Localizing Object-level Shape Variations with Text-to-Image Diffusion Models

Or Patashnik[1] Daniel Garibi[1] Idan Azuri[2] Hadar Averbuch-Elor[1] Daniel Cohen-Or[1]

[1]Tel-Aviv University [2]Independent Researcher

https://orpatashnik.github.io/local-prompt-mixing/

Abstract

Text-to-image models give rise to workflows which often begin with an exploration step, where users sift through a large collection of generated images. The global nature of the text-to-image generation process prevents users from narrowing their exploration to a particular object in the image. In this paper, we present a technique to generate a collection of images that depicts variations in the shape of a specific object, enabling an object-level shape exploration process. Creating plausible variations is challenging as it requires control over the shape of the generated object while respecting its semantics. A particular challenge when generating object variations is accurately localizing the manipulation applied over the object's shape. We introduce a prompt-mixing technique that switches between prompts along the denoising process to attain a variety of shape choices. To localize the image-space operation, we present two techniques that use the self-attention layers in conjunction with the cross-attention layers. Moreover, we show that these localization techniques are general and effective beyond the scope of generating object variations. Extensive results and comparisons demonstrate the effectiveness of our method in generating object variations, and the competence of our localization techniques.

1. Introduction

Text-to-image diffusion models have recently shown unprecedented image quality and diversity [35, 41, 39], and have opened a new era in image synthesis. Still, the control over the generated image is limited, resulting in a tedious selection procedure, where users sample numerous initial seed noises, from which they can select a preferred one. Images generated from different initial noise with the same text prompt share semantics, but the shape, appearance, and location of the generated shapes may differ greatly. The uncontrolled global changes realized by such a sampling process, do not allow users to interact with the generated image and narrow down their open-ended exploration process. In particular, the lack of object-level control of the user with the generated image hinders the user's ability to focus on refining specific objects during their exploration.

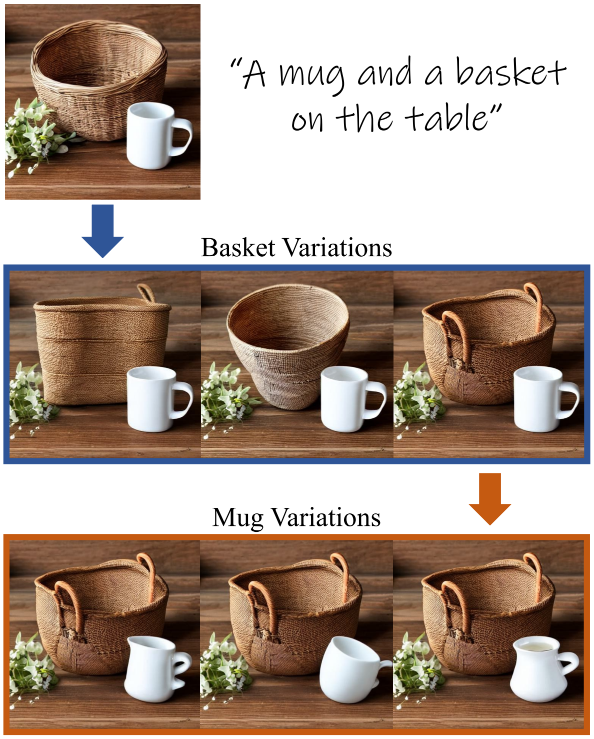

Figure 1. Our method generates shape variations of an object. Here, given the text prompt, we generate variations of the basket. After selecting a preferred basket, we generate variations of the mug. We develop general techniques for localizing modifications.

A possible approach for interacting with the generated image as a means to explore the shape and appearance of a specific object in the image is to use text-guided image inpainting [39] or SDEdit [29]. These methods mainly excel in changing the texture of an object and are therefore more suitable for textural exploration. However, changing the shape of an object, particularly if other regions in the image should be preserved, is significantly more challenging, and these methods struggle to achieve that without affecting the entire image. Figure 2 demonstrates the lack of shape variations with the above approaches.

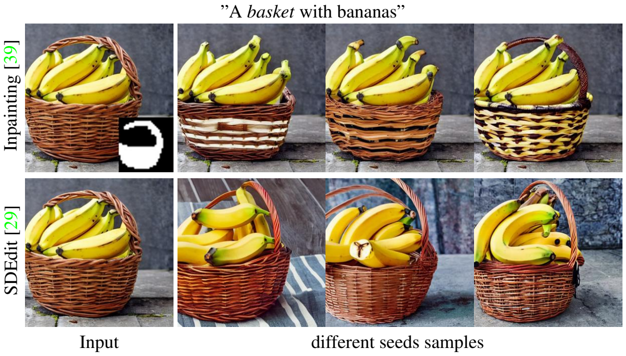

Figure 2. Inpainting and SDEdit struggle to achieve significant shape variations of the basket. Inpainting is restricted to the mask while SDEdit struggles at localizing changes, see the bananas.

In this paper, we deal with object-level shape exploration, where the user attains a gallery with variations of a specific object in the image, without having to provide any additional input. Specifically, acknowledging that previous works have struggled to perform geometric manipulations, we focus on object-level shape variations, which are automatically generated and presented to the user, see Figure 1.

We introduce a prompt-mixing technique, where a mix of different prompts is used along the denoising process. This approach is built on the premise that the denoising process is an innate coarse-to-fine synthesis, which roughly consists of three stages. In the first stage, the general configuration or layout is drafted. In the second stage, the shapes of the objects are formed. Finally, in the third stage, their fine visual details are generated.

Common in text-based image editing techniques, a challenge arises when localizing the modification of the object. Hence, we develop two novel means to localize edits. First, to preserve the shapes of other objects, we inject a localized self-attention map from the original image into the newly generated image. This injection leads to a rough alignment between the two images. Next, to further preserve appearance (*e.g.*, image background), we automatically extract labeled segmentation maps of both the original and generated images. The segmentation maps allow applying the edits locally in selected segments only. Finally, during the last denoising steps, we seamlessly blend all segments together.

