Imagic: Text-Based Real Image Editing with Diffusion Models

Bahjat Kawar* [1,2] Shiran Zada* [1] Oran Lang[1] Omer Tov[1]
Huiwen Chang[1] Tali Dekel[1,3] Inbar Mosseri[1] Michal Irani[1,3]
[1]Google Research [2]Technion [3]Weizmann Institute of Science

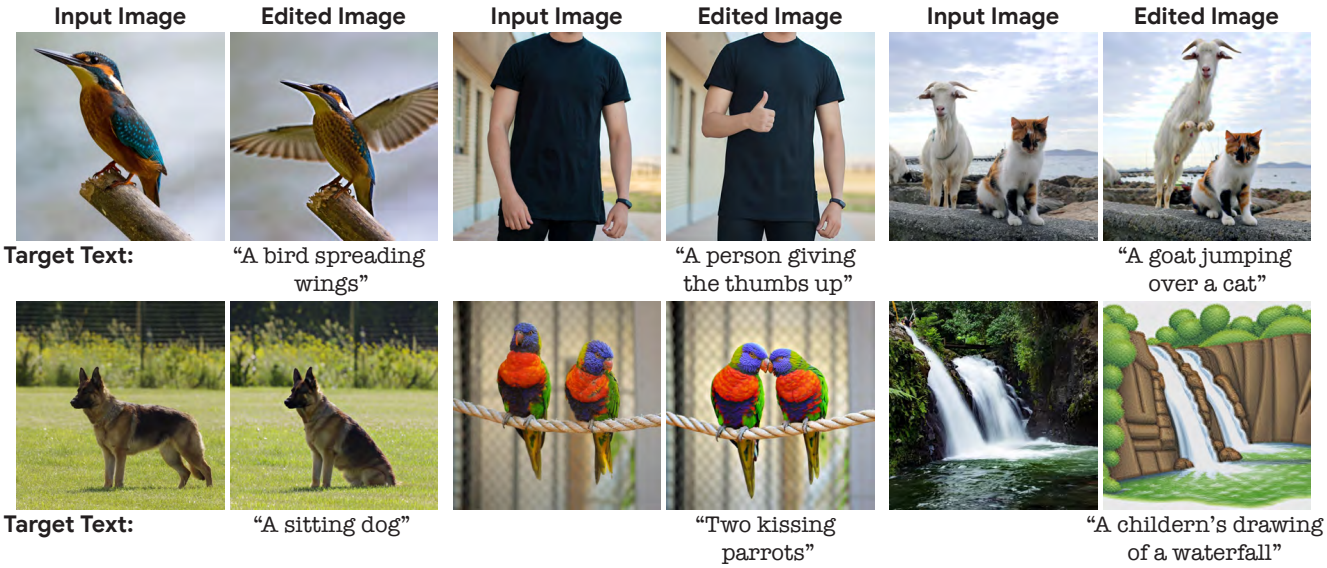

Figure 1. **"Imagic" – Editing a single real image.** Our method can perform various text-based semantic edits on a single real input image, including highly complex non-rigid changes such as posture changes and editing multiple objects. Here, we show pairs of 1024×1024 input (real) images, and edited outputs with their respective target texts.

Abstract

Text-conditioned image editing has recently attracted considerable interest. However, most methods are currently either limited to specific editing types (e.g., object overlay, style transfer), or apply to synthetically generated images, or require multiple input images of a common object. In this paper we demonstrate, for the very first time, the ability to apply complex (e.g., non-rigid) text-guided semantic edits to a single real image. For example, we can change the posture and composition of one or multiple objects inside an image, while preserving its original characteristics. Our method can make a standing dog sit down or jump, cause a bird to spread its wings, etc. – each within its single high-resolution natural image provided by the user. Contrary to previous work, our proposed method requires only a single input image and a target text (the desired edit). It operates on real images, and does not require any additional inputs (such as image masks or additional views of the object). Our method, which we call "Imagic", leverages a pre-trained text-to-image diffusion model for this task. It produces a text embedding that aligns with both the input image and the target text, while fine-tuning the diffusion model to capture the image-specific appearance. We demonstrate the quality and versatility of our method on numerous inputs from various domains, showcasing a plethora of high quality complex semantic image edits, all within a single unified framework.

1. Introduction

Applying non-trivial semantic edits to real photos has long been an interesting task in image processing [35]. It has attracted considerable interest in recent years, enabled by the considerable advancements of deep learning-based systems. Image editing becomes especially impres-

* Equal contribution.
The first author performed this work as an intern at Google Research.

