Continuous Layout Editing of Single Images with Diffusion Models

ZHIYUAN ZHANG*, City University of Hong Kong, China
ZHITONG HUANG*, City University of Hong Kong, China
JING LIAO†, City University of Hong Kong, China

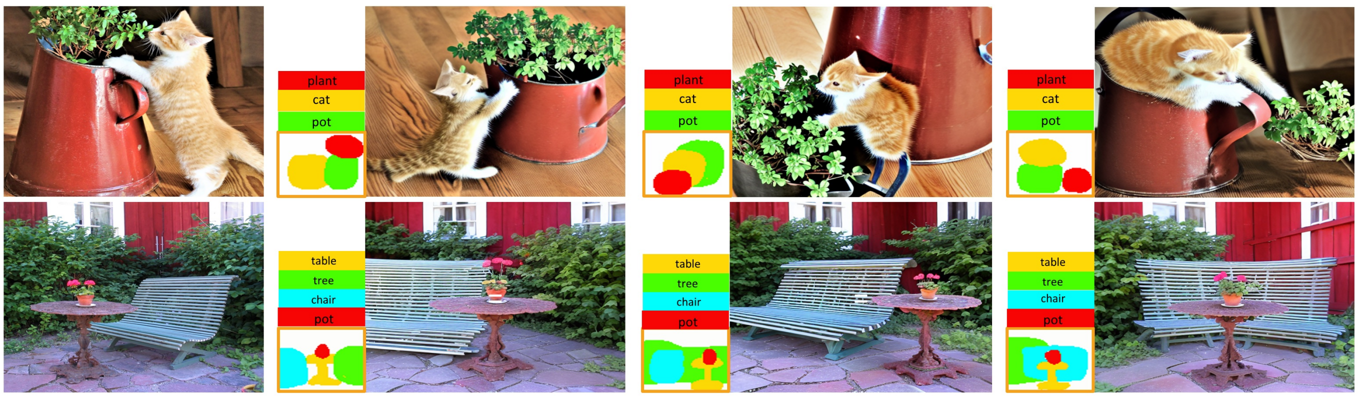


Fig. 1. Our method can continuously edit the layout of a single image with multiple objects. The first column shows the input image, followed by three columns of edited results. The edited results preserve the visual properties of the input image while remaining faithful to the layout map provided by the user.

Recent advancements in large-scale text-to-image diffusion models have enabled many applications in image editing. However, none of these methods have been able to edit the layout of single existing images. To address this gap, we propose the first framework for layout editing of a single image while preserving its visual properties, thus allowing for continuous editing on a single image. Our approach is achieved through two key modules. First, to preserve the characteristics of multiple objects within an image, we disentangle the concepts of different objects and embed them into separate textual tokens using a novel method called masked textual inversion. Next, we propose a training-free optimization method to perform layout control for a pre-trained diffusion model, which allows us to regenerate images with learned concepts and align them with user-specified layouts. As the first framework to edit the layout of existing images, we demonstrate that our method is effective and outperforms other baselines that were modified to support this task. Our code will be freely available for public use upon acceptance at *https://bestzzhang.github.io/continuous-layout-editing*.

CCS Concepts: • **Computing methodologies** → **Image manipulation**; *Graphics systems and interfaces*; *Neural networks.*

1 INTRODUCTION

Recent advances in text-to-image generation have made significant progress through the use of diffusion models trained on large-scale datasets [Nichol et al. 2021; Rombach et al. 2022; Saharia et al. 2022]. However, due to the inherent ambiguity of text and its limitations in expressing precise spatial relationships in the image space, controlling the layout of generated images is still a challenge for these large-scale text-to-image models. To address this issue, some latest methods have been proposed to enable layout control in image generation. These methods are typically based on pre-trained diffusion models, which either incorporate layout guidance as a new condition through fine-tuning [Avrahami et al. 2023; Li et al. 2023; Zhang and Agrawala 2023] or optimize the noise diffusion process on-the-fly to achieve layout control [Bar-Tal et al. 2023].

Despite the success of existing layout control methods [Avrahami et al. 2023; Bar-Tal et al. 2023; Li et al. 2023; Zhang and Agrawala 2023] in generating new images with controlled layouts, they are unable to rearrange and edit the layout of existing images. In practice, users may want to continuously edit the positions of objects in an existing image without altering its visual properties. For example, as illustrated in the first example of Figure 1, a user may want to experiment with different layout options to find the best arrangement of a cat and a pot in an image. However, previous methods do not support this functionality since their layout control does not take into account the input image, and a new image with different cat and pot will be generated for each specified layout. To fill this gap, we propose the first framework for continuous layout editing of single input images.

One of the key challenges in continuous layout editing is preserving the visual properties of the input image, which requires learning concepts for multiple objects within a single image and using the learned concepts to regenerate new images under different layouts. While some pioneer textual inversion methods [Gal et al. 2022; Ruiz et al. 2023] have proposed fine-tuning a text token embedding of pre-trained text-to-image diffusion models to learn the concept of an object from multiple images containing the same object, they are limited in their ability to learn multiple objects within a single image. To overcome this limitation, we propose a novel approach, called **masked textual inversion**, that disentangles the concepts

*Both authors contributed equally to this research.
†Corresponding author.

Authors' addresses: Zhiyuan Zhang, zzhang452-c@my.cityu.edu.hk, City University of Hong Kong, Hong Kong SAR, China; Zhitong Huang, luckyhzt@gmail.com, City University of Hong Kong, Hong Kong SAR, China; Jing Liao, jingliao@cityu.edu.hk, City University of Hong Kong, Hong Kong SAR, China.

of different objects within a single image and embeds them into separate tokens. By adding masks to the regions of each object, our method ensures that the visual characteristics of each object are effectively learned by the corresponding token embedding.

After learning the concepts of multiple objects within a single image, the next challenge is to control the positions of these objects to align with the desired layout. P2P [Hertz et al. 2022] suggests that the cross-attention of a pretrained text-to-image diffusion model can represent the position of the generated object associated with the corresponding text token and Attend-and-Excite [Chefer et al. 2023] further utilizes the cross-attention to ensure the generation of objects. Inspired by these papers, we propose a novel, **training-free layout editing** method that iteratively optimizes the cross-attention during the diffusion process. This optimization is guided by a region loss that prioritizes the alignment of the specified object with its designated region in the layout by encouraging higher cross-attention between the object's text embedding and its corresponding region than with any other region in the image. Our approach enables precise and flexible control over the positions of objects in the image, without requiring additional training or fine-tuning of the pre-trained diffusion model.

