Blended Diffusion for Text-driven Editing of Natural Images Supplementary Material

Omri Avrahami[1] Dani Lischinski[1] Ohad Fried[2]
[1]The Hebrew University of Jerusalem [2]Reichman University

1. Additional Examples

In this section we provide additional examples of the applications and the failure cases that were mentioned in the main paper. In addition, we show that our method naturally supports an iterative editing process. Lastly, we demonstrate the naïve blending approach (main paper, Section 4.2.1).

1.1. Applications — Additional Examples

We provide additional examples for the applications mentioned in the paper: Figures 1 to 3 demonstrate the ability of our method to add new objects to an existing image, where Figures 1 and 2 show that different results can be obtained for the same text prompt, while Figure 3 shows results obtained using a variety of prompts. Figure 4 demonstrates the ability to remove or replace objects in an exiting image, while Figure 5 demonstrates the ability to alter an existing object in an image. Figures 6 and 7 demonstrate the ability to replace the background of an image. Figure 8 demonstrates more examples of scribble-guided editing, and Figure 9 demonstrates text-guided image extrapolation.

1.2. Iterative Editing

The synthesis results that are given by our method are at times exactly what the user envisioned, but they can also be different from the user's intent or might include unwanted artifacts. Unlike other text-driven image editing techniques that operate on the entire image (e.g., StyleCLIP [14]), our method is region-based, thus allowing the user to progressively refine their result in an *incremental* editing session.

Figure 10 demonstrates such an editing session. At first, the user starts by replacing the background, as described in Section 5.3 in the main paper, and obtains a result that is mostly satisfactory, but is not perfect: there are two unwanted generated objects on the grass that the user wishes to remove. In addition, the user decides that the initial mask used in the previous step was not accurate enough, causing a mismatch between the generated grass and the grass from the original scene. The user then provides additional masks, without a text prompt, causing our method to inpaint these areas, yielding the final result. Figures 11 to 13 demonstrate more editing sessions. Each of the sessions utilizes a variety of editing types: adding, changing and removing objects and backgrounds, scribble-guided edits, and clip-art-guided edits. Out method is compositional by design, and does not require any modifications to support such mixed editing sessions.

Unless stated otherwise, all the results in the main paper and in this supplemental document are *without* such incremental refinements — we show the raw results with no further user interaction.

1.3. Failure Cases

Figure 14 demonstrates the susceptibility of our model to typographic attacks [8]. Figure 15 demonstrates synthesis of objects which appear natural on their own, but possess the wrong size compared to the rest of the photo.

1.4. Naïve blending example

As discussed in Section 4.2.1 of the paper, naïve blending of the input image and the diffusion-synthesized result inside the masked area yields an unnatural result, as can be seen in Figure 16.

1.5. High-resolution generation

Most results presented in the paper use an unconditional DDPM model of resolution 256×256, producing generated images of that resolution. Nevertheless, we are not constrained to this resolution, as can be seen in Figure 10 in the main paper and in Figure 9 in this supplementary document (for more details read Section 2.5.2). We can also use OpenAI's unconditional 512×512 version of the model [12], by feeding the one-hot encoding with zeroes vector (similarly to [2]). Demonstration of using the higher resolution model for blended diffusion can be seen in Figure 17.

1.6. Comparison to DDIM

Our method uses Denoising Diffusion Probabilistic Models (DDPMs). Recently, Song et al. propose Denoising Diffusion Implicit Models (DDIMs) [16], a fast sampling

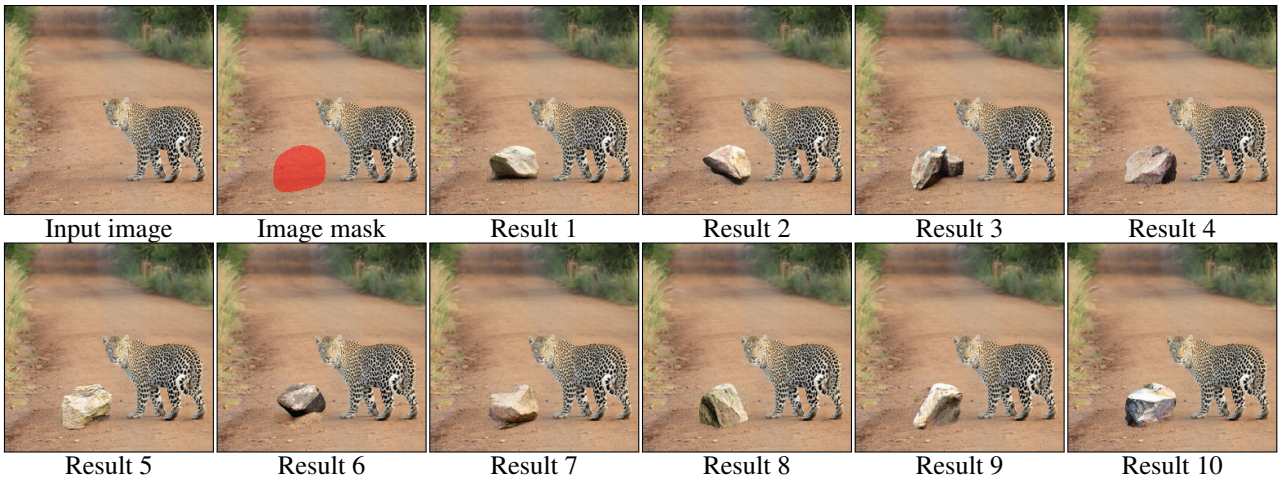


Figure 1. **Adding a new object (multiple results for the same input):** Given the input image, mask and text description "rock", our model is able to generate multiple plausible results.

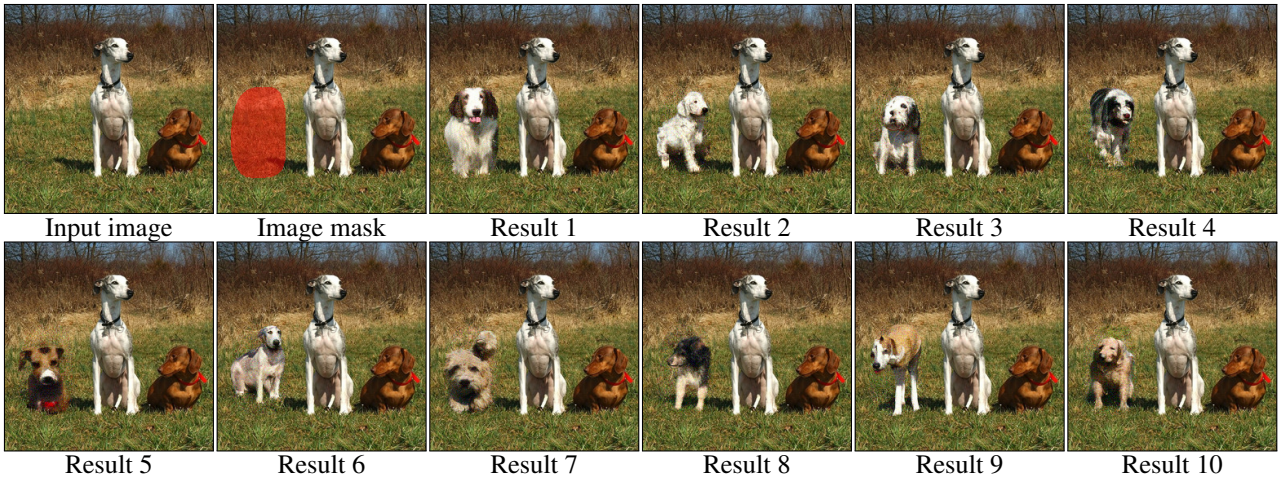


Figure 2. **Adding a new object (multiple results for the same input):** Given the input image, mask, and text description "a dog", our model is able to generate multiple plausible results. Some results are better (first row) than others (second row).

algorithm for DDPMs that produces a new implicit model with the same marginal noise distributions, but deterministically maps noise to images. Nichol et al. [10] showed that DDIMs produce better samples than DDPMs with fewer than 50 sampling steps, but worse samples when using 50 or more steps. In order to check the effect of using DDIM instead of DDPM we first adjusted the DDIM version of the guided-diffusion algorithm [5] with Blended Diffusion in Algorithm 1. As we can see experimentally in Figure 18, the same holds for image generation using Blended Diffusion: DDPMs produce better results than DDIMs when using 100 diffusion steps, but worse results when using less than 50 diffusion steps.

