

Click2Mask: Local Editing with Dynamic Mask Generation

Omer Regev, Omri Avrahami, Dani Lischinski

The Hebrew University of Jerusalem

Abstract

Recent advancements in generative models have revolutionized image generation and editing, making these tasks accessible to non-experts. This paper focuses on local image editing, particularly the task of adding new content to a loosely specified area. Existing methods often require a precise mask or a detailed description of the location, which can be cumbersome and prone to errors. We propose Click2Mask, a novel approach that simplifies the local editing process by requiring only a single point of reference (in addition to the content description). A mask is dynamically grown around this point during a Blended Latent Diffusion (BLD) process, guided by a masked CLIP-based semantic loss. Click2Mask surpasses the limitations of segmentation-based and fine-tuning dependent methods, offering a more user-friendly and contextually accurate solution. Our experiments demonstrate that Click2Mask not only minimizes user effort but also delivers competitive or superior local image manipulation results compared to SoTA methods, according to both human judgement and automatic metrics. Key contributions include the simplification of user input, the ability to freely add objects unconstrained by existing segments, and the integration potential of our dynamic mask approach within other editing methods.

1 Introduction

Recent advances in generative models have revolutionized image generation and editing capabilities, enabling both streamlined workflows and accessibility for non-experts. The latest approaches utilize natural language to manipulate images either globally – altering the content or style of the entire image – or locally – adding, removing, or modifying specific objects within a limited image region.

In this work, we focus on local editing, specifically on the task of adding new content in a local area. To accomplish such edits, some existing methods require users to provide explicit precise masks (Avrahami, Lischinski, and Fried 2022; Ramesh et al. 2022; Avrahami, Fried, and Lischinski 2023; Wang et al. 2023b; Xie et al. 2022), which is tedious and may yield unexpected results due to lack of mask precision. Other methods describe the desired manipulations in natural language, as an edit instruction (Brooks, Holynski,

Project page: <https://omeregev.github.io/click2mask>



Figure 1: **Comparisons to SoTA models.** A comparison of Emu Edit (Sheynin et al. 2023), MagicBrush (Zhang et al. 2023) and DALL-E 3 (Betker et al. 2023) with our model **Click2Mask**. In each example, the top prompt was given to the other models, while Click2Mask received the simpler bottom prompt, in addition to the blue dot (mouse click) on the input. Other models completely change the image, or the background, fail to edit, or produce unrealistic results.

and Efros 2023; Sheynin et al. 2023), or by providing a caption and the desired change (Bar-Tal et al. 2022; Kawar et al. 2023; Hertz et al. 2022; Tumanyan et al. 2022). These methods also require user expertise, and their results may suffer from ambiguous or imprecise prompts. Moreover, they fail to ensure that the changes to the image are confined to a local area, or that they occur at all, as demonstrated in Figure 1.

To overcome the aforementioned shortcomings, we introduce Click2Mask, a novel approach that simplifies user interaction by requiring only a single point of reference rather than a detailed mask or a description of the target area. The provided point gives rise to a mask that dynamically evolves through a Blended Latent Diffusion (BLD) process (Avrahami, Lischinski, and Fried 2022; Avrahami, Fried, and Lischinski 2023), where the evolution is guided by a semantic loss based on Alpha-CLIP (Sun et al. 2023). This process (Figure 2) enables local edits that are both precise and contextually relevant (Figures 1 and 3).

Unlike segmentation-based methods that depend on pre-existing objects (Couairon et al. 2022; Xie et al. 2023; Wang et al. 2023a; Zou et al. 2024), Click2Mask does not confine

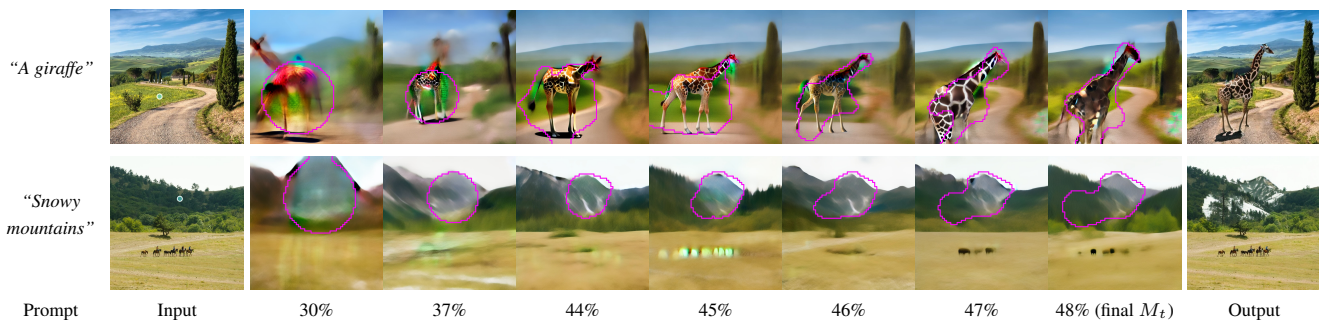


Figure 2: **Mask evolution.** A visualization of the mask evolution throughout the diffusion process. Leftmost image is input with clicked point, rightmost image is the final Click2Mask output. Intermediate images are decoded latents \tilde{z}_{fg} at several diffusion steps, where the purple outline depicts the contour of current (upscaled) mask M_t . Percentages indicate the step out of 100 diffusion steps, with the last being the final evolved mask.

the edit area to the boundaries of an existing segment. Furthermore, in contrast to editing approaches that require fine-tuning the diffusion model (Wang et al. 2023b; Xie et al. 2022; Kawar et al. 2023; Avrahami et al. 2023), we employ pre-trained models, and only perform context dependent optimization on the mask.

Our experiments demonstrate that Click2Mask not only reduces the effort required by users but also achieves competitive or superior results compared to state-of-the-art methods in local image manipulation.

In summary, our contributions are: (i) Reduction of user effort by eliminating the need for precise mask outlines, or overly descriptive prompts. (ii) Ability to add objects in a free-form manner, unconstrained by boundaries of existing objects or segments. (iii) Our dynamically evolving mask approach is not a stand-alone method, but rather it can be embedded as a mask generation of the fine-tuning step within other methods that internally employ a mask, such as Emu Edit (Sheynin et al. 2023) which currently generates multiple masks (a precise mask using DINO (Caron et al. 2021) and SAM (Kirillov et al. 2023), an expanded version of it, and a bounding box), and filters the best result from multiple images produced using these masks.

2 Related Work

In recent years, much work has been done on image generation, with diffusion-based models (DMs) (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Dhariwal and Nichol 2021; Rombach et al. 2022; Ramesh et al. 2022; Saharia et al. 2022) facilitating a host of SoTA text-guided image editing methods and capabilities.

Mask-based approaches. Text-guided image manipulation may naturally be limited to a specific region using a mask. In the context of DMs this was first explored in Blended Diffusion (Avrahami, Lischinski, and Fried 2022), where a user-provided mask is used to blend images throughout a denoising process with a text-guided noisy image. This approach was later incorporated into Latent Diffusion (Rombach et al. 2022) by performing the blending in latent space. The resulting Blended Latent Diffusion (BLD) method (Avrahami, Fried, and Lischinski 2023)

serves as the basis for our work and described in more detail in Section 3. GLIDE (Nichol et al. 2022), Imagen Editor (Wang et al. 2023b) and SmartBrush (Xie et al. 2022) fine-tuned the DM for image inpainting, by obscured training images or by conditioning on a mask. However, user-provided masks have a major disadvantage: the success of the edit depends on the exact shape of the mask, which can be tedious and time-consuming for a user to create.

Mask-free approaches. Both Text2Live (Bar-Tal et al. 2022), which generates a composite layer, and Imagic (Kawar et al. 2023), which interpolates target text and optimized source embeddings, fine-tune the generative model for each image, which is quite costly, contrary to our work. Several works use attention injection, such as Plug-and-Play (Tumanyan et al. 2022) and Prompt-to-Prompt (Hertz et al. 2022), where the latter requires a time-consuming caption of the input image, unlike our method. Most of these methods focus on altering a certain object (by replacement, removal or style change), or applying global changes (style or content), in contrast to our focus on adding objects freely.

Instruction-based approaches. Other methods can add objects in a free manner. InstructPix2Pix (Brooks, Holynski, and Efros 2023) (subsequently fine-tuned by MagicBrush (Zhang et al. 2023)) produces (instruction, image) pairs, used to train an instruction-conditioned DM. Emu Edit (Sheynin et al. 2023) is a more recent model trained on a wide range of learned task embeddings to enable instruction-based image editing, however, it is not publicly available. DALL-E3 (Betker et al. 2023) is also proprietary, and modifies the entire image as demonstrated in Figure 1. DALL-E3 and DALL-E 2 (Ramesh et al. 2022) apparently support masked inpainting, but we are unaware of a publicly available way to apply it to general real images. MGIE (Fu et al. 2024) train a DM, utilizing a MLLM to derive expressive instructions. These methods require the user to specify the desired localization in words, which has a few shortcomings. On the user’s side, this requires effort, and it can be difficult or impossible to describe the precise location. From the model’s side, failure to visually ground the text-specified location may fail to perform the desired edit, and/or make unintended changes in other locations instead.

