DragonDiffusion: Enabling Drag-style Manipulation on Diffusion Models

Chong Mou[1] Xintao Wang[2] Jiechong Song[1] Ying Shan[2] Jian Zhang[†1]
[1]School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University [2]ARC Lab, Tencent PCG

Figure 1. We propose DragonDiffusion, an efficient classifier-guidance-like method that can perform various real-image editing tasks, including object moving, object resizing, object appearance replacement, and content dragging. Our method is built based on the pre-trained Stable Diffusion without model fine-tuning or training.

Abstract

Despite the ability of existing large-scale text-to-image (T2I) models to generate high-quality images from detailed textual descriptions, they often lack the ability to precisely edit the generated or real images. In this paper, we propose a novel image editing method, ***DragonDiffusion****, enabling* ***Drag****-style manipulation* ***on Diffusion*** *Models. Specifically, we construct classifier guidance based on the strong correspondence of intermediate features in the diffusion model. It can transform the editing signals into gradients via feature correspondence loss to modify the intermediate representation of the diffusion model. Based on this guidance strategy, we also build a multi-scale guidance to consider both semantic and geometric alignment. Moreover, a cross-branch self-attention is added to maintain the consistency between the original image and the editing result. Our method, through an efficient design, achieves various editing modes for the generated or real images, such as object moving, object resizing, object appearance replacement, and content dragging. It is worth noting that all editing and content preservation signals come from the image itself, and the model does not require fine-tuning or additional modules. Our source code will be available at* https://github.com/MC-E/DragonDiffusion.

† Corresponding author.

1. Introduction

Thanks to the large-scale training data and huge computing power, generative models have developed rapidly, especially large-scale text-to-image (T2I) diffusion models [29, 27, 23, 26, 10, 43, 22, 42], which aims to generate images conditioned on a given text/prompt. However, this generative capability is often diverse, and it is challenging to design suitable prompts to generate images consis-

tent with what the user has in mind, let alone further editing based on existing images.

Compared to image generation, image editing has broader application demands. Methods based on GANs [1, 2, 3] are widely used in the image editing domain due to the compact and editable latent space (e.g., StyleGAN [17]). Recently, DragGAN [24] proposes a point-to-point dragging scheme, which can achieve refined content dragging. However, it is constrained by the capacity and generalization of GAN models. Compared to GAN models, Diffusion [14] has higher stability and superior generation quality. In this paper, we aim to investigate whether the diffusion model can achieve a similar drag-style ability. This ability should be a more generalized editing capability, not limited to point dragging, such as object moving, object resizing, and cross-image content dragging.

In implementation, the primary challenge lies in the lack of a concise and modifiable latent space amenable to editing. Numerous diffusion-based image editing methods (*e.g.*, Prompt2Prompt [13], [12], [5]) are built based on the correspondence between intermediate text and image features. They find that the cross-attention map between the feature of words and object has a notable local similarity, which can be used as an editing medium. Recently, self-guidance [11] proposes a differentiable approach that employs cross-attention maps to locate and calculate the size of objects within images. Then, gradient backpropagation is utilized to edit these properties. However, the correspondence between text and image features is weak, heavily relying on the design of prompts. Moreover, in complex or multi-object scenarios, text struggles to build accurate local similarity with a specific object. In this paper, we aim to explore a more fine-grained editable space than text-image correspondence for generalized image editing tasks.

In the large-scale T2I diffusion generation process, besides the strong correspondence between text features and intermediate image features, there is also a strong correspondence between intermediate image features. This characteristic has been explored in DIFT [37], which demonstrates that this feature correspondence is high-level, facilitating point-to-point correspondence of relevant content in different images. Therefore, we are intrigued by the possibility of utilizing this strong correspondence between image features to achieve image editing. In this paper, we present our solution. Specifically, our method involves two sets of features (*i.e.*, guidance features and generation features) during the diffusion process. We use the guidance features as the target, employing strong image feature correspondence to constrain and edit the generation features. Additionally, the content consistency between the edited result and the original image is also maintained through the strong image feature correspondence. Here, we also notice that there is a concurrent work, Drag-Diffusion [30], studying this issue. It utilizes LORA [28] to maintain consistency with the original image and optimizes one intermediate step in the diffusion process to perform editing. Unlike Drag-Diffusion, our method is based on classifier-guidance [9], and all editing and content consistency signals come from the image itself, without the need for fine-tuning or training the model. In addition, we use the intermediate feature correspondence to explore generalized image editing capabilities, such as object moving, object resizing, object appearance replacement, and content dragging. In summary, the contributions of this paper are as follows:

- We propose a classifier-guidance image editing strategy based on the strong correspondence of intermediate features in diffusion models. In this design, we also study the roles of the feature in different layers and develop a multi-scale feature matching scheme that considers both semantic and geometric correspondence.
- All content editing and preservation signals in our proposed method come from the image itself. It allows for a direct translation of T2I generation ability in diffusion models to image editing tasks without the need for any model fine-tuning or training.
- Extensive experiments demonstrate that our DragonDiffusion can perform various fine-grained image editing tasks, including object moving, object resizing, object appearance replacement, and content dragging.