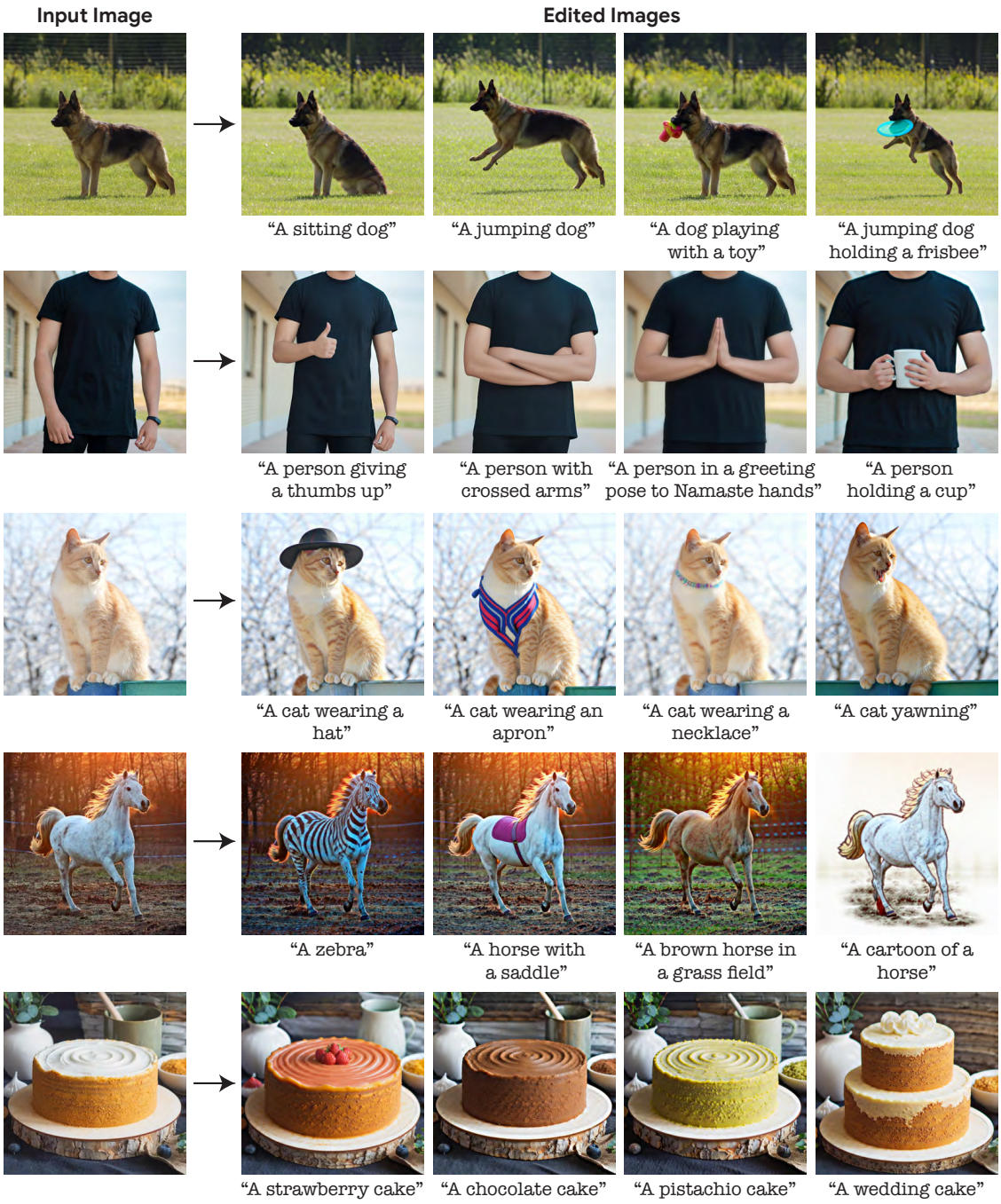

Figure 2. **Different target texts applied to the same image.** Imagic edits the same image differently depending on the input text.

sive when the desired edit is described by a simple natural language text prompt, since this aligns well with human communication. Many methods were developed for text-based image editing, showing promising results and continually improving [7, 9, 28]. However, the current leading methods suffer from, to varying degrees, several drawbacks: (i) they are limited to a specific set of edits such as painting over the image, adding an object, or transferring style [6, 28]; (ii) they can operate only on images from a specific domain or synthetically generated images [16, 36]; or (iii) they require auxiliary inputs in addition to the input image, such as image masks indicating the desired edit location, multiple images of the same subject, or a text describing the original image [6, 13, 40, 44].

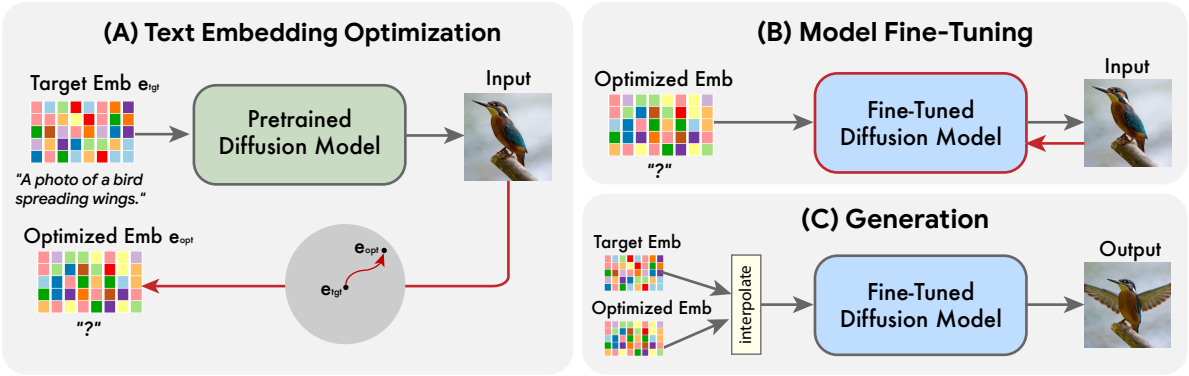

Figure 3. **Schematic description of Imagic.** Given a real image and a target text prompt, we encode the target text and get the initial text embedding $\mathbf{e}_{tgt}$, then optimize it to reconstruct the input image, obtaining $\mathbf{e}_{opt}$. We then fine-tune the generative model to improve fidelity to the input image while fixing $\mathbf{e}_{opt}$. Finally, we interpolate $\mathbf{e}_{opt}$ with $\mathbf{e}_{tgt}$ to generate the edit result.

In this paper, we propose a semantic image editing method that mitigates all the above problems. Given only an input image to be edited and a single text prompt describing the target edit, our method can perform sophisticated non-rigid edits on real high-resolution images. The resulting image outputs align well with the target text, while preserving the overall background, structure, and composition of the original image. For example, we can make two parrots kiss or make a person give the thumbs up, as demonstrated in Figure 1. Our method, which we call *Imagic*, provides the first demonstration of text-based semantic editing that applies such sophisticated manipulations to a single real high-resolution image, including editing multiple objects. In addition to these complex changes, Imagic can also perform a wide variety of edits, including style changes, color changes, and object additions.