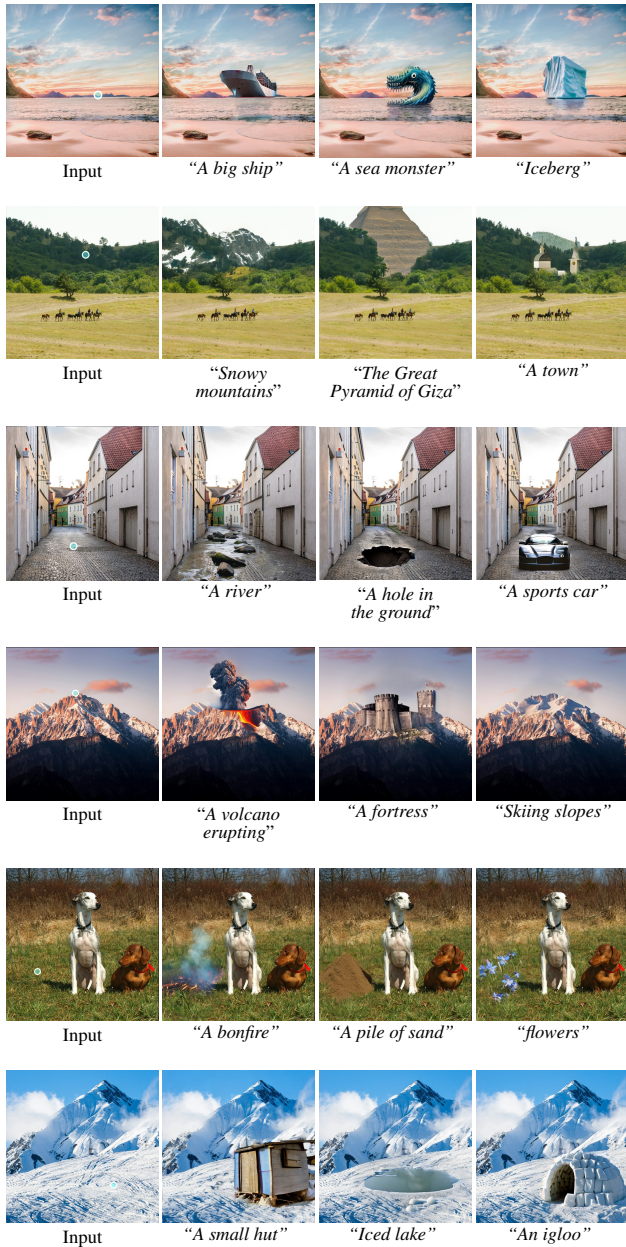


Figure 3: **Examples of Click2Mask outputs.** The leftmost column is the input image with clicked point. The other columns are **Click2Mask** outputs given the prompts below.

Segmentation-based approaches. Segmentation methods have been utilized to overcome the need for a precise user-provided mask. DiffEdit (Couairon et al. 2022) and Edit Everything (Xie et al. 2023) generate segmentation-based masks by utilizing conditionings on diffusion steps, or SAM (Kirillov et al. 2023), but require an input image caption, which is painstaking. InstructEdit (Wang et al. 2023a), which uses Grounding DINO (Liu et al. 2023) and SAM to generate a mask, does not require one, but requires a description of the object to alter. This can cause errors due to failure of the model to localize. InstDiffEdit (Zou et al. 2024) gen-

erates masks based on attention maps during denoising.

The segmentation-based methods, however, suffer from a few limitations: (i) Such models need to “lock” on an existing object or segment; consequently, in most cases they alter objects, but do not add new free-form ones, which is our focus. (ii) These methods typically require the user to provide an input caption or a description of the altered object.

In contrast to all the above, our work enables *adding* objects to real images (as opposed to merely altering existing ones), without having to provide a precise mask, to describe the input image, or target image, and without being constrained to boundaries of existing objects or segments. We aim to enable edits where the manipulated area is not well-defined in advance, and a free-form alteration is required.

3 Blended Latent Diffusion

Blended Latent Diffusion (BLD) (Avrahami, Fried, and Lischinski 2023) is a method for local text-guided image manipulation, based on Latent Diffusion Models (LDMs) (Rombach et al. 2022) and Blended Diffusion (Avrahami, Lischinski, and Fried 2022). Given a source image x , a guiding text prompt p , and a binary mask m , the model blends the source latents (obtained by DDIM inversion (Song, Meng, and Ermon 2020)) with the prompt-guided latents throughout the LDM process, to derive a blended final output.

Initially, inputs are converted to a latent space. A variational auto-encoder (Kingma and Welling 2013) with encoder $E(\cdot)$ and decoder $D(\cdot)$, encodes x to latent space, s.t. $z_{init} = E(x)$. In addition, m is downsampled to m_{latent} in order to meet latent spatial dimensions.

In each BLD step t , the following occurs:

1. The latent resulting from the previous step, z_{t+1} , undergoes denoising conditioned by the prompt p , to yield z_{fg} (we refer to the generated content as *foreground*, or *fg*).
2. The original image latent z_{init} is noised to step t , yielding z_{bg} (we refer to the original content as *background*, *bg*).
3. The next step z_t is obtained by blending z_{fg} and z_{bg} using m_{latent} :

$$z_t = z_{fg} \odot m_{latent} + z_{bg} \odot (1 - m_{latent}) \quad (1)$$

where \odot denotes element-wise multiplication.

After the final step, the output z_0 is decoded to obtain the final edited image $\hat{x} = D(z_0)$.

However, because information is lost during the VAE encoding, the decoded final output \hat{x} , might exhibit some artifacts when the unmasked region has important fine-detailed content (such as faces, text, etc.). Avrahami et al. (2023) solve this issue by optionally fine-tuning the decoder weights for each image after the denoising steps, and using these weights to infer the final result. In our experiments, we found that this optional background preservation process is no longer necessary (possibly due to improvements in the Stable Diffusion VAE), and a final blending with Gaussian feathering suffices (Figure 13 in appendix).

4 Method

Given an image, a text prompt, and a user-indicated location (e.g., via a mouse click), our goal is to modify the image

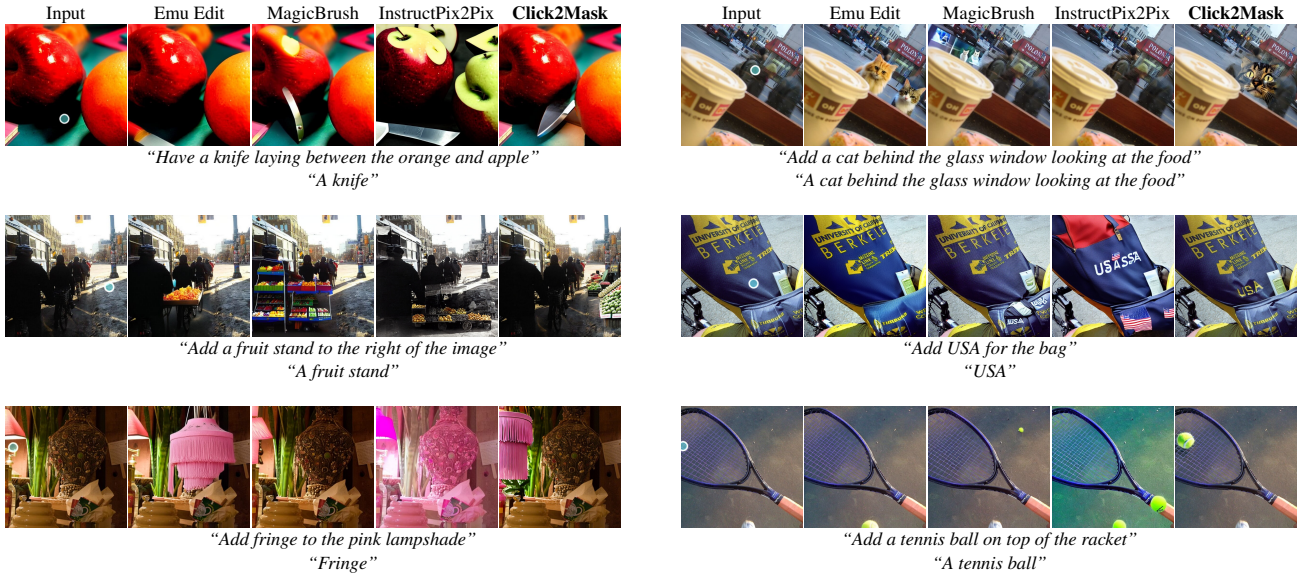


Figure 4: **Comparisons with SoTA methods.** Comparisons of Emu Edit (Sheynin et al. 2023), MagicBrush (Zhang et al. 2023) and InstructPix2Pix (Brooks, Holynski, and Efros 2023) with our model **Click2Mask**. Upper prompts were given to baselines, and lower ones to Click2Mask. The inputs contain the clicked point given to Click2Mask. See Figure 8 and appendix for additional comparisons.

according to the prompt in an unspecified area roughly surrounding the provided point. We utilize Blended Latent Diffusion (BLD) (Avrahami, Fried, and Lischinski 2023) as our image editing backbone, but rather than providing it with a fixed mask at the outset, we evolve a mask dynamically throughout the diffusion process. We initialize the process with a large mask around the indicated point, and gradually *contract* the mask towards the center, while guiding the rate of contraction along the mask boundary using a semantic alignment loss based on Alpha-CLIP (Sun et al. 2023).