2. Implementation Details

For all the experiments reported in this paper we used a pre-trained CLIP model [15] and a pre-trained guided-diffusion model [5]:

- For the CLIP model we used ViT-B/16 as a backbone for the Vision Transformer [6] that was released by OpenAI [11].
- For the diffusion model we used an unconditional model of resolution 256×256 [12].

Both of these models were released under MIT license and were developed using PyTorch [13]. All the input images in this paper are real images (i.e., not synthesized), except the ones in Figure 5 of the main paper, which were

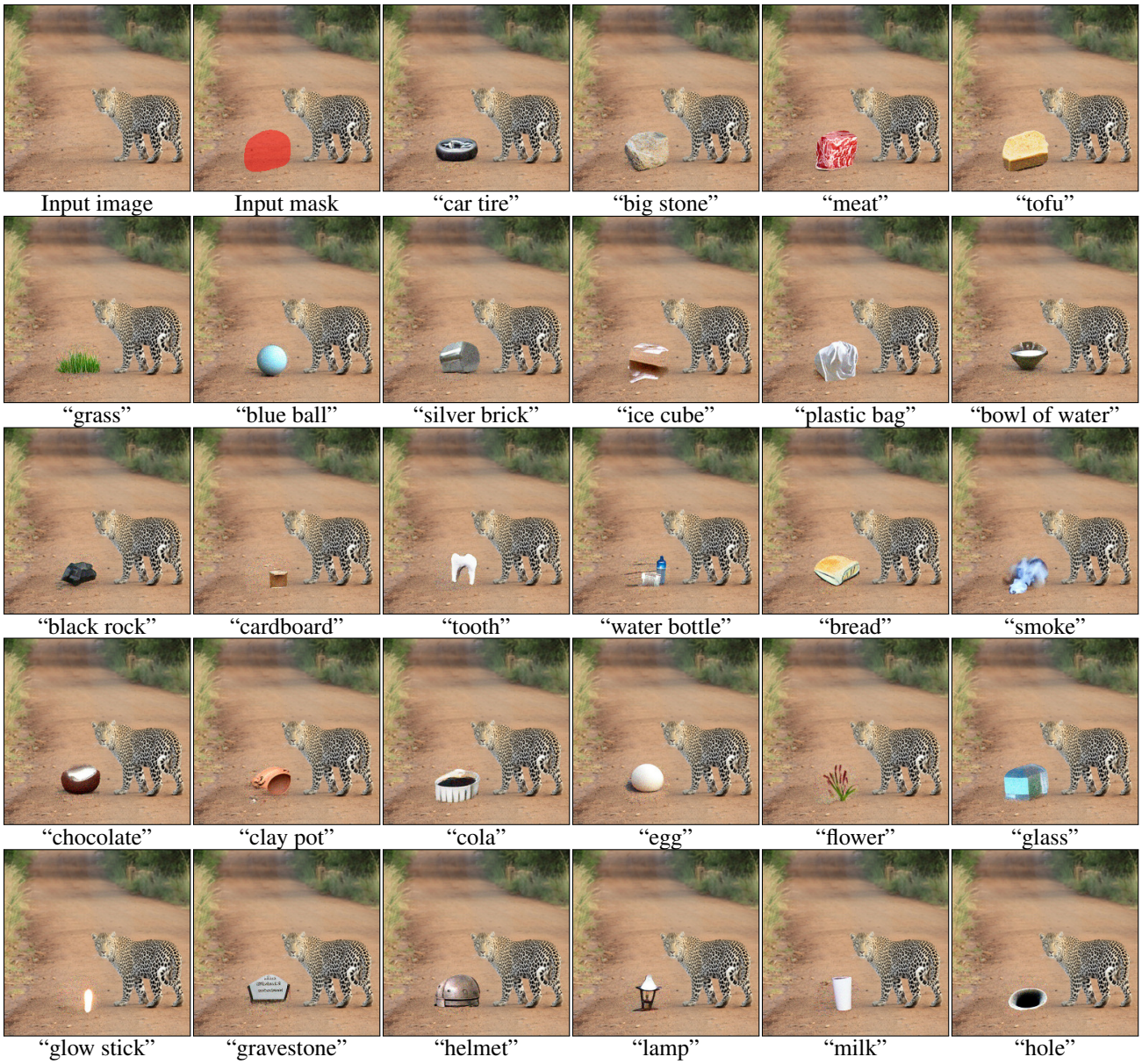


Figure 3. **Adding a new object (different prompts):** Given an input image and mask, our model is able to generate different objects corresponding to different text descriptions.

generated by Bau et al. [1]. All images were released freely under a Creative Commons license.

2.1. Hyperparamters

We used the CLIP model as-is, without changing any parameters. In addition, we did not utilize any prompt engineering techniques as described by Radford et al. [15].

We used the following hyperparameters in the guided-diffusion model across the different experiments (both in our model and in the baselines):

- **Fast sampling speed:** We follow the fast sampling speed from [10] which showed that 100 sampling steps are sufficient to achieve near-optimal FID score [9] on ImageNet [4]. This scheme reduces the sampling time to 27 seconds, for more details see Section 2.3.
- **Number of diffusion steps:** In most of our experiments we set the number of diffusion steps to $k = 75$, allowing the model to change the input image in a sufficient manner. Exceptions are scribble-based editing ($k = 60$) and background editing ($k = 67$).

In Algorithm 2 we use the following hyperparameters:

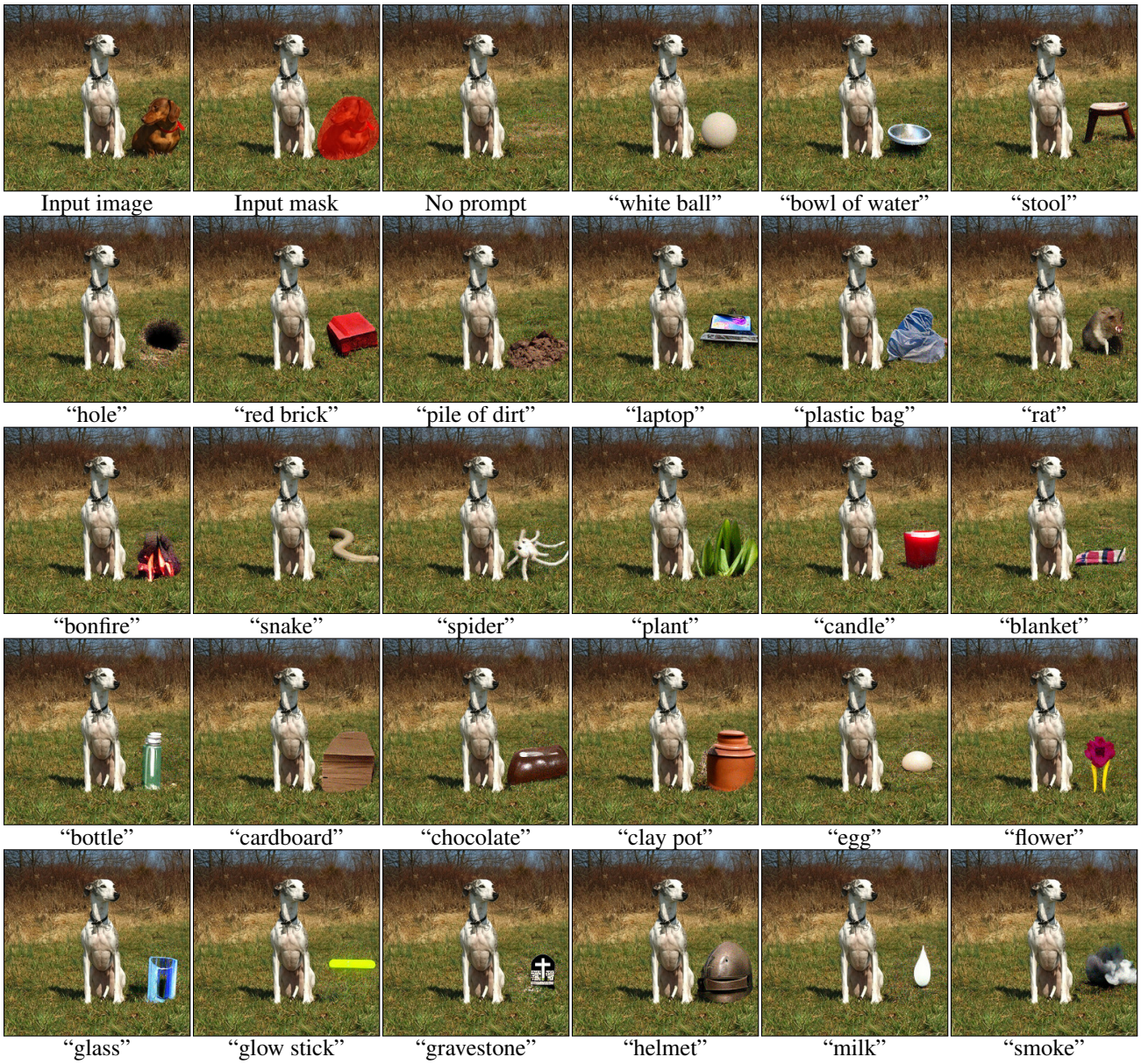


Figure 4. **Removing/replacing a foreground object:** Given an input image and a mask, we demonstrate inpainting of the masked region using different guiding texts. When no prompt is given, the result is similar to traditional image inpainting.

