A. Additional Results

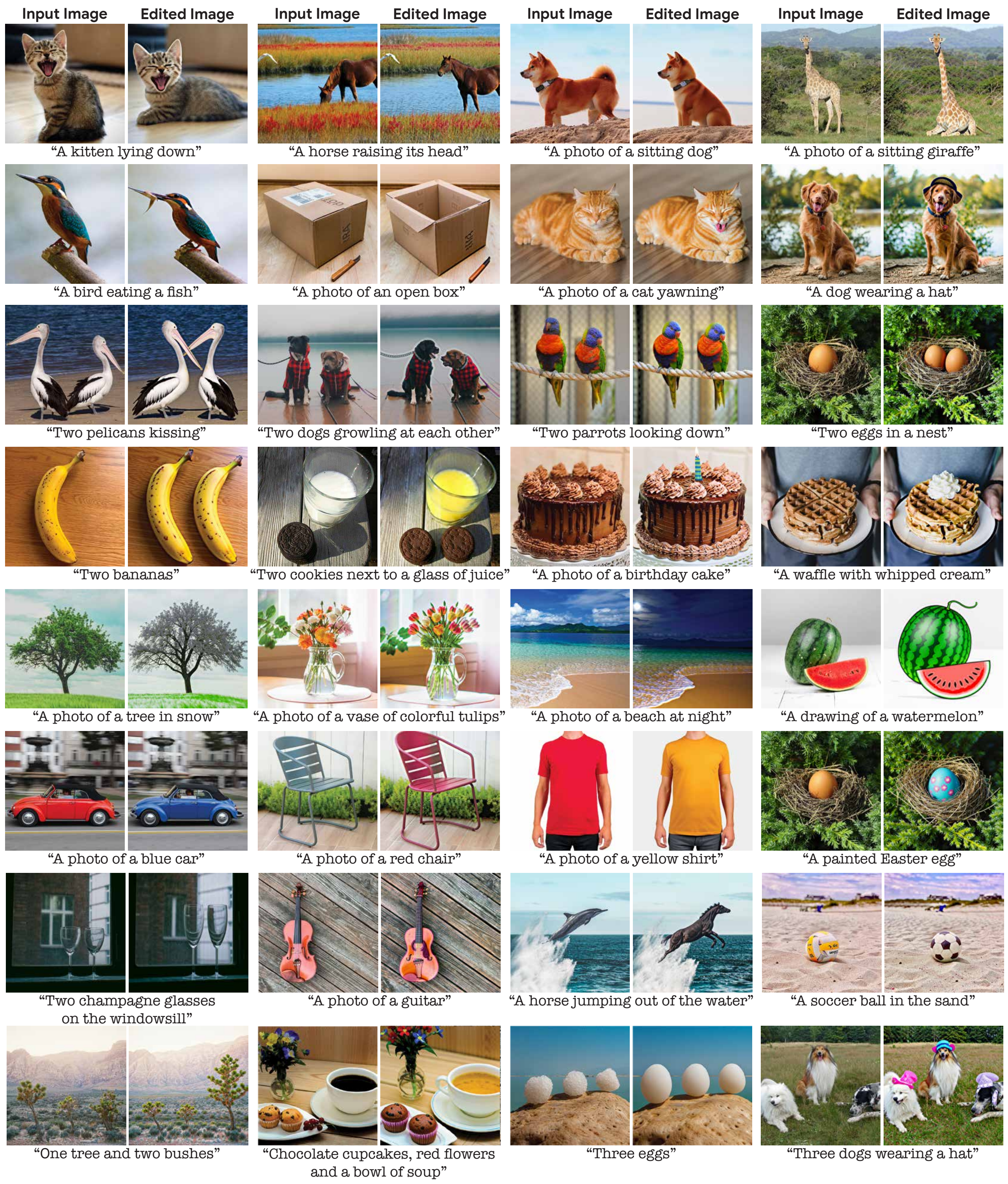

Input Image
Edited Image
Input Image
Edited Image
Input Image
Edited Image
Input Image
Edited Image
"A kitten lying down"
"A horse raising its head"
"A photo of a sitting dog"
"A photo of a sitting giraffe"
"A bird eating a fish"
"A photo of an open box"
"A photo of a cat yawning"
"A dog wearing a hat"
"Two pelicans kissing"
"Two dogs growling at each other"
"Two parrots looking down"
"Two eggs in a nest"
"Two bananas"
"Two cookies next to a glass of juice"
"A photo of a birthday cake"
"A waffle with whipped cream"
"A photo of a tree in snow"
"A photo of a vase of colorful tulips"
"A photo of a beach at night"
"A drawing of a watermelon"
"A photo of a blue car"
"A photo of a red chair"
"A photo of a yellow shirt"
"A painted Easter egg"
"Two champagne glasses on the windowsill"
"A photo of a guitar"
"A horse jumping out of the water"
"A soccer ball in the sand"
"One tree and two bushes"
"Chocolate cupcakes, red flowers and a bowl of soup"
"Three eggs"
"Three dogs wearing a hat"

