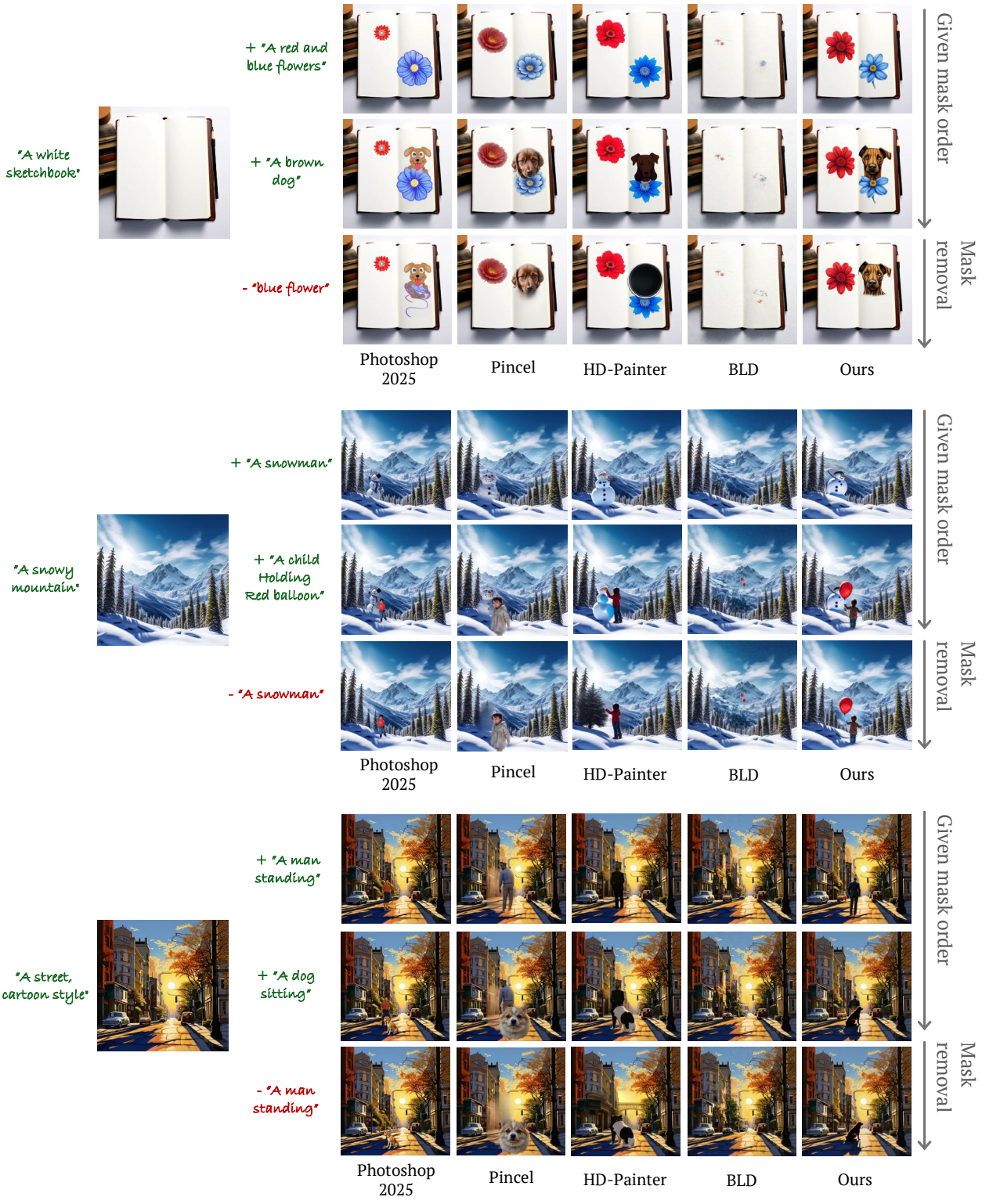


Figure 16. **Comparison under improved editing scenario.** Ours maintain the background well compared to other commercial products [1, 37] or baselines [3, 32].