We demonstrate that without any costly optimization process, our method offers the user the ability to generate object-level shape variations, while remaining faithful to the original image, either generated or real. Moreover, we show that our localization techniques are beneficial to not only object shape variations, but also to generic local image editing methods. We demonstrate the improved results achieved by integrating our localization techniques into existing image editing methods. Extensive experiments are conducted to show that our approach can create more diverse results with larger shape changes and better content preservation compared to alternative methods.

2. Related Work

2.1. Text-Guided Image Generation

Text-to-image synthesis is a longstanding problem in computer vision and computer graphics. Early works were GAN-based [48, 37, 38, 50] and were trained on small-scale datasets, typically of a single class. Recently, with the rapid progress in diffusion models [45, 22, 15], auto-regressive models [49, 16, 17, 9], and the availability of gigantic text-image datasets [43], large-scale text-to-image models [31, 39, 36, 41, 42, 5, 24] have lead to a huge leap in performance. Our work uses the publicly available Stable Diffusion model based on Latent Diffusion Models [39].

Large-scale text-to-image models allow the user to generate a gallery of images for a given text prompt. The control over the generated image, however, is limited, with attributes such as image composition, object shape, color, and texture changing depending on the arbitrary randomly sampled initial noise. Thus, recent works have introduced additional spatial conditions to the model such as segmentation maps [3, 7], bounding boxes [26, 39], keypoints [26] and other visual conditions [47, 51, 23]. Such conditions offer spatial control, but no object-level control. Alternatively, to gain object-level control, numerous text-guided image editing methods have been recently developed.

2.2. Text-Guided Image Editing

Generative models are a powerful tool for image editing [1, 44, 33, 20, 6]. With the increased performance of text-to-image diffusion models, many methods [29, 21, 10] have utilized them for text-guided image editing. A simple approach adds noise to the input image and then denoises it with a guiding prompt [4, 2, 29, 13]. To localize the edit, a user-defined mask is required [39]. Another approach manipulates internal representations of the model (*e.g.*, attention maps) during the generation process [21, 46, 32] to preserve the image layout. Other methods operate in the text encoder latent space [18] and possibly fine-tune the generator [40, 25, 19], or train a model on image pairs [8].

Recently, it has been shown that one can change an object's semantics by switching the text prompt along the denoising process [27]. This method shares similarities with our method since they also mix prompts. However, their method is limited to appearance modifications as a means to change the semantics and explicitly preserve the object's shape. In contrast, our work focuses on object geometric modifications. It should be noted that unlike all the editing methods above, generating object shape variations is not an editing task per-se, as it aims at generating multiple object-level variations for a given image while preserving the semantics of the object. Furthermore, our introduced editing localization techniques are complementary to the editing methods mentioned above as we later demonstrate.

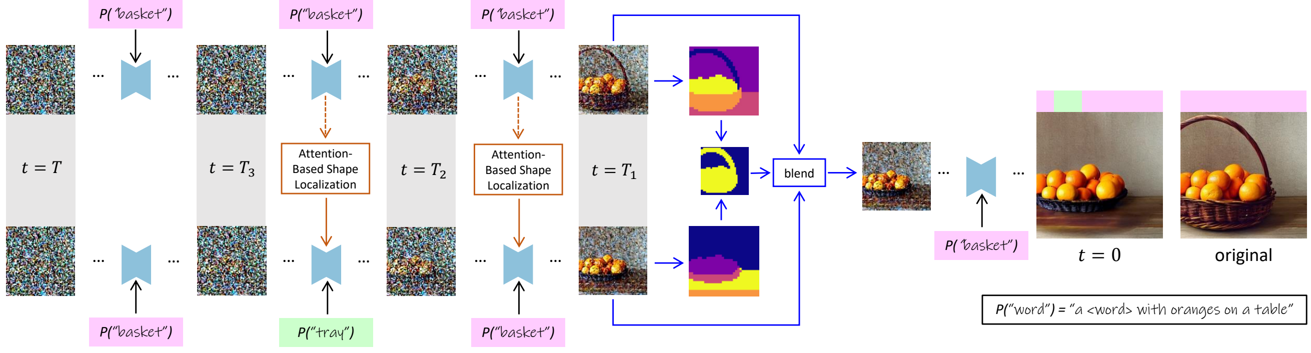

Figure 3. Given a reference image, and its corresponding denoising process, our full pipeline consists of three main building blocks. We perform Mix-and-Match in the timestamp intervals $[T, T_3], [T_3, T_2], [T_2, 0]$ using the prompt $P(w)$. For example, during the intervals $[T, T_3], [T_2, 0]$ we set $w =$ "basket", while during the interval $[T_3, T_2]$ we set $w =$ "tray". During the denoising process, we apply our attention-based shape localization technique to preserve other objects' structures (here, "table"). We do so by selectively injecting the self-attention map from the reference denoising process. At $t = T_1$, we apply controllable background preservation by segmenting the reference and the newly generated images, blend them, and proceed the denoising process.

3. Preliminaries

Latent Diffusion Models We demonstrate our method applied over the publicly available Stable Diffusion model which is built over the Latent Diffusion Models (LDM) architecture [39]. In LDM, a diffusion model operates in the latent space of a pretrained autoencoder.

The denoising network is implemented as a UNET and consists of self-attention layers followed by cross-attention layers. At each timestep t, the noised spatial code z_t is passed as input to the denoising network. The intermediate features of the network, denoted by $\phi(z_t)$, receive information from the self and cross-attention layers. The attention mechanism consists of three main components: Keys (K), Queries (Q), and Values (V). The Keys and the Queries together form an attention map, which is multiplied by the Values. In this work, we utilize the attention maps of both the self and cross-attention layers.

