# Prompt-to-Prompt Image Editing with Cross Attention Control

Amir Hertz*[1,2], Ron Mokady*[1,2], Jay Tenenbaum[1], Kfir Aberman[1], Yael Pritch[1], and Daniel Cohen-Or*[1,2]

[1] *Google Research*
[2] *The Blavatnik School of Computer Science, Tel Aviv University*

## Abstract

Recent large-scale text-driven synthesis models have attracted much attention thanks to their remarkable capabilities of generating highly diverse images that follow given text prompts. Such text-based synthesis methods are particularly appealing to humans who are used to verbally describe their intent. Therefore, it is only natural to extend the text-driven image synthesis to text-driven image editing. Editing is challenging for these generative models, since an innate property of an editing technique is to preserve most of the original image, while in the text-based models, even a small modification of the text prompt often leads to a completely different outcome. State-of-the-art methods mitigate this by requiring the users to provide a spatial mask to localize the edit, hence, ignoring the original structure and content within the masked region. In this paper, we pursue an intuitive *prompt-to-prompt* editing framework, where the edits are controlled by text only. To this end, we analyze a text-conditioned model in depth and observe that the cross-attention layers are the key to controlling the relation between the spatial layout of the image to each word in the prompt. With this observation, we present several applications which monitor the image synthesis by editing the textual prompt only. This includes localized editing by replacing a word, global editing by adding a specification, and even delicately controlling the extent to which a word is reflected in the image. We present our results over diverse images and prompts, demonstrating high-quality synthesis and fidelity to the edited prompts.

## 1 Introduction

Recently, large-scale language-image (LLI) models, such as Imagen [38], DALL·E 2 [33] and Parti [48], have shown phenomenal generative semantic and compositional power, and gained unprecedented attention from the research community and the public eye. These LLI models are trained on extremely large language-image datasets and use state-of-the-art image generative models including auto-regressive and diffusion models. However, these models do not provide simple editing means, and generally lack control over specific semantic regions of a given image. In particular, even the slightest change in the textual prompt may lead to a completely different output image.

To circumvent this, LLI-based methods [28, 4, 33] require the user to explicitly mask a part of the image to be inpainted, and drive the edited image to change in the masked area only, while matching the background of the original image. This approach has provided appealing results, however, the masking procedure is cumbersome, hampering quick and intuitive text-driven editing. Moreover, masking the image content removes important structural information, which is completely ignored in the inpainting process. Therefore, some editing capabilities are out of the inpainting scope, such as modifying the texture of a specific object.

In this paper, we introduce an intuitive and powerful *textual editing* method to semantically edit images in pre-trained text-conditioned diffusion models via *Prompt-to-Prompt* manipulations. To do so, we dive deep into the cross-attention layers and explore their semantic strength as a handle to control the generated image.

*Performed this work while working at Google.


Figure 1: Our method provides variety of *Prompt-to-Prompt* editing capabilities. The user can tune the level of influence of an adjective word (top-left), replace items in the image (top-right), specify a style for an image (bottom-left), or make further refinements over the generated image (bottom-right). The manipulations are infiltrated through the cross-attention mechanism of the diffusion model without the need for any specifications over the image pixel space.

Specifically, we consider the internal *cross-attention maps*, which are high-dimensional tensors that bind pixels and tokens extracted from the prompt text. We find that these maps contain rich semantic relations which critically affect the generated image.

Our key idea is that we can edit images by injecting the cross-attention maps during the diffusion process, controlling which pixels attend to which tokens of the prompt text during which diffusion steps. To apply our method to various creative editing applications, we show several methods to control the cross-attention maps through a simple and semantic interface (see fig. 1). The first is to change a single token's value in the prompt (e.g., "dog" to "cat"), while fixing the cross-attention maps, to preserve the scene composition. The second is to globally edit an image, e.g., change the style, by adding new words to the prompt and freezing the attention on previous tokens, while allowing new attention to flow to the new tokens. The third is to amplify or attenuate the semantic effect of a word in the generated image.

Our approach constitutes an intuitive image editing interface through editing only the textual prompt, therefore called *Prompt-to-Prompt*. This method enables various editing tasks, which are challenging otherwise, and does not requires model training, fine-tuning, extra data, or optimization. Throughout our analysis, we discover even more control over the generation process, recognizing a trade-off between the fidelity to the edited prompt and the source image. We even demonstrate that our method can be applied to real images by using an existing inversion process. Our experiments and numerous results show that our method enables seamless editing in an intuitive text-based manner over extremely diverse images.

## 2 Related work

Image editing is one of the most fundamental tasks in computer graphics, encompassing the process of modifying an input image through the use of an auxiliary input, such as a label, scribble, mask, or reference image. A specifically intuitive way to edit an image is through textual prompts provided by the user. Recently, text-driven image manipulation has achieved significant progress using GANs [15, 8, 19–21], which are known for their high-quality generation, in tandem with CLIP [32], which consists of a semantically rich joint image-text representation, trained over millions of text-image pairs. Seminal works [29, 14, 46, 2] which combined these components were revolutionary, since they did not require extra manual labor, and produced

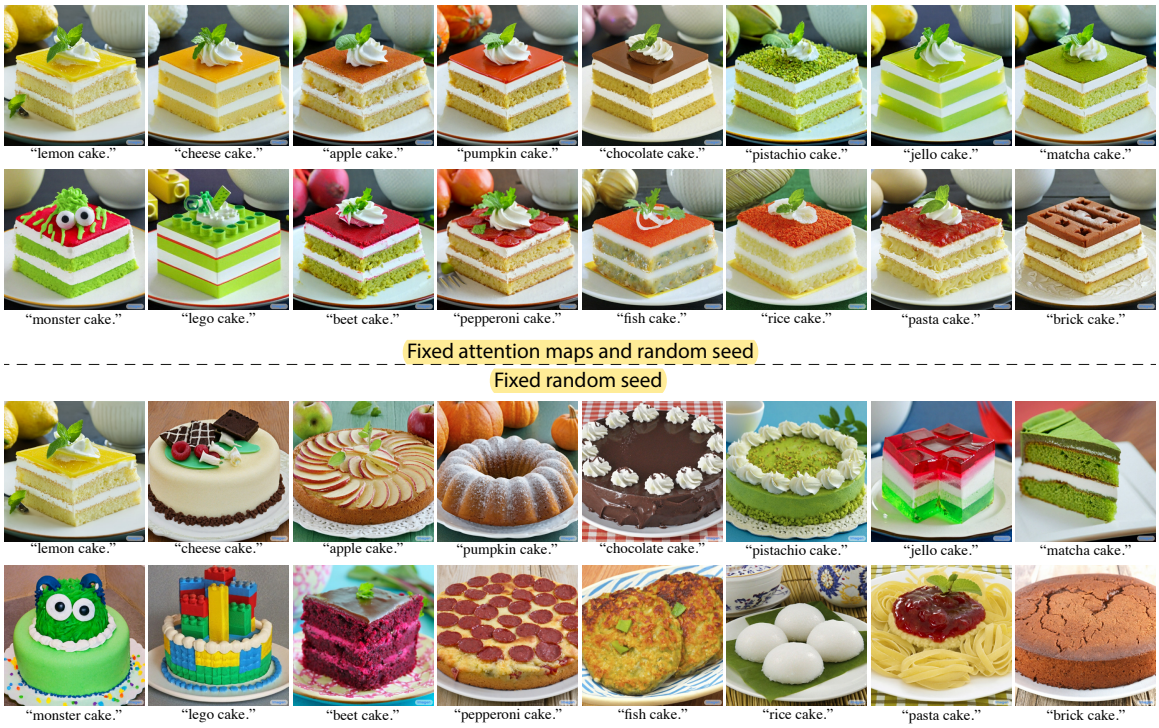



Figure 2: Content modification through attention injection. We start from an original image generated from the prompt "lemon cake", and modify the text prompt to a variety of other cakes. On the top rows, we inject the attention weights of the original image during the diffusion process. On the bottom, we only use the same random seeds as the original image, without injecting the attention weights. The latter leads to a completely new structure that is hardly related to the original image.

highly realistic manipulations using text only. Bau et al. [7] further demonstrated how to use masks provided by the user, to localize the text-based editing and restrict the change to a specific spatial region. However, while GAN-based image editing approaches succeed on highly-curated datasets [27], e.g., human faces, they struggle over large and diverse datasets.