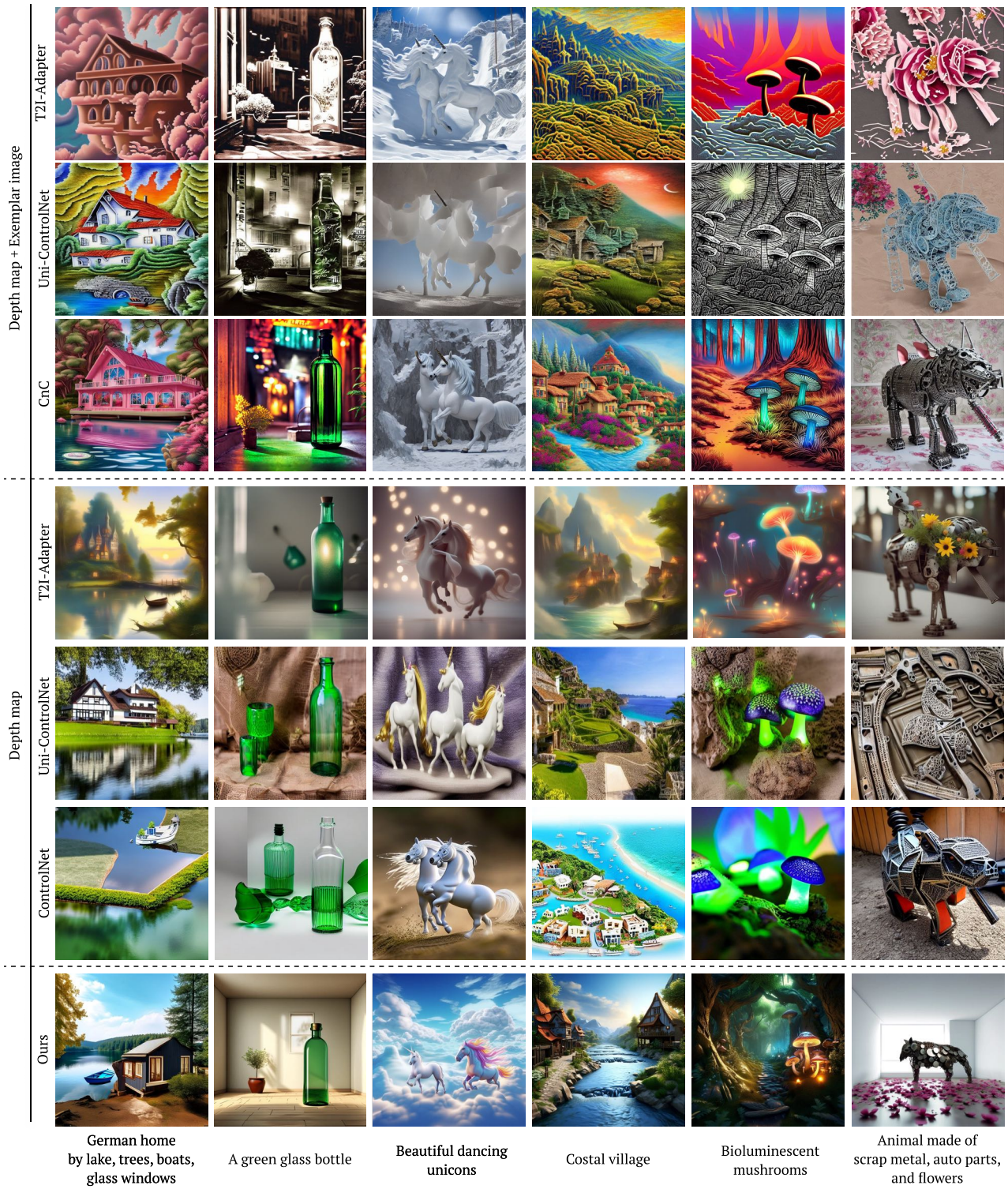


Figure 17. **Comparison with depth-aware text-to-image approaches.** The approaches in the first three rows utilize a depth map, exemplar image, and text prompt. The approaches in the next three rows get a depth map and text prompt. Our approach rivals the baseline approaches without using depth maps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