Extensive experiments and perceptual studies have demonstrated that our proposed method is effective in editing the layout of single images and outperforms other baseline methods (modified to perform this task). We also provide a user interface for interactive layout editing to assist in the design process. In summary, our contributions to the field are as follows:

- We propose the first framework which supports continuous layout editing of single images.
- We present a masked textual inversion method to learn disentangled concepts of multiple objects within single images.
- We propose a training-free optimization method to perform layout control with diffusion models.

2 RELATED WORKS

In this section, we will review works related to our method, which includes diffusion models, image editing with diffusion models through textual inversion, and layout control with diffusion models.

2.1 Diffusion Models

Diffusion models have become one of the most popular generative models due to their impressive quality in image generation. The original DDPM [Ho et al. 2020] simulates a Markovian process where Gaussian noise is added to clean images x_0 to create the noisy image x_t in the forward process. Then, a model is trained to predict and remove the noise in x_t to generate images. To accelerate the denoising process, DDIM [Song et al. 2020] converts DDPM into a non-Markovian process, which requires no additional training.

Recently, text-to-image diffusion models [Balaji et al. 2022; Ramesh et al. 2022; Saharia et al. 2022] trained on large-scale datasets have gained significant attention due to their ability to generate diverse high-quality images with text prompts. Among them, Stable Diffusion [Rombach et al. 2022] operates the diffusion process in latent space instead of pixel space, allowing it to generate high-resolution images.

2.2 Image Editing with Textual Inversion

By leveraging the power of pretrained text-to-image diffusion models, many image editing methods have been derived. Among them, a major category is to learn the concepts of objects or styles into text tokens and then generate new images with the extracted concepts. Pioneer works in this category include Textual inversion[Gal et al. 2022] and DreamBooth[Ruiz et al. 2023]. Textual inversion [Gal et al. 2022] embeds concepts of objects into pseudo-words, while DreamBooth [Ruiz et al. 2023] further finetunes the UNet to learn more details. However, these two methods can only extract a single common concept from multiple images. Multi-Concept [Bar-Tal et al. 2023] extends to learn multiple objects and explores the most effective layers in UNet to be finetuned; however, each concept still needs to be learned from multiple images. Overall, a method that can learn multiple concepts from a single image is still under exploration, and our masked textual inversion fills this gap.

2.3 Layout Control with Diffusion Models

Due to the sparsity and ambiguity of text descriptions, it is difficult to precisely control the layout of generated images by pretrained text-to-image diffusion models. To address this limitation, some layout control methods [Nichol et al. 2021; Rombach et al. 2022; Saharia et al. 2022] based on diffusion models have been proposed, which can be divided into two categories.

The first category requires finetuning the pretrained text-to-image model to incorporate layout guidance as an extra condition besides text. Spatext [Avrahami et al. 2023] proposes to convert the concept of each object into CLIP image features with unCLIP [Ramesh et al. 2022], which are then stacked at the target positions of that object to form a spatio-textual representation. This layout condition is concatenated with the noisy latent to control the layout during the denoising process. ControlNet [Zhang and Agrawala 2023] inserts additional conditions, such as the semantic maps for layout control, by utilizing a trainable copy of the original UNet model. The conditional copy and the original model are fused in intermediate layers to generate a conditioned output. GLIGEN [Li et al. 2023] adds additional trainable gated self-attention layers that take the information of layout conditions (i.e., bounding boxes) to control the layout of the generated image. A common limitation of these methods is that additional modules are added to the original UNet, and datasets of paired data (e.g., images and corresponding semantic maps) are required to finetune the diffusion model and added modules. Also, their capabilities are restricted by the training data.

The second category explores training-free layout control with on-the-flight optimization. The representative work is MultiDiffusion [Bar-Tal et al. 2023], which denoises different crops of each object locally and then fuses the results globally for each denoising step. Compared with computing multiple denoising directions for each object, which may cause artifacts and discontinuities at the boundary of objects, our training-free layout editing method directly denoises the whole image and optimizes the image latent for layout control, to avoid the gaps and discontinuities among multiple denoising directions.

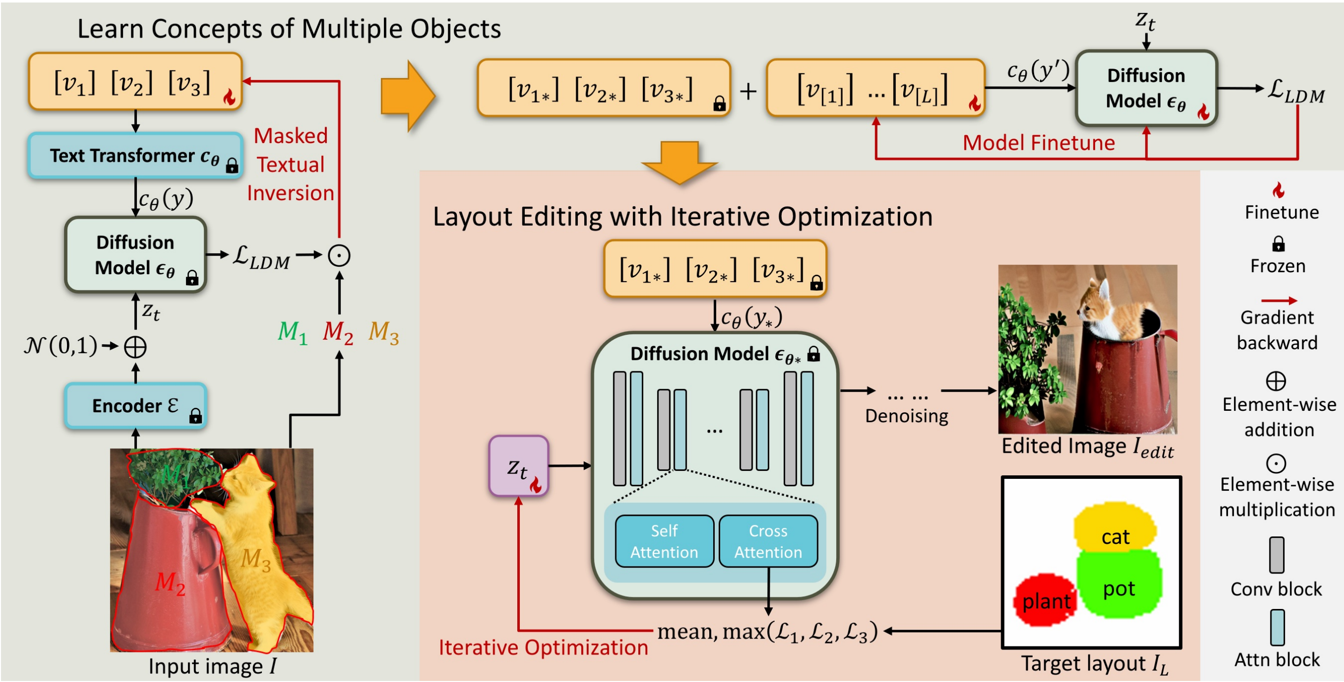


Fig. 2. Overall framework of our method.