To achieve this feat, we take advantage of the recent success of text-to-image diffusion models [40, 43, 46]. Diffusion models are powerful state-of-the-art generative models, capable of high quality image synthesis [12, 18]. When conditioned on natural language text prompts, they are able to generate images that align well with the requested text. We adapt them in our work to edit real images instead of synthesizing new ones. We do so in a simple 3-step process, as depicted in Figure 3: We first optimize a text embedding so that it results in images similar to the input image. Then, we fine-tune the pre-trained generative diffusion model, conditioned on the optimized embedding to better reconstruct the input image. Finally, we linearly interpolate between the target text embedding and the optimized one, resulting in a representation that combines both the input image and the target text. This representation is then passed to the generative diffusion process with the fine-tuned model, which outputs our final edited image.

To illustrate the prowess of Imagic, we conduct several experiments, applying our method on numerous images from various domains. Our method produces impressive results across all of our experiments, outputting high quality images that both resemble the input image to a high degree, and align well with the requested target text. These results showcase the generality, versatility, and quality of Imagic. We additionally conduct an ablation study, highlighting the effect of each element of our proposed method. When compared to recent and concurrent approaches suggested in the literature, Imagic exhibits significantly better editing quality and faithfulness to the original image, especially when tasked with highly sophisticated non-rigid edits.

We summarize our main contributions as follows:

1. We present the first text-based semantic image editing technique that allows for complex non-rigid edits on a single real input image, while preserving its overall structure and composition.

2. We apply a semantically meaningful linear interpolation between two text embedding sequences, uncovering strong compositional capabilities of text-to-image diffusion models.

3. We demonstrate our method on different editing types (*e.g.*, pose changes, multiple object editing) on images from various domains, showcasing the quality and versatility of our method.

2. Related Work

Following recent advancements in image synthesis quality [21–24], many works utilized the latent space of pre-trained generative adversarial networks (GANs) to perform a variety of image manipulations [3, 15, 30, 36, 49, 50]. Applying such manipulations on real images requires a latent space representation corresponding to each given image, such that feeding the representation to the generative

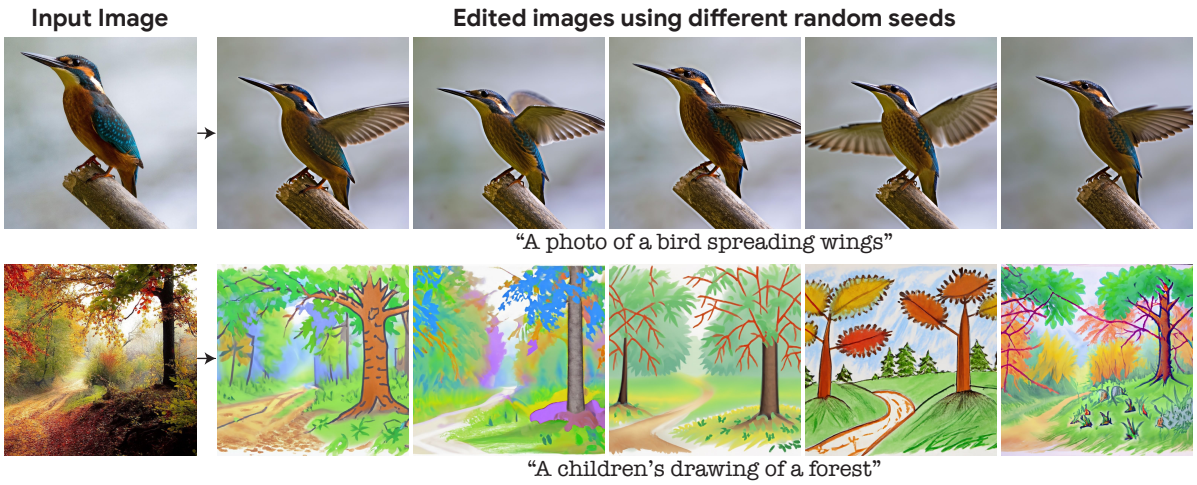

Figure 4. **Multiple edit options.** Imagic utilizes a probabilistic model, enabling it to generate multiple options with different random seeds.

model will result in an image similar to the input. This process of inverting a real image to the latent space is commonly referred to as "inversion", and is usually divided into optimization-based techniques [1,2] and encoder-based techniques [4,41,56]. To improve fidelity to the input image without deteriorating the editing quality, later works modify the generative model as well, creating a dedicated model for the given input [5, 8, 42]. In addition to GAN-based methods, some techniques utilize other deep learning-based systems for image editing [7, 11].

More recently, diffusion models were utilized for similar image manipulation tasks, showcasing remarkable results. SDEdit [32] adds intermediate noise to an image (possibly augmented by user-provided brush strokes), then denoises it using a diffusion process conditioned on the desired edit, which is limited to global edits. DiffusionCLIP [28] utilizes language-vision model gradients, DDIM inversion [52], and model fine-tuning to edit images using a domain-specific diffusion model. It was also suggested to edit images by synthesizing data in user-provided masks, while keeping the rest of the image intact [6, 33]. Liu et al. [31] guide a diffusion process with a text and an image, synthesising images similar to the given one, and aligned with the given text. Hertz et al. [16] alter a text-to-image diffusion process by manipulating cross-attention layers, providing more fine-grained control over generated images, and can edit real images in cases where DDIM inversion provides meaningful attention maps. Textual Inversion [13] and DreamBooth [44] synthesize novel views of a given subject given 3–5 images of the subject and a target text (rather than edit a single image), with DreamBooth requiring additional generated images for fine-tuning the models. In this work, we provide the first text-based semantic image editing tool that operates on a single real image, maintains high fidelity to it, and applies sophisticated non-rigid edits given a single free-form text prompt.