Figure 11. **Wide range of editing types.** *Additional* 1024 × 1024*-pixel pairs of original (left) and edited (right) images using our method (with target texts). Editing types include posture changes, composition changes, multiple object editing, object additions, object replacements, style changes, and color changes.*

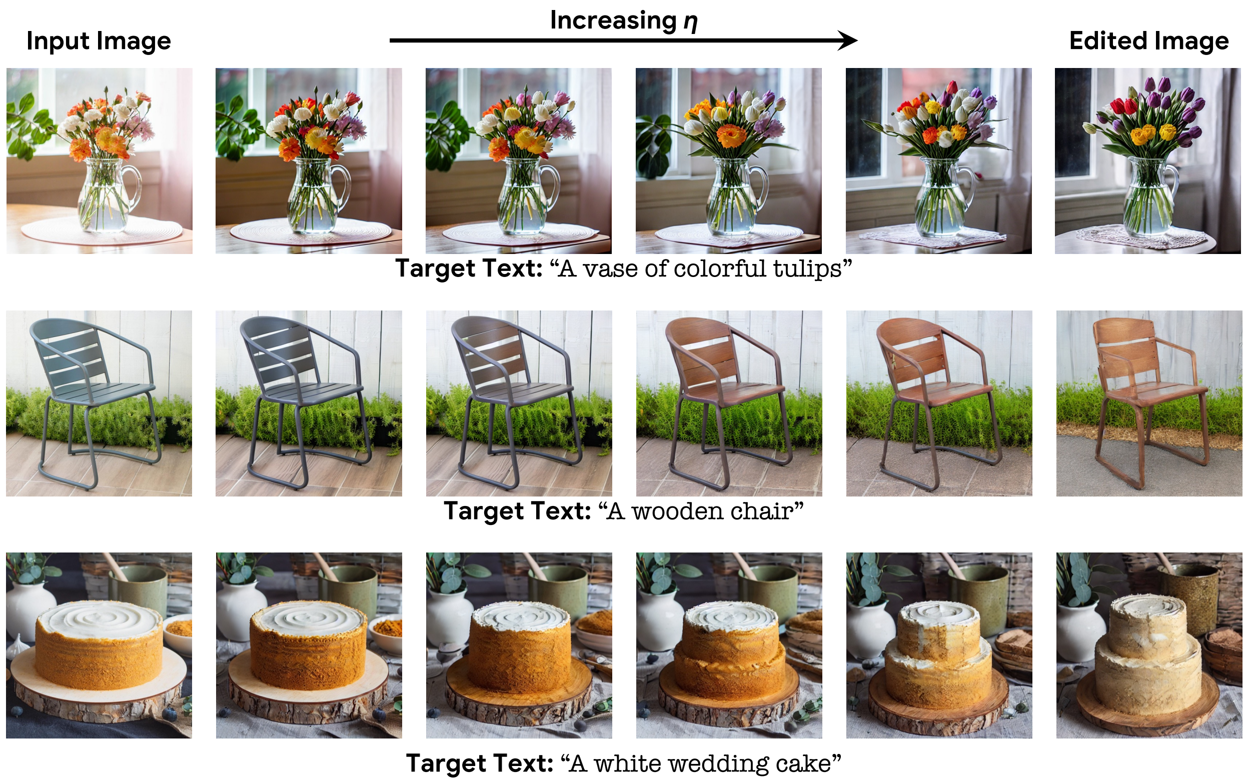

Input Image
Increasing η
Edited Image
Target Text: "A vase of colorful tulips"
Target Text: "A wooden chair"
Target Text: "A white wedding cake"

Figure 12. **Smooth interpolation.** *Additional results for smooth interpolation between the input image and the edited image using Imagic with Stable Diffusion (See animated GIFs in the supplementary material zip file).*