- **Number of extending augmentations:** We found that setting this to $N = 16$ was sufficient to mitigate the adversarial example phenomena.

- **Number of total repetitions:** As explained in Section 4.2.3, we generate several results and rank them using the CLIP model. In our experiments, we generate 64 samples and choose the best ones. For more details on inference time see Section 2.3.

2.2. Extending Augmentations

Given an input image x, in the resolution of the diffusion model (256×256 in our case), we first resize it to the input size of the CLIP model (224×224) along with its input mask. Next, we create N copies of this image and perform a different random projective transformation on each copy, along with the same transformation on the corresponding mask (see Figure 19). Finally, we calculate the gradients using the CLIP loss w.r.t each one of the transformed copies and average all the gradients. This way, an adversarial manipulation is much less likely, as it would have to “fool”

Figure 5. **Altering a part of an existing foreground object:** Given an input image and a mask, we aim to alter the foreground object corresponding to the guiding text "body of a standing dog". Multiple plausible results are generated, some more plausible than others. (The first two rows are better than the bottom two rows.)

Algorithm 1 DDIM blended diffusion: given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, and *CLIP* model

Input: source image x, target text description d, input mask m, diffusion steps k, number of extending augmentations N
Output: edited image $\widehat{x}$ that differs from input image x inside area m according to text description d
$x_k \sim \mathcal{N}(\sqrt{\bar{\alpha}_k}x_0, (1-\bar{\alpha}_k)\mathbf{I})$
for all t from k to 0 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $\widehat{x}_0 \leftarrow \frac{x_t}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t,t)}{\sqrt{\bar{\alpha}_t}}$
  $\widehat{x}_{0,aug} \leftarrow \mathit{ExtendingAugmentations}(\widehat{x}_0, N)$
  $\nabla_{text} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \nabla_{\widehat{x}_{0,aug}} \mathcal{D}_{CLIP}(\widehat{x}_{0,aug}, d, m)$
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1-\bar{\alpha}_t}\nabla_{text}$
  $x_{fg} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$
  $x_{bg} \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$
  $x_{t-1} \leftarrow x_{fg} \odot m + x_{bg} \odot (1-m)$
end for
return x_{-1}

CLIP under multiple transformations.

As mentioned in Section 5.2 we performed an ablation study for the extending augmentations. Figure 20 demonstrates the importance of the augmentations: the same random seed is used in two runs, one with and the other without augmentations. We can see that the images generated with the use of augmentations are more visually plausible and are more coherent than the ones generated without the augmentations. (This is an extended version of Figure 7 from the main paper.)

2.3. Inference Time

We report synthesis time for a single image using one NVIDIA A10 GPU:

- Our method (Algorithm 2) & Local CLIP-guided diffusion (Algorithm 1): 27 seconds.
- *PaintByWord++*: 78 seconds.

Figure 6. **Background replacement:** Given a source image and a mask of the background, the model is able to replace the background according to the text description. Note that the famous landmarks are not meant to accurately appear in the new background, but serve as an inspiration for the image completion.

Original paint by word [1] did not release their code and did not mention the run-time.

In practice, as described in Section 4.2.3, we generate several results for the same inputs and use the best ones. Instead of generating them sequentially, we accelerate the generation process using two techniques:

1. **Batch generation:** Instead of generating a single image in each diffusion pass, we multiplied the input several times and generated several instances on the same pass. Because of the stochasticity of the diffusion process, each result is different.
2. **Parallel generation:** Because each of the generation processes is independent, we can distribute the generation across multiple GPUs. In our experiments, we concurrently used 4 NVIDIA A10 GPUs.

Using the above accelerations, we generate 64 synthesis results in about 6 minutes — less than 6 seconds per image.

2.4. Comparison with Baselines

PaintByWord Because the models and code that was used by Bau et al. [1] are currently unavailable, we used as input the images and masks extracted from their paper.

Figure 7. **Background replacement:** Given a source image and a mask of the background, the model is able to replace the background corresponding to the text description. Note that the famous landmarks are not meant to accurately appear in the new background, but serve as an inspiration for the image completion.

PaintByWord++ We adapted the VQGAN+CLIP [3] implementation to support masks using the same $\mathcal{D}_{CLIP}$ loss from Equation (6). We used the VQGAN [7] model that was trained on ImageNet with reduction factor $f = 16$. For the latent optimization, we used the Adam optimizer with a learning rate of 0.1 for 500 steps. We found that constraining the optimization of the latent space z only to the corresponding mask area, the same way it was done by Bau et al. [1], improved the background preservation.

2.5. Implementation Details for Applications

In this section, we provide the implementation details for scribble-guided editing and text-guided image extrapolation applications.

2.5.1 Scribble-guided editing

In order to create the results that are demonstrated in Figure 9 of the main paper, the user first scribbles on the input image, then masks the scribble area (the masking can also be done automatically by taking the scribbles area and di-