2. Related Work

2.1. Diffusion Models

In recent years, the diffusion model [14] has achieved great success in the community of image synthesis. It is designed based on thermodynamics [32, 34], including a diffusion process and a reverse process. In the diffusion process, a natural image $\mathbf{X}_0$ is converted to a Gaussian distribution $\mathbf{X}_T$ by adding random Gaussian noise with T iterations. Each step of adding noise is defined as:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1},\ t \in [1, T], \tag{1}$$

where $\beta_t \in [0, 1]$ is a gradually increasing hyperparameter. $\boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(0, \mathbf{I})$ is the random Gaussian noise. The reverse process is to recover $\mathbf{x}_0$ from $\mathbf{x}_T$ by several denoising steps. Therefore, the diffusion model is training a denoiser, conditioned on the current noisy image and time step:

$$L(\theta) = \mathbb{E}_{\mathbf{x}_0, t, \epsilon\sim\mathcal{N}(0,1)}\left[||\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)||_2^2\right], \tag{2}$$

where θ is the model parameters of the denoiser.

Recently, some text-conditioned diffusion models (*e.g.*, GLID [23] and SD [27]) have been proposed, which mostly inject text condition into the denoiser through a cross-attention strategy.

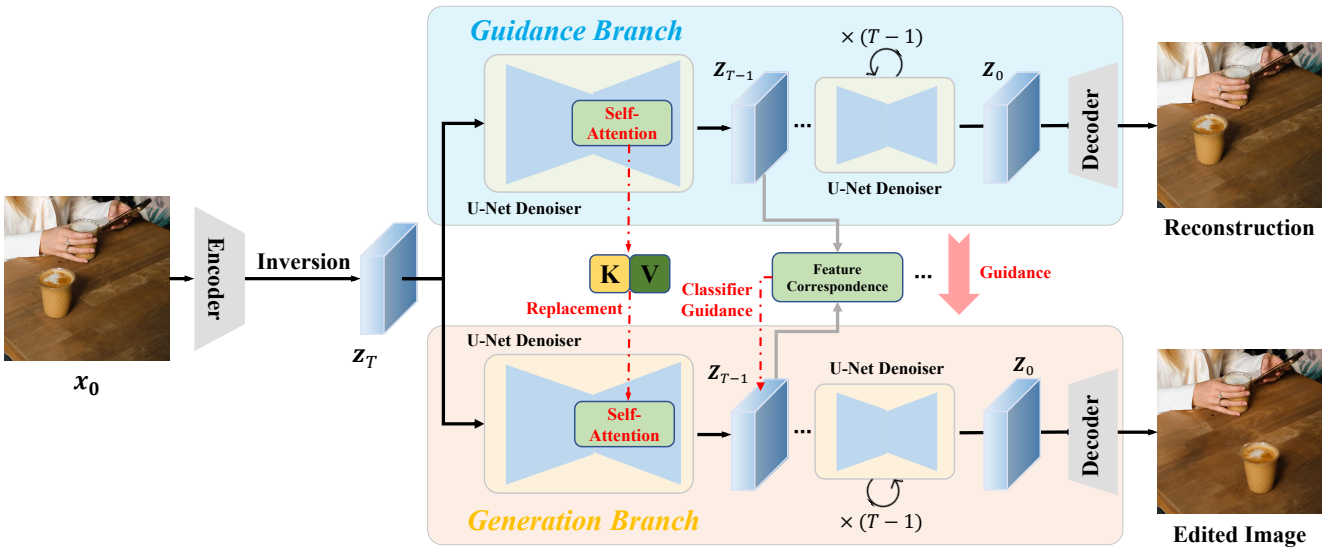


Figure 2. Illustration of our model design. Our proposed method consists of two branches, *i.e.*, the guidance branch and the generation branch. The guidance branch provides editing and consistency guidance to the generation branch through the correspondence of intermediate features. Our DragonDiffusion is built based on Stable Diffusion [27], without model fine-tuning or training.

2.2. Classifier guidance in Diffusion Model

From a continuous perspective [36], diffusion models can be viewed as a score function, *i.e.*, $\nabla \mathbf{x}_t \log q(\mathbf{x}_t)$, that samples from the corresponding distribution [35] according to Langevin dynamics [32, 34]. The conditional diffusion process, on the other hand, can be seen as using a joint score function, *i.e.*, $\nabla \mathbf{x}_t \log q(\mathbf{x}_t, y)$, to sample from a more enriched distribution, where y is the external condition. The joint score function can be further decomposed into:

$$\nabla \mathbf{x}_t \log q(\mathbf{x}_t, y) = \nabla \mathbf{x}_t \log q(\mathbf{x}_t) + \nabla \mathbf{x}_t \log q(y|\mathbf{x}_t), \quad (3)$$

where the first term is the original unconditional diffusion denoiser, and the second term corresponds to the classifier guidance to be added to the diffusion process, also known as the energy function. The energy function can be selected based on the generation target, such as a classifier [9] to specify the category of generation results.