To obtain more expressive generation capabilities, Crowson et al. [9] use VQ-GAN [12], trained over diverse data, as a backbone. Other works [5, 22] exploit the recent Diffusion models [17, 39, 41, 17, 40, 36], which achieve state-of-the-art generation quality over highly diverse datasets, often surpassing GANs [10]. Kim et al. [22] show how to perform global changes, whereas Avrahami et al. [5] successfully perform local manipulations using user-provided masks for guidance.

While most works that require only text (i.e., no masks) are limited to global editing [9, 23], Bar-Tal et al. [6] proposed a text-based localized editing technique without using any mask, showing impressive results. Yet, their techniques mainly allow changing textures, but not modifying complex structures, such as changing a bicycle to a car. Moreover, unlike our method, their approach requires training a network for each input.

Numerous works [11, 16, 42, 25, 26, 30, 31, 34, 49, 9, 13, 36] significantly advanced the generation of images conditioned on plain text, known as text-to-image synthesis. Several large-scale text-image models have recently emerged, such as Imagen [38], DALL-E2 [33], and Parti [48], demonstrating unprecedented semantic generation. However, these models do not provide control over a generated image, specifically using text guidance only. Changing a single word in the original prompt associated with the image often leads to a completely different outcome. For instance, adding the adjective "white" to "dog" often changes the dog's shape. To overcome this, several works [28, 4] assume that the user provides a mask to restrict the area in which the changes are applied.

Unlike previous works, our method requires textual input only, by using the spatial information from the internal layers of the generative model itself. This offers the user a much more intuitive editing experience of modifying local or global details by merely modifying the text prompt.

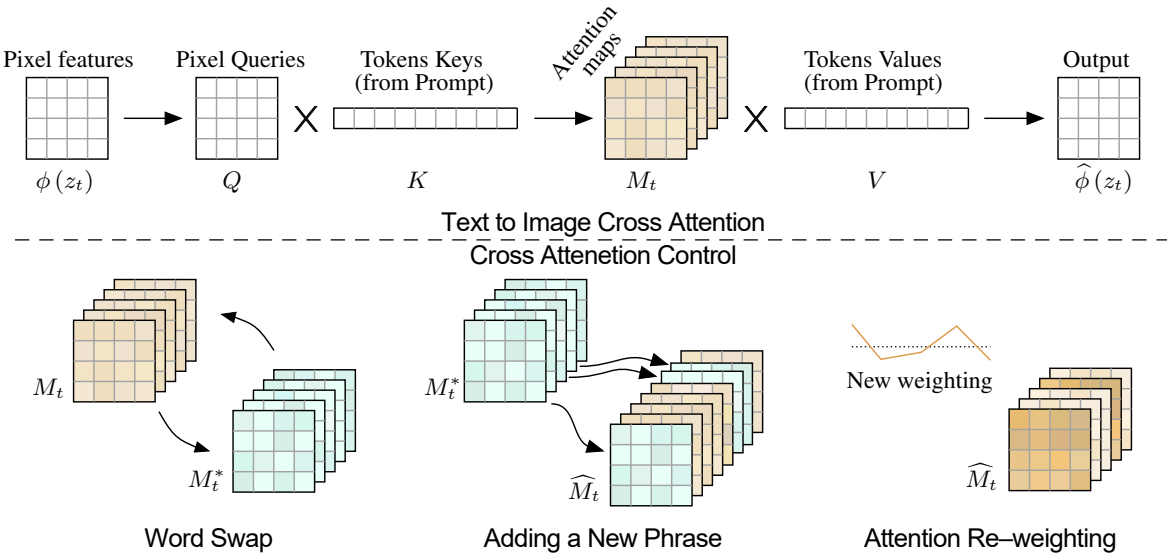



Figure 3: Method overview. Top: visual and textual embedding are fused using cross-attention layers that produce spatial attention maps for each textual token. Bottom: we control the spatial layout and geometry of the generated image using the attention maps of a source image. This enables various editing tasks through editing the textual prompt only. When swapping a word in the prompt, we inject the source image maps $M_t$, overriding the target image maps $M_t^*$, to preserve the spatial layout. Where in the case of adding a new phrase, we inject only the maps that correspond to the unchanged part of the prompt. Amplify or attenuate the semantic effect of a word achieved by re-weighting the corresponding attention map.

## 3 Method

Let $\mathcal{I}$ be an image which was generated by a text-guided diffusion model [38] using the text prompt $\mathcal{P}$ and a random seed $s$. Our goal is editing the input image guided only by the edited prompt $\mathcal{P}^*$, resulting in an edited image $\mathcal{I}^*$. For example, consider an image generated from the prompt "my new bicycle", and assume that the user wants to edit the color of the bicycle, its material, or even replace it with a scooter while preserving the appearance and structure of the original image. An intuitive interface for the user is to directly change the text prompt by further describing the appearance of the bikes, or replacing it with another word. As opposed to previous works, we wish to avoid relying on any user-defined mask to assist or signify where the edit should occur. A simple, but an unsuccessful attempt is to fix the internal randomness and regenerate using the edited text prompt. Unfortunately, as fig. 2 shows, this results in a completely different image with a different structure and composition.

Our key observation is that the structure and appearances of the generated image depend not only on the random seed, but also on the *interaction* between the pixels to the text embedding through the diffusion process. By modifying the pixel-to-text interaction that occurs in *cross-attention* layers, we provide Prompt-to-Prompt image editing capabilities. More specifically, injecting the cross-attention maps of the input image $\mathcal{I}$ enables us to preserve the original composition and structure. In section 3.1, we review how cross-attention is used, and in section 3.2 we describe how to exploit the cross-attention for editing. For additional background on diffusion models, please refer to appendix A.

### 3.1 Cross-attention in text-conditioned Diffusion Models

We use the Imagen [38] text-guided synthesis model as a backbone. Since the composition and geometry are mostly determined at the $64 \times 64$ resolution, we only adapt the text-to-image diffusion model, using the super-resolution process as is. Recall that each diffusion step $t$ consists of predicting the noise $\epsilon$ from a noisy image $z_t$ and text embedding $\psi(\mathcal{P})$ using a U-shaped network [37]. At the final step, this process yields the generated image $\mathcal{I} = z_0$. Most importantly, the interaction between the two modalities occurs during the noise prediction, where the embeddings of the visual and textual features are fused using Cross-attention layers that produce spatial attention maps for each textual token.