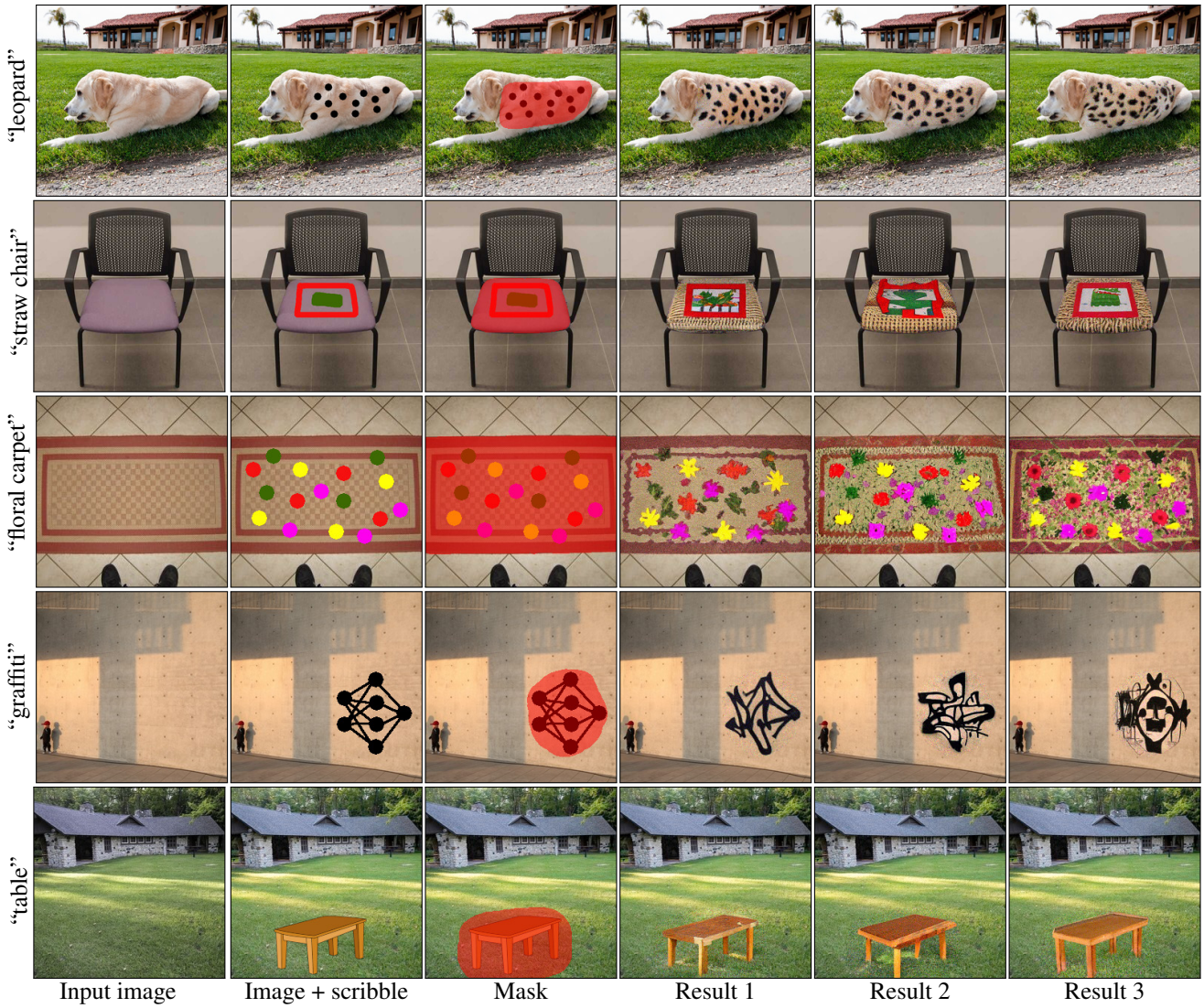


Figure 8. **Scribble-guided editing:** Users scribble a rough shape of the object they want to insert, mark the edited area, and provide a guiding text. The model uses the scribble as a general shape and color reference, transforming it to match the guiding text. Note that the scribble patterns can also change. In the last example, we embedded a clip art of a table instead of a manual scribble, it shows the effectiveness of our model to transform unnatural clip arts into real-looking objects.

lating it by morphological operations), then provides a text prompt and uses the same algorithm as for object altering.

An important hyper-parameter for this application is the number of target diffusion steps k in Algorithm 2. Figure 21 demonstrates the effect of changing this parameter: when diffusing for a longer period (e.g., 80 diffusion steps out of 100), only the main red color of the blanket is kept, the blanket shading is more realistic, and the results are more diverse. When diffusing for a shorter period (e.g., 20 diffusion steps out of 100), the scribble is hardly modified.

2.5.2 Text-guided image extrapolation

In order to extend the image beyond its original resolution, we gradually predict the unknown parts of the image in a sequential manner. Figure 22 demonstrates the building process: at each stage, (2) we translate the image $\frac{1}{4}$ to the opposite of the desired direction and fill the missing area using standard reflection padding, (4) then we inpaint the new area guided by the text description, using the regular algorithm for foreground editing. (5-7) We repeat the process 3 times until we have a new image. The new image is still a bit noisy — due of the gradual inpainting, each synthesis result is noisier than the previous one because of the chaining

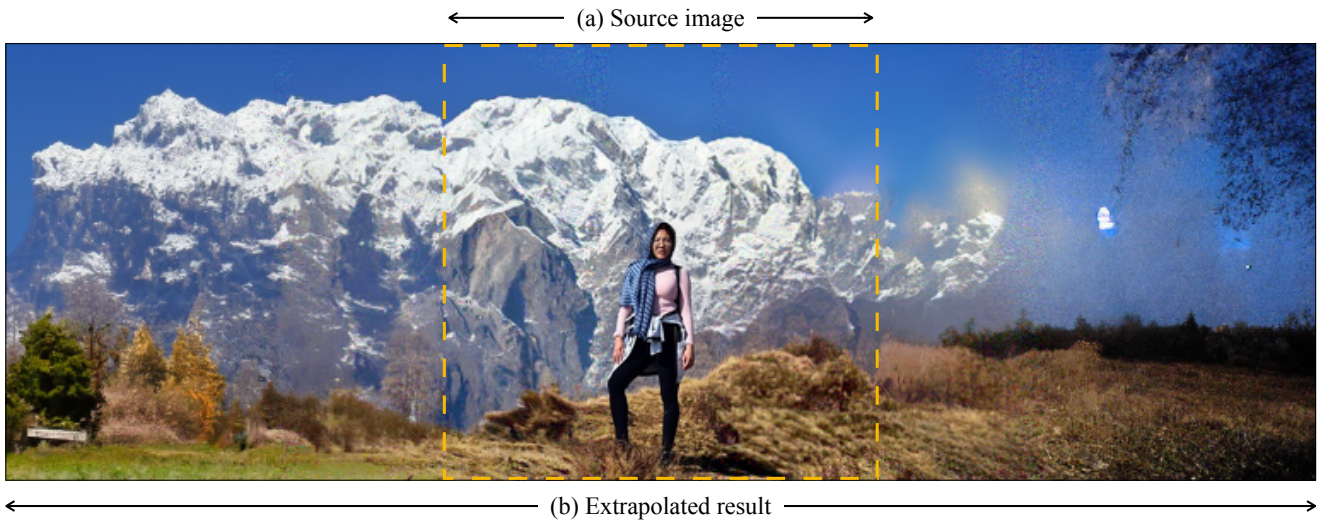


Figure 9. **Text-guided image extrapolation:** The user provides an image and two text descriptions that guide the extrapolation to the left ("sunny day" in this example) and to the right ("dark night").

Figure 10. **Result refinement:** The initial synthesis result of our model can be further refined. For example, here the user first masks a rough area in the source image and replaces the background using the prompt "New York City". Next, they wish to remove two unwanted objects from the generated result and to further refine the rough mask that was used in the first stage. They provide additional masks and no guiding text in this case (to perform inpainting) in order to obtain the final result.

of the natural image statistics. In order to mitigate it, (8) we denoise this image using the diffusion process again. We repeat the same process in the other direction. Our output can have an arbitrarily large image resolution.

We also notice that gradual diffusion steps are beneficial: we diffuse the first quarter for a small number of diffusion steps, and then in each step, we enlarge the number of diffusion steps.

2.6. Ranking Implementation Details

We utilized the ranking algorithm that is explained in Section 4.2.3 in the main paper using 64 synthesis results. As described in Section 6 in the main paper, the ranking is not perfect because it takes into account only the generated area. In addition, the ranking is not accurate enough in the resolution of single images: the top-ranked image isn't always better than the second one, etc. Nonetheless, the top 20% of the images are almost always better than the bottom 20%. In practice, we generate 64 results and choose manually from the top 10 images ordered by their ranking (in both the baselines and our method). Figure 23 demonstrates the effectiveness of the ranking algorithm.

3. User Study

In order to evaluate our model quantitatively, we conducted a user study. The only results of the Paint By Word model on general images (albeit GAN-generated) that were available are the ones in their paper. Hence, we chose to conduct the user study on these images (along with their corresponding masks). The study was conducted on 35 participants.

The participants were shown each time the inputs to the model (image, mask and text description) along with the model prediction, and were asked to rate the prediction, on a scale of 1–5, for one of the following criteria:

Figure 11. **Editing session mix example:** The user can use several editing operations consecutively. For example, as the first step, the user masks the hair of the person and provides the guiding text "curly blond hair". As the second step, the user masks the tie and provides the guiding text "shiny purple tie". At the last step, the user scribbles red dots on the jacket, masks the jacket, and provides the guiding text "floral jacket".

1. The overall realism of the prediction.
2. The amount of background preservation of the prediction in the unedited area.
3. The correspondence of the edited image to the guiding text description.

The questions were randomly ordered, and the participant had the ability to go back and edit their previous ratings until submission.

Mean user study scores are presented in Table 1 of the main paper. The difference between conditions is statistically significant (Kruskal-Wallis test, $p < 10^{-130}$). Further analysis using Tukey's honestly significant difference procedure [17] shows that the improvement shown by our method is statistically significant vs. all other conditions (Table 1).