Classifier guidance has been applied to numerous controllable image generation tasks, such as sketch-guided generation [38], mask-guided generation [31], universal guided generation [41, 6], and image editing [11]. These methods, based on classifier guidance, inspire us to transform editing signals into gradients through score functions, achieving fine-grained image editing.

2.3. Image Editing

Image editing methods traditionally targeted a translation between image domains [15, 16, 19]. Numerous editing approaches [1, 2, 3] invert images into a latent space of StyleGAN [17] and then edit specific content (*e.g.*, hair and age) by manipulating latent vectors. Recently, DragGAN [24] proposes a point-to-point dragging scheme, which can achieve more refined content dragging. Diffusion [14], as a more stable generative model compared to GANs, has led to several diffusion-based image editing methods [4, 13, 18, 20, 7]. Most of them use text as the edit signal. For example, Prompt2Prompt [13] achieves specific object editing by replacing the correspondence between text features and intermediate features. SDEdit [20] performs image editing by adding noise to the original image and then denoising under new text conditions. InstructionP2P [7] achieves image editing by fine-tuning the model and using text as an editing instruction. Recently, Self-guidance [11] transforms editing signals into gradients through the correspondence between text and intermediate features to achieve image editing. However, the correspondence between image and text is coarse-grained. How to perform fine-grained and generalized image editing with diffusion models is still an open challenge.

3. Method

3.1. Preliminary: Stable Diffusion

In this paper, we implement our method based on the recent state-of-the-art T2I diffusion model (*i.e.*, Stable Diffusion (SD) [27]). SD is a latent diffusion model (LDM), which contains an autoencoder and an UNet denoiser. The autoencoder can convert natural images $\mathbf{x}_0$ into latent space $\mathbf{z}_0$ and then reconstruct them. The diffusion process of SD is conducted in the latent space. The training objective of SD is the same as that of common diffusion models (*i.e.*, Eq. 4), except that the denoiser operates on the latent $\mathbf{z}_t$ instead of the image $\mathbf{x}_t$. During inference, $\mathbf{Z}_T$ is generated

from random Gaussian distribution. The final result $\mathbf{z}_0$, as the clean latent, is fed into the decoder of the autoencoder to generate the natural image $\mathbf{x}_0$. In the conditional part, SD utilizes the pre-trained CLIP [25] text encoder to embed text inputs as embedding sequences $\mathbf{y}$.

3.2. Overview

The objective of our DragonDiffusion is to achieve fine-grained image editing of real images by SD, which involves two issues: changing the content to be edited and preserving other content. For example, if a user wants to move the bread in an image, the generated result only needs to change the position of the bread, while the appearance of the bread and other image content should not change. In this paper, inspired by DIFT [14], we utilize the strong correspondence of intermediate features in diffusion models to address both issues simultaneously. An overview of our design is presented in Fig. 2. First, we invert the original image $\mathbf{x}_0$ to the latent representation $\mathbf{z}_T$ through the reverse diffusion process [33, 21]. Then, we input $\mathbf{z}_T$ into two parallel branches, *i.e.*, the guidance branch and the generation branch. The guidance branch is the standard diffusion generation process, which can reconstruct $\mathbf{x}_0$. The generation branch needs to generate the corresponding editing result according to the demand. To preserve the content of the original image, we utilize the correspondence between the intermediate features of the two branches, transferring the content information from the guidance branch to the generation branch through a cross-branch self-attention design. Similarly, using the strong features correspondence, we design a score function [36, 35] that transforms the editing signal into gradients through classifier guidance [9], modifying the intermediate representation $\mathbf{z}_t$ of the generation branch. Our entire editing process only applies the correspondence of intermediate features in diffusion models, without the need for model fine-tuning or training.

3.3. Classifier-guidance-based Editing Design

In this article, inspired by classifier guidance [9], we aim to update the intermediate representation (*i.e.*, $\mathbf{z}_t$) of the diffusion process by transforming editing signals into gradients through score functions, thereby achieving image editing.