Cross-Attention Layers in LDM Text guidance in LDM is performed using the cross-attention mechanism. Specifically, denoting the text encoding by c, $Q = f_Q(\phi(z_t))$, $K = f_K(c)$, and $V = f_V(c)$ are obtained using learned linear layers f_Q, f_K, f_V. Each token in the text prompt corresponds to an attention map formed by the Queries and the Keys, which is multiplied by each token's Values. Therefore, intuitively, the Keys and the Queries control the placement of each token, while the Values control its shape and appearance, as we later show in the supplementary materials. It is important to note, however, that these components are not fully disentangled.

By the definition of the cross-attention mechanism, it can be observed that the encoding of the text prompt is fed only to f_K and f_V, and therefore the Keys and the Values are the only components affected directly by the text prompt.

Self-Attention Layers in LDM Self-attention layers model the relation between each pixel to all the other pixels. In LDM, each such pixel corresponds to a patch in the final generated image. Previous works [21, 46] have shown that self-attention strictly controls the image layout and objects' shapes in it, and is therefore useful to preserve the input image structure in image editing. In our work, we aim to modify the object of interest and preserve the remainder of the image. Thus, injecting the entire self-attention map does not allow for shape variations on the object of interest as was demonstrated in the above works.

4. Prompt-Mixing

To generate object variations, we propose a method that operates during inference time and does not require any optimization or model training. Given a text prompt P and an object of interest, represented by a word w, we manipulate the denoising process to obtain object-level variations.

The key enabler for generating shape variations for an object is our prompt-mixing technique. In prompt-mixing, different prompts are used in different time intervals of the denoising process. Specifically, we define three timestep intervals, $[T, T_3], [T_3, T_2], [T_2, 0]$, and use a different prompt in each interval to guide the denoising process. We denote by $P_{[t,t']}$ the prompt used in interval $[t, t']$. This technique is based upon insights related to the coarse-to-fine nature of the denoising process. These insights were also mentioned in [5, 10, 47, 14, 11], and we further analyze them in Section 4.1.

Note that common to image editing techniques, localizing the manipulated region is crucial for a successful result. In Section 5, we introduce two generic techniques for achieving localized editing, and use them in our full pipeline for object variations illustrated in Figure 3.

4.1. Denoising Diffusion Process Stages

We analyze the timestep intervals defined above, to show the type of attributes in the image controlled by each interval. This analysis demonstrates the coarse-to-fine nature of the denoising process. We show the results of the analysis in Figure 4. In each row, the three leftmost images were generated using the prompt $P(w)$ = "Two $\langle w \rangle$ on the street" along the entire denoising process, where w represents a different word in each image. All images were generated using the same initial noise. For the two rightmost images in each row we apply prompt-mixing. Specifically, we alter w in the input prompt $P(w)$ in each time interval. As mentioned earlier, the input prompt is fed into the cross-attention layers, and directly affects the Keys and Values. We use the altered prompt $P(w')$ to compute the Values, while using the original prompt $P(w)$ to compute the Keys. This design choice is explained in the supplementary materials.

In the fourth column of each row, we use $P_{[T,T_3]}(w_1)$ and $P_{[T_3,0]}(w_2)$. As can be seen, we obtain an image containing w_2 (pyramids, mugs), with the layout and background of the images containing the balls (w_1). In the fifth column, we use $P_{[T,T_3]}(w_1)$, $P_{[T_3,T_2]}(w_2)$, and $P_{[T_2,0]}(w_3)$. As observed, we now obtain an image containing w_2 (pyramids, mugs), with the layout and background of the images containing the balls (w_1), and the fine visual details (*e.g.*, texture) of w_3 (fluffies, metals). We conclude that the first interval, $[T, T_3]$, controls the image layout, the second $[T_3, T_2]$ controls the shapes of the objects, and the third $[T_2, 0]$ controls fine visual details (*e.g.*, texture). We provide additional examples in the supplementary materials.

4.2. Object Variations

Mix-and-Match Let $P(w)$ denote the prompt of the original image, where w denotes the word corresponding to the object of interest. To generate shape variations of the object of interest, we perform prompt-mixing where $P_{[T,T_3]}(w) = P_{[T_2,0]}(w)$. This is a special case of prompt-mixing, which we term *Mix-and-Match*, since we perform mixing in the second interval, and match the prompts between the first and third intervals. Formally, our shape variations are achieved by using $P_{[T,T_3]}(w)$, $P_{[T_3,T_2]}(w')$, and $P_{[T_2,0]}(w)$, where w' is a "proxy" word (explained below).

Mix-and-Match allows keeping the original image layout, formed in the first interval, the shape of w', set in the second interval, and the fine visual details of the original object represented by w during the third interval.

Proxy Words Here we describe our scheme for determining the proxy words. Intuitively, a proxy word represents a semantically close object to the object of interest, and whose shape is rather different. Motivated by the use of CLIP [34] for extracting text encodings in Stable Diffusion, we use CLIP's text space for finding the set of proxy words.

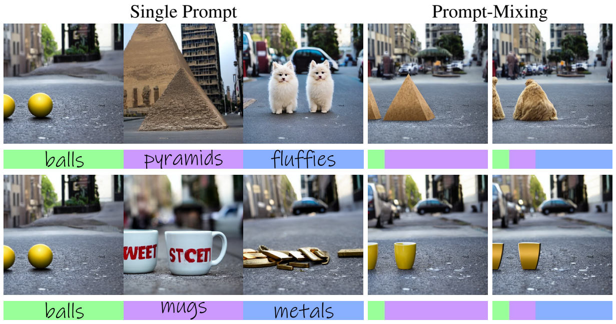

Figure 4. Prompt-mixing. For the prompt "Two $\langle w \rangle$ on the street", the colored bars under the images on the right represent the corresponding word used along the denoising process.