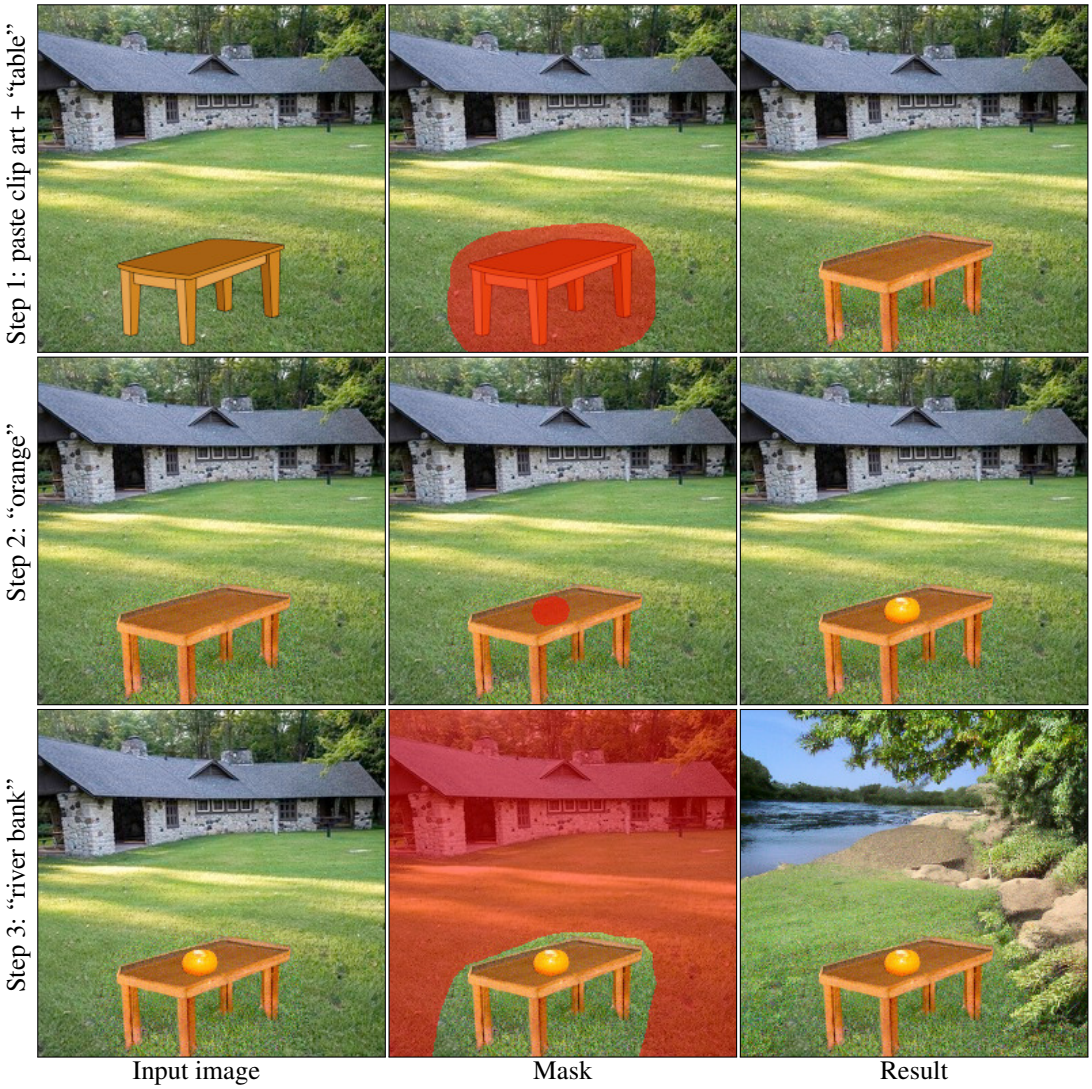


Figure 12. **Editing session mix example:** The user can use several editing operations consecutively. For example, here the user starts by pasting a clip art of a table on the image, then masks the relevant area and provides the guiding text "table" to get a more natural looking table. In the second stage, the user masks an area on the previous synthesis result and provides the guiding text "orange". In the last stage, the user masks the background of the previous synthesis result and provides the guiding text "river bank" to get the final synthesis result.

Method 1	**Method 2**	Realism p-value	Background preservation p-value	Text match p-value
Local CLIP GD [2]	Ours	**0.003**	**<0.001**	**<0.001**
Local CLIP GD [2]	*PaintByWord* [1]	0.435	0.578	**<0.001**
Local CLIP GD [2]	*PaintByWord++* [1, 3]	**<0.001**	0.106	**<0.001**
Ours	*PaintByWord* [1]	**<0.001**	**<0.001**	**<0.001**
Ours	*PaintByWord++* [1, 3]	**<0.001**	**<0.001**	**<0.001**
PaintByWord [1]	*PaintByWord++* [1, 3]	**<0.001**	0.719	0.704

Table 1. **User study statistical analysis:** We use Tukey's honestly significant difference procedure [17] to test whether the differences between mean scores in our user study are statistically significant. Significant results in bold. Our results are statistically better than all other methods on all the measured conditions.

Figure 13. **Editing session mix example:** The user can use several editing operations consecutively. As a first step, the user masks the chair and provides the guiding text “dresser”. Next, the user scribbles a rough shape of a lamp on the result of the previous step, masks the area of the lamp, and provides the guiding text “ceiling lamp”. Finally, the user masks an area over the wall in the previous result, and provides the guiding text “window” to obtain the final result.

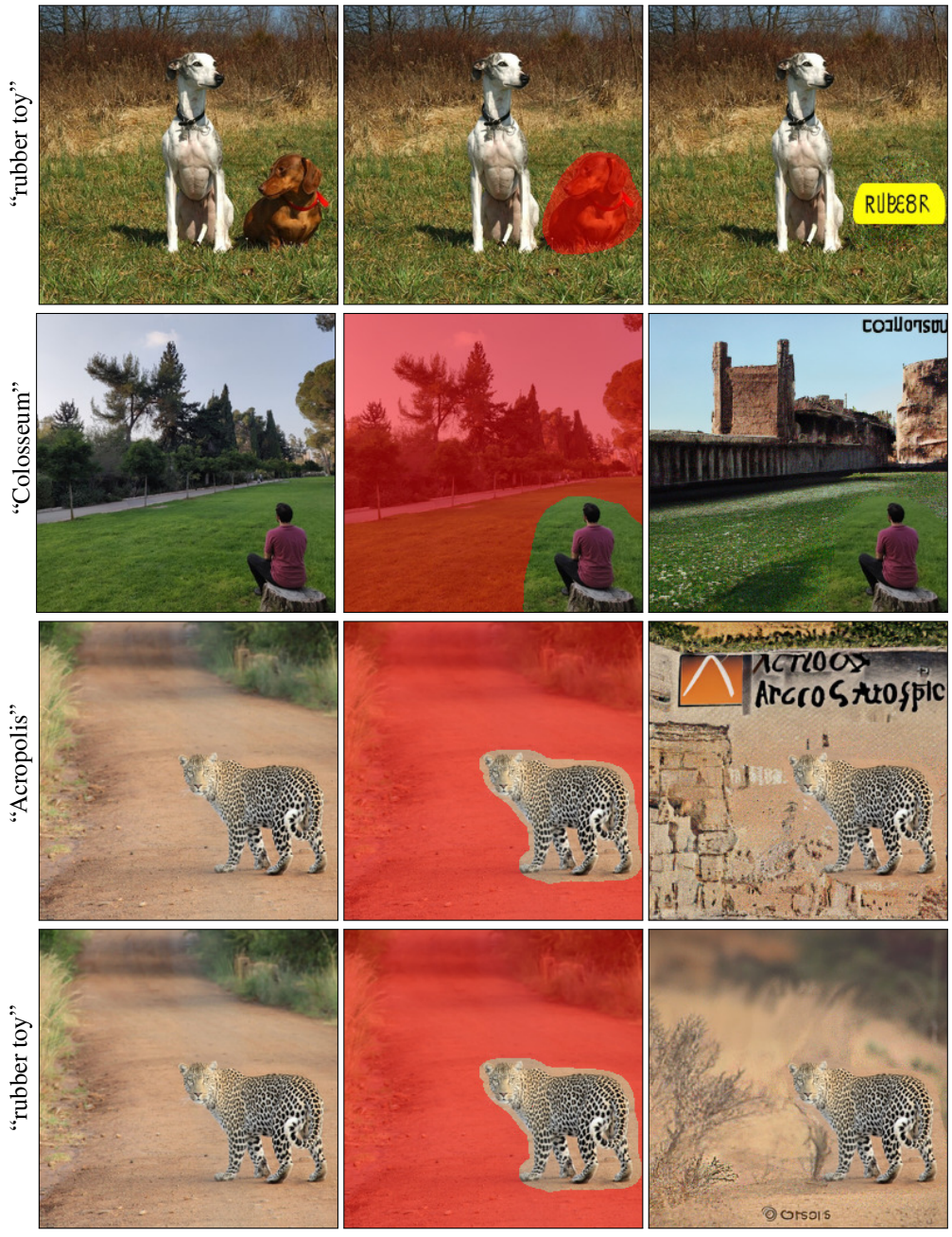
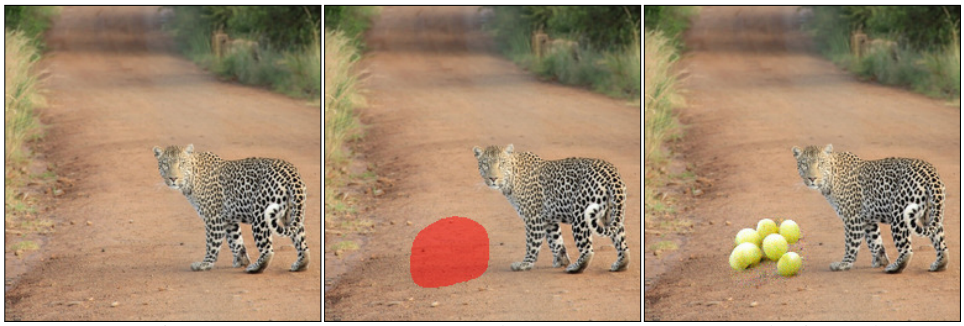