3.3.1 Score Function

As illustrated in Eq. 4, to utilize classifier guidance, we first need to construct a score function that matches the target. The recent work, DIFT [37], discovers that the intermediate features of diffusion models have a strong correspondence, which can be used for point-to-point matching between different images. Inspired by this work, in each iteration, we use the same denoiser to map the intermediate representations (*i.e.*, $\mathbf{z}_t^{gen}$, $\mathbf{z}_t^{gud}$) of the two branches to the feature space (*i.e.*, $\mathbf{F}_t^{gen}$, $\mathbf{F}_t^{gud}$). The subscripts "*gen*" and "*gud*" represent the generation branch and the guidance branch, respectively. Note that the features here come from the decoder in the denoiser. $\mathbf{F}_t^{gud}$ contains the features of the original image, and $\mathbf{F}_t^{gen}$ contains the features of the edited image. Here, we use two masks (*i.e.*, $\mathbf{m}^{gud}$ and $\mathbf{m}^{gen}$) to represent the positions of certain content in the original and edited images, respectively. Based on the strong correspondence between the features, the two regions in $\mathbf{F}_t^{gen}$ and $\mathbf{F}_t^{gud}$ need to have high similarity. Here, we utilize the cosine distance ($cos(\cdot) \in [-1, 1]$) to measure the similarity and normalize it to $[0, 1]$:

$$\mathcal{S}(\mathbf{m}^{gen}, \mathbf{m}^{gud}) = \frac{cos(\mathbf{F}_t^{gen}[\mathbf{m}^{gen}],\ Sg(\mathbf{F}_t^{gud}[\mathbf{m}^{gud}])) + 1}{2}, \tag{4}$$

where Sg is the gradient clipping operation. The larger the value, the higher the similarity. $[\cdot]$ represents retrieving values in non-zero regions. When we want to constrain the content appearing in the position of $\mathbf{m}^{gud}$ to appear in the target position $\mathbf{m}^{gen}$, our optimization goal is to make the similarity in Eq. 4 as large as possible.

In addition to editing, we hope that other areas of the editing result remain consistent with the original image. Given a mask $\mathbf{m}^{share}$, marking areas with no editing, the similarity between the editing result and the original image in these areas can also be defined using the cosine similarity as $\mathcal{S}(\mathbf{m}^{share}, \mathbf{m}^{share})$. Finally, the loss function, combining editing and content preserving, is defined as:

$$\mathcal{L} = \frac{w_e}{\alpha + \beta \cdot \mathcal{S}(\mathbf{m}^{gen}, \mathbf{m}^{gud})} + \frac{w_p}{\alpha + \beta \cdot \mathcal{S}(\mathbf{m}^{share}, \mathbf{m}^{share})}, \tag{5}$$

where α and β are two hyper-parameters. w_e and w_p are two weights to balance the editing and consistency parts. Finally, the joint score function in Eq. 6 can be written as:

$$\begin{aligned} &\nabla \mathbf{z}_t^{gen} \text{logq}(\mathbf{z}_t^{\text{gen}}, \mathbf{m}^{\text{gen}}, \mathbf{m}^{\text{share}}) = \\ &\nabla \mathbf{z}_t^{gen} \text{logq}(\mathbf{z}_t^{\text{gen}}) + \nabla \mathbf{z}_t^{\text{gen}} \text{logq}(\mathbf{m}^{\text{gen}}, \mathbf{m}^{\text{share}} | \mathbf{z}_t^{\text{gen}}). \end{aligned} \tag{6}$$

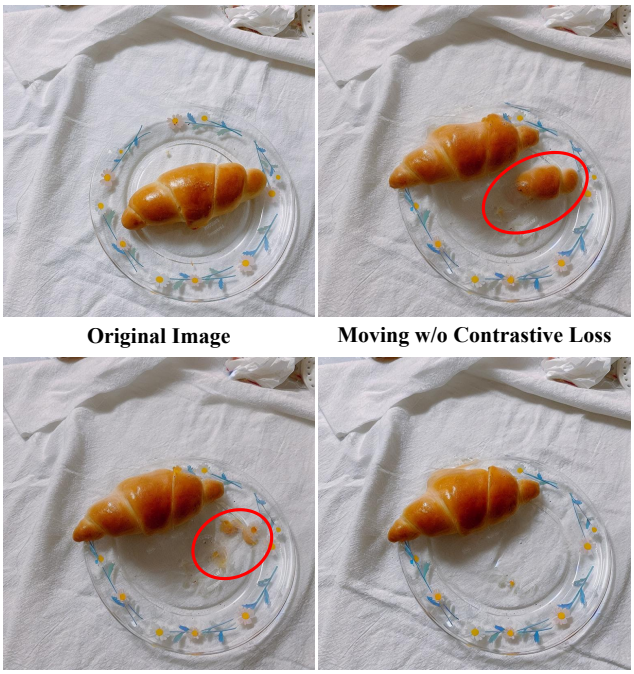
The classifier guidance $\nabla \mathbf{z}_t^{gen} \text{logq}(\mathbf{m}^{\text{gen}}, \mathbf{m}^{\text{share}} | \mathbf{z}_t^{\text{gen}})$ can be computed as $\eta \frac{d\mathcal{L}}{d\mathbf{z}_t^{gen}}$, where η is the learning rate.