Both categories of methods still focus on the layout control of generated images, which cannot be used to edit the layout of existing images. Our proposed framework targets this gap.

3 METHODOLOGY

Preliminaries. Our method is implemented with Stable Diffusion [Rombach et al. 2022], a large-scale text-to-image model. Therefore, before discussing our method, we first introduce Latent Diffusion Models (LDMs) [Rombach et al. 2022], which is the theory of Stable Diffusion. LDMs consist of two key stages. In the first stage, the encoder of an autoencoder maps the image to latent space $z_0 = \mathcal{E}(I)$, and a decoder maps it back to image $\mathcal{D}(\mathcal{E}(I)) \approx I$. In the second stage, a diffusion model ϵ_θ is trained to denoise the noised latent $z_t = \sqrt{\alpha_t} z_0 + \sqrt{1-\alpha_t}\epsilon$, where α_t is a factor to determine noise level for each timestep t, and $\epsilon \sim \mathcal{N}(0,1)$ is Gaussian noise. Then the diffusion model is trained to predict the added Gaussian noise with the LDM loss [Ho et al. 2020]:

$$\mathcal{L}_{LDM} := \mathbb{E}_{z_0 \in \mathcal{E}(I), y, \epsilon \sim \mathcal{N}(0,1), t}[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))\|_2^2] \tag{1}$$

where y is the input text condition, and c_θ is the text encoder.

Overview. As shown in Fig. 2, our methods can be divided into two stages. In the first stage, we learn the concepts of multiple objects from a single input image I into text tokens $v_1, v_2, ..., v_N$ with masked textual inversion, where the regions of each object are specified by masks $M_1, M_2, ..., M_N$. We further learn the details of the objects by finetuning the diffusion model ϵ_θ and optimizing the appended text tokens $v_{[1]}, ..., v_{[L]}$. After the first stage, we get the optimized text tokens for objects $y_* = [v_{1*}, ..., v_{N*}]$ and the finetuned model $\epsilon_{\theta*}$. Then, in the second stage, we rearrange the positions of the objects according to the user-specified layout map I_L through a training-free layout editing method with optimization:

$$I_{edit} = \texttt{Layout-control}(I, c_\theta(y_*), \epsilon_{\theta*}, I_L) \tag{2}$$

3.1 Learn Concepts of Multiple Objects within Single Image

To rearrange the layout of an input image, we first need to extract the concepts of multiple objects within the single input image to best preserve their visual characteristics, such as shape, color, and texture. We propose using masked textual inversion to learn and embed the concept of each individual object into a unique text token. We then fine-tune the diffusion model to better grasp the detailed texture of the learned objects.

3.1.1 Masked textual inversion. Original technique of text inversion only supports learning the concept of a single object from a set of images (typically 3-5). However, in our applications, we need to learn multiple concepts from a single image. As observed in [Avrahami et al. 2023], the latent vector $z_0 = \mathcal{E}(I)$ encoded from the input image with the autoencoder [Rombach et al. 2022] has local property in the spatial dimension and the encoder performs like a down-sampler. Therefore, we can disentangle the concepts of different objects by simply applying a spatial mask. Instead of calculating the loss of the whole latent, we only propagate the loss within the region of the object to update the corresponding text token:

$$v_{k*} = \arg\min_{v_k} \mathbb{E}_{z_0 \in \mathcal{E}(I), y, \epsilon \sim \mathcal{N}(0,1), t} [M_k \odot \|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))\|_2^2] \tag{3}$$

where M_k is the mask of the k-th object ($k = 1, ..., N$ is the index of the N objects), v_k is the corresponding text token of the k-th object, and the input text condition y consists of text tokens $[v_1, ..., v_N]$. The mask can either be generated coarsely by hand or automatically using CLIP Segmentation [Lüddecke and Ecker 2022]. We repeat this process independently for each of the N objects to optimize the text tokens for each object. To avoid overfitting, we only run each optimization for 200 steps, which is much less than the original textual inversion 3000-5000 steps).

3.1.2 Model fintuning. A single text token can only store limited information of an object, which may cause obvious distortion or artifacts during sampling. Therefore, we propose further fine-tuning the denoising network ϵ_θ to better grasp the detailed texture of the objects. In multi-concept customization [Kumari et al. 2023], it was discovered that fine-tuning the key and value projections in cross-attention layers of the denoising network is the most effective way to achieve this. We set the input text condition to the optimized tokens. To avoid overfitting, we further append additional L trainable tokens at the end of the text condition and apply prior preservation loss as in [Kumari et al. 2023]. The training objective is as follows:

$$v_{[1:L]*}, \epsilon_{\theta*} = \underset{v_{[1:L]}, \epsilon_\theta}{\arg\min}\ \mathbb{E}_{z_0 \in \mathcal{E}(I), y', \epsilon \sim \mathcal{N}(0,1), t} \left[\| \epsilon - \epsilon_\theta(z_t, t, c_\theta(y')) \|_2^2 \right] \quad (4)$$

where $y' = [v_{1*}, ..., v_{N*}, v_{[1:L]}]$ and $v_{[1:L]}$ are the trainable appended tokens. Only key and value projections of cross-attention layers in $\epsilon_{\theta*}$ are finetuned.

3.2 Training-Free Layout Editing

With the learned concepts of multiple objects $v_{1*}, ..., v_{N*}$ and the fine-tuned model $\epsilon_{\theta*}$, we rearrange the positions of the objects to edit the layout. A straightforward way to control the layout is to add a new layout condition to a stable diffusion model, as in [Avrahami et al. 2023; Li et al. 2023; Zhang and Agrawala 2023]. However, this approach requires further fine-tuning with a additional dataset. Instead, we propose a training-free method to control the layout to avoid dataset collection.