3. Imagic: Diffusion-Based Real Image Editing

3.1. Preliminaries

Diffusion models [18, 51, 53, 58] are a family of generative models that has recently gained traction, as they advanced the state-of-the-art in image generation [12, 26, 54, 57], and have been deployed in various downstream applications such as image restoration [25, 45], adversarial purification [10, 34], image compression [55], image classification [61], and others [14, 27, 37, 48, 59].

The core premise of these models is to initialize with a randomly sampled noise image $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, then iteratively refine it in a controlled fashion, until it is synthesized into a photorealistic image $\mathbf{x}_0$. Each intermediate sample $\mathbf{x}_t$ (for $t \in \{0, \ldots, T\}$) satisfies

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t, \tag{1}$$

with $0 = \alpha_T < \alpha_{T-1} < \cdots < \alpha_1 < \alpha_0 = 1$ being hyperparameters of the diffusion schedule, and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$. Each refinement step consists of an application of a neural network $f_\theta(\mathbf{x}_t, t)$ on the current sample $\mathbf{x}_t$, followed by a random Gaussian noise perturbation, obtaining $\mathbf{x}_{t-1}$. The network is trained for a simple denoising objective, aiming for $f_\theta(\mathbf{x}_t, t) \approx \boldsymbol{\epsilon}_t$ [18, 51]. This leads to a learned image distribution with high fidelity to the target distribution, enabling stellar generative performance.

This method can be generalized for learning conditional distributions – by augmenting the denoising network with an auxiliary input $\mathbf{y}$, the network $f_\theta(\mathbf{x}_t, t, \mathbf{y})$ and its resulting diffusion process can faithfully sample from a data distribution conditioned on $\mathbf{y}$. The conditioning input $\mathbf{y}$ can be a low-resolution version of the desired image [47] or a class

label [19]. Furthermore, $\mathbf{y}$ can also be on a text sequence describing the desired image [40, 43, 46]. By incorporating knowledge from large language models (LLMs) [39] or hybrid vision-language models [38], these *text-to-image diffusion models* have unlocked a new capability – users can generate realistic high-resolution images using only a text prompt describing the desired scene. In all these methods, a low-resolution image is first synthesized using a generative diffusion process, and then it is transformed into a high-resolution one using additional auxiliary models.

3.2. Our Method

Given an input image $\mathbf{x}$ and a target text which describes the desired edit, our goal is to edit the image in a way that satisfies the given text, while preserving a maximal amount of detail from $\mathbf{x}$ (*e.g.*, small details in the background and the identity of the object within the image). To achieve this feat, we utilize the text embedding layer of the diffusion model to perform semantic manipulations. Similar to GAN-based approaches [36, 42, 56], we begin by finding meaningful representation which, when fed through the generative process, yields images similar to the input image. We then tune the generative model to better reconstruct the input image and finally manipulate the latent representation to obtain the edit result.

More formally, as depicted in Figure 3, our method consists of 3 stages: (i) we optimize the text embedding to find one that best matches the given image in the vicinity of the target text embedding; (ii) we fine-tune the diffusion models to better match the given image; and (iii) we linearly interpolate between the optimized embedding and the target text embedding, in order to find a point that achieves both image fidelity and target text alignment. We now turn to describe each step in more detail.

Text embedding optimization. The target text is first passed through a text encoder [39], which outputs its corresponding text embedding $\mathbf{e}_{tgt} \in \mathbb{R}^{T\times d}$, where T is the number of tokens in the given target text, and d is the token embedding dimension. Then, we freeze the parameters of the generative diffusion model f_θ, and optimize the target text embedding $\mathbf{e}_{tgt}$ using the denoising diffusion objective [18],

$$\mathcal{L}(\mathbf{x}, \mathbf{e}, \theta) = \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - f_\theta(\mathbf{x}_t, t, \mathbf{e})\|_2^2\right], \tag{2}$$

where $\mathbf{x}$ is the input image, $t \sim Uniform[1, T]$, $\mathbf{x}_t$ is a noisy version of $\mathbf{x}$ obtained using $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and Equation 1, and θ are the pre-trained diffusion model weights. This results in a text embedding that matches our input image as closely as possible. We run this process for relatively few steps, in order to remain close to the initial target text embedding, obtaining $\mathbf{e}_{opt}$. This proximity enables meaningful linear interpolation in the embedding space, which does not exhibit linear behavior when the embeddings are more distant.

Model fine-tuning. Note that the obtained optimized embedding $\mathbf{e}_{opt}$ does not necessarily lead to the input image $\mathbf{x}$ exactly when passed through the generative diffusion process, as our optimization runs for a small number of steps (see top left image in Figure 5). Therefore, in the second stage of our method, we close this gap by optimizing the model parameters θ using the same loss function presented in Equation 2, while freezing the optimized embedding. This process shifts the model to fit the input image $\mathbf{x}$ at the point $\mathbf{e}_{opt}$. In parallel, we optimize any auxiliary diffusion models present in the underlying generative method, such as super-resolution models. We optimize them with the same reconstruction loss, but conditioned on the target text embedding. The optimization of these auxiliary models ensures the preservation of high-frequency details from $\mathbf{x}$ that are not present in the base resolution.

Text embedding interpolation. Since the generative diffusion model was trained to fully recreate the input image $\mathbf{x}$ at the optimized embedding $\mathbf{e}_{opt}$, we use it to apply the desired edit by advancing in the direction of the target text embedding $\mathbf{e}_{tgt}$. More formally, our third stage is a simple linear interpolation between $\mathbf{e}_{tgt}$ and $\mathbf{e}_{opt}$. For a given hyperparameter $\eta \in [0, 1]$, we obtain

$$\bar{\mathbf{e}} = \eta \cdot \mathbf{e}_{tgt} + (1 - \eta) \cdot \mathbf{e}_{opt}, \tag{3}$$

which is the embedding that represents the desired edited image. We then apply the base generative diffusion process using the fine-tuned model, conditioned on $\bar{\mathbf{e}}$. This results in a low-resolution edited image, which is then super-resolved using the fine-tuned auxiliary models, conditioned on the target text. This generative process outputs our final high-resolution edited image $\bar{\mathbf{x}}$.

