EasyDrag: Efficient Point-based Manipulation on Diffusion Models

Xingzhong Hou[1,2] Boxiao Liu[3] Yi Zhang[3] Jihao Liu[3,4] Yu Liu[3] Haihang You[1]*
[1]State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences
[2]School of Computer Science and Technology, University of Chinese Academy of Sciences
[3]SenseTime Research [4]CUHK MMLab
{houxingzhong, youhaihang}@ict.ac.cn {liuboxiao, zhangyi17}@sensetime.com
jihaoliu@link.cuhk.edu.hk liuyuisanai@gmail.com

Figure 1. DragGAN fails in both cases due to limited model capacity. Similarly, DragDiffusion relies on LoRA fine-tuning and hand-drawn masks to achieve high-quality results, and SDE-Drag also fails to address these two cases effectively. In contrast, our approach ensures precise manipulation and detail preservation without fine-tuning or masks.

Abstract

Generative models are gaining increasing popularity, and the demand for precisely generating images is on the rise. However, generating an image that perfectly aligns with users' expectations is extremely challenging. The shapes of objects, the poses of animals, the structures of landscapes, and more may not match the user's desires, and this applies to real images as well. This is where point-based image editing becomes essential. An excellent image editing method needs to meet the following criteria: user-friendly interaction, high performance, and good generalization capability. Due to the limitations of StyleGAN, DragGAN exhibits limited robustness across diverse scenarios, while DragDiffusion lacks user-friendliness due to the necessity of LoRA fine-tuning and masks. In this paper, we introduce a novel interactive point-based image editing framework, called EasyDrag, that leverages pretrained diffusion models to achieve high-quality editing outcomes and user-friendship. Extensive experimentation demonstrates that our approach surpasses DragDiffusion in terms of both image quality and editing precision for point-based image manipulation tasks. The code will be available on https://github.com/Ace-Pegasus/EasyDrag.

*Corresponding author.

1. Introduction

In the past few years, diffusion models [9, 24, 27] represent a groundbreaking advancement in the domain of generative modeling. In particular, text-to-image diffusion models [24, 25, 31] can produce remarkable images based on textual prompts. Recently, such models have also been extended for image editing [3, 7, 13, 18, 23, 33]. Since diffusion models depend on textual input, image elements, including objects and styles, can be altered with a high degree of quality and diversity guided by edited prompts.

However, text-guided image editing cannot precisely direct changes in the image, and it is unpredictable for a specific point in the image that one might want to move. Therefore, point-guided image editing becomes essential. A recent study DragGAN [21] introduces "drag" operation for precise image editing, which has received widespread attention for its point-based interaction. Nonetheless, it encounters limitations tied to the capacity and adaptability of StyleGAN [11, 12] as shown in Fig. 1. In contrast, diffu-

sion models [9, 24, 27] exhibit better stability and superior generative quality. DragDiffusion [26] enables interactive point-based image editing on diffusion models with LoRA fine-tuning [10] but introduces high time consumption at the same time.

What capabilities should point-based image editing possess? **User-friendly interaction**, **good performance** and **effective generalization**. User-friendly interaction implies ease of use and short response times for users, while good performance refers to precise dragging with maintaining identity. Effective generalization requires the method to work with a single model for various images. Unfortunately, existing methods have not been able to simultaneously exhibit these capabilities. DragGAN lacks generalization of dragging general domain image, while DragDiffusion is less user-friendly due to its reliance on LoRA fine-tuning and masks. In this paper, we propose a novel point-based image editing framework built on diffusion models, which can concurrently fulfill these requirements.

In contrast to DragDiffusion, our method do not require any fine-tuning of pre-trained models, which significantly reduces the response time for users. Since the generative domain of diffusion models are extremely huge and lack of LoRA fine-tuning presents a certain degree of challenge for diffusion-based dragging, we introduce a novel motion supervision method to better align with the feature space, which can get precise point tracking on diffusion models. Similarly, the edited latent often generates images that are inconsistent with the original due to the broad generation space. Built on this, we propose reference guidance during the denoising process, to preserve identity consistency with the input image. Furthermore, we design a method for auto mask generation and achieve simplified dragging interaction, which meets the need of user-friendliness. As shown in Fig. 1, the performance of DragDiffusion relies on LoRA fine-tuning and masks, while our method does not require these and achieves better performance.