More formally, as illustrated in fig. 3(Top), the deep spatial features of the noisy image $\phi(z_t)$ are projected to a query matrix $Q = \ell_Q(\phi(z_t))$, and the textual embedding is projected to a key matrix $K = \ell_K(\psi(\mathcal{P}))$ and

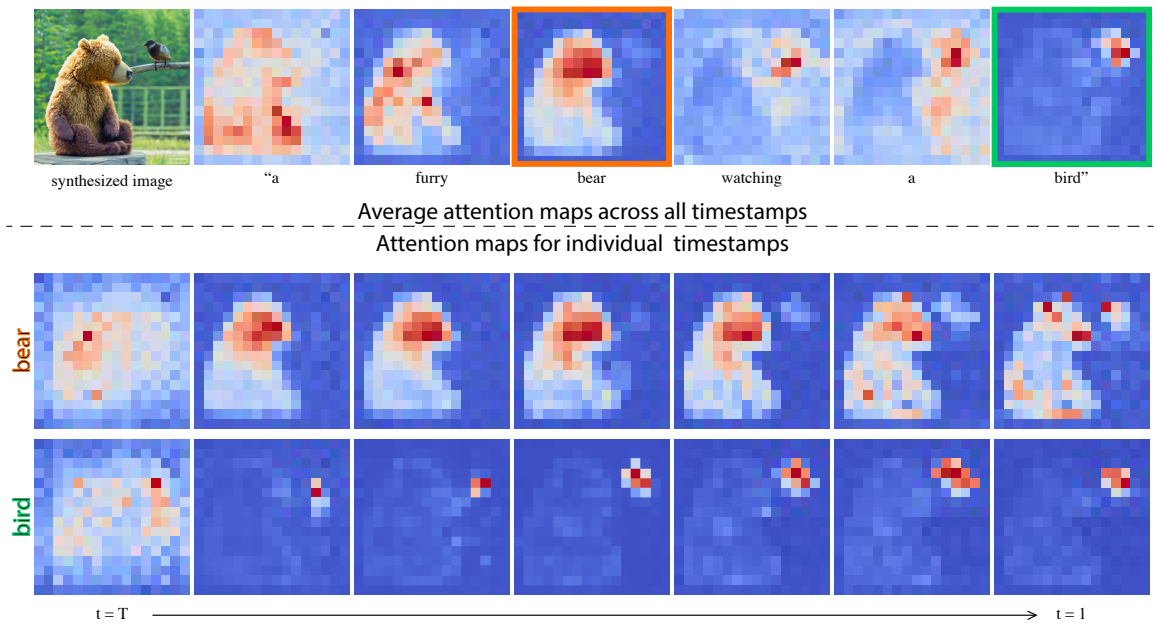



Figure 4: Cross-attention maps of a text-conditioned diffusion image generation. The top row displays the average attention masks for each word in the prompt that synthesized the image on the left. The bottom rows display the attention maps from different diffusion steps with respect to the words “bear” and “bird”.

a value matrix $V = \ell_V(\psi(\mathcal{P}))$, via learned linear projections $\ell_Q, \ell_K, \ell_V$. The *attention maps* are then

$$M = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right), \tag{1}$$

where the cell $M_{ij}$ defines the weight of the value of the $j$-th token on the pixel $i$, and where $d$ is the latent projection dimension of the keys and queries. Finally, the cross-attention output is defined to be $\widehat{\phi}(z_t) = MV$, which is then used to update the spatial features $\phi(z_t)$.

Intuitively, the cross-attention output $MV$ is a weighted average of the values $V$ where the weights are the attention maps $M$, which are correlated to the *similarity* between $Q$ and $K$. In practice, to increase their expressiveness, multi-head attention [44] is used in parallel, and then the results are concatenated and passed through a learned linear layer to get the final output.

Imagen [38], similar to GLIDE [28], conditions on the text prompt in the noise prediction of each diffusion step (see appendix A.2) through two types of attention layers: i) cross-attention layers. ii) hybrid attention that acts both as self-attention and cross-attention by simply concatenating the text embedding sequence to the key-value pairs of each self-attention layer. Throughout the rest of the paper, we refer to both of them as cross-attention since our method only intervenes in the cross-attention part of the hybrid attention. That is, only the last channels, which refer to text tokens, are modified in the hybrid attention modules.

### 3.2 Controlling the Cross-attention

We return to our key observation — the spatial layout and geometry of the generated image depend on the *cross-attention* maps. This interaction between pixels and text is illustrated in fig. 4, where the average attention maps are plotted. As can be seen, pixels are more *attracted* to the words that describe them, e.g., pixels of the bear are correlated with the word “bear”. Note that averaging is done for visualization purposes, and attention maps are kept separate for each head in our method. Interestingly, we can see that the structure of the image is already determined in the early steps of the diffusion process.

Since the attention reflects the overall composition, we can inject the attention maps $M$ that were obtained from the generation with the original prompt $\mathcal{P}$, into a second generation with the modified prompt $\mathcal{P}^*$. This allows the synthesis of an edited image $\mathcal{I}^*$ that is not only manipulated according to the edited prompt, but also preserves the structure of the input image $\mathcal{I}$. This example is a specific instance of a broader set of

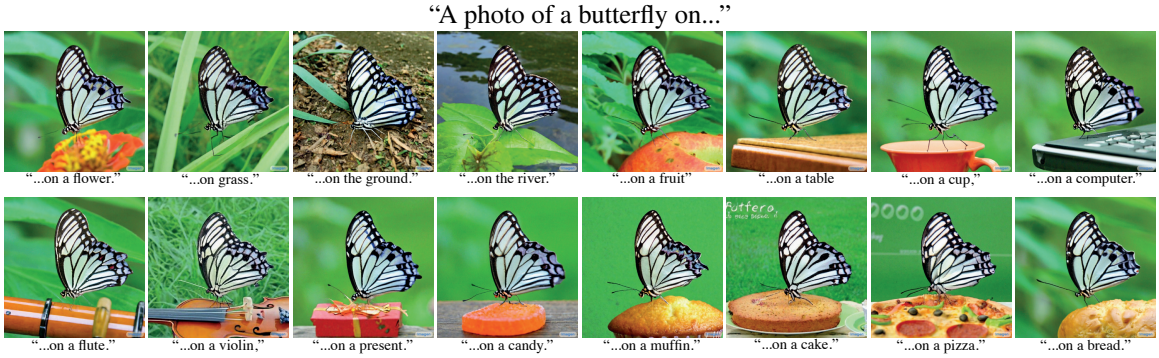



Figure 5: Object preservation. By injecting only the attention weights of the word "butterfly", taken from the top-left image, we can preserve the structure and appearance of a single item while replacing its context. Note how the butterfly sits on top of all objects in a very plausible manner.

attention-based manipulations leading to different types of intuitive editing. We, therefore, start by proposing a general framework, followed by the details of the specific editing operations.