3.3.2 Multi-scale Guidance

The decoder of the Unet denoiser contains four blocks of different scales. DIFT [37] finds that the second layer contains more semantic information, while the third layer contains more geometric information. We also studied the role of features from different layers in editing tasks, as shown in Fig. 3. In the experiment, we set $\mathbf{z}_T$ as random Gaussian noise, and we set $\mathbf{m}^{gen}$, $\mathbf{m}^{gud}$ as zeros matrixes and $\mathbf{m}^{share}$ as a ones matrix. In this way, the generation branch is guided to reconstruct the original image from the random

Original Image | Guided by the layer-1 | Guided by the layer-2 | Guided by the layer-3 | Guided by the layer-4 | Guided by the layer-2&3

Figure 3. Illustration of using features from different layers as guidance to reconstruct the original image. In this experiment, we set $\mathbf{z}_T$ as random Gaussian noise, and we set $\mathbf{m}^{gen}$, $\mathbf{m}^{gud}$ as zeros matrix and $\mathbf{m}^{share}$ as a ones matrix.



Original Image | Moving w/o Contrastive Loss

Moving w/o Inpainting Loss | Ours

Figure 4. Visualization of the roles that contrastive loss and inpainting loss play in the object movement task. The contrastive loss we designed can eliminate the multi-object phenomenon, while the inpainting loss can generate more natural content in the missing areas.

Gaussian distribution. We can find that the feature in the first layer is too high-level to reconstruct the original image accurately. The features in the fourth layer have weak feature correspondence, resulting in significant differences between the reconstructed image and the original. The features in the second and third layers are more suitable for reconstructing the original image, and each has its own specialty. The second layer of features contains more semantic information and can reconstruct images that are semantically similar to the original but with some differences in content details. The features in the third layer tend to express low-level visual features. The reconstructed images are closer to the original, but they cannot provide effective supervision for high-level texture features, resulting in blurry reconstructed images. In our design, we aim to combine these two levels (*i.e.*, high and low) of guidance and propose a multi-scale supervision approach based on the second and third layers of features. The reconstructed results in Fig. 3 also demonstrate that this combination can balance the generation of low-level and high-level visual features. Therefore, $\mathbf{F}_t^{gen}$ and $\mathbf{F}_t^{gud}$ contain two sets of features from layer 2 and layer 3.

3.3.3 Implementation Details for Each Application

Object moving. In the task of object moving, $\mathbf{m}^{gen}$ and $\mathbf{m}^{gud}$ locate the same object in different spatial positions. $\mathbf{m}^{share}$ is the complement of the union of $\mathbf{m}^{gen}$ and $\mathbf{m}^{gud}$, *i.e.*, $\mathbf{m}^{share} = Cu(\mathbf{m}^{gen} \cup \mathbf{m}^{gud})$. We define the points with a value of 1 in the binary mask as belonging to this mask. Using only the editing and preserving losses in Eq. 5 can lead to some issues, especially in the multiple objects phenomenon. As shown in the second image of Fig. 4, although the bread has been moved according to the editing signal, some of the bread content is still preserved in its original position in the generated result. Therefore, in the object moving task, we need to constrain the generated results to avoid previous image content in the original position. To address this, we added a contrastive loss to Eq. 5 to provide an additional constraint:

$$\mathcal{L}_c = w_c \cdot \mathcal{S}(\mathbf{m}^{inpaint}, \mathbf{m}^{inpaint}), \tag{7}$$

where $\mathbf{m}^{inpaint} = \mathbf{m}^{gud} - \mathbf{m}^{gen}$, *i.e.*, $\mathbf{m}^{inpaint} = \{p | p \in \mathbf{m}^{gud} \; and \; p \notin \mathbf{m}^{gen}\}$. w_c is a hyper-parameter of the loss weight.

As illustrated in the third image of Fig. 4, although the contrastive loss function can address the multi-object phenomenon, it lacks guidance during the inpainting process, resulting in somewhat disordered inpainting. Here, we design an inpainting loss, using content outside of the object as guidance to constrain the features of the inpainting re-

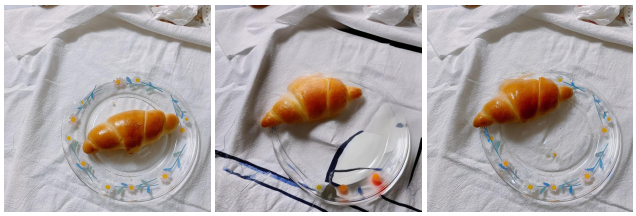


Figure 5. Visualization of the object moving with and without cross-branch self-attention.

gion. Mathematically, the loss function is defined as:

$$\begin{cases} \mathcal{L}_i = \frac{w_i}{\alpha+\beta\cdot\mathcal{S}_{glob}} \\ \mathcal{S}_{glob} = \frac{cos(\frac{\sum \mathbf{F}_t^{gen}[\mathbf{m}^{inpaint}]}{\sum \mathbf{m}^{inpaint}},\ Sg(\frac{\sum \mathbf{F}_t^{gud}[\mathbb{I}-\mathbf{m}^{gud}]}{\sum \mathbf{m}^{gud}}))+1}{2}, \end{cases} \tag{8}$$

where w_i is a hyper-parameter of the loss weight. After equipping $\mathcal{L}_c$ and $\mathcal{L}_i$, our method can effectively inpaint the gaps of the object in the original image, as shown in the fourth image of Fig. 4.