Our contributions can be summarized as follows:

- User-friendly interaction. We introduce a point-based image editing method with auto mask generation for user-friendship, simplifying the dragging process while maintaining the comparable results.
- High performance. We propose a stable motion supervision and point-tracking method that has better adaptability to the latent space of the diffusion model to achieve precise manipulation. We also introduce reference guidance during the denoising process after the optimization of the latent code for better identity preservation.
- Effective Generalization. Based on diffusion models, our approach can handle a wide range of natural images and various types of dragging, obviously outperforming GAN-based methods.
- Extensive qualitative and quantitative results demonstrate that our method produces superior and more precise point dragging outcomes with clear gaps when compared to existing point-guided image editing methods.

2. Related work

2.1. Text-guided image editing

With the success of GANs in the field of image generation, many previous image editing methods [1, 6, 29, 34] are based on StyleGAN. These editing approaches map real images into the latent space of StyleGAN ($\mathcal{W}$ space or $\mathcal{W}+$ space) and then achieve image editing by manipulating the latent vectors. However, StyleGAN generates images in a highly restricted domain, and the image quality is not always high. This makes it challenging to successfully invert many real images to a satisfactory latent code. Recently, diffusion models have enabled image synthesis at high quality. An increasing number of image editing tasks [7, 17, 18] have shifted towards being based on diffusion models. Previous image editing methods that rely on diffusion models, such as SDEdit [16] and blended diffusion [2], cannot fully leverage the capabilities of accurate diffusion inversion as they introduced random noise into the input image to create a noisy initial state. Prompt-to-Prompt [7] is the first to accomplish extensive text-guided image editing without the need for diffusion model refinement, including local editing even when a known mask is unavailable. Null-text inversion [18] successfully reconstructs real images by optimizing the null-text embedding at each prediction step, which decides the unconditional prediction. Most recently, pix2pix-zero [22] proposes to learn editing directions in the textual embedding space for specific image translation tasks. In contrast to these methods, our work enables users to conduct image control through point-guided editing.

2.2. Point-guided image editing

In order to facilitate fine-grained editing, several approaches have been introduced for point-based editing, including those by [5, 21, 26, 30]. DragGAN [21] introduces an interactive method for seamless point-based image editing, incorporating two innovative components: the optimization of latent codes to progressively move multiple handle points to their desired positions and a precise point tracking mechanism to accurately follow the path of these handle points. However, the results of DragGAN are constrained by the generative capacity of StyleGAN. Recently, DragDiffusion [26] extends the editing framework of DragGAN to diffusion models with LoRA fine-tuning. Conversely, LoRA fine-tuning significantly extends the waiting time, which is unfriendly to the users. DragonDiffusion [19] utilizes an energy function to guide the editing and a memory bank for editing consistency. SDE-Drag [20] proposes an SDE-based approach to formulate the image