As discovered in P2P [Hertz et al. 2022], the cross-attention in the denoising network of text-to-image diffusion models can reflect the positions of each generated object specified by the corresponding text token, which is calculated from:

$$A_l = \sigma(Q_l(z_t^l) K_l(y)^T) \quad (5)$$

where A_l is the cross-attention at layer l of the denoising network, Q_l, K_l are the query and key projections, σ is the softmax operation along the dimension of text embedding y, and z_t^l is the intermediate feature of the image latent. The calculated attention A_l has the size of $h_l \times w_l \times d$, where h_l and w_l is the spatial dimension of the feature z_t^l and d is the length of input text tokens. More specifically, for each text token, we could get an attention map of size $h_l \times w_l$ which reflects the relevance to its concept. For example, in the attention map with the text "cat", the positions within the area containing the cat should have larger values than other positions.
Therefore, we could optimize z_t towards the target that the desired area of the object has large values. Previous study [Hertz et al. 2022] has shown that the layers of resolution 16×16 contain the most meaningful semantic information. Therefore, we choose l to be the layers with $h_l = w_l = 16$. As shown in Fig. 3, for each layer l, each channel of the cross attention A_l represents the spatial relevance to the corresponding text token. For example, if we want to optimize the position of the object represented by v_{2*}, we can extract the corresponding channel $A_{l,2}$ and multiply it with the target region mask of v_{2*} (i.e., the yellow mask). Finally, the region loss can be calculated with the summation of the values within the mask (yellow region) and of all positions (black and yellow regions):

$$\mathcal{L}_k = 1 - \frac{\sum_i (M_k^* \odot \sum_l A_{l,k})}{\sum_i \sum_l A_{l,k}} \quad (6)$$

$$\mathcal{L} = \frac{1}{N} \sum_{k=1}^{N} \mathcal{L}_k + \max(\mathcal{L}_1, ..., \mathcal{L}_N) \quad (7)$$

where $\mathcal{L}_k$ is the loss for the k-th object with target position specified by M_k^*, $A_{l,k}$ is the cross-attention map with the optimized token v_{k*} (from Eqn. (3)) at layer l, and i is the spatial position in $A_{l,k}$. We use

ALGORITHM 1: Denoising process with layout control

Input: The sequence of optimized text tokens for objects y_*, the initial image I^*, a set of object masks M, optimization learning rate α_t, a set of thresholds $\{Q_t\}$, the timestep to stop optimization and blending t_{opt} and t_{bld}.
Output: An edited Image I

1 Encode input image: $z_0^* = \mathcal{E}(I^*)$;
2 Initialize with Gaussian noise: $z_T = \mathcal{N}(0, I)$;
3 **for** $t = T, ..., 1$ **do**
    // Iterative optimization:
4   **if** $t \geq t_{opt}$ **then**
5   Get cross attention: $A \leftarrow \epsilon_{\theta*}(z_t, c_\theta(y_*), t)$;
6   **for** $k = 1, ..., N$ **do**
7             $A_{l,k} \leftarrow A_l[:, :, k]$;
8   Calculate $\mathcal{L}_k$ with M_k as in Eqn. (6);
9   **end**
10   Calculate mean-max loss $\mathcal{L}$ as in Eqn. (7);
11   **if** $\mathcal{L} > 1 - Q_t$ **then**
12   Update z_t with: $z_t \leftarrow z_t - \alpha_t \cdot \nabla_{z_t} \mathcal{L}$;
13   **if** Reach maximum optimization steps **then**
14   **go to** 22;
15   **end**
16   **else**
17   **go to** 5;
18   **end**
19   **end**
20   **end**
    // Background Blending:
21   **if** $t \geq t_{bld}$ **then**
22   Add noise to original latent: $z_t^* = addnoise(z_0^*, t)$;
23   Get mask of background: $M_{bg} = 1 - \sum_k M_k$;
24   Blending: $z_t \leftarrow M_{bg} \odot z_t^* + (1 - M_{bg}) \odot z_t$;
25   **end**
26     $z_{t-1} \leftarrow denoising(z_t, c_\theta(y_*), t)$;
27 **end**
28 Decode the edited image: $I = \mathcal{D}(z_0)$

the mean value together with the maximum value of $\mathcal{L}_k$, so that the model can control the positions of each object and simultaneously focus on the object with a large loss.

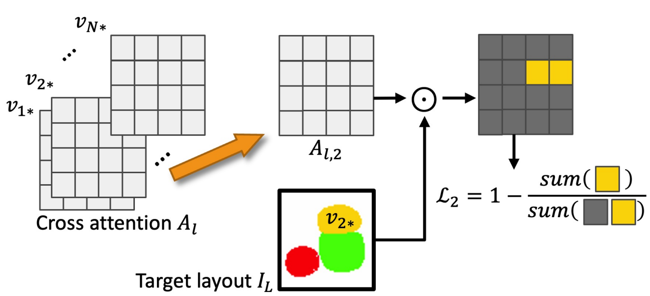


Fig. 3. Region loss calculated from cross attention. It encourages higher cross-attention between the object's text embedding and its corresponding region than with any other region.

We optimize the latent z_t with the loss $\mathcal{L}$ in Eqn. (7) only at large timesteps, i.e., $t >= 0.5$, which is enough to fix the layout of the generated image. We apply iterative optimization for $t = 1.0, 0.8, 0.6$ with maximum steps and early stopping. For other timesteps, we only update z_t for one single step. Although the model has memorized the background during the process mentioned in Sec. 3.1, it still introduces distortions to the background during the optimization for layout control. To preserve the original background, we start to blend with the original input image at the area without objects as in [Avrahami et al. 2022], for timesteps $t >= 0.7$. The detailed algorithm for our layout control method during the denoising process is shown in Algo. 1.

3.3 Implementation Details

We optimize each token for different objects using masked textual inversion for 200 steps with a batch size of 4. The learning rate is set to 0.002, and it takes approximately 40 seconds for each token on a single Nvidia V100 GPU. Next, we fine-tune the model using all the optimized tokens concatenated with 1-3 additional rare tokens for 800 steps (for 2 or 3 objects) or 1200 steps (for 4 objects), with a batch size of 4. The learning rate is set to 0.0002, and it takes around 3-4 minutes on an Nvidia V100 GPU. To sample images, we use DDIM sampling [Song et al. 2020] with 50 steps. For layout control, we optimize z_t with a learning rate that decreases from 20 to 15 for t values from 1.0 to 0.5. We apply iterative optimization for 40 iterations with early stopping thresholds of 0.4, 0.3, and 0.2 when t is 1.0, 0.8, and 0.6, respectively. It takes around 37 seconds to generate an image with a new layout.