Let $DM(z_t, \mathcal{P}, t, s)$ be the computation of a single step $t$ of the diffusion process, which outputs the noisy image $z_{t-1}$, and the attention map $M_t$ (omitted if not used). We denote by $DM(z_t, \mathcal{P}, t, s)\{M \leftarrow \widehat{M}\}$ the diffusion step where we override the attention map $M$ with an additional given map $\widehat{M}$, but keep the values $V$ from the supplied prompt. We also denote by $M_t^*$ the produced attention map using the edited prompt $\mathcal{P}^*$. Lastly, we define $Edit(M_t, M_t^*, t)$ to be a general edit function, receiving as input the $t$'th attention maps of the original and edited images during their generation.

Our general algorithm for controlled image generation consists of performing the iterative diffusion process for both prompts simultaneously, where an attention-based manipulation is applied in each step according to the desired editing task. We note that for the method above to work, we must fix the internal randomness. This is due to the nature of diffusion models, where even for the same prompt, two random seeds produce drastically different outputs. Formally, our general algorithm is:

**Algorithm 1:** Prompt-to-Prompt image editing

1 **Input:** A source prompt $\mathcal{P}$, a target prompt $\mathcal{P}^*$, and a random seed $s$.
2 **Output:** A source image $x_{src}$ and an edited image $x_{dst}$.
3 $z_T \sim N(0, I)$ a unit Gaussian random variable with random seed $s$;
4 $z_T^* \leftarrow z_T$;
5 **for** $t = T, T-1, \ldots, 1$ **do**
6 $\quad z_{t-1}, M_t \leftarrow DM(z_t, \mathcal{P}, t, s)$;
7 $\quad M_t^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s)$;
8 $\quad \widehat{M}_t \leftarrow Edit(M_t, M_t^*, t)$;
9 $\quad z_{t-1}^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s_t)\{M \leftarrow \widehat{M}_t\}$;
10 **end**
11 **Return** $(z_0, z_0^*)$

Notice that we can also define image $\mathcal{I}$, which is generated by prompt $\mathcal{P}$ and random seed $s$, as an additional input. Yet, the algorithm would remain the same. For editing real images, see section 4. Also, note that we can skip the forward call in line 7 by applying the edit function inside the diffusion forward function. Moreover, a diffusion step can be applied on both $z_{t-1}$ and $z_t^*$ in the same batch (i.e., in parallel), and so there is only one step overhead with respect to the original inference of the diffusion model.

We now turn to address specific editing operations, filling the missing definition of the $Edit(M_t, M_t^*, t)$ function. An overview is presented in fig. 3(Bottom).

**Word Swap.** In this case, the user swaps tokens of the original prompt with others, e.g., $\mathcal{P}$ ="a big red bicycle" to $\mathcal{P}^*$ ="a big red car". The main challenge is to preserve the original composition while also addressing the content of the new prompt. To this end, we inject the attention maps of the source image into the generation with the modified prompt. However, the proposed attention injection may over constrain the

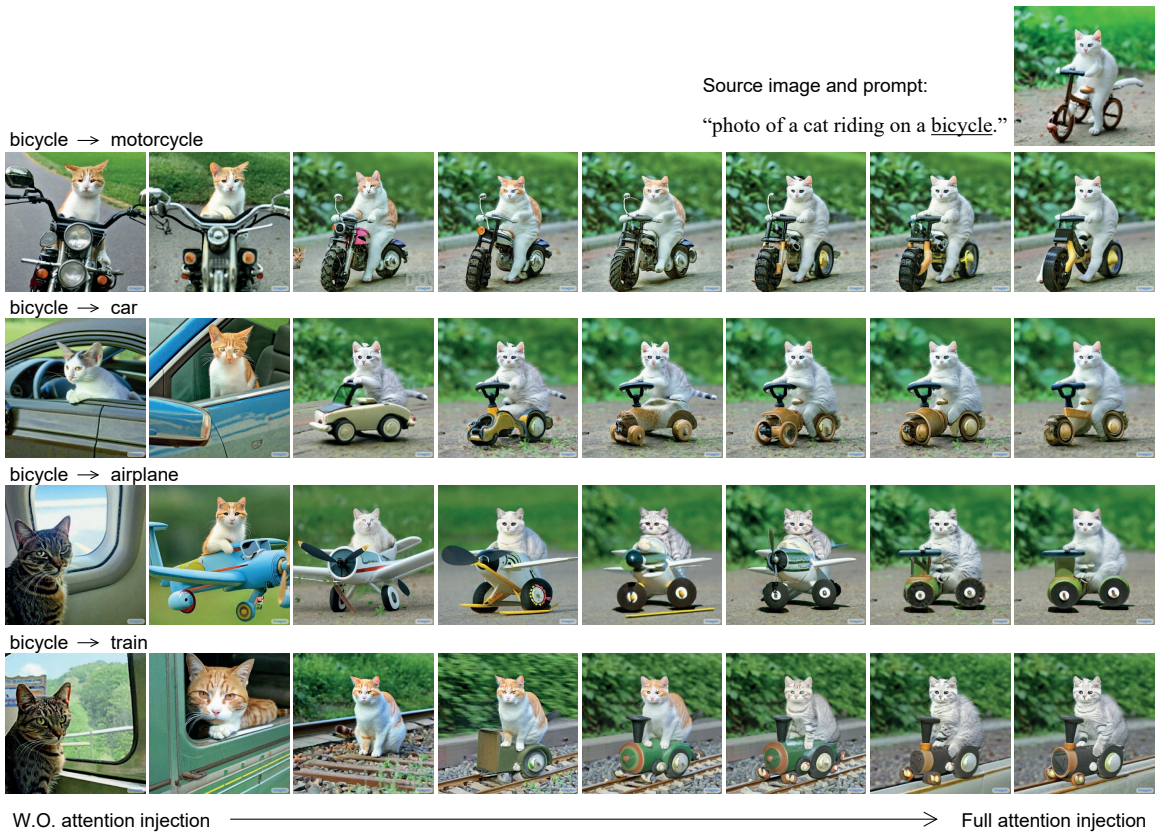



Figure 6: Attention injection through a varied number of diffusion steps. On the top, we show the source image and prompt. In each row, we modify the content of the image by replacing a single word in the text and injecting the cross-attention maps of the source image ranging from 0% (on the left) to 100% (on the right) of the diffusion steps. Notice that on one hand, without our method, none of the source image content is guaranteed to be preserved. On the other hand, injecting the cross-attention throughout all the diffusion steps may over-constrain the geometry, resulting in low fidelity to the text prompt, e.g., the car (3rd row) becomes a bicycle with full cross-attention injection.

geometry, especially when a large structural modification, such as "car" to "bicycle", is involved. We address this by suggesting a softer attention constrain:

$$Edit(M_t, M_t^*, t) := \begin{cases} M_t^* & \text{if } t < \tau \\ M_t & \text{otherwise.} \end{cases}$$

where $\tau$ is a timestamp parameter that determines until which step the injection is applied. Note that the composition is determined in the early steps of the diffusion process. Therefore, by limiting the number of injection steps, we can guide the composition of the newly generated image while allowing the necessary geometry *freedom* for adapting to the new prompt. An illustration is provided in section 4. Another natural relaxation for our algorithm is to assign a different number of injection timestamps for the different tokens in the prompt. In case the two words are represented using a different number of tokens, the maps can be duplicated/averaged as necessary using an alignment function as described in the next paragraph.