3.3. Implementation Details

While our framework is general, our current implementation is based on Imagen [46], a state-of-the-art text-to-image generative model. Imagen consists of three separate text-conditioned diffusion models: (i) a generative diffusion model for 64×64-pixel images; (ii) a super-resolution (SR) diffusion model turning 64×64-pixel images into 256×256 ones; and (iii) another SR model transforming 256×256-pixel images into the 1024×1024 resolution. By cascading these three models [19] and using classifier-free guidance [20], Imagen constitutes a powerful text-guided image generation scheme.

We optimize the text embedding using the 64×64 diffusion model and the Adam [29] optimizer for 100 steps and a fixed learning rate of 0.001. We then fine-tune the

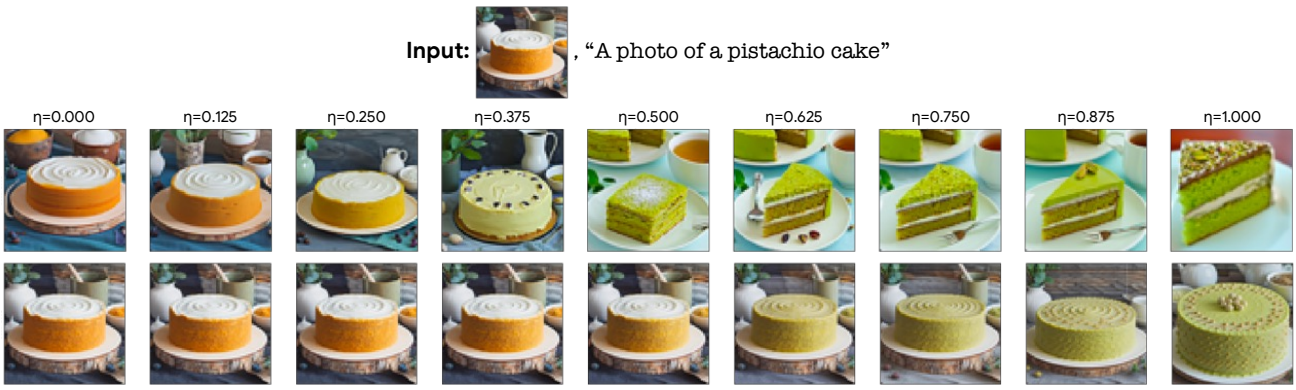

Figure 5. **Embedding interpolation.** Varying η with the same seed, using the pre-trained (top) and fine-tuned (bottom) models.

64×64 diffusion model by continuing Imagen's training for 1500 steps for our input image, conditioned on the optimized embedding. In parallel, we also fine-tune the $64 \times 64 \rightarrow 256 \times 256$ SR diffusion model using the target text embedding and the original image for 1500 steps, in order to capture high-frequency details from the original image. We find that fine-tuning the $256 \times 256 \rightarrow 1024 \times 1024$ model adds little to no effect to the results, therefore we opt to use its pre-trained version conditioned on the target text. This entire optimization process takes around 8 minutes per image on two TPUv4 chips.

Afterwards, we interpolate the text embeddings according to Equation 3. Because of the fine-tuning process, using $\eta = 0$ will generate the original image, and as η increases, the image will start to align with the target text. To maintain both image fidelity and target text alignment, we choose an intermediate η, usually residing between 0.6 and 0.8 (see Figure 8). We then generate with Imagen [46] with its provided hyperparameters. We find that using the DDIM [52] sampling scheme generally provides slightly improved results over the more stochastic DDPM scheme.

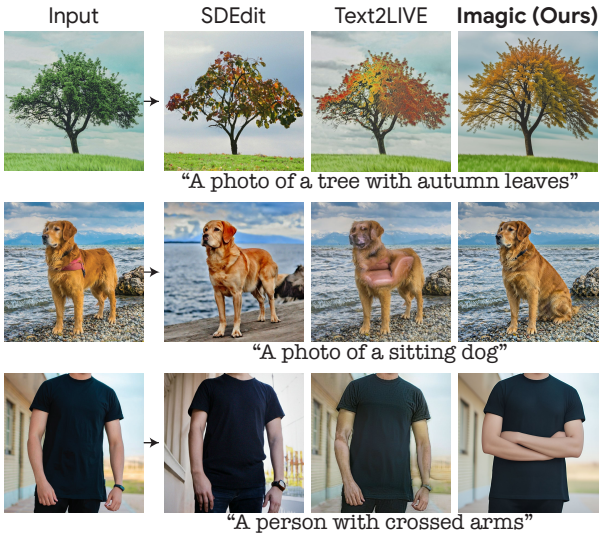

Figure 6. **Method Comparison.** We compare SDEdit [32] and Text2LIVE [7] to our method. Imagic successfully applies the desired edit (including complex non-rigid edits), while preserving the original image details well.

4. Experiments

4.1. Qualitative Evaluation

In order to test it, we apply our method on a multitude of real images from various domains, with simple text prompts describing different editing categories such as: style, appearance, color, posture, and composition. We collect high-resolution free-to-use images from Unsplash and Pixabay. After optimization, we generate each edit with 5 random seeds and choose the best result. Imagic shows impressive results, and it is able to apply various editing categories on any general input image and text, as we show in Figure 1 and Figure 7. We experiment with different text prompts for the same image in Figure 2, showing the versatility of Imagic. Since the underlying generative diffusion model that we utilize is probabilistic, our method can generate different results for a single image-text pair. We show multiple options for an edit using different random seeds in Figure 4, slightly tweaking η for each seed. This stochasticity allows the user to choose among these different options, as natural language text prompts can generally be ambiguous and imprecise.