Given a word w, we seek to find the k most similar tokens $\{w'_1, ..., w'_k\}$ to w. To this end, we consider all tokens t of CLIP's tokenizer and embed each to the CLIP embedding space using prompts of the form $P_{\text{sim}}(t)$ = "A photo of a $\langle t \rangle$". We then take the k tokens with the smallest CLIP-space distances to the encoding of $P_{\text{sim}}(w)$, the prompt representing our object of interest. This gives us the k tokens with the closest semantic meaning to w. To take into account the input prompt context, we rank these k tokens according to CLIP's distance between $P(t)$ and $P(w)$. Finally, we define the top m tokens as proxy words. For each proxy word, we perform Mix-and-Match to obtain an image with a variation of the object of interest.

Note, that since we consider embeddings at the token level, some proxy words may not have semantic meaning alone. However, we observe that they still provide meaningful variations when used in our Mix-and-Match technique due to their close proximity in CLIP's embedding space.

5. Edit Localization

As mentioned above, localizing the edit is especially challenging when changing the object's shape. To this end, we present two techniques that assist in localizing the edit from two different aspects. These localization techniques are crucial for successfully generating object shape variations. As we shall show, other known editing methods can also benefit from integrating them, leading to better localized manipulations.

5.1. Attention-Based Shape Localization

To preserve the shapes of objects in the image, we introduce a shape localization technique based on injecting information from the self-attention maps of the source image into the self-attention maps of the generated image. In the object variations pipeline, we apply this technique to objects that we aim to preserve. Injecting the full self-attention maps, even for a few steps, accurately preserves the structure of the original image, but at the same time prevents noticeable shape changes in the object we aim to change.

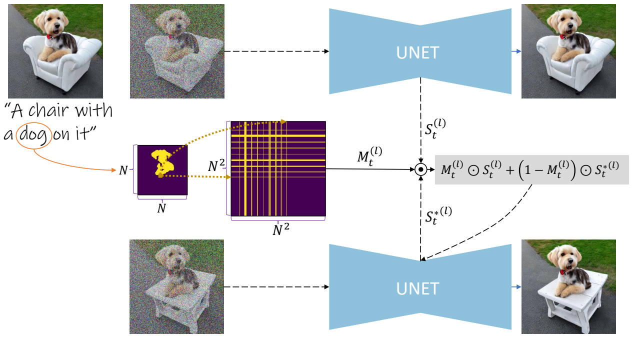
"A chair with a dog on it"
UNET
UNET
N
N
N^2
N^2
$S_t^{(l)}$
$M_t^{(l)}$
$S_t^{*(l)}$
$M_t^{(l)} \odot S_t^{(l)} + (1 - M_t^{(l)}) \odot S_t^{*(l)}$

Figure 5. Attention-based shape localization. Refer to Section 5.1 for more details.

Our technique, depicted in Figure 5, revolves around a selective injection of self-attention maps. Consider a specific self-attention layer l in the denoising network, which receives features of dimension $N \times N$, and the attention map formed by this layer, $S_t^{(l)}$, whose dimensions are $N^2 \times N^2$. The value $S_t^{(l)}[i,j]$ in the map indicates the extent to which pixel j affects pixel i. In other words, row i of the map shows the degree to which each pixel impacts pixel i, while column j displays the degree to which pixel j impacts all other pixels in the image. To preserve the shape of an object, we inject the rows and columns of the self-attention map that correspond to the pixels containing the object of interest. Specifically, for a given denoising timestep t, the self-attention layer l, and the self-attention map $S_t^{(l)}$, we define a corresponding mask $M_t^{(l)}$ by:

$$M_t^{(l)}[i,j] = \begin{cases} 1 & i \in O_t^{(l)} \text{ or } j \in O_t^{(l)} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $O_t^{(l)}$ is the set of pixels corresponding to the object we aim to preserve. We explain later how we find $O_t^{(l)}$. After defining the mask $M_t^{(l)}$, the self-attention map in the newly generated image is changed to be:

$$S_t^{*(l)} \leftarrow M_t^{(l)} \cdot S_t^{(l)} + (1 - M_t^{(l)}) S_t^{*(l)}, \tag{2}$$

where $S_t^{(l)}$ and $S_t^{*(l)}$ are the self-attention maps of the original and the newly generated images, respectively. Additional mask controls are presented in the supplementary.

To find the pixels in which an object is located (*i.e.* defining the set of pixels $O_t^{(l)}$), we leverage the cross-attention maps. These maps model the relations between each pixel in the image and each of the prompt's tokens. For an object we aim to preserve, we consider the cross-attention map of the corresponding token in the prompt. We then define the set $O_t^{(l)}$ of the object's pixels to be pixels with high activation in the cross-attention map by setting a fixed threshold.

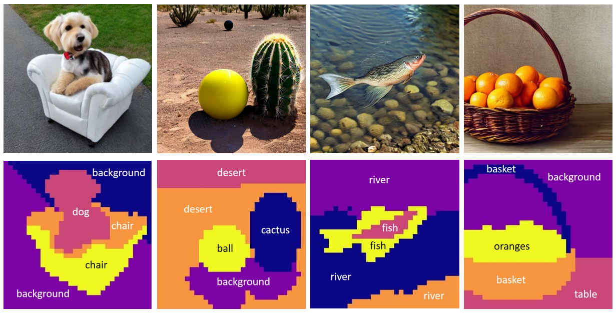

Figure 6. Segmentation maps obtained using our self-attention clustering technique. Each segment's label is determined by the cross-attention maps of the prompt nouns.

5.2. Controllable Background Preservation

As shown in previous works [46, 21], self-attention injection preserves mainly structures. Therefore, we introduce a *controllable background preservation* technique which preserves the appearance of the background and possibly some user-defined objects, specified by their corresponding nous in the input prompt. We give the user control to set the user-defined objects to be preserved since different images may require different configurations. For example, in Figure 3, a user may want to preserve the oranges if the basket's size fits, while in other cases where the size of the basket is changed, it is desirable to change the oranges to properly fill a basket with a modified size.