**Adding a New Phrase.** In another setting, the user adds new tokens to the prompt, e.g., $\mathcal{P}$ ="a castle next to a river" to $\mathcal{P}^*$ ="children drawing of a castle next to a river". To preserve the common details, we apply the attention injection only over the common tokens from both prompts. Formally, we use an alignment function $A$ that receives a token index from target prompt $\mathcal{P}^*$ and outputs the corresponding token index in $\mathcal{P}$ or *None* if there isn't a match. Then, the editing function is given by:


Figure 7: Editing by prompt refinement. By extending the description of the initial prompt, we can make local edits to the car (top rows) or global modifications (bottom rows).

$$\left(Edit\left(M_t, M_t^*, t\right)\right)_{i,j} := \begin{cases} \left(M_t^*\right)_{i,j} & \text{if } A(j) = None \\ \left(M_t\right)_{i,A(j)} & \text{otherwise.} \end{cases}$$

Recall that index $i$ corresponds to a pixel value, where $j$ corresponds to a text token. Again, we may set a timestamp $\tau$ to control the number of diffusion steps in which the injection is applied. This kind of editing enables diverse Prompt-to-Prompt capabilities such as stylization, specification of object attributes, or global manipulations as demonstrated in section 4.

**Attention Re–weighting.** Lastly, the user may wish to strengthen or weakens the extent to which each token is affecting the resulting image. For example, consider the prompt $\mathcal{P}$ = "a fluffy red ball", and assume we want to make the ball more or less fluffy. To achieve such manipulation, we scale the attention map of the assigned token $j^*$ with parameter $c \in [-2, 2]$, resulting in a stronger/weaker effect. The rest of the attention maps remain unchanged. That is:

$$\left(Edit\left(M_t, M_t^*, t\right)\right)_{i,j} := \begin{cases} c \cdot \left(M_t\right)_{i,j} & \text{if } j = j^* \\ \left(M_t\right)_{i,j} & \text{otherwise.} \end{cases}$$

As described in section 4, the parameter $c$ allows fine and intuitive control over the induced effect.

## 4 Applications

Our method, described in section 3, enables intuitive text-only editing by controlling the spatial layout corresponding to each word in the user-provided prompt. In this section, we show several applications using this technique.

**Text-Only Localized Editing.** We first demonstrate localized editing by modifying the user-provided prompt without requiring any user-provided mask. In fig. 2, we depict an example where we generate an image using


Figure 8: Image stylization. By adding a style description to the prompt while injecting the source attention maps, we can create various images in the new desired styles that preserve the structure of the original image.

the prompt "lemon cake". Our method allows us to retain the spatial layout, geometry, and semantics when replacing the word "lemon" with "pumpkin" (top row). Observe that the background is well-preserved, including the top-left lemons transforming into pumpkins. On the other hand, naively feeding the synthesis model with the prompt "pumpkin cake" results in a completely different geometry (3rd row), even when using the same random seed in a deterministic setting (i.e., DDIM [40]). Our method succeeds even for a challenging prompt such as "pasta cake." (2nd row) — the generated cake consists of pasta layers with tomato sauce on top. Another example is provided in fig. 5 where we do not inject the attention of the entire prompt but only the attention of a specific word – "butterfly". This enables the preservation of the original butterfly while changing the rest of the content. Additional results are provided in the appendix (fig. 13).

As can be seen in fig. 6, our method is not confined to modifying only textures, and it can perform structural modifications, e.g., change a "bicycle" to a "car". To analyze our attention injection, in the left column we show the results without cross-attention injection, where changing a single word leads to an entirely different outcome. From left to right, we then show the resulting generated image by injecting attention to an increasing number of diffusion steps. Note that the more diffusion steps in which we apply cross-attention injection, the higher the fidelity to the original image. However, the optimal result is not necessarily achieved by applying the injection throughout all diffusion steps. Therefore, we can provide the user with even better control over the fidelity to the original image by changing the number of injection steps.

Instead of replacing one word with another, the user may wish to add a new specification to the generated image. In this case, we keep the attention maps of the original prompt, while allowing the generator to address the newly added words. For example, see fig. 7 (top), where we add "crushed" to the "car", resulting in the generation of additional details over the original image while the background is still preserved. See the appendix (fig. 14) for more examples.

**Global editing.** Preserving the image composition is not only valuable for localized editing, but also an important aspect of global editing. In this setting, the editing should affect all parts of the image, but still retain the original composition, such as the location and identity of the objects. As shown in fig. 7 (bottom), we retain the image content while adding "snow" or changing the lightning. Additional examples appear in fig. 8, including translating a sketch into a photo-realistic image and inducing an artistic style.

**Fader Control using Attention Re-weighting.** While controlling the image by editing the prompt is very effective, we find that it still does not allow full control over the generated image. Consider the prompt "snowy mountain". A user may want to control the *amount* of snow on the mountain. However, it is quite difficult to describe the desired amount of snow through text. Instead, we suggest a *fader* control [24], where the user controls the magnitude of the effect induced by a specific word, as depicted in fig. 9. As described in section 3, we achieve such control by re-scaling the attention of the specified word. Additional results are in the appendix (fig. 15).