Figure 14. **Typographic failure:** Our model inherits CLIP [15] susceptibility to typographic attacks [8]. Instead of generating an object or a scene, the model might generate a textual description.



Input image Mask Synthesis result

Figure 15. **Out of proportion synthesis:** We show a failure case in which our method generates objects the look natural by themselves, but with the wrong proportion to the rest of the scene. For the guiding text "grapes", the synthesized result contains grapes which are huge compared to the leopard and to the rest of the scene.

Input image Input mask Initial prediction Naïve blend

Figure 16. **Naïve Blending:** When providing the model the input image and mask with the text prompt "a dog", and without using the background preservation loss — the result is a dog whose head is inside the mask, but most of the dog's body is outside the mask. Blending such a result with the input image using the input mask we obtain an unnatural result.

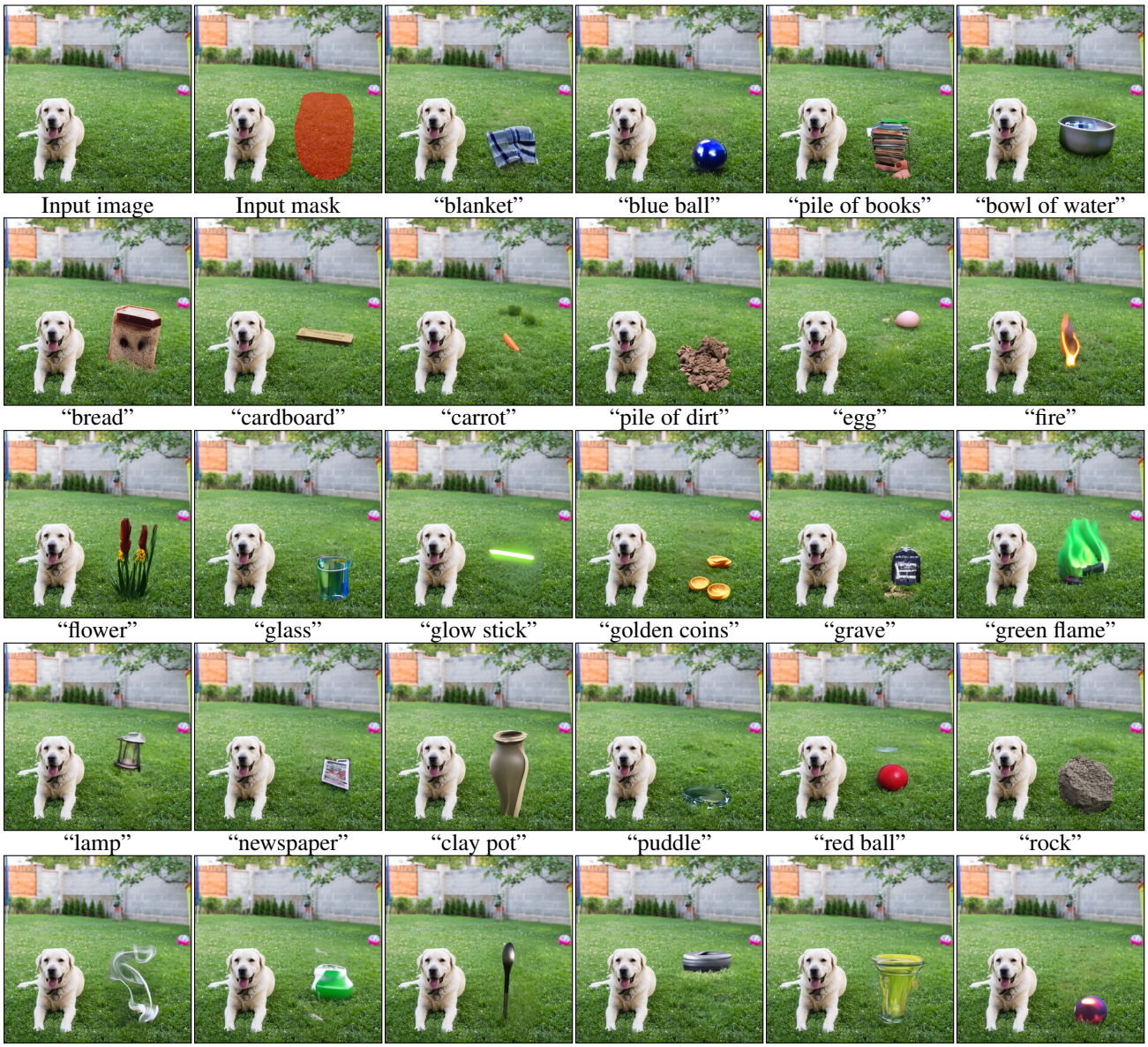


Figure 17. **High resolution results:** Given an input image of and mask, our model is able to generate different objects corresponding to different text descriptions. Results were produced using 512×512 DDPM model.