4 EXPERIMENTS

To evaluate our proposed method, we first compare it qualitatively and quantitatively with several baselines, and then conduct a user study to compare them perceptually. Additionally, we conduct several ablation studies to validate the effectiveness of each important component of our method and demonstrate its application in continuous layout editing, which was not feasible with previous methods.

4.1 Baselines

Since there are no existing methods that perform the same task as ours, which is to edit the layout of existing images, we compare our method with four baselines that are designed and modified from existing methods: image-level manipulation, noised latent-level manipulation, and two variations of combining existing layout control methods with textual inversion, which include GLIGEN [Li et al. 2023] with textual inversion [Gal et al. 2022] and MultiDiffusion [Bar-Tal et al. 2023] with Dreambooth [Ruiz et al. 2023].

Image-level manipulation. We first crop and paste the objects from the original input image onto a blank image at the positions specified in the target layout. Since the target layout may specify objects with different widths and heights, we scale the cropped patches to match the desired size. Next, we use stable diffusion inpainting to fill in the blank areas.

Latent-level manipulation. Instead of cropping and pasting on the image-level, we perform a similar process on the noised latent from DDIM inversion [Song et al. 2020]. We initialize a random noise as the "canvas" and prepare a "source image" by adding noise to the original input image until $t = 0.7$ using DDIM inversion. Then, we copy objects in the "source image" and paste them onto the "canvas" following the target layout without scaling, as resizing in latent space can lead to distorted results. Finally, we use the DDIM scheduler to denoise the "canvas" and obtain the result..

GLIGEN with textual inversion. Although GLIGEN [Li et al. 2023] can perform layout control, it can only generate new images with target layouts. Therefore, to adapt it to our task, we need to add an additional step of textual inversion [Gal et al. 2022] to learn the appearances of the objects in existing images. After textual inversion, we use the learned text tokens to sample the image, where the layout is controlled with GLIGEN.

MultiDiffusion with Dreambooth. MultiDiffusion [Bar-Tal et al. 2023] is a training-free method for layout control, but it cannot be directly used for existing image layout editing. Therefore, before using it, we adopt Dreambooth [Ruiz et al. 2023] to learn the concepts of the objects in the input image, and convert them into multiple text tokens. Then, we provide the prompt with the learned text tokens and corresponding masks to MultiDiffusion to perform our task of layout editing.

4.2 Qualitative Comparison

Fig. 4 illustrates the qualitative comparisons between our method and four baselines. The image-level manipulation method (column c) produces less realistic results as it only copies and resizes objects to the target positions without natural editing. For example, in row 4, the size of the horse is significantly larger than that of the tree, violating the proper order of sizes. Moreover, noticeable artifacts appear around objects due to imperfect cropping and inpainting, as evidenced by the distorted chair armrests in row 3 and the artifacts on the horse's back in row 4. The noise-level manipulation (column d) can maintain the basic layout but cannot retain the visual features of original objects since DDIM inversion cannot reconstruct the input image, resulting in the loss of the "dog" appearance in row 2. GLIGEN with textual inversion (column e) also fails to retain the visual properties of objects since the textual inversion on the

Fig. 4. Qualitative comparison with other baseline methods. From left to right: (a) Input images; (b) Target layout; (c) Image-level manipulation; (d) Latent-level manipulation; (e) GLIGEN + textual inversion; (f) MultiDiffusion + Dreambooth; (g) Ours

whole image cannot disentangle the concepts of multiple objects within a single image. In row 5, for instance, the concept of the "pot" is not correctly learned. Additionally, GLIGEN is pre-trained with a dataset to perform layout control, which cannot be perfectly adapted to learned concepts from a particular image, leading to some misalignment with the specified layout, as shown in row 4. MultiDiffusion with Dreambooth (column f) has better layout alignment than GLIGEN with textual inversion because MultiDiffusion is a training-free method that can better cooperate with learned concepts. However, Dreambooth is designed for learning concepts of single objects from multiple images and still cannot disentangle multiple objects present in a single image, causing significant changes in object characteristics, as shown by the car in row 1 and the dog in row 2. Furthermore, it introduces artifacts around the boundaries of objects because MultiDiffusion denoises each subregion with different diffusion processes. In row 4, for instance, the regions of the "horse" and the "tree" are separately generated, resulting in artifacts around the objects and discontinuity between objects, such as the yellow background around the "horse" and the cyan background around the "tree". Our method (column g) achieves the best results by effectively controlling the layout of objects while retaining the visual features of the input images. It also produces the highest quality and most harmonious images among all four baselines.

Table 1. Quantitative comparison with other baseline methods. Our method achieves the best performance on both metrics: visual similarity to the input image and alignment with the specified layout.

	Visual similarity ↑	Layout alignment ↑
Image-level manipulation	0.57	0.0068
Latent-level manipulation	0.41	0.0071
GLIGEN with textual inversion	0.34	0.0027
MultiDiffusion with Dreambooth	0.53	0.0047
Ours	**0.61**	**0.0099**

4.3 Quantitative Comparison

We conduct a quantitative study to further evaluate the preservation of visual properties and layout alignment of our method and compare it with baselines. To measure the visual similarity between the edited result and the input image, we calculate one minus the CLIP distance between the input image and edited image. A higher score indicates that the objects in the edited image have a more similar visual appearance to the input image. For layout alignment, we calculate the CLIP distance between the image and text prompt. Specifically, for an object in the layout map, we erase the corresponding region in the edited image and fill it with black color. We then calculate the change in CLIP distance between the image (before and after erasing) and the text token corresponding to the object. If the CLIP distance drops dramatically after erasing, it indicates that the object is placed in the correct position after layout editing. This process is repeated for each object, and an average score is calculated. The experiment shows that our method has the best performance in both metrics: the least image-to-image distance (0.61) and the largest change in image-to-text distance (0.0099).