This iterative process results in a mask whose shape and size are determined by both the text prompt, the content, and the structure of the original input image. Furthermore, the shape of the mask adjusts itself to the emerging object, as the mask’s evolution is determined by the gradients obtained by the semantic alignment loss (see Section 4.1), which in turn depend on the shape of the object being generated (see Figure 2 for mask evolution illustration, and Figure 6 for examples of generated masks). Once the mask has settled into its final form, we run BLD once more, using the final mask to generate the final result. Our method is outlined in Algorithm 1 and illustrated in Figure 5.

4.1 Dynamic Mask Evolution

Given an image x , a text prompt p , and a user-provided location c , we aim to modify x , so as to align with p , in proximity to c . We start by encoding the input image $z_{init} = E(x)$. We also create a 2D potential height-field Φ in latent space, which is initialized to a Gaussian around c .

We now perform the BLD process, where at each step t we obtain a binary mask M_t by thresholding the potential Φ using a threshold τ . The mask evolves dynamically through the BLD process, since the threshold τ and the potential Φ are

both updated at each step: the threshold τ increases, while the potential Φ is elevated in important areas to ensure they remain above the threshold. This prevents the mask from shrinking in spatial areas that emerge as important for alignment of the generated new content with the guiding prompt p . As a consequence, the mask evolves into a shape determined by the newly generated object.

Commencing the blending at 25% of the diffusion steps, the initial threshold value τ_{init} is relatively low, such that M_t is sufficiently large at the beginning ($\sim 16\%$ of the image). This enables BLD to capture the desired edit, as demonstrated in Figure 7 (this idea was originally introduced in BLD to cope with the case of small or thin input masks). On the other hand, we prevent the generation of overly large masks, that would result in large scale changes that might not blend seamlessly with the original content, by increasing τ geometrically, and by initiating the potential elevation at a later stage (40% out of total diffusion steps). However, we initiate the potential elevation at an early enough stage when the blended image is still noisy and can be modified. We stop mask evolution when spatial structure is close to be determined (at 50% of total diffusion steps).

The potential elevation is obtained by generating the estimated final image \tilde{x}_0 at each step, and calculating the cosine distance between the CLIP (Radford et al. 2021) embeddings of \tilde{x}_0 and the guidance prompt p . \tilde{x}_0 is obtained by blending a predicted final foreground latent \tilde{z}_{fg} , with the original latent background z_{init} :

$$\tilde{z}_0 = \tilde{z}_{fg} \odot M_t + z_{init} \odot (1 - M_t) \quad (2)$$

The decoded $\tilde{x}_0 = D(\tilde{z}_0)$ is passed alongside the current mask M_t and the prompt p to Alpha-CLIP to focus on the area of M_t . The gradient of the cosine distance with respect

5 Results

Algorithm 1: Click2Mask

Given: models $LDM = \{noise(z, t), denoise(z, p, t) \rightarrow (z_t, z_0)\}$, VAE = $\{E(x), D(z)\}$, BLD = $\{(x, p, m, t) \rightarrow z_t\}$, Alpha-CLIP = $\{\alpha_{CLIP}(x, m, p) \rightarrow Sim_{CLIP}\}$, and **hyper parameters** $\{\tau_{n...l}, lr\}$ with schedulers $\{n, r, k, l\}$

Input: input image x , text prompt p , target coordinates c
Output: edited image \hat{x} that matches the prompt p in proximity of c , and complies to x outside edited region

```

 $\Phi = Gaussian(c)$ 
 $z_{init} = E(x)$ 
 $z_n \sim noise(z_{init}, n)$ 
for all  $t$  from  $n$  to  $l$  do
   $z_{bg} \sim noise(z_{init}, t)$ 
   $z_{fg}, \tilde{z}_{fg} \sim denoise(z_t, p, t)$ 
   $G = 0$ 
  if  $t < r$  then
     $\tilde{z}_0 = \tilde{z}_{fg} \odot M_t + z_{init} \odot (1 - M_t)$ 
     $S_t \sim \alpha_{CLIP}(D(\tilde{z}_0), upscale(M_t), p)$ 
     $G \sim |gradients(S_t, M_t)|$ 
     $z_{fg} \sim BLD(x, p, M_t, t)$ 
  end if
  if  $t < k$  and  $S_t > S_{t+1}$  then exit loop
   $M_t = (\Phi + G * lr) > \tau_t$ 
   $z_t = z_{fg} \odot M_t + z_{bg} \odot (1 - M_t)$ 
end for
 $\hat{z} \sim BLD(x, p, M_t, 0)$ 
return  $D(\hat{z})$ 

```

to the latent mask pixels is then calculated by backpropagating through the CLIP embeddings and the decoder. The larger the absolute gradient of the cosine distance (i.e. CLIP loss) with respect to a pixel in M_t , the more significant this location is for the alignment of the generated content to the prompt p . Adding the absolute gradient values G to Φ , elevates important areas in the Φ height-field (around M_t 's contour for stable evolution – Figures 14 and 15 in appendix).

Halfway through the mask evolution steps, we utilize the Alpha-CLIP loss as an optional stopping point for M_t 's evolution, if the loss did not decrease in subsequent steps.

After each update of M_t , we restart the BLD process, letting it proceed from the beginning to the current step t , using the mask M_t as a fixed mask. This is to allow pixels that were added (or removed) in M_t to affect the generated image from the beginning (see Figure 16 in appendix).

We then apply Equation (1) and blend z_{fg} with z_{bg} using the mask M_t , which provides z_{t-1} , the input to next step.

After all mask evolution steps have been completed, we perform a final BLD run using the final M_t with several seeds to obtain several candidate results, where the best one is filtered by Alpha-CLIP. As noted earlier, rather than fine-tuning the VAE decoder weights to preserve the original background details outside the mask, we employ instead a simple Gaussian mask feathering when blending the BLD output and the original input image (in pixel space).

Given that our method is mask-free, we compare ourselves to mask-free image editing methods, with the slight difference being that a *clicked point* replaces any parts of the prompt that describe the edit location. To begin with, we compare to MagicBrush, which is the SoTA method among the open-source models. In addition, we compare to Emu Edit, which is the SoTA among closed-source models. Since we are unable to run Emu Edit ourselves, we must rely on the Emu Edit Benchmark (Sheynin et al. 2023), which includes images generated by Emu Edit. This benchmark contains images with several categories of editing instructions, such as adding objects, removing objects, altering style, etc., and can be filtered by these categories. As our focus is adding objects to images, we filtered the dataset by the “addition” instruction. This resulted in 533 items, from which we randomly sampled an evaluation subset of 100 samples.

We perform the following fixed routine for each sample: (i) Removed the word that instructs addition (e.g., “Add”, “Insert”), (ii) removed the part that describes the edit location, and instead (iii) clicked on the image to direct the editing location. For instance, the instruction “Add a black baseball cap to the man on the left” becomes “A black baseball cap” (non-localized instruction).

Following Emu Edit (Sheynin et al. 2023) and BLD (Avrahami, Fried, and Lischinski 2023), each sample run produces multiple results internally (comprising of three mask evolutions, each followed by a batch of 8 outputs), and outputs the best result, as determined automatically using Alpha-CLIP scoring.

To evaluate our results, we compared these 100 outputs generated by Click2Mask, with the outputs generated by Emu Edit and by MagicBrush (which ran with the original edit instructions). We conducted the evaluation through a user study (Section 5.1), as well as through automatic metrics (Section 5.2). In both cases, our method outperformed the SoTA methods.

5.1 Human Evaluation

We conducted a user study, where participants were given a random batch of survey items out of 200 total items (100 items comparing to each model). Each item included an input image, the original edit instruction, and a pair of edited images: one generated by our model, and the other generated by either Emu Edit or MagicBrush. Participants were asked to rank which of the edited images performed better according to three criteria: executing the instruction, not adding any other edits or artifacts, and generating a realistic image. The survey was completed by 149 participants. Each of the 200 items was rated by at least 5 users, where the average rate was 15.67 users in Emu Edit, and 8.06 users in MagicBrush.