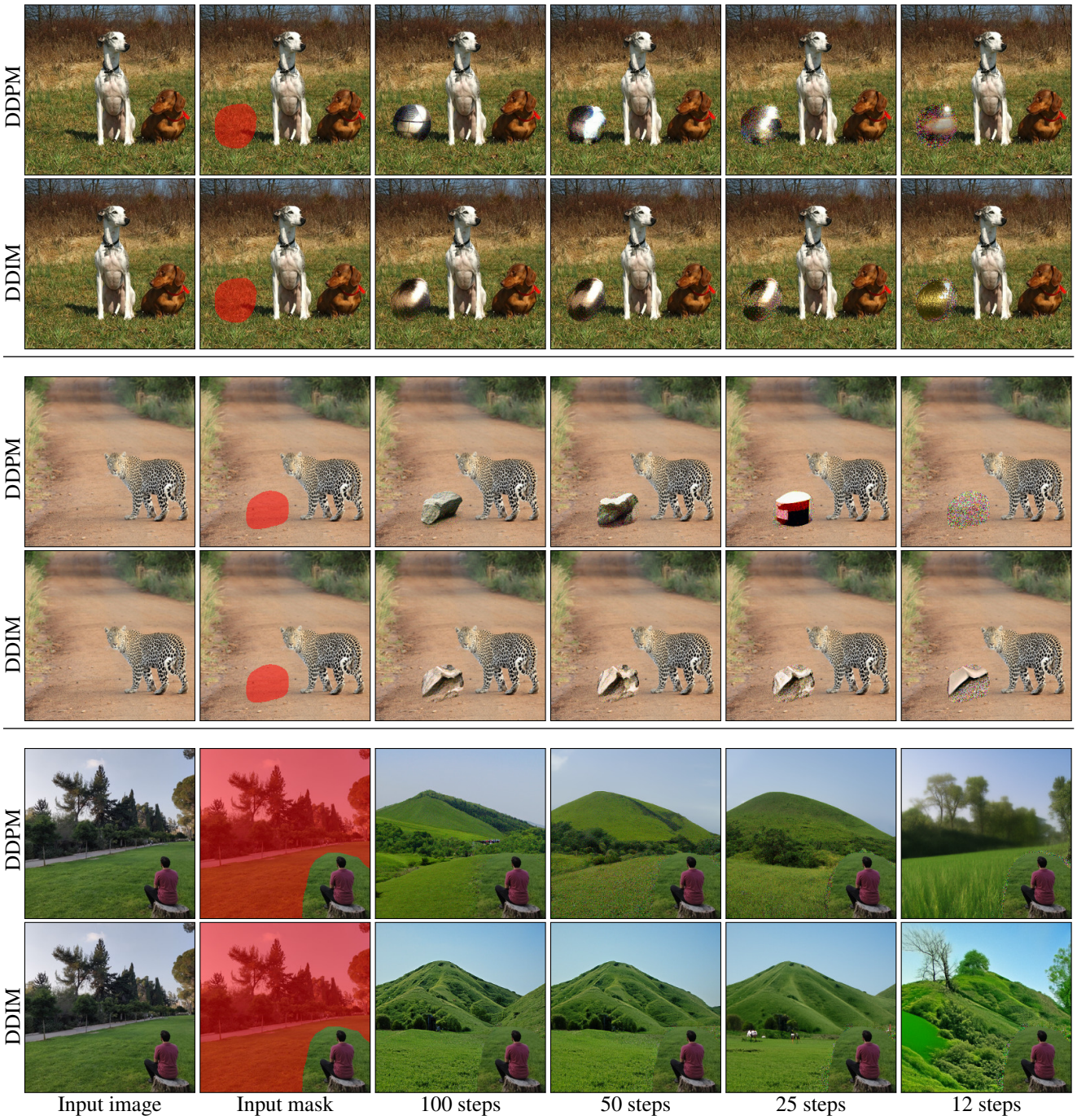


Figure 18. **Blended Diffusion DDPM VS Blended Diffusion DDIM comparison:** The part corresponds to the editing text "a shiny ball", the middle part to "a rock" and the bottom part to "green hills". As we can see, DDPM produces better results when using 100 diffusion steps, whereas it produces worse results in less than 50 diffusion steps.

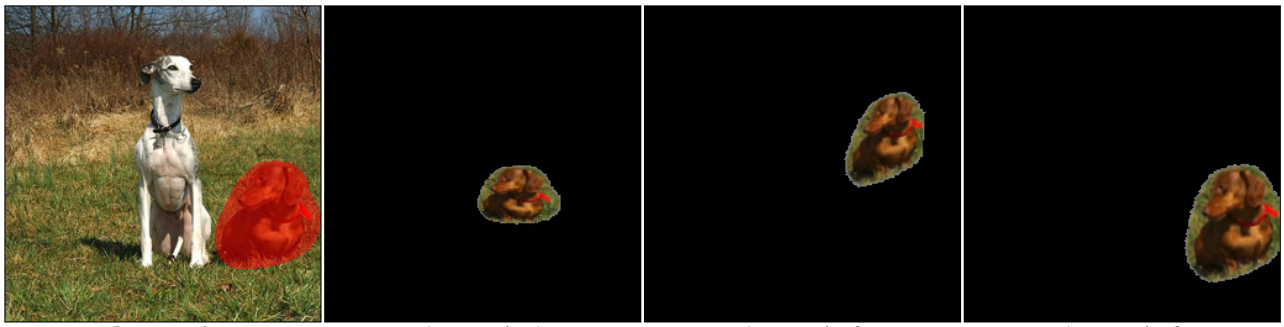


Figure 19. **Extending augmentation example:** Given an input image and mask, we augment the masked area in the image using various projective transformations.

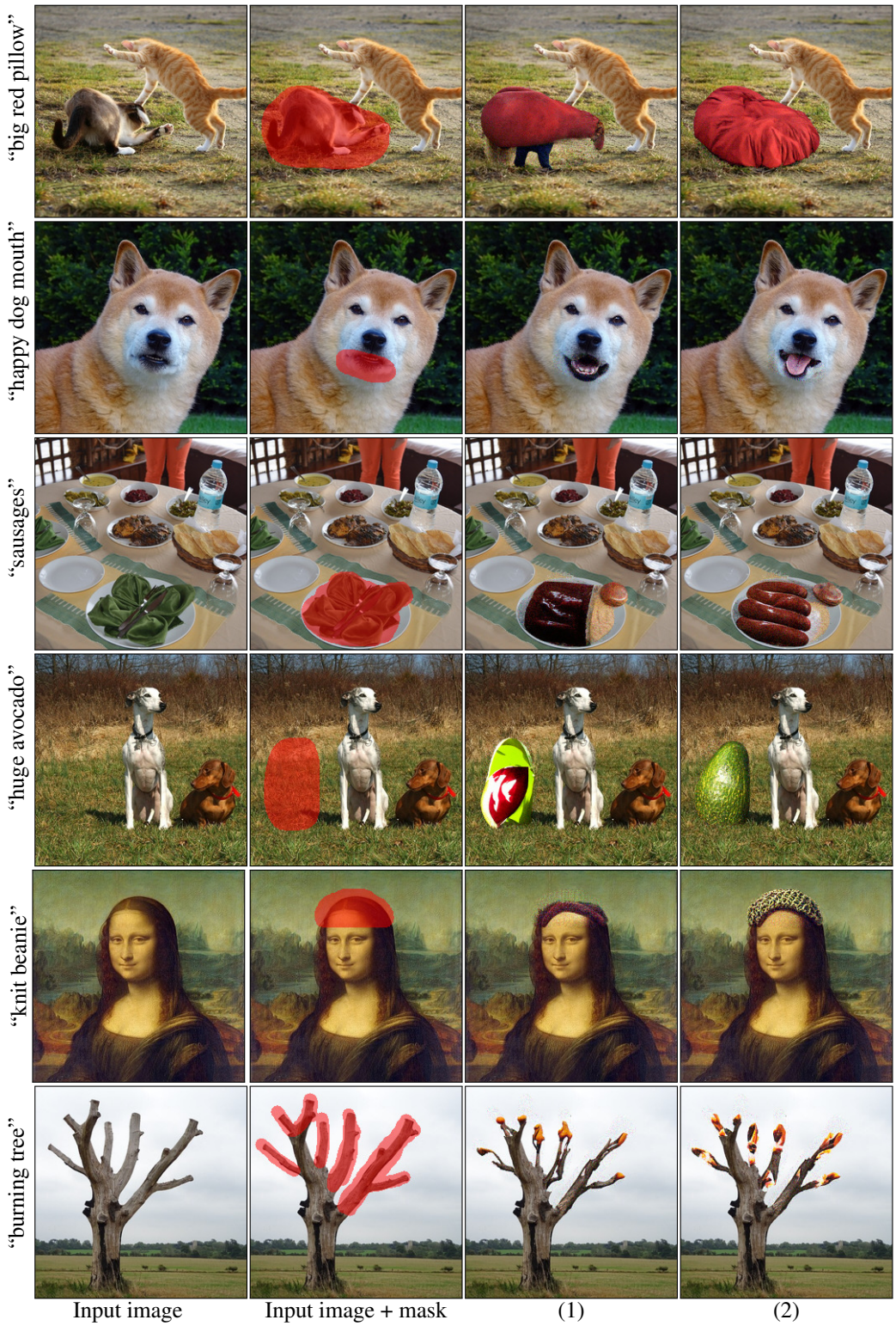


Figure 20. **Extending augmentations ablation:** In order to assess the importance of the extending augmentation technique, we used the same random seeds for the same inputs to ensure that the results would differ in the use of augmentations. As we can see, (2) using extending augmentations makes the resulting images more natural and coherent with the background in comparison to (1) not using extending augmentations.

Figure 21. **Scribble-guided editing diffusion steps effect:** when the diffusion steps are large (e.g. $k = 80$), the resulting images are more realistic and diverse but do not preserve the colors of the input scribble, on the other hand, when the diffusions steps are low (e.g. $k = 20$), the resulting images are almost identical to the input scribble.