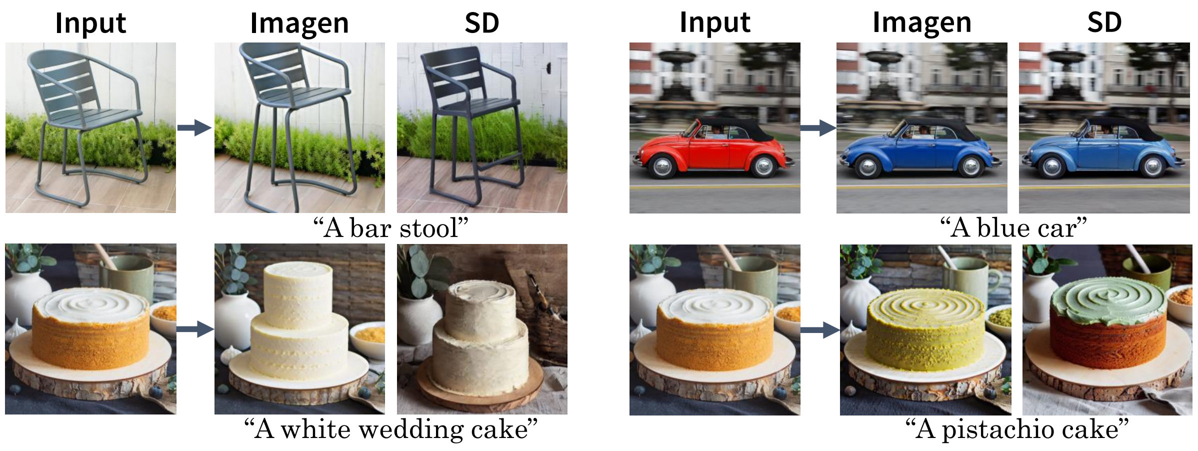


Figure 13. **Imagen vs Stable Diffusion.** *Imagic's formulation is agnostic to the diffusion model choice. We show multiple examples of the same requested edit applied with either Imagen or Stable Diffusion.*

B. Ablation Study

In the paper, we performed ablation studies on model fine-tuning and interpolation intensity. Here we present a discussion on the necessity of text embedding optimization, and additional ablation studies on the number of text embedding optimization steps and our method's sensitivity to varying random seeds.

Text embedding optimization Our method consists of three main stages: text embedding optimization, model fine-tuning, and interpolation. In the paper, we tested the value that the latter two stages add to our method. For the final two stages to