In order to compare Click2Mask vs. Emu Edit, as well as Click2Mask vs. MagicBrush, while taking into account “ties” (ratings stating equal performance on an item, or items with equal ratings to both methods), we analyzed the results using the following metrics: (A) The percentage of items in which each method was preferred by the majority, disregarding ties. (B) For each item we counted if the majority voted

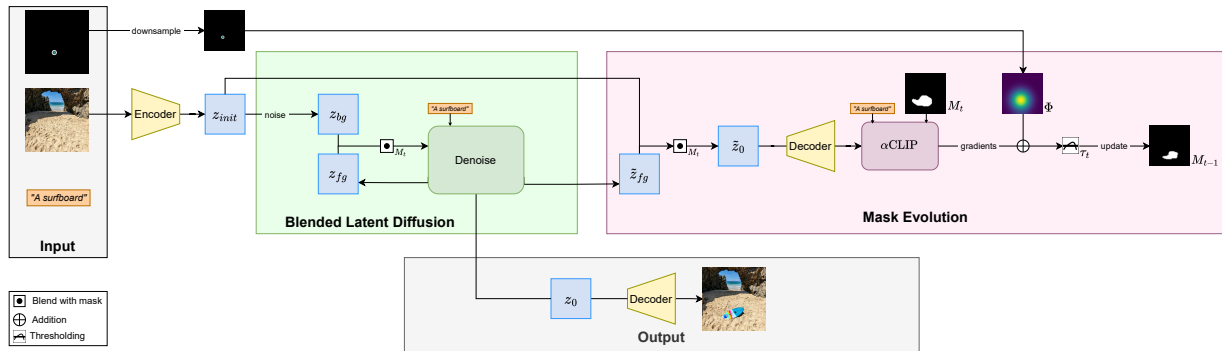


Figure 5: **Click2Mask**: An illustration of our method as described in Algorithm 1. The **green block** is BLD process, performing diffusion steps while blending noised input latents with text guided latents. The **pink block** is the mask evolution process, where we utilize Alpha-CLIP to evaluate the gradients with respect to the mask M_t pixels, using them to update M_t , obtaining M_{t-1} .

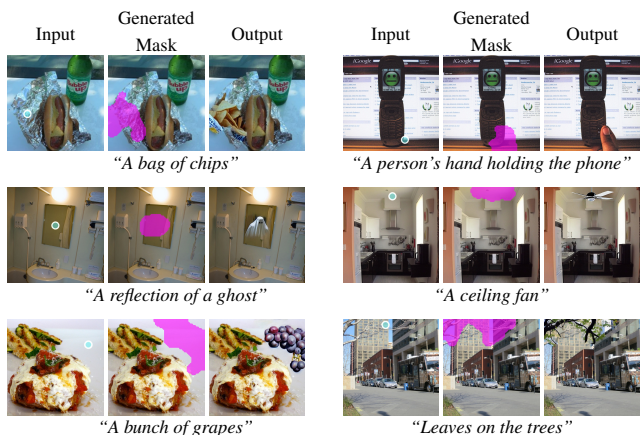


Figure 6: **Examples of generated masks**. For each triplet, given an input image with clicked point (left) and a prompt (below), a purple overlay shows the generated mask (middle). The rightmost image is Click2Mask output.

for a tie, and if so marked it as a “tied item”. For the other “non-tied items”, we conducted the same majority vote analysis described in A. (C) The number of total ratings for each method. In each parameter our method surpassed the closed-source SoTA method Emu Edit, and the open-sourced SoTA MagicBrush, as shown in Table 1. See Figures 1, 4 (and Figures 17, 18, 19, 20 in appendix) for qualitative comparisons to the baselines alongside with InstructPix2Pix, and Figure 8 with a detailed comparison. Statistical significance analysis can be found in the appendix.

5.2 Automatic Metrics

Utilizing the input captions and output captions (describing the desired output) provided in Emu Edit benchmark, a variety of metrics were used to assess each method’s outputs on the sampled items: (i) Directional CLIP (Gal et al. 2022) similarity ($CLIP_{direct}$) to measure correspondence of the change between input and output images, with the change between input and output captions. (ii) CLIP sim-

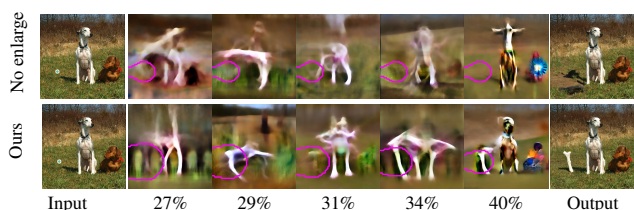


Figure 7: **Ablation study: No mask enlargement at early stages**. As explained in Section 4, we begin with a large mask (about 16% of the image), to capture the desired edit in M_t . Top row: M_t (marked by purple contours on decoded \tilde{z}_{fg} s, throughout diffusion steps indicated by percentages below) evolves without a large starting point, and the diffusion identifies and guides the white dog to the prompt “Huge bone”, while the small M_t fails capturing the bone. Bottom row: Click2Mask’s enlarged M_t captures the guided area even though the dog is also initially identified as the bone.

ilarity between output image and output caption, to measure the compliance between desired and generated outputs ($CLIP_{out}$). (iii) Mean L1 pixel distance between input and output images, to measure the amount of change in the entire image (L1). (iv) In addition, we present a new metric, Edited Alpha-CLIP ($\alpha CLIP_{edit}$).

Edited Alpha-CLIP. Besides evaluating the images *globally*, it is beneficial to evaluate the *edited region*. We offer an Edited Alpha-CLIP procedure to overcome the lack of input or output masks in Emu Edit and MagicBrush: we extract a mask specifying the edited area in the generated image, and calculate the Alpha-CLIP similarity between the masked generated image and the instruction (removing words describing addition and edit locations as mentioned in Section 5). See Appendix A.4 and Figure 12 in appendix for details and extracted masks demonstrations.

Table 2 shows that our method surpassed both Emu Edit and MagicBrush in all metrics: higher scores in all CLIP-based metrics, indicating stronger similarities, and lower L1 distance indicating better compliance with input image.



Figure 8: **Failure cases of baselines.** Baselines suffer occasionally from replacing an existing object instead of adding one (\Leftrightarrow), misplacing the object ($\leftarrow P$), modifying other objects (\wedge), altering the image globally (\blacksquare), or failing to produce an edit (\emptyset). For additional comparisons to baselines, see Figure 4 and appendix.

5.3 Ablation Study

We conducted several ablation studies to analyze the impact of various components on the overall performance of our model. Figure 7 demonstrates the need for a sufficiently large mask on early diffusion steps. See additional ablation studies in Appendix A.2 accompanied by Figures 13 to 16.

6 Model Limitations

During the evolution process, our model encounters a difficulty converging to a small, finely detailed mask shape (e.g. a dog collar). Additionally, since text guidance in Stable Diffusion is not spatially driven, BLD sometimes has difficulty adding the desired object to the masked area when a similar object is nearby in the unmasked area (e.g., adding a Bigfoot next to a person). Since we use BLD as our backbone, we sometimes encounter this problem. However, we have considerably improved it in comparison to BLD by optimizing the progressive mask shrinking process, and applying it across all objects, not just thin objects, as part of our mask evolution process. Moreover, in comparison to other SOTA methods, they often fail to add the desired object even if a

| | (A) | (B) | (C) | |
|------------|---------------|--------------|--------------------------|---------------|
| Method | % Majority | % Tied items | % Majority from non-tied | # Total votes |
| Emu Edit | 42.86% | 35% | 47.69% | 416 |
| Click2Mask | 57.14% | | 52.31% | 465 |
| MagicBrush | 16.30% | 27% | 15.07% | 148 |
| Click2Mask | 83.70% | | 84.93% | 362 |

Table 1: **Human evaluation results.** Comparisons of (A): % of items each method received majority votes, disregarding ties. (B): % of items the majority voted as tie (left), and % of items – out of the other non-tied items – each method received majority votes (right). (C): Total votes. Refer to Section 5 for details.

| Method | CLIP _{direct} \uparrow | CLIP _{out} \uparrow | α CLIP _{edit} \uparrow | L1 \downarrow |
|------------|-----------------------------------|--------------------------------|--|-----------------|
| Emu Edit | 0.150 | 0.331 | 0.186 | 0.046 |
| MagicBrush | 0.095 | 0.324 | 0.166 | 0.049 |
| Click2Mask | 0.204 | 0.334 | 0.195 | 0.027 |

Table 2: **Automatic metrics results.** Evaluation using automatic metrics. CLIP_{direct} measures consistency between changes (from input to output) in images and captions, CLIP_{out} measures similarity between output image and output caption, α CLIP_{edit} measures similarity to the non-localized instruction in the edited area, and L1 measures the alignment with the input image. See Section 5 for details.

similar one is not present, and our method outperforms them in both cases. See Figure 9 for examples of these cases.