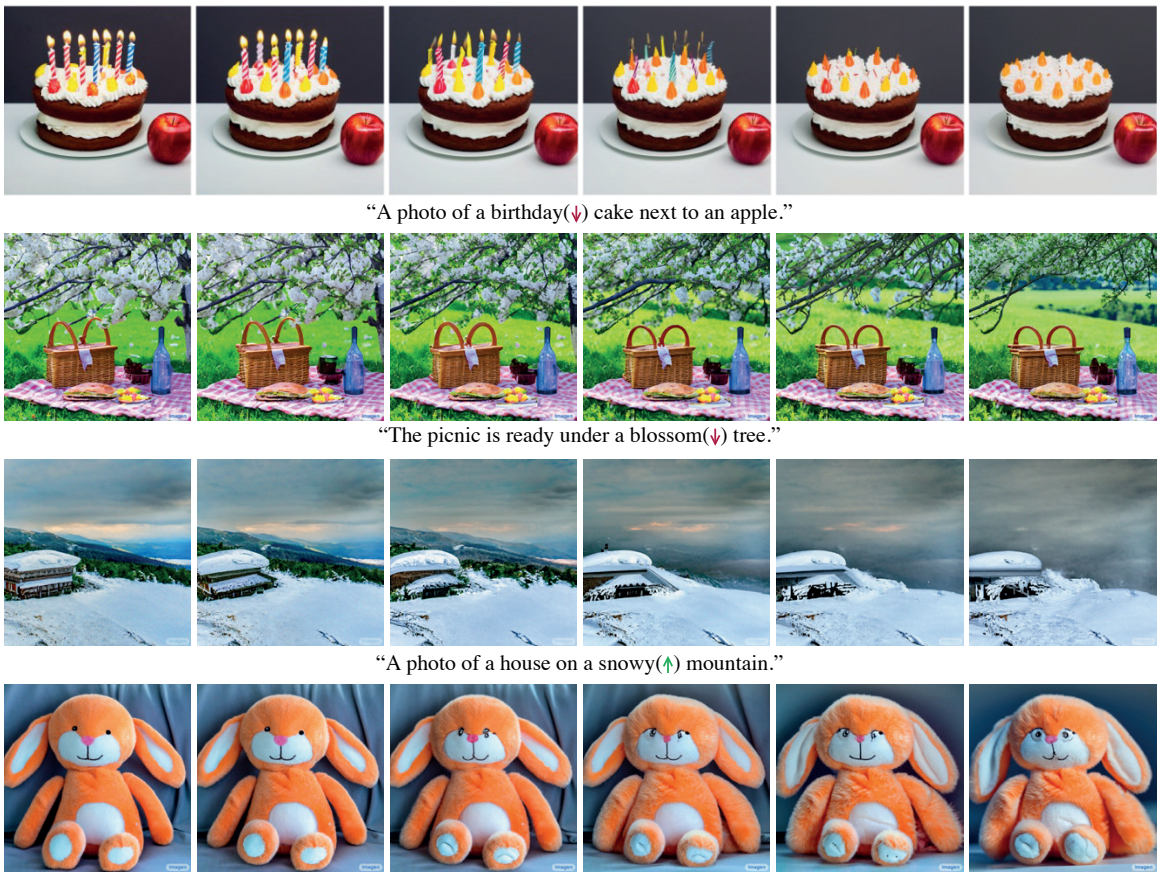



Figure 9: Text-based image editing with fader control. By reducing (top rows) or increasing (bottom) the cross-attention of the specified words (marked with an arrow), we can control the extent to which it influences the generated image.

**Real Image Editing.** Editing a real image requires finding an initial noise vector that produces the given input image when fed into the diffusion process. This process, known as *inversion*, has recently drawn considerable attention for GANs, e.g., [51, 1, 3, 35, 50, 43, 45, 47], but has not yet been fully addressed for text-guided diffusion models.

In the following, we show preliminary editing results on real images, based on common inversion techniques for diffusion models. First, a rather naïve approach is to add Gaussian noise to the input image, and then perform a predefined number of diffusion steps. Since this approach results in significant distortions, we adopt an improved inversion approach [10, 40], which is based on the deterministic DDIM model rather than the DDPM model. We perform the diffusion process in the reverse direction, that is $x_0 \longrightarrow x_T$ instead of $x_T \longrightarrow x_0$, where $x_0$ is set to be the given real image.

This inversion process often produces satisfying results, as presented in fig. 10. However, the inversion is not sufficiently accurate in many other cases, as in fig. 11. This is partially due to a distortion-editability tradeoff [43], where we recognize that reducing the classifier-free guidance [18] parameter (i.e., reducing the prompt influence) improves reconstruction but constrains our ability to perform significant manipulations.

To alleviate this limitation, we propose to restore the unedited regions of the original image using a mask, directly extracted from the attention maps. Note that here the mask is generated with no guidance from the user. As presented in fig. 12, this approach works well even using the naïve DDPM inversion scheme (adding noise followed by denoising). Note that the cat's identity is well-preserved under various editing operations, while the mask is produced only from the prompt itself.

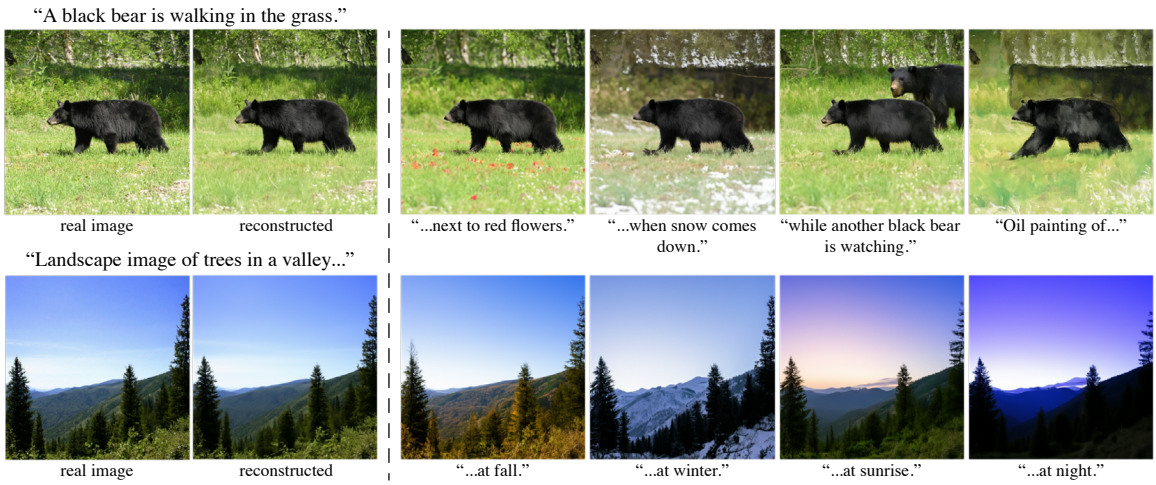



Figure 10: Editing of real images. On the left, inversion results using DDIM [40] sampling. We reverse the diffusion process initialized on a given real image and text prompt. This results in a latent noise that produces an approximation to the input image when fed to the diffusion process. Afterward, on the right, we apply our Prompt-to-Prompt technique to edit the images.

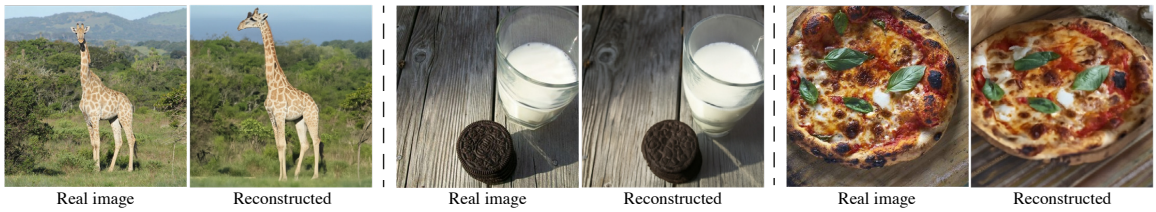



Figure 11: Inversion Failure Cases. Current DDIM-based inversion of real images might result in unsatisfied reconstructions.

## 5 Conclusions

In this work, we uncovered the powerful capabilities of the cross-attention layers within text-to-image diffusion models. We showed that these high-dimensional layers have an interpretable representation of spatial maps that play a key role in tying the words in the text prompt to the spatial layout of the synthesized image. With this observation, we showed how various manipulations of the prompt can directly control attributes in the synthesized image, paving the way to various applications including local and global editing. This work is a first step towards providing users with simple and intuitive means to *edit* images, leveraging textual semantic power. It enables users to navigate through a semantic, textual, space, which exhibits incremental changes after each step, rather than producing the desired image from scratch after each text manipulation.

While we have demonstrated semantic control by changing only textual prompts, our technique is still subject to a few limitations to be addressed in follow-up work. First, the current inversion process results in a visible distortion over some of the test images. In addition, the inversion requires the user to come up with a suitable prompt. This could be challenging for complicated compositions. Note that the challenge of inversion for text-guided diffusion models is an orthogonal endeavor to our work, which will be thoroughly studied in the future. Second, the current attention maps are of low resolution, as the cross-attention is placed in the network's bottleneck. This bounds our ability to perform even more precise localized editing. To alleviate this, we suggest incorporating cross-attention also in higher-resolution layers. We leave this for future works since it requires analyzing the training procedure which is out of our current scope. Finally, we recognize that our current method cannot be used to spatially move existing objects across the image and also leave this kind of control for future work.