4.2. Comparisons

We compare Imagic to the current leading general-purpose techniques that operate on a single input real-world image, and edit it based on a text prompt. Namely, we compare our method to Text2LIVE [7] and SDEdit [32]. We use Text2LIVE's default provided hyperparameters. We feed it with a text description of the target object (*e.g.*, “dog”) and one of the desired edit (*e.g.*, “sitting dog”). For

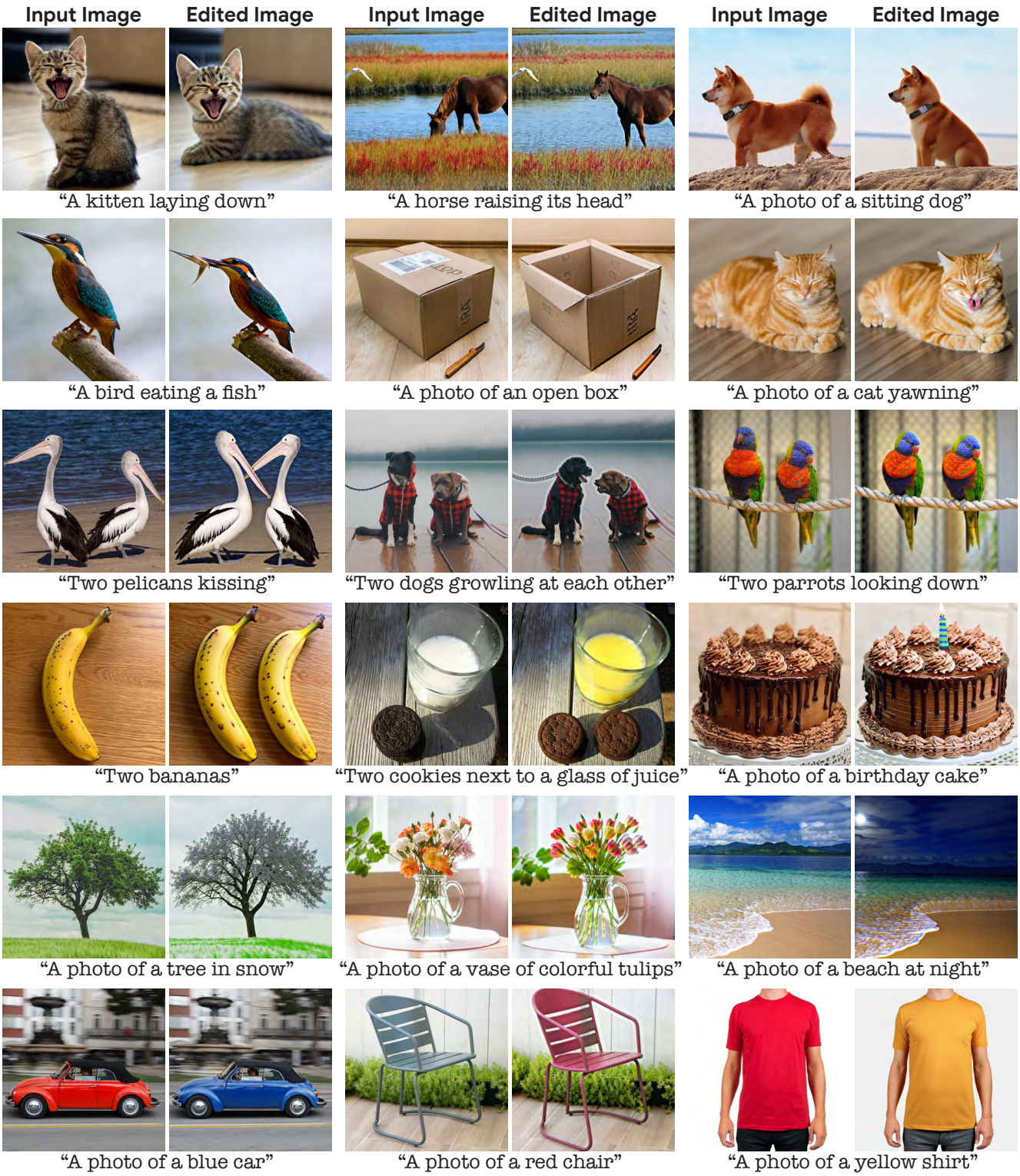

Figure 7. **Wide range of editing types.** 1024 × 1024-pixel pairs of original (left) and edited (right) images using our method (with target texts). Editing types include posture changes, composition changes, multiple object editing, object additions, style changes, and color changes.

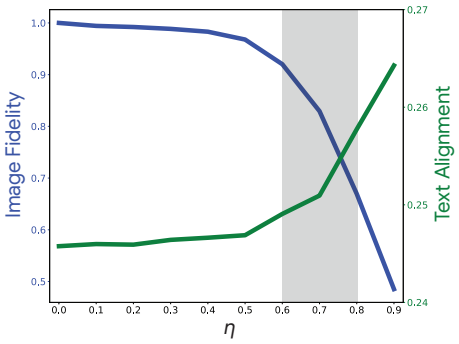

Figure 8. **Editability–fidelity tradeoff.** CLIP score (indicating target text alignment) and $1-$LPIPS (indicating fidelity to the original image) as functions of η, averaged over 150 image-text pairs using different random seeds. Edited images tend to match both the input image and text in the highlighted area.