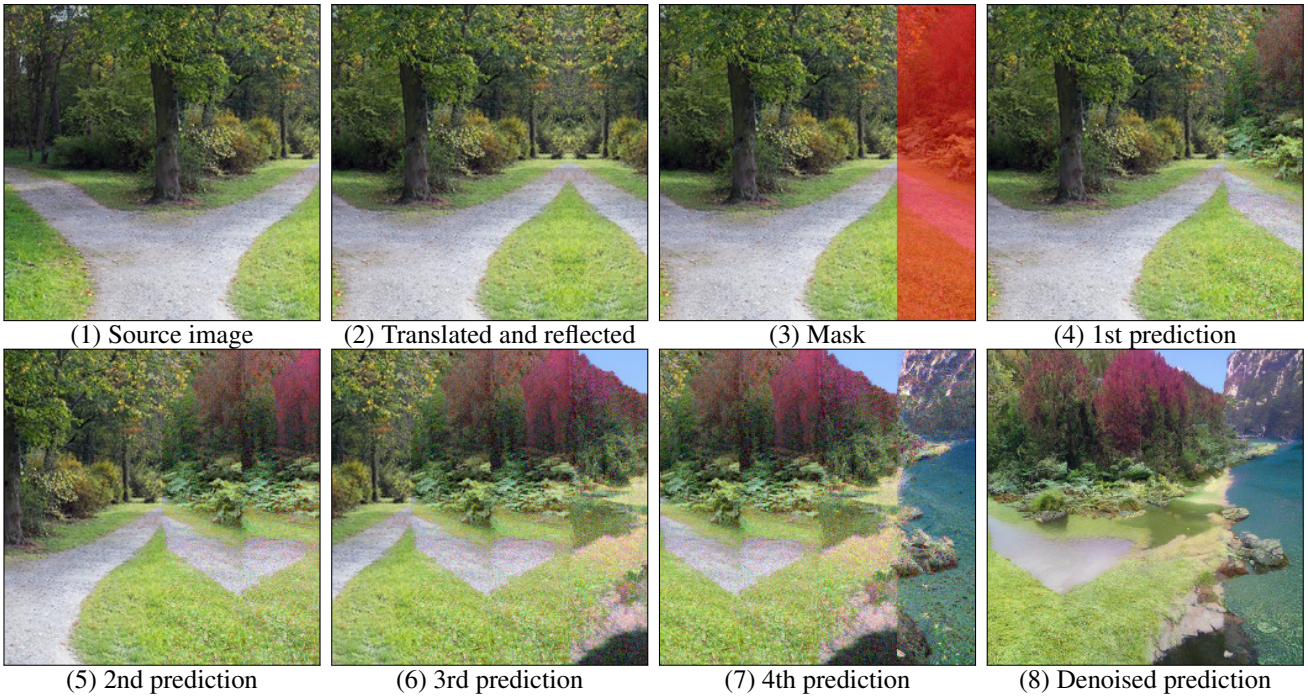


Figure 22. **Text-guided image extrapolation:** We aim to extrapolate the source image (1) to the right according to the guiding text "heaven". We start by (2) translating the image to the left by $\frac{1}{4}$ of the input resolution, and filling the missing area with reflection padding. Then we mask the new area (3) and predict the missing part (4) using the foreground altering algorithm. We perform this process 3 more times (5-7) to get a noisy prediction (7). In order to denoise it, we do the same process with a mask that covers the entire image and get the denoised result (8) that we can stitch to the source image. Notice that we can reach an arbitrary resolution using this method.

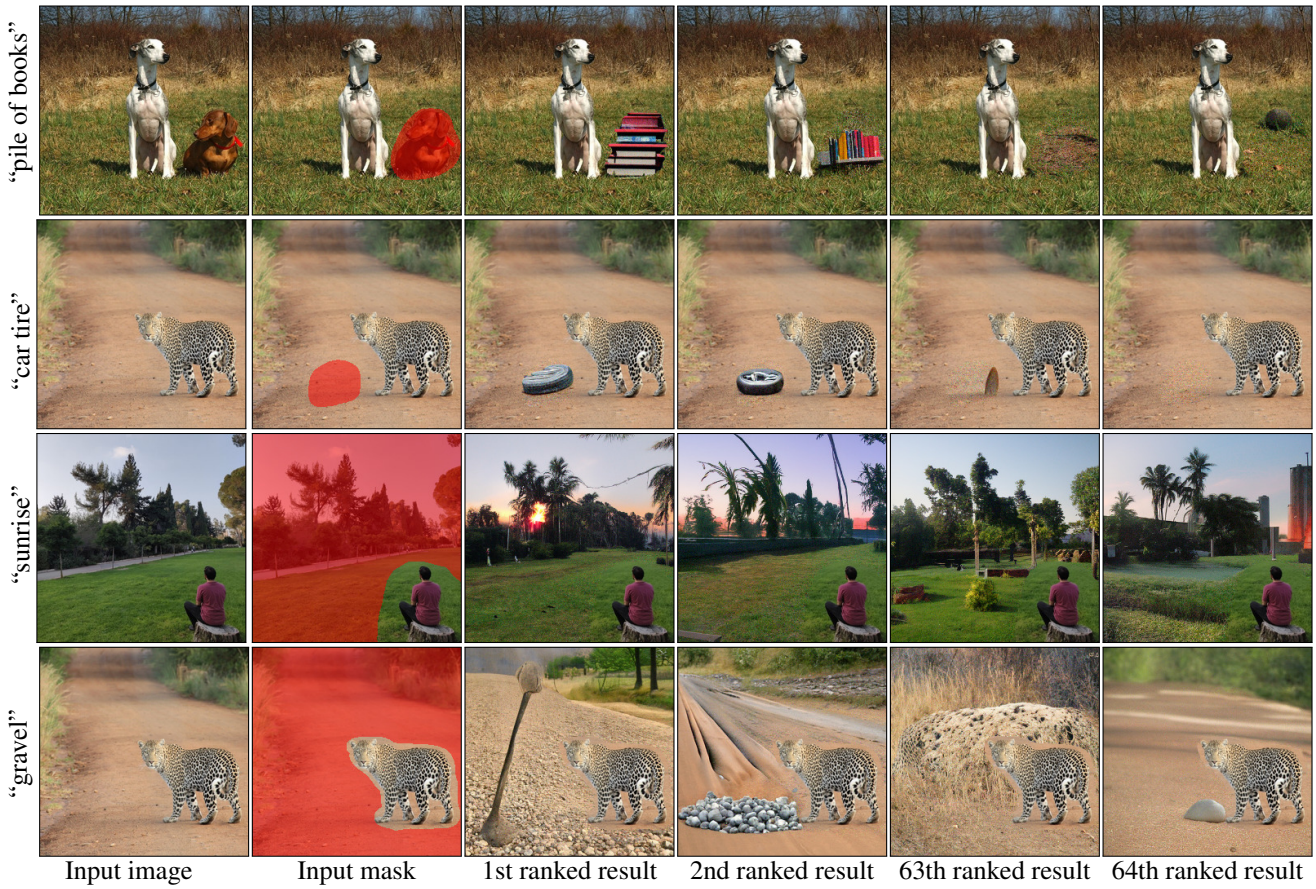


Figure 23. **Ranking algorithm effectiveness:** We generate 64 synthesis results and rank them using CLIP. We found that this method only roughly ranks the results: the top 20% are consistently better than the bottom 20%, but in the resolution of a single image, this is not the case — the first result isn't always better than the second one.

References

[1] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *arXiv preprint arXiv:2103.10951*, 2021. 3, 6, 7, 11

[2] Katherine Crowson. CLIP guided diffusion HQ 256x256. https://colab.research.google.com/drive/12a_Wrfi2_gwwAuN3VvMTwVMz9TfqctNj. 1, 11

[3] Katherine Crowson. VQGAN+CLIP. https://colab.research.google.com/drive/1L8oL-vLJXVcRzCFbPwOoMkPKJ8-aYdPN. 7, 11

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 2

[7] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 7

[8] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. https://distill.pub/2021/multimodal-neurons. 1, 13

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 3

[10] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2, 3

[11] OpenAI. CLIP Github. https://github.com/openai/CLIP. 2

[12] OpenAI. Guided Diffusion Github. https://github.com/openai/guided-diffusion. 1, 2

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 2

[14] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 1

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2, 3, 13

[16] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[17] John W Tukey. Comparing individual means in the analysis of variance. *Biometrics*, pages 99–114, 1949. 10, 11