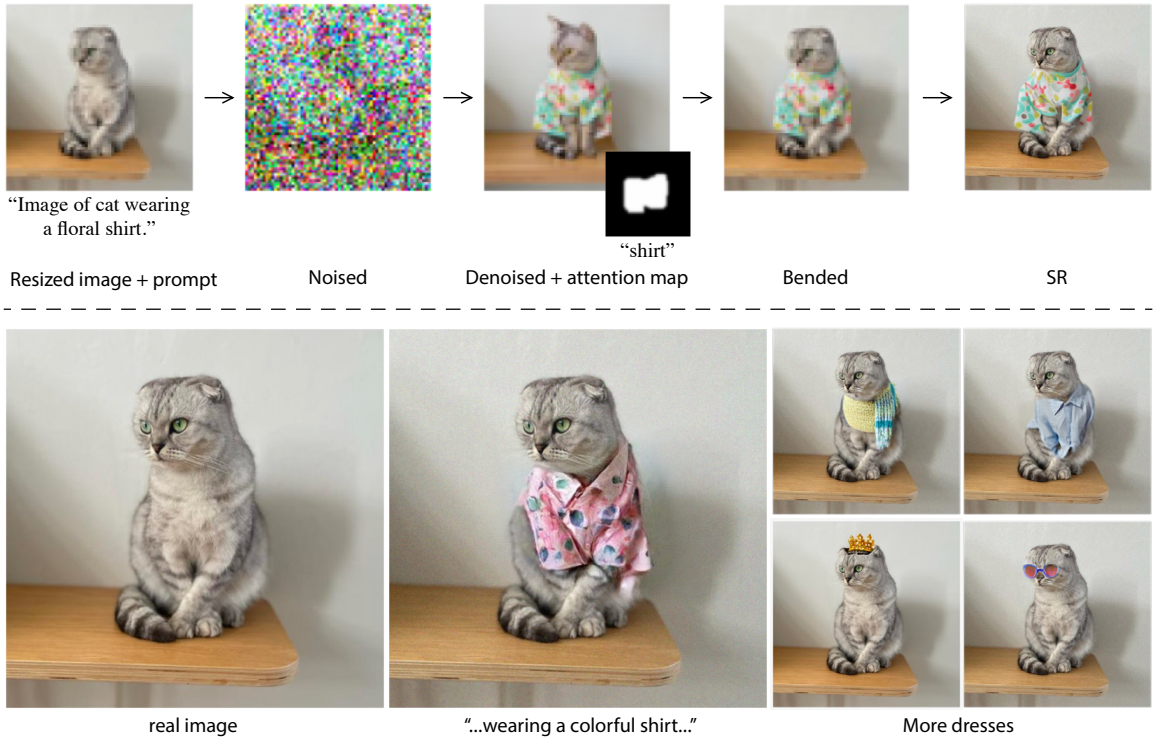



Figure 12: Mask-based editing. Using the attention maps, we preserve the unedited parts of the image when the inversion distortion is significant. This does not require any user-provided masks, as we extract the spatial information from the model using our method. Note how the cat's identity is retained after the editing process.

## 6 Acknowledgments

We thank Noa Glaser, Adi Zicher, Yaron Brodsky and Shlomi Fruchter for their valuable inputs that helped improve this work, and to Mohammad Norouzi, Chitwan Saharia and William Chan for providing us with their support and the pretrained models of Imagen [38]. Special thanks to Yossi Matias for early inspiring discussion on the problem and for motivating and encouraging us to develop technologies along the avenue of intuitive interaction.

## References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] Rameen Abdal, Peihao Zhu, John Femiani, Niloy J Mitra, and Peter Wonka. Clip2stylegan: Unsupervised extraction of stylegan edit directions. *arXiv preprint arXiv:2112.05219*, 2021.

[3] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18511–18521, 2022.

[4] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022.

[5] 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.

[6] 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.

[7] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word, 2021.

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[9] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. *arXiv preprint arXiv:2204.08583*, 2022.

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

[11] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

[12] 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.

[13] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. *arXiv preprint arXiv:2203.13131*, 2022.

[14] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[16] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

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

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

[19] 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*, 34:852–863, 2021.

[20] 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.

[21] 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.

[22] 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.

[23] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. *arXiv preprint arXiv:2112.00374*, 2021.

[24] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. *Advances in neural information processing systems*, 30, 2017.

[25] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. *Advances in Neural Information Processing Systems*, 32, 2019.

[26] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12174–12182, 2019.

[27] Ron Mokady, Omer Tov, Michal Yarom, Oran Lang, Inbar Mosseri, Tali Dekel, Daniel Cohen-Or, and Michal Irani. Self-distilled stylegan: Towards generation from internet photos. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings*, pages 1–9, 2022.

[28] 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.

[29] 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.

[30] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Learn, imagine and create: Text-to-image generation from prior knowledge. *Advances in neural information processing systems*, 32, 2019.

[31] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019.

[32] 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.

[33] 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.

[34] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[35] 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)*, 2022.

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[38] 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, Tim Salimans, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[39] 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.

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

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

[42] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*, 2020.

[43] 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.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[45] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. *ArXiv*, abs/2109.06590, 2021.

[46] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2256–2265, 2021.

[47] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey, 2021.

[48] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

[49] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6199–6208, 2018.

[50] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. *arXiv preprint arXiv:2004.00049*, 2020.

[51] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pages 597–613. Springer, 2016.

# A Background

## A.1 Diffusion Models

Diffusion Denoising Probabilistic Models (DDPM) [39, 17] are generative latent variable models that aim to model a distribution $p_\theta(x_0)$ that approximates the data distribution $q(x_0)$ and easy to sample from. DDPMs model a "forward process" in the space of $x_0$ from data to noise.[†] This process is a Markov chain starting from $x_0$, where we gradually add noise to the data to generate the latent variables $x_1, \ldots, x_T \in X$. The sequence of latent variables therefore follows $q(x_1, \ldots, x_t \mid x_0) = \prod_{i=1}^{t} q(x_t \mid x_{t-1})$, where a step in the forward process is defined as a Gaussian transition $q(x_t \mid x_{t-1}) := N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ parameterized by a schedule $\beta_0, \ldots, \beta_T \in (0, 1)$. When $T$ is large enough, the last noise vector $x_T$ nearly follows an isotropic Gaussian distribution.

An interesting property of the forward process is that one can express the latent variable $x_t$ directly as the following linear combination of noise and $x_0$ without sampling intermediate latent vectors:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}w, \;\; w \sim N(0, I), \tag{2}$$

where $\alpha_t := \prod_{i=1}^{t}(1-\beta_i)$.

In order to sample from the distribution $q(x_0)$, we define the dual "reverse process" $p(x_{t-1} \mid x_t)$ from isotropic Gaussian noise $x_T$ to data by sampling the posteriors $q(x_{t-1} \mid x_t)$. Since the intractable reverse process $q(x_{t-1} \mid x_t)$ depends on the unknown data distribution $q(x_0)$, we approximate it with a parameterized Gaussian transition network $p_\theta(x_{t-1} \mid x_t) := N(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$. The $\mu_\theta(x_t, t)$ can be replaced [17] by predicting the noise $\varepsilon_\theta(x_t, t)$ added to $x_0$ using equation 2.

Under this definition, we use Bayes' theorem to approximate

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\varepsilon_\theta(x_t, t)\right). \tag{3}$$

[†]This process is called "forward" due to its procedure progressing from $x_0$ to $x_T$.