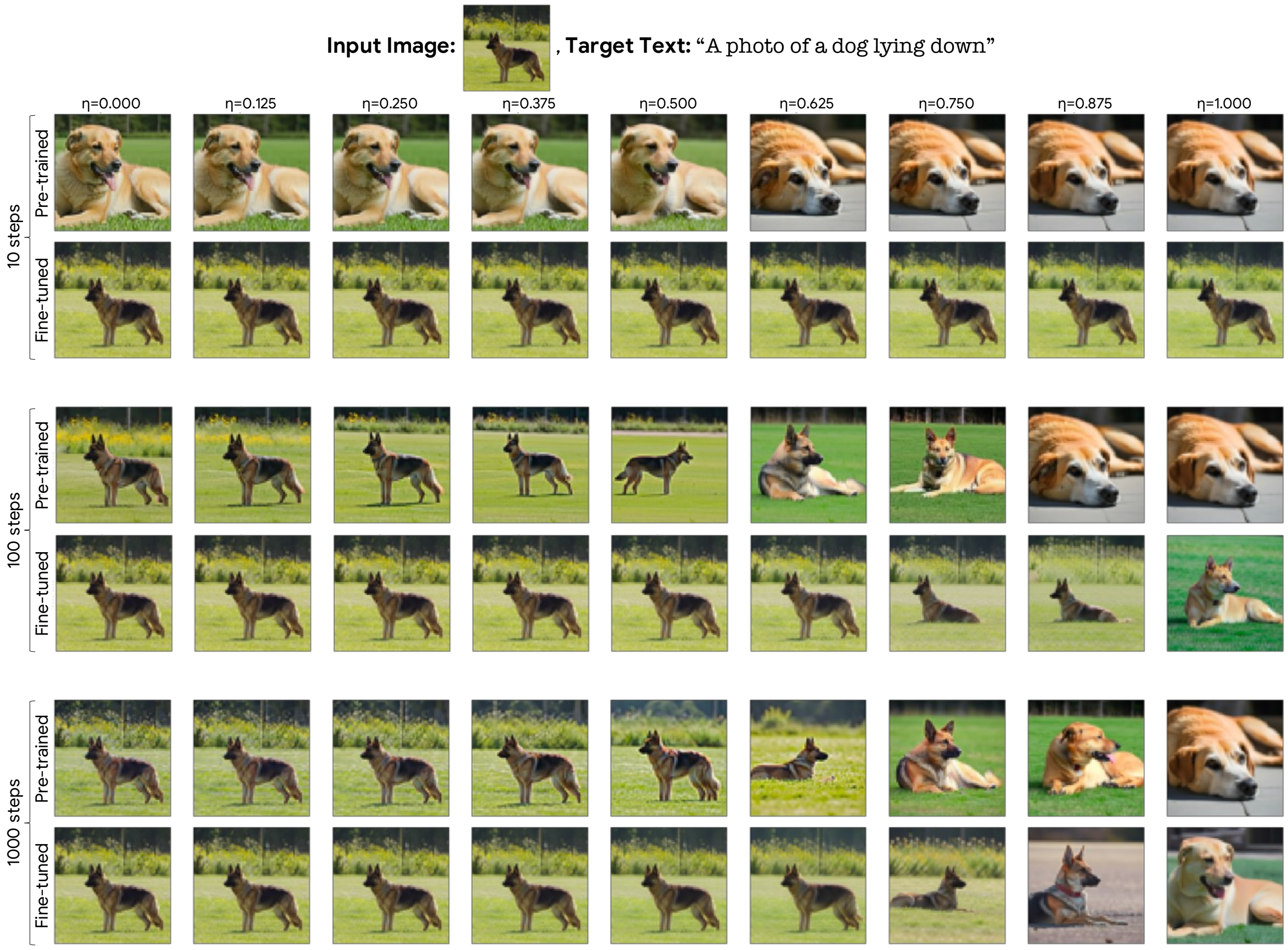


Figure 14. **Ablation for number of embedding optimization steps.** *Editing results for varying η and number of text embedding optimization steps, with and without fine-tuning (fixed seed).*

work well, the first one needs to provide two text embeddings to interpolate between: a "target" embedding and a "source" embedding. Naturally, one might be inclined to ask the user for both a target text describing the desired edit, and a source text describing the input image, which could theoretically replace the text embedding optimization stage. However, besides the additional required user input, this option may be rendered impractical, depending on the architecture of the text embedding model. For instance, Imagen [53] uses the T5 language model [46]. This model outputs a text embedding whose length depends on the number of tokens in the text, requiring the two embeddings to be of the same length to enable interpolation. It is highly impractical to request the user to provide that, especially since sentences may have a different number of tokens even if they have the same number of words (depending on the tokenizer used). Therefore, we opt not to test this option, and defer the pursuit of cleverer alternatives to future work. Moreover, this dependence on the number of tokens prevents optimizing the model once per image, and then editing it for any text prompt.

Number of text embedding optimization steps We evaluate the effect of the number of text embedding optimization steps on our editing results, both with and without model fine-tuning. We optimize the text embedding for 10, 100, and 1000 steps, then fine-tune the 64×64 diffusion model for 1500 steps separately on each optimized embedding. We fix the same random seed and assess the editing results for η ranging from 0 to 1. From the visual results in Figure 14, we observe that a 10-step optimization remains significantly close to the initial target text embedding, thereby retaining the same semantics in the pre-trained model, and imposing the reconstruction of the input image on the entire interpolation range in the fine-tuned model. Conversely, optimizing for 100 steps leads to an embedding that captures the basic essence of the input image, allowing for