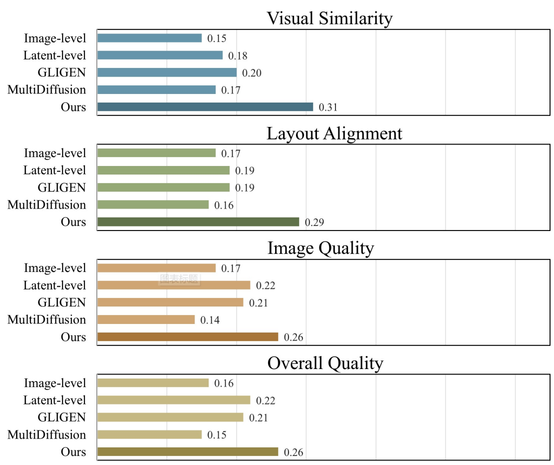


Fig. 5. Results of the user study on visual similarity, layout alignment, image quality, and overall quality, respectively.

4.4 User Study

In this part, we perform a user study to verify the effectiveness and quality of our method. We compare our method with four baselines mentioned in Sec. 4.1, *i.e.,* Image-level manipulation, Latent-level manipulation, GLIGEN with textual inversion, and MultiDiffusion with Dreambooth.

The user study consists of 16 questions. For each of the questions, we first show the original input image and the target layout to the user. Then we show the five images edited by four compared methods and our method in random order. Finally, the user is asked to make four selections regarding four different factors:

- Visual similarity: to choose the image whose generated objects have the highest similarity to the objects in the original input image.
- Layout alignment: to select the image whose layout is best aligned with the target layout.
- Image quality: to select the image with the highest quality and photorealism.
- Overall quality: to choose the best result considering all the three factors above.

Among the 16 questions, we also set a validation question where the 5 edited images consist of a ground-truth image with four unrelated images. The user has to make more than 3 correct selections out of 4, to be considered as a valid questionnaire.

We finally collected 42 questionnaires, among which 31 are valid by passing the validation questions. Among the 31 valid participants, 5 users are below 20 years old, 17 range from 20 to 30 years of age, 6 are between 30 and 40 years old, and 3 are above 40 years old. The result of the user study are shown in Fig. 5, and we find that our method outperforms other methods in all the four factors with a preferred rate of 31% in visual similarity, 29% in layout alignment, 26% in image quality, and 26% in overall quality.

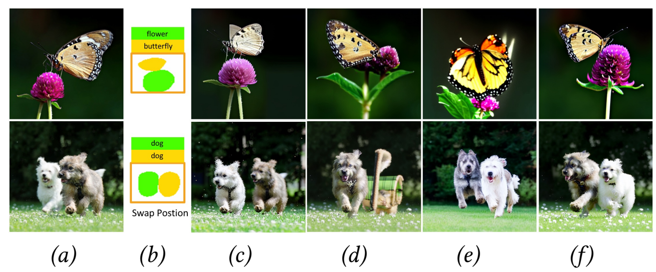


Fig. 6. Ablation study on different textual inversion methods. From left to right: (a) Input images; (b) Target layouts; (c) Inversion with Dreambooth [Ruiz et al. 2023]; (d) Textual inversion [Gal et al. 2022] + finetune [Kumari et al. 2023]; (e) Masked textual inversion w/o finetune; (f) Our full inversion method with masked textual inversion and finetune.

4.5 Ablation Study

4.5.1 Inversion Methods. In this part, we compare the inversion method with and without a mask. We selected Dreambooth [Ruiz et al. 2023] and textual inversion [Gal et al. 2022] + finetune as representatives of the methods without a mask. For Dreambooth, we jointly trained the model on multiple concepts with a single image and prompt pair, while the textual inversion + finetune method first runs the textual inversion to update each token on the entire image and then uses the updated tokens to fine-tune the cross-attention layers of the model. The results show that if a single image contains multiple objects, the methods without a mask, including both Dreambooth and textual inversion, cannot precisely learn and disentangle the concepts of different objects, which may result in the loss of visual properties. For example, in Fig. 6, row 1, the shape and color of the flower change significantly in columns (c) and (d). Another issue with these methods without a mask is learning incorrect objects. For instance, in Fig. 6, row 2, columns (c) and (d), the information about the white and brown dogs is not encoded into two separate tokens. Therefore, both methods cannot correctly swap the position of the two dogs. The textual inversion + finetune method (d) even generates a trolley-like object instead of a dog.

In our method, we optimize the text tokens using masked textual inversion followed by model finetuning, as described in Sec. 3.1. By applying the mask, the information of different objects in the image is correctly disentangled. In Fig. 6, row 2, columns (e) and (f), the two dogs successfully swap positions. Moreover, adding finetuning after our masked textual inversion can further preserve visual details, including colors and textures.

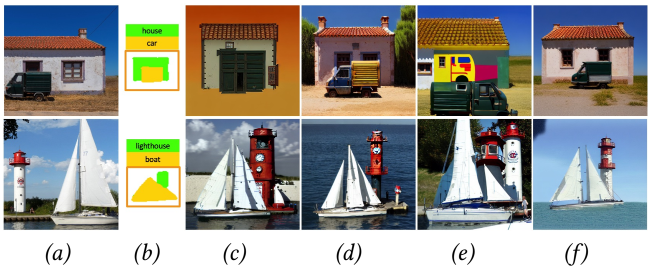


Fig. 7. Ablation study on differen layout control methods. From left to right: (a) Input images; (b) Target layouts; (c) Our inversion + ControlNet [Zhang and Agrawala 2023]; (d) Our inversion + GLIGEN [Li et al. 2023]; (e) Our inversion + MultiDiffusion [Bar-Tal et al. 2023]; (f) Our full methods.

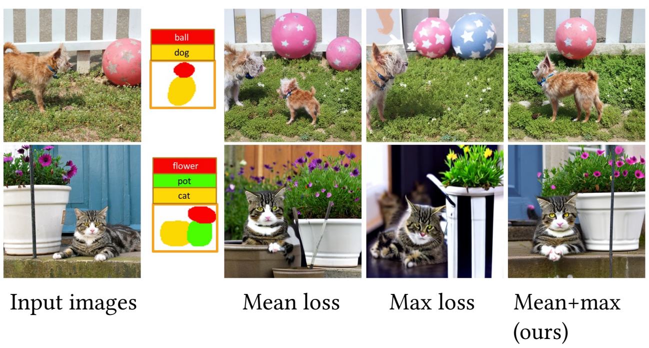


Fig. 8. Ablation study on optimization loss of layout control.