Once we have a trained $\varepsilon_\theta(x_t, t)$, we can using the following sample method

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t z, \;\; z \sim N(0, I). \tag{4}$$

We can control $\sigma_t$ of each sample stage, and in DDIMs [40] the sampling process can be made deterministic using $\sigma_t = 0$ in all the steps. The reverse process can finally be trained by solving the following optimization problem:

$$\min_\theta L(\theta) := \min_\theta E_{x_0 \sim q(x_0), w \sim N(0,I), t} \left\| w - \varepsilon_\theta(x_t, t) \right\|_2^2,$$

teaching the parameters $\theta$ to fit $q(x_0)$ by maximizing a variational lower bound.

### A.2 Cross-attention in Imagen

Imagen [38] consists of three text-conditioned diffusion models: A text-to-image $64 \times 64$ model, and two super-resolution models – $64 \times 64 \rightarrow 256 \times 256$ and $256 \times 256 \rightarrow 1024 \times 1024$. These predict the noise $\varepsilon_\theta(z_t, c, t)$ via a U-shaped network, for $t$ ranging from $T$ to 1. Where $z_t$ is the latent vector and $c$ is the text embedding. We highlight the differences between the three models:

- $64 \times 64$ – starts from a random noise, and uses the U-Net as in [10]. This model is conditioned on text embeddings via both cross-attention layers at resolutions $[16, 8]$ and hybrid-attention layers at resolutions $[32, 16, 8]$ of the downsampling and upsampling within the U-Net.
- $64 \times 64 \rightarrow 256 \times 256$ – conditions on a naively upsampled $64 \times 64$ image. An efficient version of a U-Net is used, which includes Hybrid attention layers in the bottleneck (resolution of 32).
- $256 \times 256 \rightarrow 1024 \times 1024$ – conditions on a naively upsampled $256 \times 256$ image. An efficient version of a U-Net is used, which only includes cross-attention layers in the bottleneck (resolution of 64).

## B Additional results

We provide additional examples, demonstrating our method over different editing operations. fig. 13 show word swap results, fig. 14 show adding specification to an image, and fig. 15 show attention re-weighting.


Figure 13: Additional results for Prompt-to-Prompt editing by word swap.

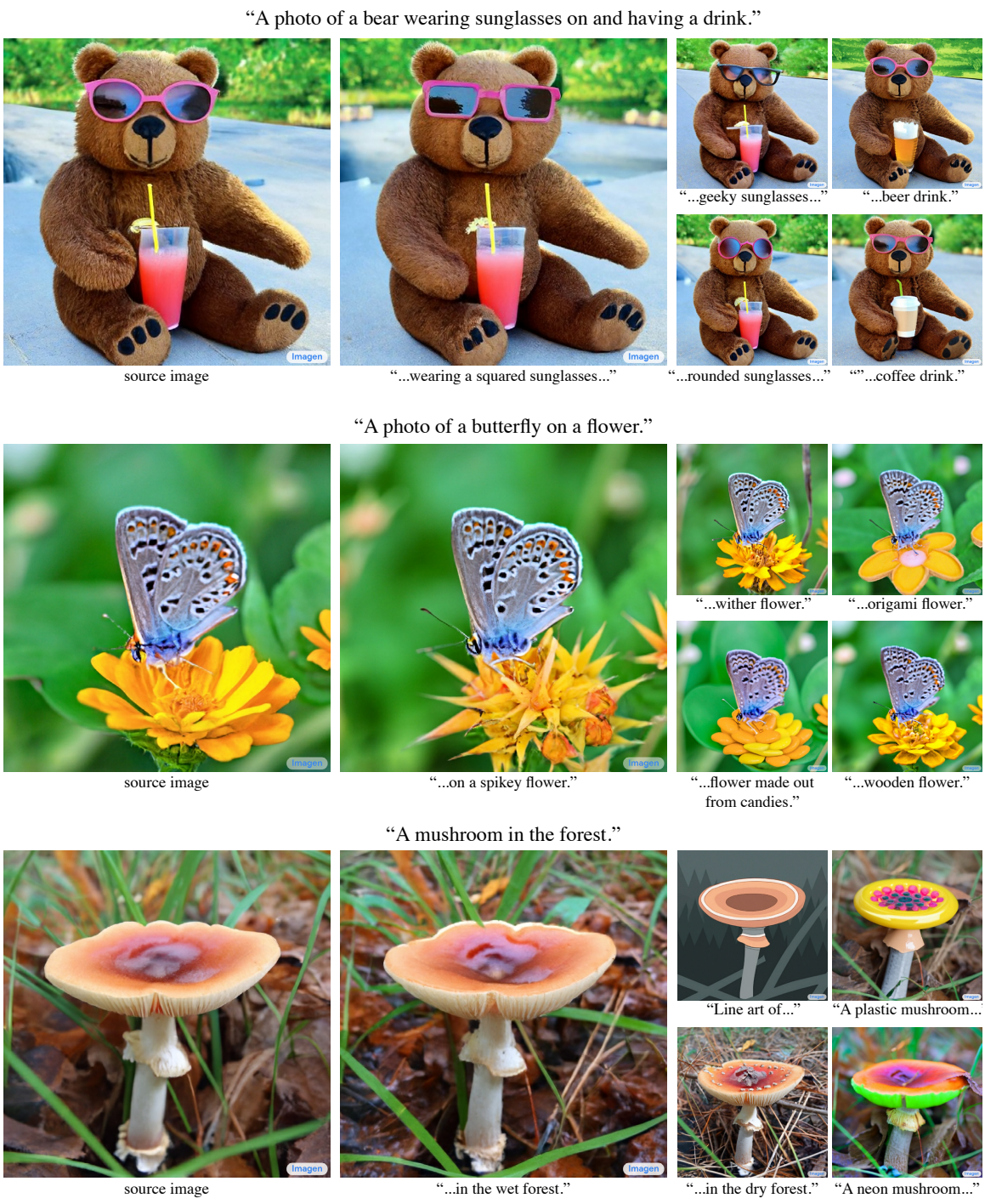


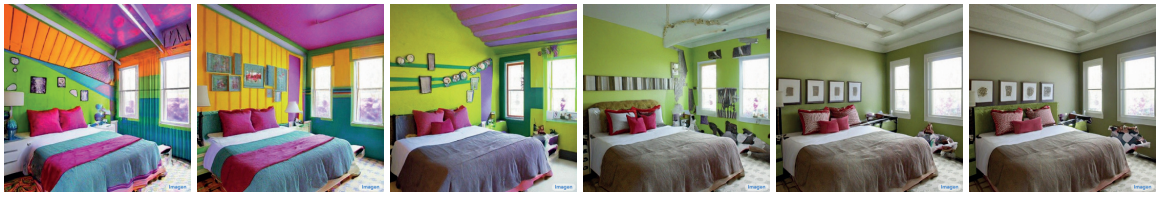

Figure 14: Additional results for Prompt-to-Prompt editing by adding a specification.

"A tiger is sleeping(↑) in a field."

"A smiling(↑) teddy bear."

"Photo of a cubic(↓) sushi."

"The modern(↓) city."



"My colorful(↓) bedroom."

"Photo of a field of poppies at night(↓)."

Figure 15: Additional results for Prompt-to-Prompt editing by attention re-weighting.