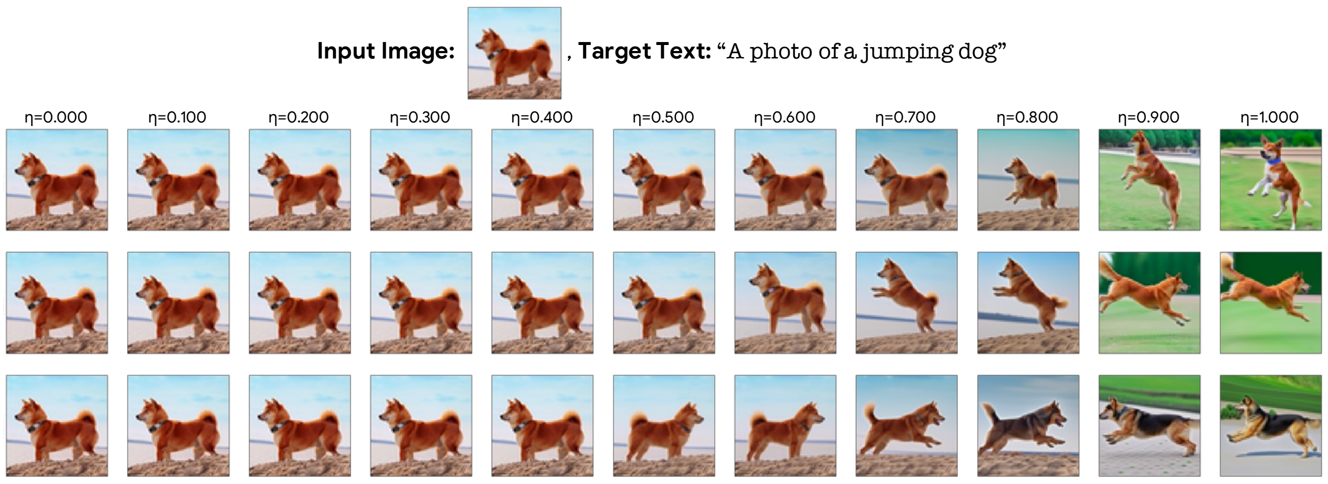


Figure 15. **Different seeds.** *Varying η values and different seeds produce different results for the same input.*

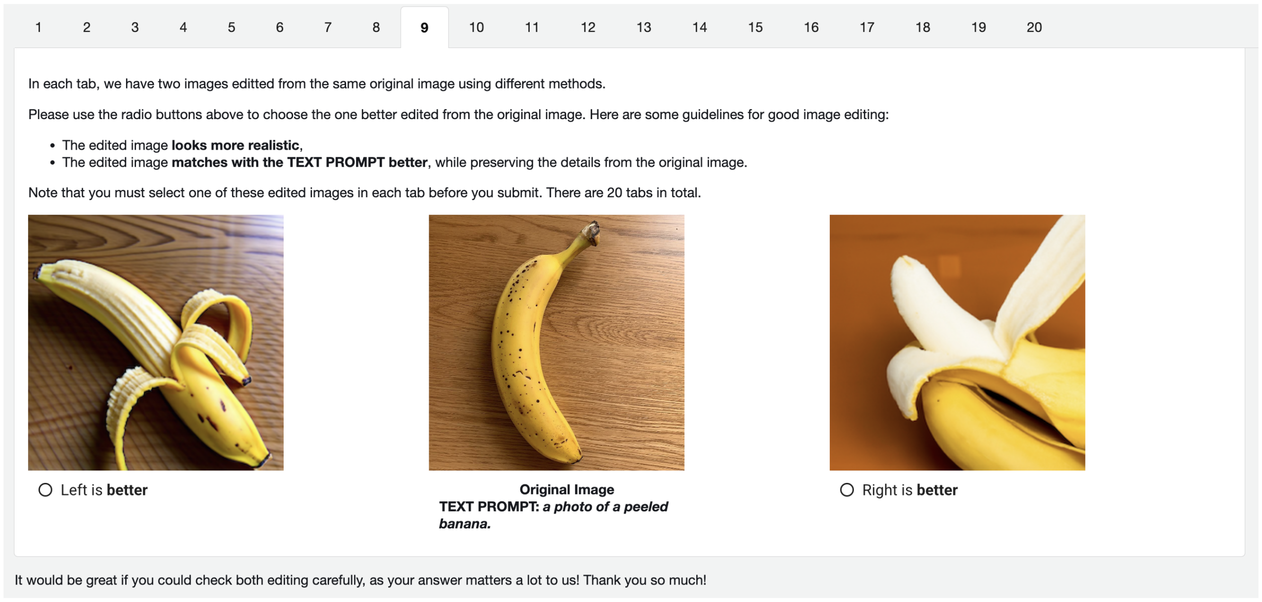
meaningful interpolation. However, the embedding does not completely recover the image, and thus the interpolation fails to apply the requested edit in the pre-trained model. Fine-tuning the model leads to an improved image reconstruction at $\eta = 0$, and enables the intermediate η values to match both the target text and the input image. Optimizing for 1000 steps enhances the pre-trained model performance slightly, but offers no discernible improvement after fine-tuning, sometimes even degrading it, in addition to incurring an added runtime cost. Therefore, we opt to apply our method using 100 text embedding optimization steps and 1500 model fine-tuning steps for all examples shown in the paper.