Object resizing. In this task, we use interpolation to transform $\mathbf{m}^{gud}$ and $\mathbf{F}_t^{gud}$ to the target size, and then extract the intermediate feature $\mathbf{F}_t^{gud}[\mathbf{m}^{gud}]$ as the feature of the object after resizing. To guide the generation branch to produce a target object with the same size, we perform local resizing on $\mathbf{m}^{gen}$. Then, we use $\mathbf{F}_t^{gud}[\mathbf{m}^{gud}]$ to supervise and guide the features within this region. Local resizing refers to interpolating the input and then restoring it to its original size with center cropping/expansion. Finally, in this task, Eq. 4 is reformulated as:

$$\mathcal{S}(\mathbf{m}^{gen}, \mathbf{m}^{gud}) = \frac{cos(\mathbf{F}_t^{gen}[\mathcal{C}(\mathcal{R}(\mathbf{m}^{gen}))],\ Sg(\mathcal{R}(\mathbf{F}_t^{gud})[\mathcal{R}(\mathbf{m}^{gud})])) + 1}{2}, \tag{9}$$

where $\mathcal{R}$ and $\mathcal{C}$ represent the interpolation and center cropping/expansion operation, respectively. The other constraints remain consistent with default.

Appearance replacement. This task aims to replace the appearance between objects of the same category. Similar to the inpainting loss (*i.e.*, Eq. 8) in object moving, we use the features mean of the corresponding region to represent the object appearance. Therefore, the guidance branch will involve the diffusion of two guidance images, the original image and the appearance reference image. The appearance reference image only edits the generation through gradients, generated from appearance similarity. We use $\mathbf{F}_t^{app}$ and $\mathbf{m}^{app}$ to represent the intermediate features of the appearance reference image and the mask corresponding to the reference object, respectively. Therefore, the appearance similarity is defined as:

$$\mathcal{S}_{app}(\mathbf{m}^{gud}, \mathbf{m}^{gen}) = \frac{cos(\frac{\sum \mathbf{F}_t^{gen}[\mathbf{m}^{gen}]}{\sum \mathbf{m}^{gen}},\ Sg(\frac{\sum \mathbf{F}_t^{app}[\mathbf{m}^{app}]}{\sum \mathbf{m}^{app}})) + 1}{2}. \tag{10}$$

The other constraints remain consistent with default.

Point dragging. In this task, we want to drag the image content via a specific point in the image. In this case, $\mathbf{m}^{gen}$ and $\mathbf{m}^{gud}$ denote the destination and starting points, as well as their adjacent points within a small range surrounding them. Unlike the previous tasks, the $\mathbf{m}^{share}$ here is manually defined. The gradient guidance comes from Eq. 5 without other specific designs.

3.4. Cross-branch Self-attention

To maintain consistency between the generated result and the original image, we use two strategies: DDIM inversion [33] and a cross-branch self-attention design. For DDIM inversion, we can also use the more accurate Null-text inversion [21] to improve consistency. However, it is still challenging to maintain high consistency between the editing result and the original image solely through DDIM inversion. Here, inspired by the consistency preservation in some video and image editing works [40, 39, 8], we design a cross-branch self-attention guidance. Specifically, we replace the key and value in the self-attention module of the denoiser in the generation branch with the corresponding key and value from the guidance branch. Note that since the feature correspondence in denoiser encoder is relatively weak [14], we only use this operation in the denoiser decoder. The modified self-attention module is defined as:

$$\begin{cases} \mathbf{Q} = \mathbf{W}_Q^{gen} * \mathbf{F}^{gen};\ \mathbf{K} = \mathbf{W}_K^{gud} * \mathbf{F}^{gud};\ \mathbf{V} = \mathbf{W}_V^{gud} * \mathbf{F}^{gud} \\ Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \end{cases} \tag{11}$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are learnable projection matrices. $*$ refers to the convolution operator. A comparison of our method with and without cross-branch self-attention is shown in Fig. 5. One can see that the design can effectively close the distance between the generated result and the original image.