4.5.2 *Layout Control Methods.*

After verifying the effectiveness of our masked textual inversion, we compared our layout control method with three other methods: ControlNet [Zhang and Agrawala 2023], GLIGEN [Li et al. 2023], and MultiDiffusion [Bar-Tal et al. 2023]. For the training-based methods ControlNet and GLIGEN, the generated objects fail to keep some visual features in the input images, such as the "car" in row 1 and the "lighthouse" in row 2 of Fig. 7. This is because those models are pretrained with datasets that may not be well-adapted to our learned concepts, especially considering that our finetuning after masked textual inversion may further change the parameters of the cross-attention layers, which can affect the ability of the pretrained model. As for the training-free method MultiDiffusion, it can better incorporate learned concepts, but it tends to yield some artifacts around the objects as it denoises each sub-region separately and fails to fuse them smoothly. For example, there is a yellow painting on the house in Fig. 7, row 1, column (e). Our iterative layout control method solves the aforementioned problems and generates the best results in (f)

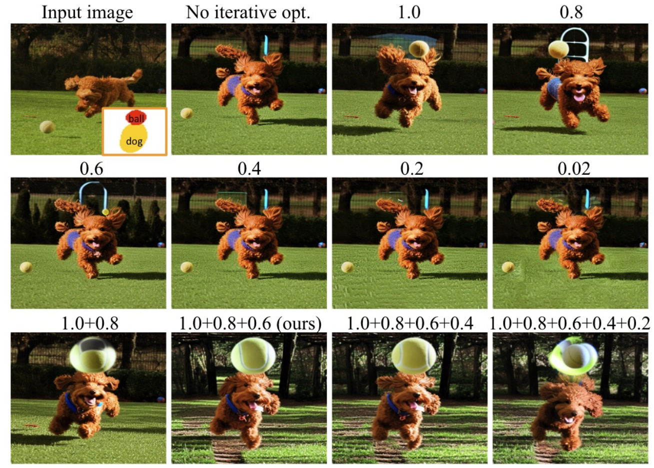


Fig. 9. Ablation study on iterative optimization. The number above the image indicates the denoising step on which iterative optimization is applied. If more than one number is labeled, iterative optimization is applied to multiple denoising steps.

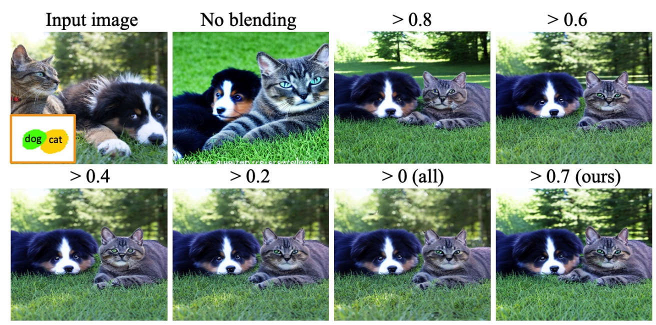


Fig. 10. Ablation study on blending steps. The number above the image indicates the number of steps where blending is applied.

4.5.3 *Optimization Loss.*

In Eqn. (7), we mention that both the mean and the maximum values of $\mathcal{L}_k$ are calculated for the optimization loss. As shown in Fig. 8, if only the mean loss is applied, the controls against each object are equal, and no additional effort can be put on the difficult one. Therefore, the positions of some objects may be insufficiently controlled. For example, part of the "dog" still stays at the original position in Fig. 8, row 1, and the relative position of the "dog" and the "pot" is not correctly arranged in row 2.

On the other hand, if only the max loss is applied, the model may focus too much on single object, but ignore others. For example, the position of the "dog" is not modified in Fig. 8, row 1, and the "pot" is stretched and distorted to fit the target position in row 2. Therefore, we finally apply both the mean and max loss, which can balance the layout control of each object and simultaneously focus more on objects that are difficult to control.

4.5.4 *Iterative Optimization.*

As described in Section 3.2, we only apply iterative optimization at certain time steps. In this part, we performed experiments to verify the effect of iterative optimization