To preserve the appearance of the desired regions, at $t = T_1$ we blend the original and the generated images, taking the changed regions (*e.g.*, the object of interest) from the generated image and the unchanged regions (*e.g.*, background) from the original image. Next, we present our novel segmentation approach and describe the blending.

Self-segmentation We perform the segmentation on noised latent codes and, as such, off-the-shelf semantic segmentation methods cannot be applied. Hence, we introduce a segmentation method that segments the image based on self-attention maps, and labels each segment by considering cross-attention maps. The method is based on the premise that internal features of a generative model encode the information needed for segmentation [12, 53].

At $t = T_1$, we average the $32^2 \times 32^2$ self-attention maps from the entire denoising process. We obtain an attention map of size $32^2 \times 32^2$, reshape it to $32 \times 32 \times 1024$, and cluster the deep pixels with the K-Means algorithm, where each pixel is represented by the 1024 channels of the aggregated self-attention maps. Each resulting cluster corresponds to a semantic segment of the generated image. Several segmentation results are illustrated in Figure 6.

Having extracted the semantic segments, we match each segment with a noun in the input prompt. For each segment, we consider the normalized aggregated cross-attention maps of the prompt's nouns, and match each segment to a noun as follows. For segment i that cor-

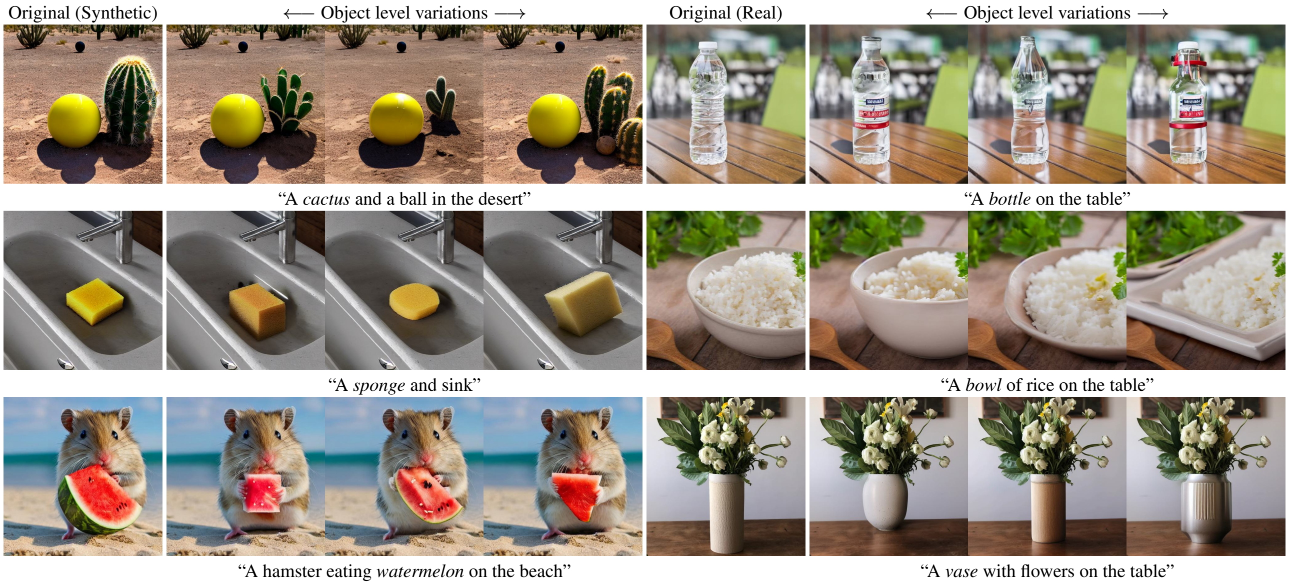
Original (Synthetic)
⟵ Object level variations ⟶
Original (Real)
⟵ Object level variations ⟶
"A *cactus* and a ball in the desert"
"A *bottle* on the table"
"A *sponge* and sink"
"A *bowl* of rice on the table"
"A hamster eating *watermelon* on the beach"
"A *vase* with flowers on the table"

Figure 7. Object-level variations for various scenes, synthetic and real (inverted [30]). For each scene, the leftmost image is the original one. The *emphasized* word corresponds to the modified object. As observed, our method generates various shape variations for each object.

responds to a binary mask M_i, and for a noun n in the prompt that corresponds to a normalized aggregated cross-attention mask A_n, we calculate a score $s(i,n) = \sum(M_i \cdot A_n)/\sum(M_i)$. We label segment i with $\arg\max_n s(i,n)$ if $\max_n s(i,n) > \sigma$ and label it as background otherwise. The threshold value σ is fixed across all our experiments.

Blending the original and generated images We use the segmentation map and the corresponding labels to overwrite relevant regions in the newly generated image. We retain pixels from the original image only if they are labeled as background or as a user-defined object in both the original and new images. This approach helps to overcome shape modifications in the object of interest, as illustrated by the example of the basket in Figure 3, where the handle region is taken from the newly generated image. After blending the latent images, we proceed with the denoising process.

6. Experiments

6.1. Object Shape Variations

We perform experiments to assess the effectiveness of our method in generating object-level shape variations, and compare it to other existing methods. Specifically, we compare our method with methods that directly provide variations for an image by changing seed (inpainting [39], SDEdit [29]), as well as recent text-guided image editing baselines (P2P [21], I-Pix2Pix [8], PnP [46], Zero-shot Image2Image Translation [32], Imagic [25]).