Figure 9: **Limitations** Top row: the evolving mask has a difficulty converging to a small fine detailed mask shape as a golden necklace. Bottom row: our Stable Diffusion backbone guides the image to a given prompt globally. When the prompt content already exists in the input image close to the generated mask (i.e. the white dog), sometimes the BLD process we incorporate does not succeed in directing the guidance into the masked region.

7 Conclusion

Click2Mask presents a novel approach for local image generation, freeing users from having to specify a mask, or describing the input or target images, and without being con-

strained to existing objects. We look forward to users applying our method with the source code that is available in the project page (see Footnote in Page 1), either to edit images or to embed the method for generating or fine-tuning masks.

References

- Avrahami, O.; Aberman, K.; Fried, O.; Cohen-Or, D.; and Lischinski, D. 2023. Break-A-Scene: Extracting Multiple Concepts from a Single Image. In *SIGGRAPH Asia 2023 Conference Papers*, SA '23. New York, NY, USA: Association for Computing Machinery. ISBN 9798400703157.
- Avrahami, O.; Fried, O.; and Lischinski, D. 2023. Blended Latent Diffusion. *ACM Trans. Graph.*, 42(4).
- Avrahami, O.; Lischinski, D.; and Fried, O. 2022. Blended Diffusion for Text-Driven Editing of Natural Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18208–18218.
- Bar-Tal, O.; Ofri-Amar, D.; Fridman, R.; Kasten, Y.; and Dekel, T. 2022. Text2live: Text-driven layered image and video editing. In *European Conference on Computer Vision*, 707–723. Springer.
- Betker, J.; Goh, G.; Jing, L.; Brooks, T.; Wang, J.; Li, L.; Ouyang, L.; Zhuang, J.; Lee, J.; Guo, Y.; Manassra, W.; Dhariwal, P.; Chu, C.; Jiao, Y.; and Ramesh, A. 2023. Improving Image Generation with Better Captions.
- Brooks, T.; Holynski, A.; and Efros, A. A. 2023. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *CVPR*.
- Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9650–9660.
- Couairon, G.; Verbeek, J.; Schwenk, H.; and Cord, M. 2022. DiffEdit: Diffusion-based semantic image editing with mask guidance. *arXiv:2210.11427*.
- Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.
- Fu, T.-J.; Hu, W.; Du, X.; Wang, W. Y.; Yang, Y.; and Gan, Z. 2024. Guiding Instruction-based Image Editing via Multimodal Large Language Models. In *The Twelfth International Conference on Learning Representations*.
- Gal, R.; Patashnik, O.; Maron, H.; Bermano, A. H.; Chechik, G.; and Cohen-Or, D. 2022. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *ACM Trans. Graph.*, 41(4).
- Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2022. Prompt-to-prompt image editing with cross attention control.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.
- Kawar, B.; Zada, S.; Lang, O.; Tov, O.; Chang, H.; Dekel, T.; Mosseri, I.; and Irani, M. 2023. Imagic: Text-Based Real Image Editing with Diffusion Models. In *Conference on Computer Vision and Pattern Recognition 2023*.
- Kingma, D. P.; and Welling, M. 2013. Auto-Encoding Variational Bayes. *arXiv:1312.6114*.
- Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. *arXiv:2304.02643*.
- Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Li, C.; Yang, J.; Su, H.; Zhu, J.; and Zhang, L. 2023. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv:2303.05499*.
- Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. *arXiv:2112.10741*.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv:1912.01703*.
- Pearson, K. 1900. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302): 157–175.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv:2103.00020*.
- Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv:2204.06125*.
- Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.
- Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494.
- Sheynin, S.; Polyak, A.; Singer, U.; Kirstain, Y.; Zohar, A.; Ashual, O.; Parikh, D.; and Taigman, Y. 2023. Emu Edit: Precise Image Editing via Recognition and Generation Tasks. *arXiv:2311.10089*.
- Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Sun, Z.; Fang, Y.; Wu, T.; Zhang, P.; Zang, Y.; Kong, S.; Xiong, Y.; Lin, D.; and Wang, J. 2023. Alpha-CLIP: A CLIP Model Focusing on Wherever You Want. arXiv:2312.03818.

Tumanyan, N.; Geyer, M.; Bagon, S.; and Dekel, T. 2022. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. arXiv:2211.12572.

Wang, Q.; Zhang, B.; Birsak, M.; and Wonka, P. 2023a. InstructEdit: Improving Automatic Masks for Diffusion-based Image Editing With User Instructions. arXiv:2305.18047.

Wang, S.; Saharia, C.; Montgomery, C.; Pont-Tuset, J.; Noy, S.; Pellegrini, S.; Onoe, Y.; Laszlo, S.; Fleet, D. J.; Soric, R.; Baldrige, J.; Norouzi, M.; Anderson, P.; and Chan, W. 2023b. Imagen Editor and EditBench: Advancing and Evaluating Text-Guided Image Inpainting. arXiv:2212.06909.

Xie, D.; Wang, R.; Ma, J.; Chen, C.; Lu, H.; Yang, D.; Shi, F.; and Lin, X. 2023. Edit Everything: A Text-Guided Generative System for Images Editing. arXiv:2304.14006.

Xie, S.; Zhang, Z.; Lin, Z.; Hinz, T.; and Zhang, K. 2022. SmartBrush: Text and Shape Guided Object Inpainting with Diffusion Model. arXiv:2212.05034.

Yates, F. 1934. Contingency Tables Involving Small Numbers and the χ^2 Test. *Supplement to the Journal of the Royal Statistical Society*, 1(2): 217–235.

Zhang, K.; Mo, L.; Chen, W.; Sun, H.; and Su, Y. 2023. MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. In *Advances in Neural Information Processing Systems*.

Zou, S.; Tang, J.; Zhou, Y.; He, J.; Zhao, C.; Zhang, R.; Hu, Z.; and Sun, X. 2024. Towards Efficient Diffusion-Based Image Editing with Instant Attention Masks. arXiv:2401.07709.

A Additional Experiments

In Appendix A.1 we start by providing additional Click2Mask generated masks examples, as well as further comparisons to the baselines Emu Edit, MagicBrush, and InstructPix2Pix. In Appendix A.2 additional ablation tests are provided. Appendix A.3 shows a statistical analysis held on the results from the user case study in Section 5.

A.1 Additional Results

Additional examples of Click2Mask generated masks can be found in Figure 10. Further results comparing to baselines Emu Edit, MagicBrush, and InstructPix2Pix are provided in Figure 17, Figure 18, Figure 19, and Figure 20. A comparison of prompt lengths with baselines is illustrated in Figure 11.

A.2 Additional Ablation Study

Figure 15 illustrates the importance of elevating potential Φ only around the area of M_t 's contour, and not across the entire image. Figure 16 demonstrates an ablation study for the rerun component. Figure 13 shows the importance of Gaussian mask feathering after the final diffusion step. Figure 14 depicts the importance of adding a surrounding receptive around M_t 's area for gradient addition. An additional ablation study can be found at Section 5.3.

A.3 Statistical Analysis

As mentioned in Section 5, we conducted a user case study between Click2Mask with both Emu Edit and MagicBrush. To determine whether our comparisons are statistically significant, we use Pearson's Chi-squared test (Pearson 1900) with Yates's continuity correction (Yates 1934). The tests show that the results are statistically significant, as can be seen in Table 3.

A.4 Edited Alpha-CLIP Mask Extraction

As mentioned in Figure 12, in order to evaluate the edited region in methods that do not have input or output masks (as Emu Edit and MagicBrush), we extract a mask which specifies this region. The mask is extracted by first calculating the L1 distance between the input image and the generated image. We then take the mean value over the RGB channels for each pixel, and further clean noise by thresholding, Min-Pooling and Max-Pooling, and creating convex hulls. This provides us with an almost exact mask of the edited region, as demonstrated in Figure 12.

B Implementation Details

B.1 Pretrained Models

The pretrained models that we have used in all the experiments described in this paper are as follows:

- Blended Latent Diffusion model from Avrahami et al. (2023).
- Text-to-image Latent Diffusion model from Rombach et al. (2022) with checkpoint <https://huggingface.co/stabilityai/stable-diffusion-2-1-base>.

- Alpha-CLIP with ViT-L/14@336px by Sun et al. (2023).
- Emu Edit benchmark from https://huggingface.co/datasets/facebook/emu_edit_test_set and Emu Edit generated images from https://huggingface.co/datasets/facebook/emu_edit_test_set_generations by Sheynin et al. (2023).
- MagicBrush by Zhang et al. (2023) results were generated with latest checkpoint `MagicBrush-epoch-52-step-4999.ckpt`.
- InstructPix2Pix results generated from <https://huggingface.co/spaces/timbrooks/instruct-pix2pix> by Brooks et al. (2023).

All the above were implemented in PyTorch (Paszke et al. 2019).