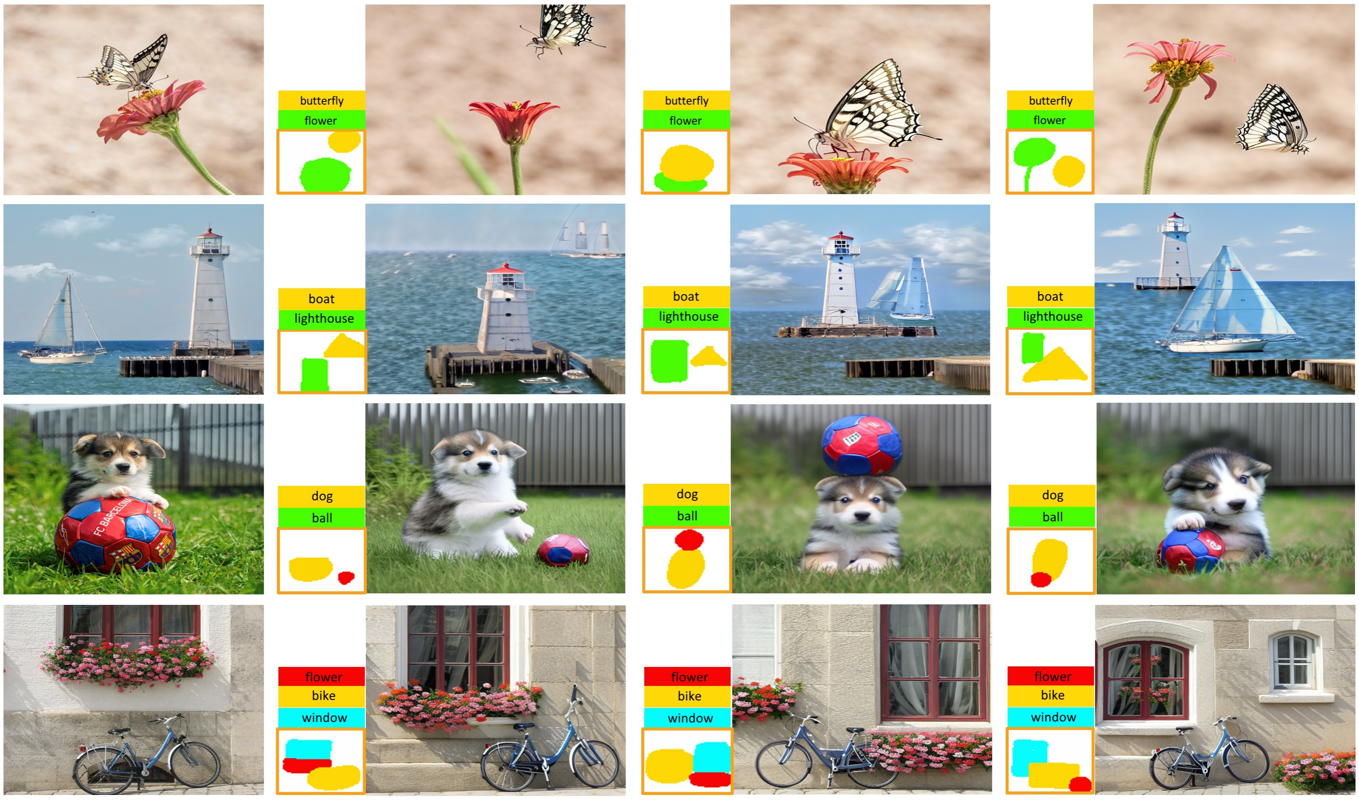


Fig. 11. Continuous layout editing with different target layouts.

and determine which time steps to apply it. As shown in Figure 9, when iterative optimization is not applied, the objects are not well aligned with the target layout.

We also found that the determination of the layout control happens at the time steps closer to the noise. If iterative optimization is applied at large time steps (e.g., $t = 1.0$, 0.8, or 0.6), it is more effective for layout control, as a ball appears in the upper part of the image. Conversely, with small time steps (e.g., $t = 0.4$, 0.2, or 0.02), the layout has little change. This implies that the layout of the objects is nearly fixed at large t and can hardly be modified when t is small. Therefore, we implement iterative optimization at relatively larger time steps, i.e., $t = 1.0 + 0.8 + 0.6$.

As shown in the image, our choice can generate a ball with the desirable size. However, fewer iterative optimization time points (e.g., $t = 1.0 + 0.8$) may not be sufficient for generating the ball completely. Conversely, too many iterative optimization time points (e.g., $t = 1.0 + 0.8 + 0.6 + 0.4$, $t = 1.0 + 0.8 + 0.6 + 0.4 + 0.2$) have little effect on the final layout but require longer optimization time and may introduce artifacts. Thus, we choose $t = 1.0 + 0.8 + 0.6$ for a balance between quality and speed.

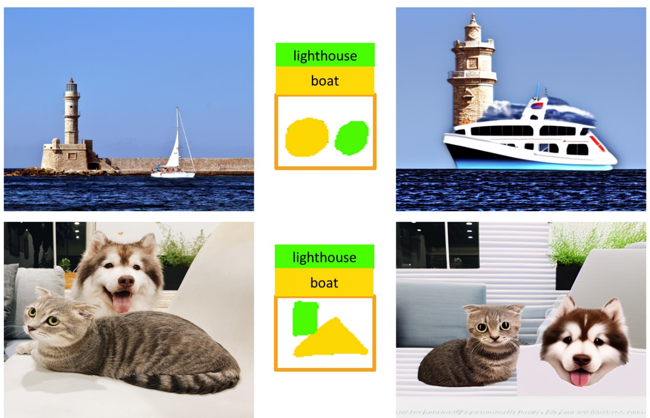
4.5.5 Blending. As described in Section 3.2, we blend the edited image with the original input at the region of the background. In this part, we perform experiments to evaluate the effect of the different number of blending steps. As shown in Figure 10, the background will be completely changed if no blending is applied because the optimization process for layout control will have a large influence on the background. Blending when t is large (e.g., $t > 0.8$) has the most substantial effect on the background, and the effect of blending starts to converge for more steps when $t > 0.6$. Therefore, we choose to apply background blending when $t > 0.7$.

4.6 Results of Continuous Editing

Our method is capable of rearranging the positions of objects in an input image to fit a target layout without altering its visual properties. This unique capability enables our method to perform continuous layout editing of single input images, which was not possible with previous methods. Figure 11 and Figure 1 demonstrate some examples of our method's effectiveness. For instance, in row 3 of Figure 11, we show how our method can change the positions of the dog and ball to fit three different layouts, creating natural interactions between them for each layout. Moreover, our method can handle images with multiple objects. We demonstrate this by showing how it can edit the positions of three objects in Figure 11, rows 4, and four objects in Figure 1, row 2, to align with different input layouts. These examples highlight the flexibility of our method to handle varying numbers of objects of different categories and sizes.

5 CONCLUSIONS & LIMITATIONS

We present the first framework that supports continuous layout editing of single images, generating high-quality results by rearranging the positions of objects in the input image to fit a user-specified layout while preserving their visual properties. A key component enabling us to learn objects from a single image is Masked Textual Inversion, which disentangles multiple concepts into different tokens. With learned objects, we propose a training-free iterative optimization method for layout control. We demonstrate that our framework outperforms other baselines, including image-level manipulation, latent-level manipulation, and combinations of existing learning and layout control methods.



Input images Edited images

Fig. 12. Failure cases: In the first row, our method fails to maintain the visual features of objects when there is a significant size difference between the input and edited images. In the second row, our method fails to generate the full body of objects when they suffer from large occlusions in the original image.

However, our method still encounters some limitations. One limitation is that it may fail to preserve the visual details of an object if the size of the object in the initial image and edited image varies significantly, as shown by the sailboat in Figure 12, row 1. Another limitation is that it may have difficulty recovering the full body of an object if the object in the input image is heavily occluded, as shown by the dog in Figure 12, row 2. We believe that these problems are caused by the limited information that can be inferred from a single image by object concept learning. To mitigate these limitations, we could augment the input images to different sizes and angles and even inpaint missing parts of the object caused by occlusion before applying our masked textual inversion. Furthermore, our layout editing is not in real-time due to the iterative sampling nature of the diffusion model. Future research directions include exploring methods to accelerate the process and supporting more applications of layout editing.

REFERENCES

Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. 2023. Spatext: Spatio-textual representation for controllable image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18370–18380.

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

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. 2022. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324* (2022).

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

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. 2023. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *arXiv preprint arXiv:2301.13826* (2023).

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618* (2022).

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626* (2022).

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

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. 2023. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1931–1941.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. 2023. GLIGEN: Open-Set Grounded Text-to-Image Generation. *CVPR* (2023).

Timo Lüddecke and Alexander Ecker. 2022. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7086–7096.

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

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

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

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22500–22510.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* 35 (2022), 36479–36494.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543* (2023).