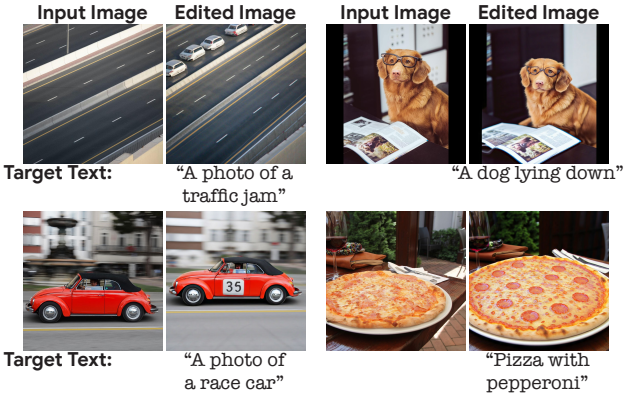

Figure 9. **Failure cases.** Examples with subpar results, such as insufficient alignment with text (top) or loss of details and camera angle changes (bottom).

SDEdit [32], we apply their proposed technique with the same Imagen [46] model and target text prompt that we use. We keep the diffusion hyperparameters from Imagen, and choose the intermediate diffusion timestep for SDEdit independently for each image to achieve the best target text alignment without drastically changing the image contents.

In Figure 6 we display editing results across the different methods. For SDEdit and Imagic, we sample 8 images using different random seeds and display the result with the best alignment to both the target text and the input image. As can be observed, our method maintains high fidelity to the input image while aptly performing the desired edits. When tasked with a complex non-rigid edit such as making a dog sit, our method significantly outperforms previous techniques. Imagic constitutes the first demonstration of such sophisticated text-based edits applied on a single real-world image.

4.3. Ablation Study

Fine-tuning and optimization We generate edited images for different η values using the pre-trained 64×64 diffusion model and our fine-tuned one, in order to gauge the effect of fine-tuning on the output quality. We use the same optimized embedding and random seed, and qualitatively evaluate the results in Figure 5. Without fine-tuning, the scheme does not fully reconstruct the original image at $\eta = 0$, and fails to retain the image's details as η increases. In contrast, fine-tuning imposes details from the input image beyond just the optimized embedding, allowing our scheme to retain these details for intermediate values of η, thereby enabling semantically meaningful linear interpolation. Thus, we conclude that model fine-tuning is essential for our method's success. Furthermore, we experiment with the number of text embedding optimization steps in the appendix. Our findings suggest that optimizing the text embedding with a smaller number of steps limits our editing capabilities, while optimizing for more than 100 steps yields little to no added value.

Interpolation intensity As can be observed in Figure 5, fine-tuning increases the η at which the model strays from reconstructing the input image. While the optimal η value may vary per input (as different edits require different intensities), we attempt to identify the region in which the edit is best applied. To that end, we apply our editing scheme with different η values, and calculate the outputs' CLIP score [17, 38] w.r.t. the target text, and their LPIPS score [60] w.r.t. the input image subtracted from 1. A higher CLIP score indicates better output alignment with the target text, and a higher $1-$LPIPS indicates higher fidelity to the input image. We repeat this process for 150 image-text inputs, and show the average results in Figure 8. We observe that for η values smaller than 0.4, outputs are almost identical to the input images. For $\eta \in [0.6, 0.8]$, the images begin to change (according to LPIPS), and align better with the text (as the CLIP score rises). Therefore, we identify this area as the most probable for obtaining satisfactory results. Note that while they provide a good sense of text or image alignment on average, CLIP score and LPIPS are imprecise measures that rely on neural network backbones, and their values noticeably differ for each different input image-text pair. As such, they are not suited for reliably choosing η for each input in an automatic way.

4.4. Limitations

We identify two failure cases of our method: In some cases, the desired edit is applied very subtly, therefore not aligning well with the target text. In other cases, the edit is applied well, but it affects extrinsic image details such

as zoom or camera angle. We show examples of these two failure cases in the first and second row of Figure 9, respectively. When the edit is not applied strongly enough, increasing η usually achieves the desired result, but it sometimes leads to a significant loss of original image details (for all tested random seeds) in a few cases. As for zoom and camera angle changes, these usually occur before the desired edit takes place, as we progress from a low η value to a large one, which makes circumventing them difficult. We demonstrate this phenomenon in Figure 11 in the appendix.

These limitations can possibly be mitigated by optimizing the text embedding or the diffusion model differently, or by incorporating cross-attention control akin to Hertz et al. [16]. We leave those options for future work. Additionally, since our method relies on a pre-trained text-to-image diffusion model, it inherits the model's generative limitations and possible biases. Therefore, unwanted artifacts are produced when the desired edit involves generating failure cases of the underlying model. For instance, Imagen is known to show substandard generative performance on human faces [46].

5. Conclusions and Future Work

We propose a novel image editing method called Imagic. Our method accepts a single image and a simple text prompt describing the desired edit, and aims to apply this edit while preserving a maximal amount of details from the image. To that end, we utilize a pre-trained text-to-image diffusion model and use it to find a text embedding that represents the input image. Then, fine-tune the diffusion model to fit the image better, and finally we interpolate linearly between the embedding representing the image and the target text embedding, obtaining a semantically meaningful mixture of them. This enables our scheme to provide edited images using the interpolated embedding. Contrary to other editing methods, our approach can produce sophisticated non-rigid edits that may alter the pose, geometry, and/or composition of the image as requested, in addition to simpler edits such as style or color. It does so while requiring the user to provide only a single image and a simple target text prompt, without the need for additional auxiliary inputs such as image masks.

Our future work may focus on further improving the method's fidelity to the input image and identity preservation, as well as its sensitivity to random seeds and the interpolation parameter η. Another intriguing research direction would be the development of an automated method for choosing η for each requested edit.

5.1. Societal Impact

The goal of our method is to enable complex editing of real world images using textual descriptions of the target edit. As such, it is less prone to societal biases of text-based generative models since it relies mostly on the input image for generation. However, as with other approaches that use generative models for image editing, such techniques might be used by malicious parties for synthesizing fake imagery to mislead viewers. To mitigate this, further research on identification of synthetically edited or generated content is needed.