For DALL·E 3 (Betker et al. 2023), we used OpenAI's ChatGPT-4o interface <https://chatgpt.com>.

All input images are real and under free public domain or Creative Commons license (including Jeremy Bishop, Isaac Maffei, Odysseas Chloridis and Cerqueira under Unsplash license; jenyalucy and Icecube11 under Pixabay license).

B.2 Our Model

When calculating the Alpha-CLIP loss to derive gradients and to pick automatically the best output out of different random seeds (as discussed in Section 5), we augmented the image to mitigate adversarial results, as discussed in (Avrahami, Lischinski, and Fried 2022), and dilate the mask region to add context.

The diffusion steps consisted of 100 steps.

To achieve unity over different samples in terms of learning rate lr and potential Φ , a normalization is performed on the saliency map (i.e., absolute gradients backpropagated from Alpha-CLIP loss function).

To reduce noise, maintain stability, and ensure a smooth mask, we perform Gaussian filtering to M_t on a number of occasions, and post-process it each update step to account for gaps that can occur due to the landscape thresholding, such as filling holes, connecting disjointed mask parts, removing noise, etc. Additionally, we reset to the initial random seed on each BLD rerun for consistency of mask evolution.

Source code of our model, which is implemented in PyTorch and runs on a GPU, is publicly available in the project page (see Footnote in Page 1).

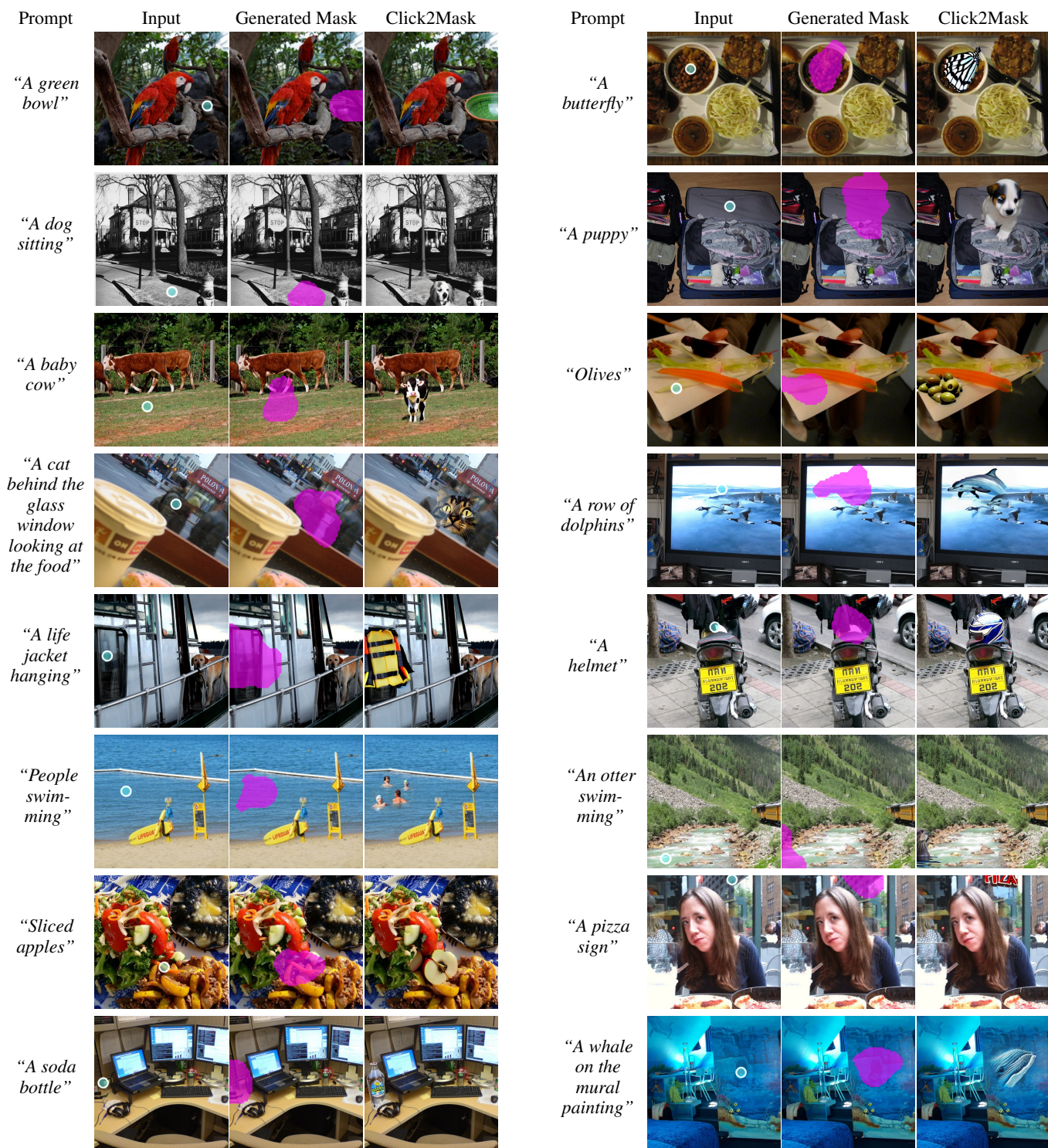


Figure 10: **Additional examples of generated masks.** In each image triplet, the leftmost image is the input with clicked point, accompanied by the given prompt on its left. The generated mask is demonstrated by a purple overlay on the input image (center image) and the rightmost image is the output of Click2Mask.

| Method 1 | Method 2 | Majority (A) p-value | Majority (B) p-value | Total votes p-value |
|------------|----------|-------------------------|-------------------------|------------------------|
| Emu Edit | Ours | $p < 10^{-21}$ | $p < 10^{-14}$ | $p < 10^{-192}$ |
| MagicBrush | Ours | $p < 10^{-19}$ | $p < 10^{-15}$ | $p < 10^{-111}$ |

Table 3: **Statistical analysis.** We use Pearson’s Chi-squared test with Yates’s continuity correction to determine whether our results are statistically significant. Majority (A) refers to the comparison of majority votes for each item disregarding ties, and majority (B) refers to the comparison of votes disregarding items that most users rated as ties. Total votes are the total ratings for each method. See Section 5 for further details.

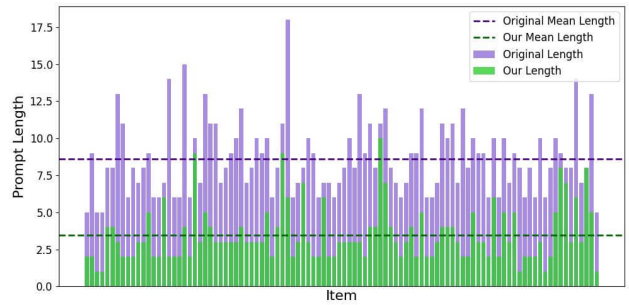


Figure 11: **Prompts word count.** An additional advantage of our method is that users can provide shorter prompts, which require less effort on their part. The bar plot above shows prompts word lengths of Emu Edit benchmark in comparison to Click2Mask. Each purple high bar represents the number of words in an item in Emu Edit benchmark, and the overlaid green low bar represents the corresponding prompt given to Click2Mask after removing the word that describes addition (e.g. “Add”, “Insert”, etc.) and the words describing the desired edit location (e.g. “on the table next to the fridge”), as explained in the fixed routine in Section 5. The 100 bars correspond to the 100 samples we compared with Emu Edit and MagicBrush, as described also in Section 5. The purple higher horizontal line is the mean prompt length in Emu Edit benchmark’s samples, while the green lower one is the mean length of Click2Mask’s shorter prompts.