It should be noted that our method differs from image editing approaches in two key aspects. First, most editing methods change the object to an object of a different class, whereas our method keeps class and offers alternative options for the same object. Second, editing methods require the user to provide specific instructions on the exact object they would like to obtain, whereas our method allows for an open-ended exploration without relying on such instructions. Therefore, we adopted two different approaches when comparing our method with image editing methods: (i) we refined the input prompt, and (ii) we replaced the word representing the object of interest with proxy words. The second approach, which involves using proxy words, can also be viewed as an ablation study for Mix-and-Match.

We separately evaluate three important aspects of our method. First, we aim at achieving high diversity in the shape of the object of interest. Second, the object class should remain the same. We term this evaluation objective as object faithfulness. Third, regions of the image other than the object of interest should be preserved.

Qualitative Experiments In Figure 7, we show a gallery of images. Additional results are provided in the supplementary materials. Observe the shape diversity in our method's results and the preservation of the original image.

In Figure 8 we compare our method with inpainting [39] and SDEdit [29] by sampling different seeds to get variations. For inpainting, we use a mask obtained from our segmentation technique. Similarly to Figure 2, we observe that applying inpainting results mainly in texture changes, while SDEdit performs small shape changes but does not preserve the background and other objects (*e.g*., cat).

Comparison to text-guided image editing methods is presented in Figure 9. For each method, we guided the editing with refined prompts ("eggchair", "stool") and with our automatic proxy words ("cart", "bed", "stool"). Note that the refined prompts were manually chosen to be types of

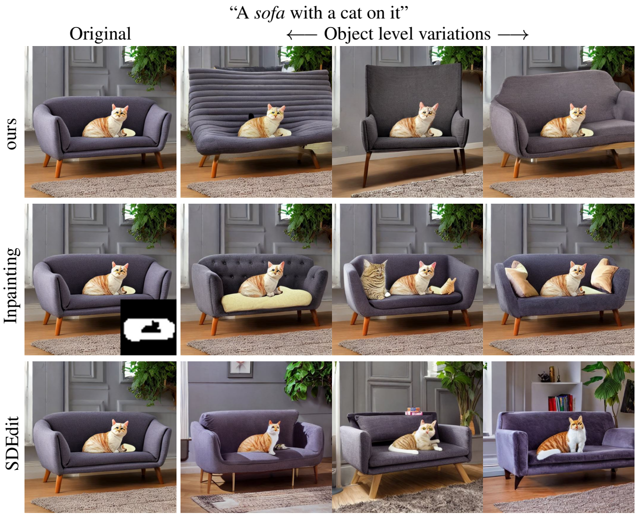

Figure 8. Comparing to methods that generate variations. As can be seen, our method generates varied shape variations, while other methods change mostly the texture.

chairs. It should be noted that refining prompts for objects that do not have subtypes (*e.g.*, basket) is more challenging. In P2P [21], we show two versions of results, which differ in the number of self-attention injection steps (10%, 40%). Additional information about the configuration of each method and comparison to additional methods are provided in the supplementary materials.

Our diverse results in Figure 9 remain faithful to the class of chairs while preserving the rest of the image. Not surprisingly, editing methods struggle at keeping the chair when a different object is specified in the prompt. For example, when replacing the chair with a cart, wheels are added. In our method, thanks to Mix-and-Match, we take the shape of the wheels but the fine visual details of the chair.

As can be seen in Figure 9, injecting self-attention maps for 40% of the denoising steps in P2P [21] prevents change in the object of interest (here, a chair). Conversely, when injecting self-attention maps for 10% of the denoising steps, P2P struggles to preserve the dog and the background. Instruct-Pix2Pix [8] results are diverse but inferior to our method in image preservation (eggchair) and faithfulness (bed, stool). Plug-and-Play [46] struggles at performing shape changes as it injects the entire self-attention maps along the denoising process. Compared to Zero-shot Image2Image Translation [32], which requires 1000 prompts with the proxy word, our method preserves the dog colors better, and allows for more diverse shapes (see the stool). We also compared our method to Imagic [25] using Imagen [41]. While Imagic produces high-quality results, our method has advantages in preserving the background and being more faithful to the class of a chair. Additionally, our method is more time-efficient than Imagic's optimization-based approach.

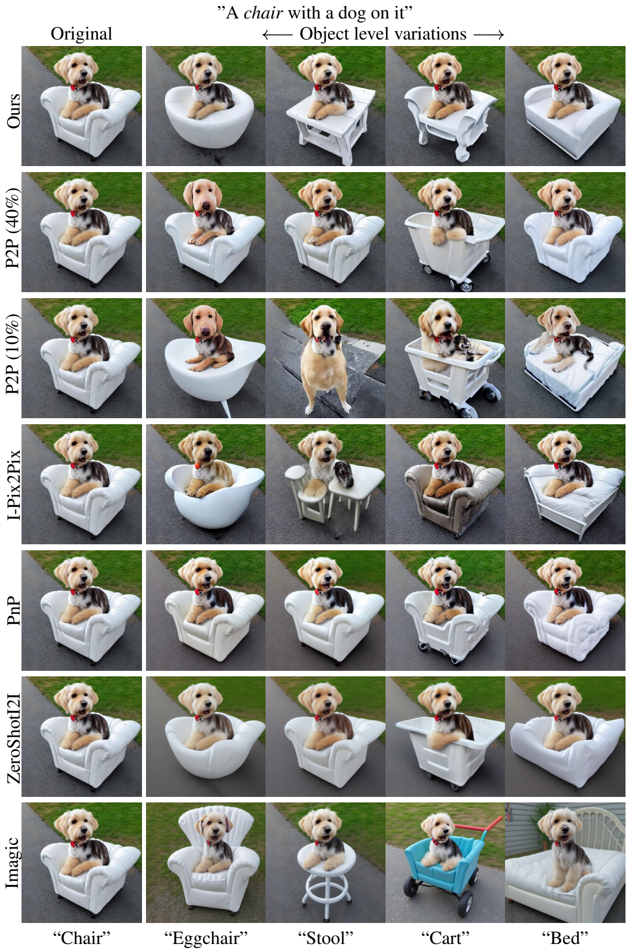

Figure 9. Comparisons to text-guided editing methods. In each column, we show the results of a different word that replaces the original word "chair" in the prompt. We apply two different percentages of self-attention map injection steps in P2P.