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.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE international conference on computer vision*, pages 4432–4441, 2019. 4

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 4

[3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows, 2020. 3

[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 4

[5] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. *arXiv preprint arXiv:2111.15666*, 2021. 4

[6] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 2, 4

[7] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2LIVE: text-driven layered image and video editing. *arXiv preprint arXiv:2204.02491*, 2022. 2, 4, 6

[8] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020. 4

[9] Amit H Bermano, Rinon Gal, Yuval Alaluf, Ron Mokady, Yotam Nitzan, Omer Tov, Oren Patashnik, and Daniel Cohen-Or. State-of-the-art in the architecture, methods and applications of stylegan. In *Computer Graphics Forum*, volume 41, pages 591–611. Wiley Online Library, 2022. 2

[10] Tsachi Blau, Roy Ganz, Bahjat Kawar, Alex Bronstein, and Michael Elad. Threat model-agnostic adversarial defense using diffusion models. *arXiv preprint arXiv:2207.08089*, 2022. 4

[11] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 4

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 3, 4

[13] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. 2, 4

[14] Jin Gao, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang. Back to the source: Diffusion-driven test-time adaptation. *arXiv preprint arXiv:2207.03442*, 2022. 4

[15] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 3

[16] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022. 2, 4, 9

[17] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 8

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3, 4, 5

[19] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 5

[20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 5

[21] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 3

[22] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 3

[25] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022. 4

[26] Bahjat Kawar, Roy Ganz, and Michael Elad. Enhancing diffusion-based image synthesis with robust classifier guidance. *arXiv preprint arXiv:2208.08664*, 2022. 4

[27] Bahjat Kawar, Jiaming Song, Stefano Ermon, and Michael Elad. JPEG artifact correction using denoising diffusion restoration models. *arXiv preprint arXiv:2209.11888*, 2022. 4

[28] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022. 2, 4

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 5

[30] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani, and Inbar Mosseri. Explaining in style: Training a gan to explain a classifier in stylespace. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 693–702, 2021. 3

[31] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. *arXiv preprint arXiv:2112.05744*, 2021. 4

[32] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 4, 6, 8

[33] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 4

[34] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, 2022. 4

[35] Byong Mok Oh, Max Chen, Julie Dorsey, and Frédo Durand. Image-based modeling and photo editing. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 433–442, 2001. 1

[36] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*, 2021. 2, 3, 5

[37] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR, 2021. 4

[38] 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. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 5, 8

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020. 5

[40] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2, 3, 5

[41] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020. 4

[42] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42(1):1–13, 2022. 4, 5

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 3, 5

[44] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arxiv:2208.12242*, 2022. 2, 4

[45] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 4

[46] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022. 3, 5, 6, 8, 9

[47] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 4

[48] Hiroshi Sasaki, Chris G Willcocks, and Toby P Breckon. Unit-ddpm: Unpaired image translation with denoising diffusion probabilistic models. *arXiv preprint arXiv:2104.05358*, 2021. 4

[49] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 3

[50] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020. 3

[51] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 4

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 4, 6

[53] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 4

[54] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020. 4

[55] Lucas Theis, Tim Salimans, Matthew D Hoffman, and Fabian Mentzer. Lossy compression with Gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022. 4

[56] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021. 4, 5

[57] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021. 4

[58] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 4

[59] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. *arXiv preprint arXiv:2112.03145*, 2021. 4

[60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 8

[61] Roland S Zimmermann, Lukas Schott, Yang Song, Benjamin A Dunn, and David A Klindt. Score-based generative classifiers. *arXiv preprint arXiv:2110.00473*, 2021. 4

A. Ablation Study

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 10, 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 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 11, we assess the effect of varying η values for different random

Figure 10. **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).

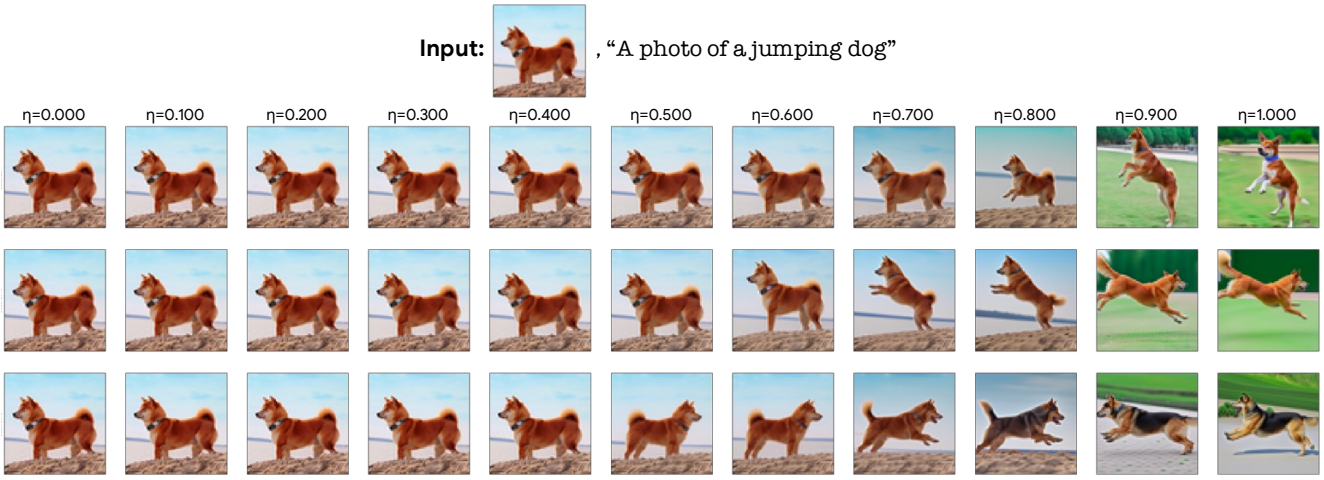

Figure 11. **Different seeds.** Varying η values and different seeds produce different results for the same input.

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 11 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 9. 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.