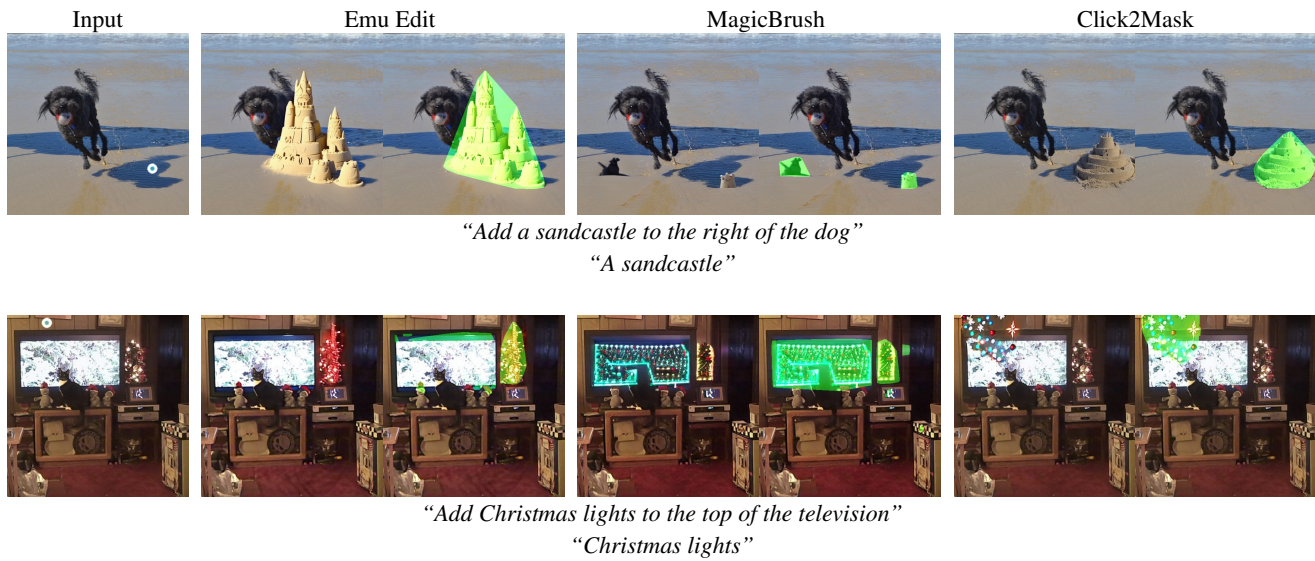


Figure 12: **Edited Alpha-CLIP**. A depiction of extracted masks as part of our Edited Alpha-CLIP metric presented in Section 5. Left column is the input with clicked point, where the text below each image row is the instruction given to Emu Edit and MagicBrush (higher text) and prompt given to Click2Mask (lower text). In each method’s pair, the left image is the output, and the right image is the mask extracted by the Edited Alpha-CLIP metric.

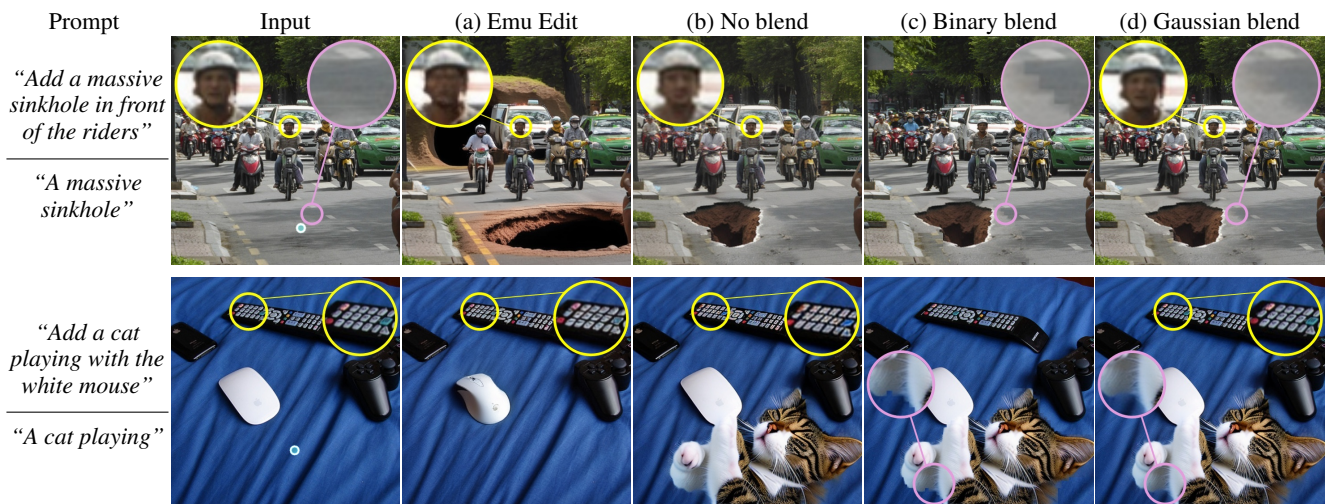


Figure 13: **Background preservation ablation study**. When decoding the final diffused latents, details are not fully preserved (b). A binary blending of the mask and the input image at pixel space will yield artifacts on pixels surrounding the mask’s contour (c). Emu Edit suffers as well from loss of details. As mentioned in Section 4, we suggest a Gaussian blend at pixel space (d), which preserves the background details, while creating a seamless blend. This also eliminates the need for a decoder weights optimization presented in BLD. Please zoom in for a vivid visual.

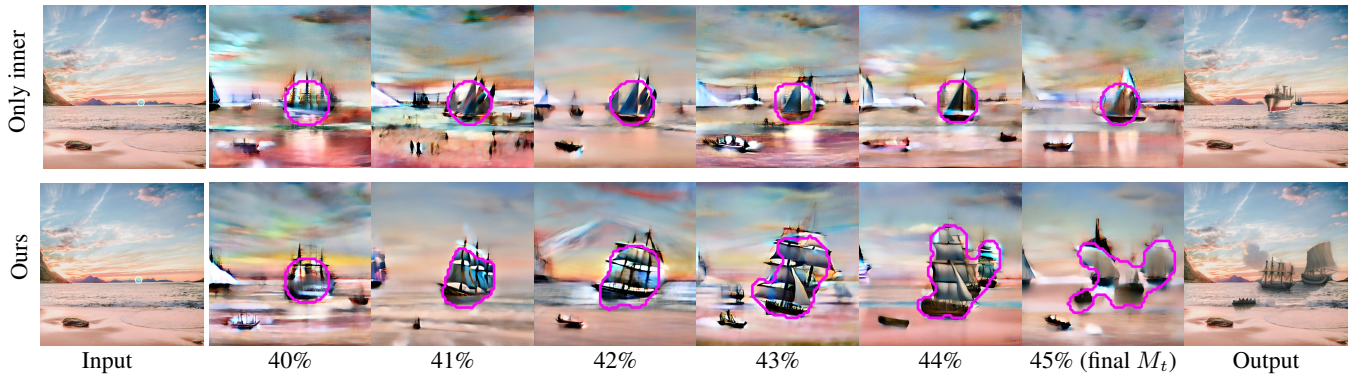


Figure 14: **Ablation study: elevating Φ only inside M_t .** Top row shows the evolution of M_t (depicted by purple contours over \tilde{z}_{fg} s during diffusion steps as indicated by percentages below) where potential Φ elevation is contained within current M_t . Bottom row depicts M_t 's evolution in Click2Mask, where a surrounding ring of M_t is also elevated in Φ . The prompt is “*Fleet of ships*”. When elevating Φ only within M_t , the mask shrinks continuously, unlike Click2Mask, where the outer ring elevation prevents the mask from shrinking in important areas, resulting in a mask shaped according to the generated objects.

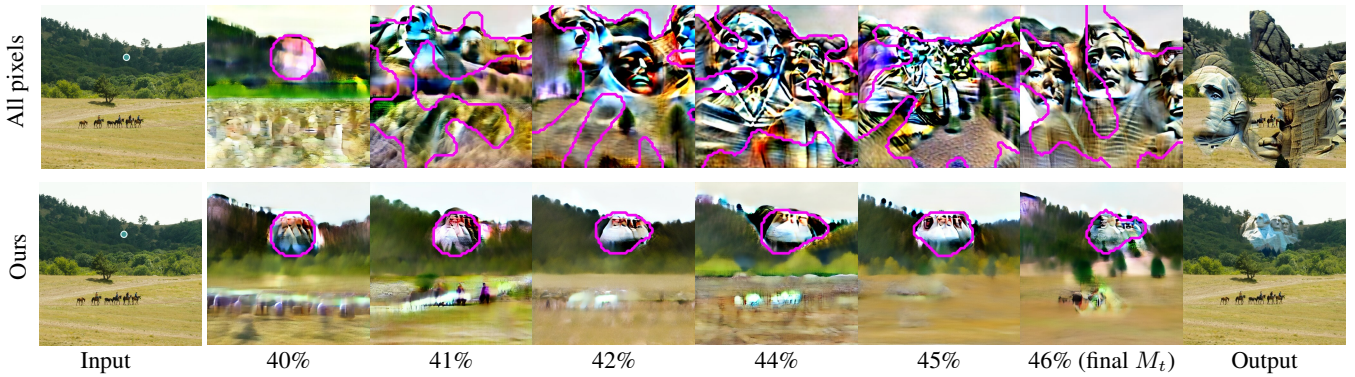


Figure 15: **Ablation study: elevating Φ on all image.** With the prompt “*Figures from Mount Rushmore*”, the top row depicts mask M_t 's evolution (shown by a purple contour over \tilde{z}_{fg} throughout the diffusion steps indicated by percentages below) when elevating potential Φ across the entire image. This results in an unstable and unsmooth mask progression, with an output disassociated from the input image. Bottom row depicts M_t 's evolution in Click2Mask, where only the surrounding area of M_t 's contour is elevated in Φ .