Different seeds Since our method utilizes a probabilistic generative model, different random seeds incur different results for the same input, as demonstrated in Figure 4. In Figure 15, we assess the effect of varying η values for different random seeds on the same input. We notice that different seeds incur viable edited images at different η thresholds, obtaining different results. For example, the first tested seed in Figure 15 first shows an edit at $\eta = 0.8$, whereas the second one does so at $\eta = 0.7$. As for the third one, the image undergoes a significant unwanted change (the dog looks to the right instead of left) at a lower η than when the edit is applied (the dog jumps). For some image-text inputs, we see behavior similar to the third seed in all of the 5 random seeds that we test. We consider these as failure cases and show some of them in Figure 10. Different target text prompts with similar meaning may circumvent these issues, since our optimization process is initialized with the target text embedding. We do not explore this option as it would compromise the intuitiveness of our method.

C. User Study Details

We perform an extensive human perceptual evaluation study with *TEdBench* (Textual Editing Benchmark), a novel benchmark containing 100 image-text input pairs for the complex non-rigid image editing task. The study was conducted using Amazon Mechanical Turk, to ensure unbiased evaluator opinions. For each evaluator, we show a randomly chosen subset of 20 images, including one image-text input pair that is shown twice. We discard all answers given by raters who answer the duplicate question differently, as they may not have paid close attention to the images. Human evaluators were shown an input image and a target text, and were asked to choose between two editing results: A random result from one of SDEdit [38], DDIB [62], or Text2LIVE [8], and our result, randomly ordered (left and right). Users were asked to choose between the left result and the right one, akin to the standard practice of Two-Alternative Forced Choice (2AFC) [8, 35, 42]. A sample screenshot of the screen shown to evaluators is provided in Figure 16. We collected 3030 answers for the comparison to SDEdit, 3131 for DDIB, and 3052 for Text2LIVE, totalling 9213 user answers.

For fairness, we apply SDEdit, Text2LIVE, and *Imagic* using a single fixed random seed, while DDIB is deterministic and thus unaffected by randomness. In *Imagic*, we choose the hyperparameter η that applies the desired edit while preserving a maximal amount of details from the original image. We choose SDEdit's intermediate diffusion timestep using the same goal. SDEdit was applied using the same Imagen model that we used, keeping its original hyperparameters. We also apply DDIB using Imagen, with a deterministic DDIM sampler, an encoder classifier-free guidance weight of 1, and a decoder classifier-free guidance weight ranging from 1 to 5 to control the editing intensity and choose the best result. Text2LIVE


1 2 3 4 5 6 7 8 **9** 10 11 12 13 14 15 16 17 18 19 20

In each tab, we have two images editted from the same original image using different methods.

Please use the radio buttons above to choose the one better edited from the original image. Here are some guidelines for good image editing:

- The edited image **looks more realistic**,
- The edited image **matches with the TEXT PROMPT better**, while preserving the details from the original image.

Note that you must select one of these edited images in each tab before you submit. There are 20 tabs in total.

○ Left is **better**

Original Image
TEXT PROMPT: *a photo of a peeled banana.*

○ Right is **better**

It would be great if you could check both editing carefully, as your answer matters a lot to us! Thank you so much!

Figure 16. **User study screenshot.** *An example screenshot of a question shown to participants in our human perceptual evaluation study.*

is applied using its default provided hyperparameters. Both DDIB and Text2LIVE had access to additional auxiliary texts describing the original image. The same hyperparameter settings were used in our qualitative comparisons as well. It is worth noting that SDEdit, DDIB, and Text2LIVE were all designed without complex non-rigid edits that preserve the remainder of the image in mind. *Imagic* is the first method to successfully target and apply such edits.

Our results show a strong user preference towards *Imagic*, with all comparisons to baselines showing a preference rate of more than 70%. We hope that *TEdBench* enables comparisons in text-based real image editing in the future, and serves as a benchmark evaluation set for future work on complex non-rigid image editing. To that end, we provide the full set of *TEdBench* images and target texts along with results for all the tested methods at the following URL: https://github.com/imagic-editing/imagic-editing.github.io/tree/main/tedbench/.

Acknowledgements

This work was done during an internship at Google Research. We thank William Chan, Chitwan Saharia, and Mohammad Norouzi for providing us with their support and access to the Imagen source code and pre-trained models. We also thank Michael Rubinstein and Nataniel Ruiz for insightful discussions during the development of this work.