4. Experiments

In this paper, our DragonDiffusion can perform various image editing tasks, including object moving, object resizing, object appearance replacement, and content dragging. In Fig. 6, we demonstrate the application of object moving and resizing. As can be seen, our method can naturally move objects within the image, and the edited objects can blend well with the other content in the original image. In Fig. 7, we present the performance of object appearance replacement. It is obvious that our method can replace

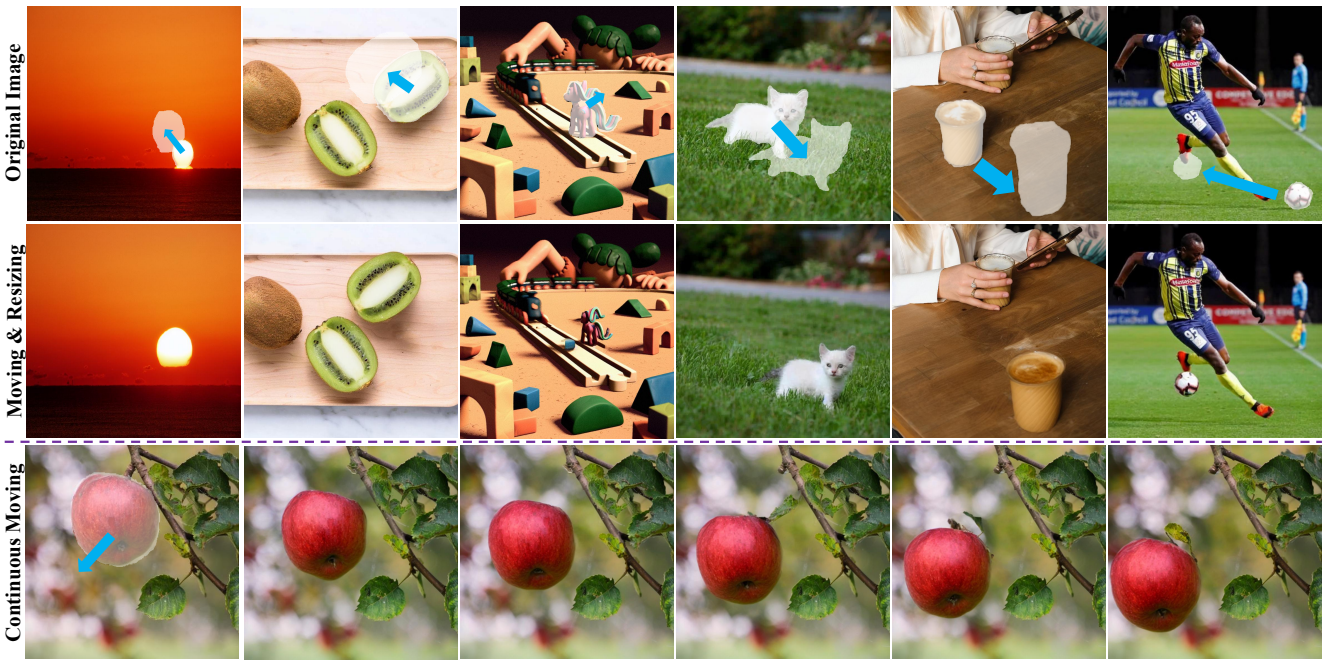


Figure 6. Visualization of our object moving and resizing applications. It can be seen that our DragonDiffusion is capable of effectively moving objects on real images, and at the same time, the region of the original object can also be well inpainted. During the object moving process, we can also selectively enlarge or shrink the object.

Figure 7. Visualization of object appearance replacement. Our method can extract the appearance features of objects within the same category from a reference image, and subsequently replace the appearance of objects in the edited image accordingly.

the appearance with that of a same-category object from a reference image while preserving the original outline. In Fig. 8, we present the content dragging performance of our method. As can be seen, our method can drag the content within the image using a single point or multiple points. The dragging results are consistent and reasonable with the editing direction, and at the same time, the content remains consistent with the original image.

5. Conclusion

Recent studies have shown that intermediate features in diffusion models exhibit strong correspondence relationships. Compared to the correspondence between text and image features, the correspondence between image and image features is more stable and fine-grained. In this paper, we aim to develop a fine-grained image editing scheme based on the strong correspondence of intermediate features in diffusion models. To this end, we design a classifier-guidance-based method to transform the editing signals into