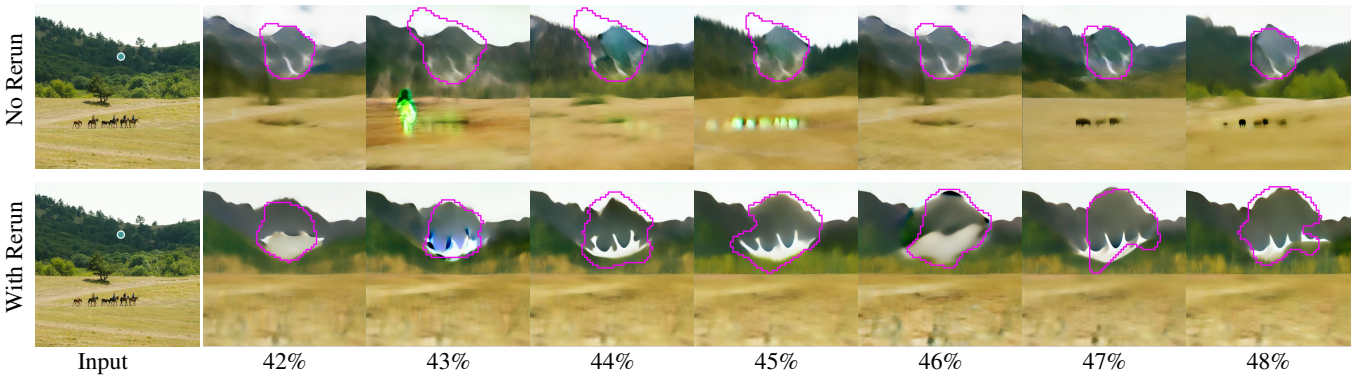


Figure 16: **Rerun ablation study.** The figure depicts the rerun procedure. The prompt is “*Snowy mountains*”, and the upper row lacks rerun, while the lower row has. In the upper row (where the purple mask contours are marked over \tilde{z}_{fg} s throughout diffusion steps, as percentages below indicate), pixels that are added to the mask M_t at advanced stages, fail to comply to the guiding prompt, since the spacial information has already been determined. Rerun allows a “refresh” of the information to be driven towards the guiding prompt.

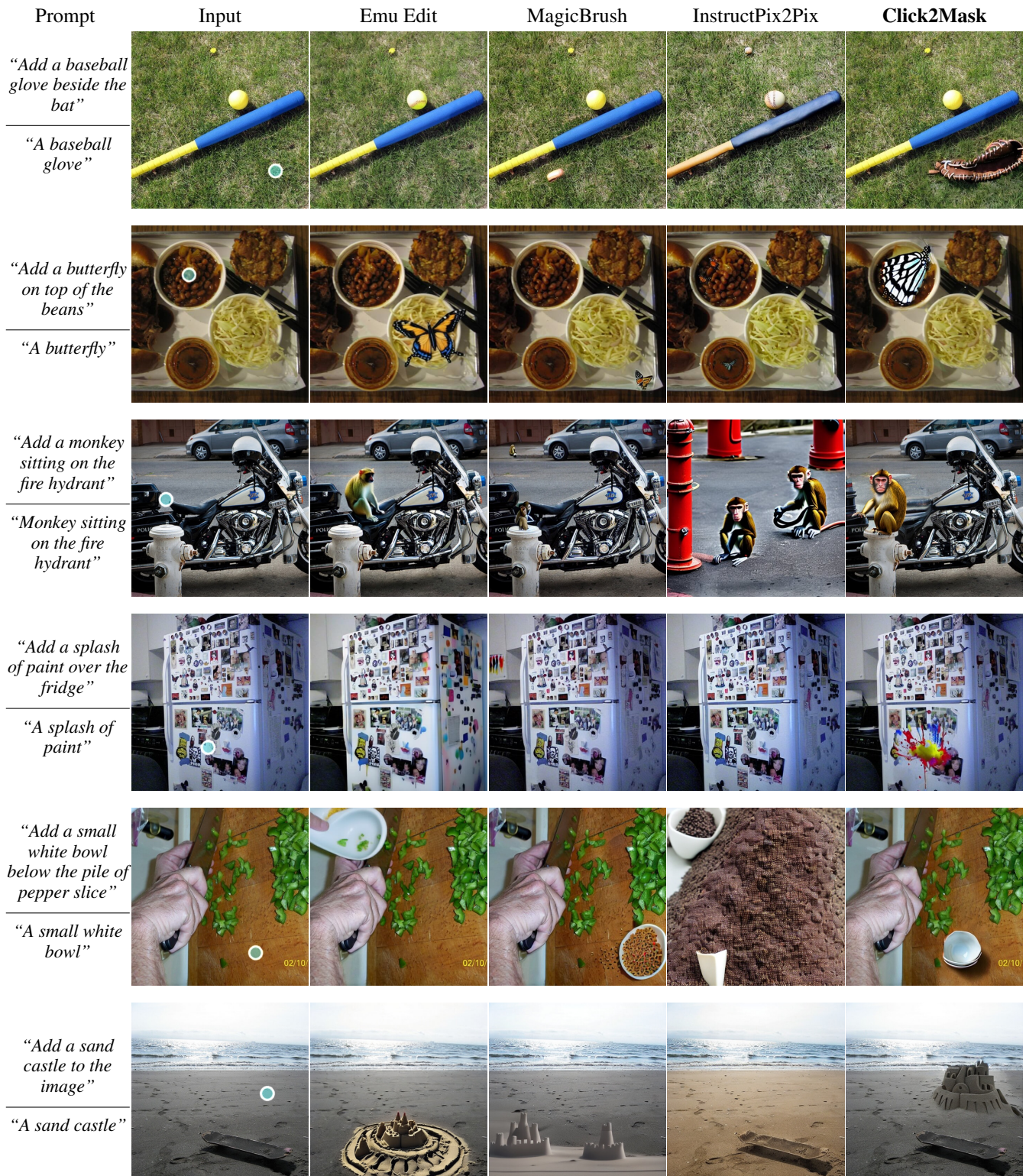


Figure 17: **Additional comparisons with SoTA methods.** Additional comparisons of Emu Edit (Sheynin et al. 2023), MagicBrush (Zhang et al. 2023) and InstructPix2Pix (Brooks, Holynski, and Efros 2023) with our model **Click2Mask**. The upper prompts were given to baselines, and the lower ones to Click2Mask. Inputs contain the clicked point from Click2Mask.

| Prompt | Input | Emu Edit | MagicBrush | InstructPix2Pix | Click2Mask |
|---|-------|----------|------------|-----------------|------------|
| <p>“Add a small pond in the front”</p> <p>“a small pond”</p> | | | | | |
| <p>“Add a smiley face on the wall between the cop and the stop sign”</p> <p>“A smiley face”</p> | | | | | |
| <p>“Add a tennis ball coming toward the man’s racquet”</p> <p>“A tennis ball”</p> | | | | | |
| <p>“Add a small teddy bear in front of the book”</p> <p>“A small teddy bear”</p> | | | | | |
| <p>“Add toys to the floor”</p> <p>“Toys”</p> | | | | | |
| <p>“Add an additional Christmas tree behind the container”</p> <p>“A Christmas tree”</p> | | | | | |

Figure 18: Additional comparisons with SoTA methods.

| Prompt | Input | Emu Edit | MagicBrush | InstructPix2Pix | Click2Mask |
|--|-------|----------|------------|-----------------|------------|
| <p>“Add a baseball in front of the batter next to his face”</p> <p>“A baseball”</p> | | | | | |
| <p>“Insert a bag of chips on the left side on the hotdog”</p> <p>“A bag of chips”</p> | | | | | |
| <p>“Add smoke to the planks”</p> <p>“Smoke”</p> | | | | | |
| <p>“In the mirror show the reflection of a ghost”</p> <p>“A reflection of a ghost”</p> | | | | | |
| <p>“Add graffiti to the orange tiles.”</p> <p>“Graffiti”</p> | | | | | |
| <p>“Add some toys in the stand”</p> <p>“Some toys”</p> | | | | | |

Figure 19: Additional comparisons with SoTA methods.

| Prompt | Input | Emu Edit | MagicBrush | InstructPix2Pix | Click2Mask |
|--|-------|----------|------------|-----------------|------------|
| <p><i>“Add a sandcastle to the right of the dog”</i></p> <p><i>“A sandcastle”</i></p> | | | | | |
| <p><i>“Add a grasshopper in the grass”</i></p> <p><i>“A grasshopper”</i></p> | | | | | |
| <p><i>“Add a hot air balloon to the background above the computer mouse”</i></p> <p><i>“A hot air balloon”</i></p> | | | | | |
| <p><i>“Add a pizza sign to the window”</i></p> <p><i>“A pizza sign”</i></p> | | | | | |
| <p><i>“Add a soda bottle to the desk”</i></p> <p><i>“A soda bottle”</i></p> | | | | | |
| <p><i>“Add a saddle to the horse”</i></p> <p><i>“A saddle”</i></p> | | | | | |

Figure 20: Additional comparisons with SoTA methods.