Quantitative Experiments Given a collection of object-level variations of an image, we measure each objective as follows. For shape diversity, we extract a mask of the object of interest by using CLIPSeg [28], and average the IoU of each pair of masks in the collection. We define the diversity as $1 - \text{IoU}$. We measure faithfulness by employing CLIP [34] and computing cosine similarity between an averaged embedding of images containing the object of interest and each of the images in the collection. To quantify image preservation we utilize LPIPS [52].

We create a dataset of 150 images with various prompts, and generate 20 variations of the object of interest with each method. More details about the construction of the dataset are provided in the supplementary materials. We test other methods with proxy words (all methods), random seeds (inpainting, SDEdit), and prompt refinement (I-Pix2Pix, P2P).

In Figure 10 we present quantitative results. Our method achieves a good trade-off between diversity, faithfulness, and image preservation. Methods with higher diversity

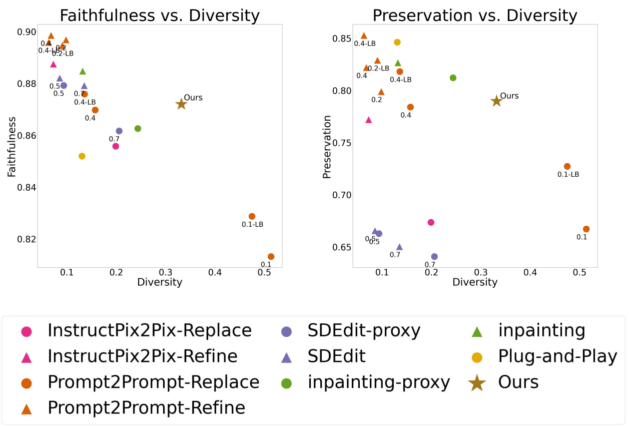

Figure 10. Quantitative comparison with other methods. The graphs above illustrate the trade-off between diversity, faithfulness, and preservation. Refer to Section 6.1 for more details.

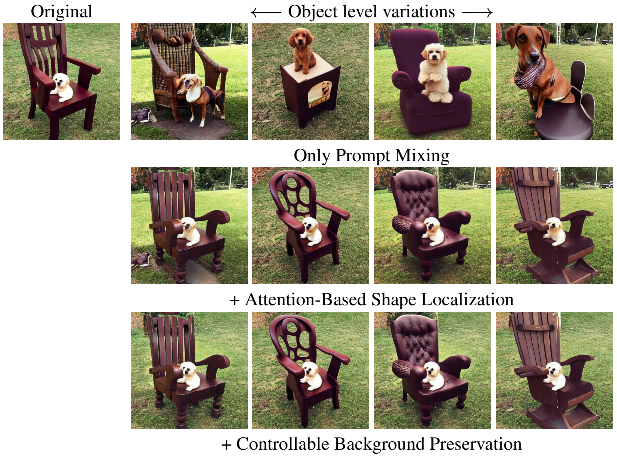
Original
⟵ Object level variations ⟶
Only Prompt Mixing
+ Attention-Based Shape Localization
+ Controllable Background Preservation

Figure 11. Ablating our full object variations pipeline. Original image was crated using the prompt “A *chair* with a dog on it”.

scores than ours do not preserve the original image and are not faithful to the object of interest, as was also demonstrated in the qualitative comparison. As shown in the graph, methods with better preservation or faithfulness scores than ours, hardly change the shape of the object.

Ablation Studies We ablate our full pipeline of generating object-level shape variations of a given image, showing the necessity of each part. We present the results in Figure 11. In the first row, we show the results of Mix-and-Match without the localization techniques. As can be seen, Mix-and-Match alone fails to preserve the dog and the background. Adding the attention-based shape localization technique, where we create a mask for the self-attention map based on the word “dog”, allows for the preservation of the dog. Finally, adding the controllable background preservation technique keeps the background of the original image.

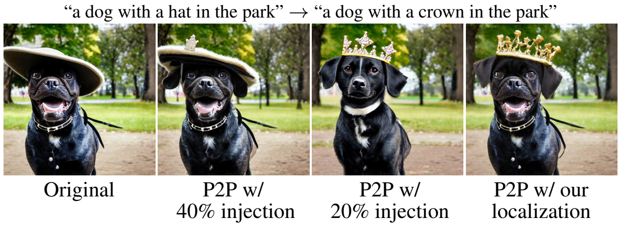

Figure 12. Comparison between P2P, which injects the entire self-attention maps, to P2P integrated with our attention-based shape localization technique. As demonstrated above, with our localization method P2P replaces the hat while preserving the dog.

6.2. Edit Localization

We integrate our localization techniques with existing text-to-image methods to show improved results when using these methods in conjunction with our techniques.

Attention-Based Shape Localization Previous methods [21, 46] have injected the entire self-attention map to preserve the shapes of the original image. To demonstrate the effectiveness of our attention-based shape localization technique, we integrate it with P2P [21] and show the results in Figure 12 where we aim to change the hat of the dog into a crown. The figure shows the original generated image on the left, followed by P2P where we injected the entire self-attention map during 40% and 20% of the denoising steps, and P2P with our localization technique. As can be seen, using P2P involves a tradeoff between accurately changing the hat into a crown and preserving the original dog. By integrating our method, we were able to selectively inject only the rows and columns that corresponded to the dog, achieving the desired transformation of the hat into a crown while preserving the original shape of the dog.