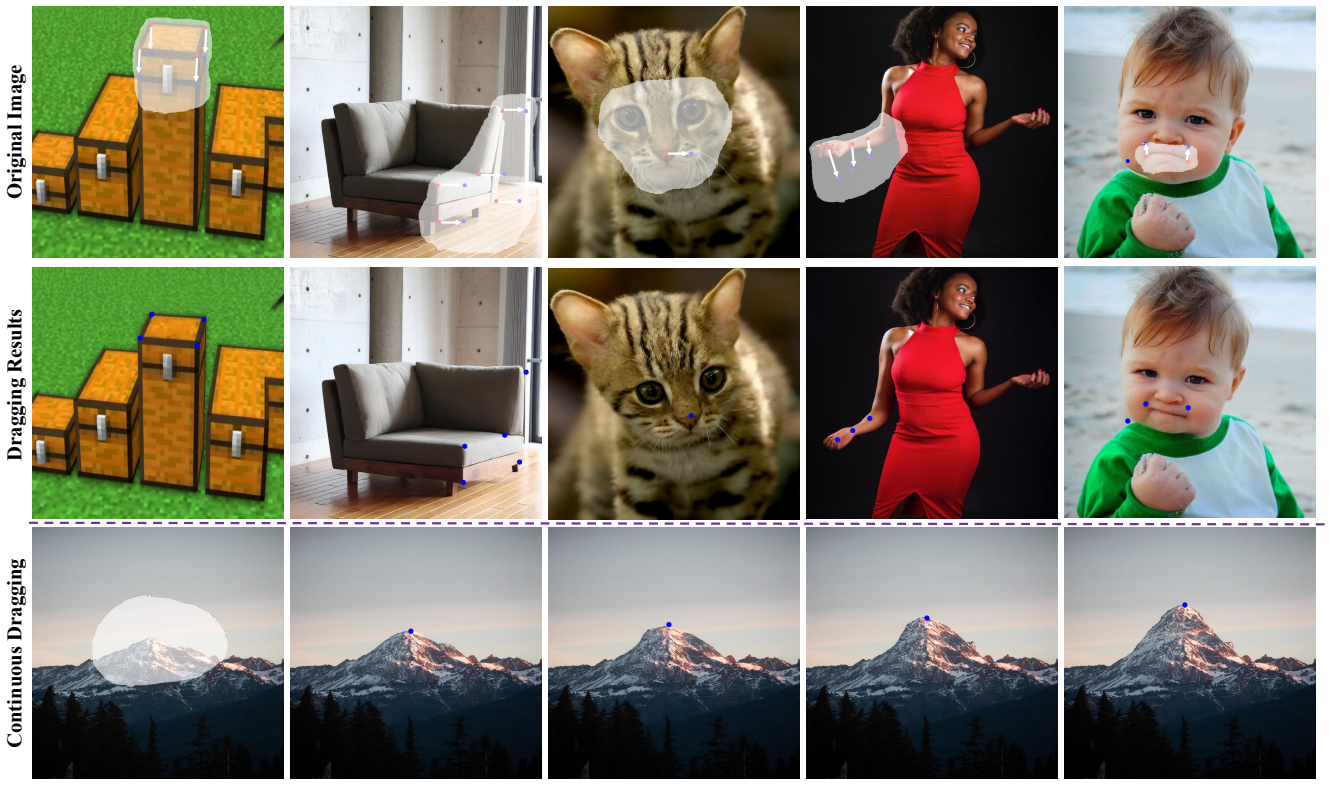


Figure 8. Visualization of content dragging. Our method allows dragging image content using one or multiple points. The results of continuous dragging demonstrate the promising editing capabilities and stability of our DragonDiffusion.

gradients via feature correspondence loss to modify the intermediate representation of the diffusion model. The feature correspondence loss is designed with multiple scales to consider both semantic and geometric alignment. Moreover, a cross-branch self-attention is added to maintain the consistency between the original image and the editing result. Extensive experiments demonstrate that our proposed DragonDiffusion can perform various image editing applications for the generated or real images, including object moving, object resizing, object appearance replacement, and content dragging. At the same time, our DragonDiffusion does not require model fine-tuning or additional modules.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 2, 3

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8296–8305, 2020. 2, 3

[3] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022. 2, 3

[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. 3

[5] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2

[6] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023. 3

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 3

[8] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *arXiv preprint arXiv:2304.08465*, 2023. 6

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2, 3, 4

[10] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 1

[11] Dave Epstein, Allan Jabri, Ben Poole, Alexei A Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *arXiv preprint arXiv:2306.00986*, 2023. 2, 3

[12] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv preprint arXiv:2212.05032*, 2022. 2

[13] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2, 3

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 3, 4, 6

[15] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018. 3

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 3

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2, 3

[18] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 3

[19] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10551–10560, 2019. 3

[20] 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. *arXiv preprint arXiv:2108.01073*, 2021. 3

[21] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 4, 6

[22] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 1

[23] Alexander Quinn 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*, pages 16784–16804. PMLR, 2022. 1, 2

[24] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. *arXiv preprint arXiv:2305.10973*, 2023. 2, 3

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763, 2021. 4

[26] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 3

[28] Simo Ryu. Low-rank adaptation for fast text-to-image diffusion fine-tuning, 2023. 2

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1

[30] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv preprint arXiv:2306.14435*, 2023. 2

[31] Jaskirat Singh, Stephen Gould, and Liang Zheng. High-fidelity guided image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5997–6006, 2023. 3

[32] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2, 3

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 4, 6

[34] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 2, 3

[35] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural*

information processing systems, 33:12438–12448, 2020. 3, 4

[36] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 3, 4

[37] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023. 2, 4

[38] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. *arXiv preprint arXiv:2211.13752*, 2022. 3

[39] Wen Wang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023. 6

[40] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *arXiv preprint arXiv:2212.11565*, 2022. 6

[41] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. *arXiv preprint arXiv:2303.09833*, 2023. 3

[42] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1

[43] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Lafite: Towards language-free training for text-to-image generation. *arXiv preprint arXiv:2111.13792*, 2021. 1