Controllable Background Preservation We test our background preservation technique with PnP [46], P2P [21], and SDEdit [29] and show the results in Figure 13. For P2P we use their local blending. To integrate each method with our technique, we first invert each image [30]. Then, we segment and label each segment of the image with our technique, and create a mask of the main object in the image. At step $t = 35$ of the denoising process we perform blending between the edited image and the original one, taking the background from the original image, and the object from the edited image.

As can be seen, our method achieves plausible segmentation maps even for inverted images. For PnP and SDEdit, our method allows editing the object while keeping the background as in the original image. In P2P we see that our technique localizes the edit better, removing the cat’s tail and feet from the image.

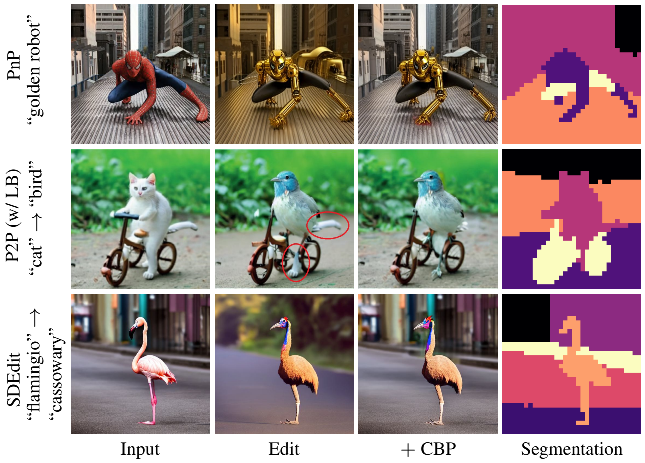

Figure 13. Our controllable background preservation (CBP) technique integrated with editing methods to localize their edits better. In P2P [21] our mask is more accurate as observed by the cat's tail and feet that are added to the bird without CBP.

7. Discussion and Conclusion

We have presented a method for exploring object-level shape variations in an image, which addresses two main technical challenges: changing the shape of an object and localizing the change. To achieve this, we introduced a Mix-and-Match technique to generate shape variations, and built upon self-attention maps injection to preserve the original image structure, while enabling changes to the object of interest. Furthermore, we demonstrated how the geometric information encoded in self-attention maps can be used for image segmentation, which allows for guiding the preservation of the background. While our method produces plausible variations of an image, we acknowledge that automatic proxy-words may fail at times. In the future, we would like to develop means to explore a continuous words space rather than the current discrete one.

Acknowledgement

We thank Ron Mokady and Matan Cohen for their early feedback and helpful suggestions. We thank Shiran Zada for assistance with Imagen comparisons. We also want to give a special thanks Yuval Alaluf and Rinon Gal for their comments and discussions throughout the writing of this paper. This work was supported by a research gift from Meta, by the Deutsch Foundation, the Yandex Initiative in AI, the Alon Fellowship, and by the Israel Science Foundation under Grant No. 2492/20.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3), May 2021. 2

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022. 2

[3] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xiaoyue Yin. Spatext: Spatio-textual representation for controllable image generation. *ArXiv*, abs/2211.14305, 2022. 2

[4] 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

[5] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2, 3

[6] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European Conference on Computer Vision*, pages 707–723. Springer, 2022. 2

[7] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *arXiv preprint arXiv:2302.08113*, 2023. 2

[8] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. November 2022. 2, 6, 7

[9] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11305–11315, 2022. 2

[10] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ArXiv*, abs/2301.13826, 2023. 2, 3

[11] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14367–14376, 2021. 3

[12] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of GANs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[13] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *ArXiv*, abs/2210.11427, 2022. 2

[14] Giannis Daras and Alex Dimakis. Multiresolution textual inversion. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. 3

[15] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021. 2

[16] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao,

Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In *Neural Information Processing Systems*, 2021. 2

[17] 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*, abs/2203.13131, 2022. 2

[18] 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

[19] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Designing an encoder for fast personalization of text-to-image models. *ArXiv*, abs/2302.12228, 2023. 2

[20] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2021. 2

[21] 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, 3, 5, 6, 7, 8, 9

[22] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. 2

[23] Lianghua Huang, Di Chen, Yu Liu, Shen Yujun, Deli Zhao, and Zhou Jingren. Composer: Creative and controllable image synthesis with composable conditions. 2023. 2

[24] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[25] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022. 2, 6, 7

[26] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. *ArXiv*, abs/2301.07093, 2023. 2

[27] Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. Magicmix: Semantic mixing with diffusion models. *arXiv preprint arXiv:2210.16056*, 2022. 2

[28] Timo Lüddecke and Alexander S. Ecker. Image segmentation using text and image prompts. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7076–7086, 2021. 7

[29] 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*, 2022. 1, 2, 6, 8

[30] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022. 6, 8

[31] 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. In *International Conference on Machine Learning*, 2021. 2

[32] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *ArXiv*, abs/2302.03027, 2023. 2, 6, 7

[33] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, October 2021. 2

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 4, 7

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 1

[36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ArXiv*, abs/2102.12092, 2021. 2

[37] Scott E. Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. *ArXiv*, abs/1610.02454, 2016. 2

[38] Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. *Advances in neural information processing systems*, 29, 2016. 2

[39] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021. 1, 2, 3, 6

[40] 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. 2022. 2

[41] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022. 1, 2, 7

[42] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. volume abs/2301.09515, 2023. 2

[43] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training

next generation image-text models. *ArXiv*, abs/2210.08402, 2022. 2

[44] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 2

[45] Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015. 2

[46] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. *arXiv preprint arXiv:2211.12572*, 2022. 2, 3, 5, 6, 7, 8

[47] Andrey Voynov, Kfir Abernan, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. 2022. 2, 3

[48] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2017. 2

[49] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Benton C. Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *ArXiv*, abs/2206.10789, 2022. 2

[50] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5908–5916, 2016. 2

[51] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[53] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021. 5