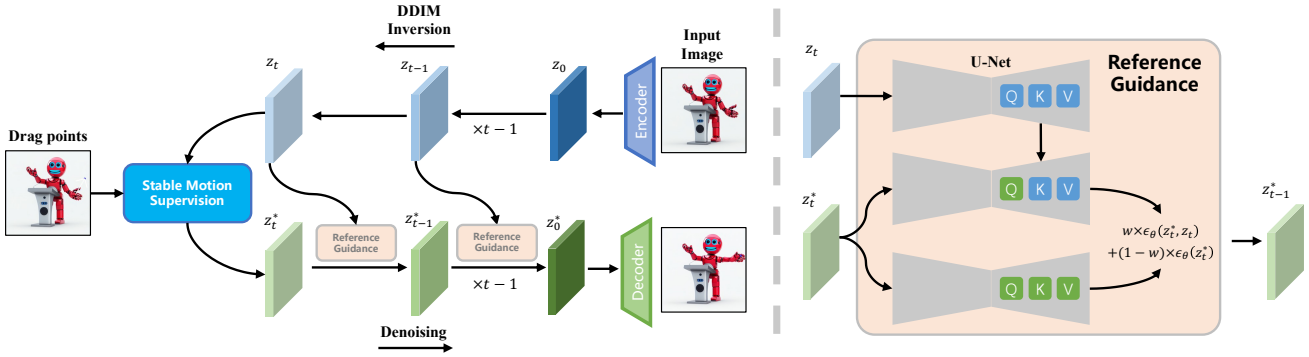


Figure 2. The pipeline of our framework. Our method generates masks automatically for precise supervision and convenience for users. To improve motion supervision over existing methods like DragGAN, we propose a stable motion supervision method based on diffusion models. The reference guidance where guidance latents are inherited from DDIM inversion preserves identity with the input image commendably.

editing. In contrast to previous works, we propose a novel point-based image editing framework, which can generate high-quality results without LoRA fine-tuning and masks.

3. Method

3.1. Preliminaries

Denoising diffusion probabilistic models (DDPM) [9] aim to map a pure noise z_T to an output image z_0, which is guided by the given conditioning prompt. During the training process, the network ϵ_θ is updated by predicting the noise ϵ from the latent variables z_t:

$$\mathcal{L}_\theta = \mathbb{E}_{z_0,\epsilon\sim N(0,I),t\sim U(1,T)}\|\epsilon - \epsilon_\theta(z_t, t, \mathcal{C})\|_2^2, \quad (1)$$

where the noised sample z_t is calculated by adding noise ϵ to z_0 according to diffusion step t and $\mathcal{C}$ is the conditioning input of ϵ_θ. During inference, we use DDIM [27] for sampling method:

$$z_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} z_t + \sqrt{\alpha_{t-1}}\left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1}\right)\epsilon_\theta(z_t), \quad (2)$$

where $\alpha_t (t = 0, 1, 2, ..., T)$ represent the noise scales at corresponding diffusion steps.

DDIM inversion. The forward process of DDIM can be expressed in terms of $\epsilon_\theta(z_t, t, \mathcal{C})$, based on the assumption that the ODE process can be inverted in the limit of small steps:

$$z_{t+1} = \sqrt{\frac{\alpha_{t+1}}{\alpha_t}} z_t + \sqrt{\alpha_{t+1}}\left(\sqrt{\frac{1}{\alpha_{t+1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1}\right)\epsilon_\theta(z_t). \quad (3)$$

Stable Diffusion. Stable Diffusion (SD) [24] presents the input image into a lower-dimensional latent space with a Variational Auto-Encoder (VAE) [15], where the input latent z_0 is defined as $E(x_0)$ and the output image x_0 is obtained by $D(z_0)$. We implement our work based on Stable Diffusion model.

Classifier-free guidance. The extent to which the prompt influences the reverse diffusion process directly affects the quality of image generation. For this purpose, Ho *et al.* [8] proposes the classifier-free guidance, which controls the degree of impact from text conditioning. Specifically, the null next-embedding $\phi = \psi(\text{""})$ is employed as a anchor point for unconditional prediction to enhance the text conditioning with guidance scale w:

$$\tilde{\epsilon}_\theta(z_t, t, \mathcal{C}, \phi) = w * \epsilon_\theta(z_t, t, \mathcal{C}) + (1 - w) * \epsilon_\theta(z_t, t, \phi). \quad (4)$$

3.2. Overview

Given an input image $\mathcal{I}$, the user should input a number of handle points $\{\boldsymbol{p}_i = (x_{p,i}, y_{p,i}) | i = 1, 2, ..., N\}$ and their corresponding target points $\{\boldsymbol{q}_i = (x_{q,i}, y_{q,i}) | i = 1, 2, ..., N\}$. The objective is to drag the content of the handle points to the target points based on image $\mathcal{I}$. The user can also input a mask $\mathbf{M}$ to specify the modifiable area.

We operate DDIM inversion with $\mathcal{C} = \psi(\mathcal{P})$ on the input image, where $\mathcal{P}$ represents the prompt, and get the intermediate results $z_T, ..., z_t, ..., z_0$, where z_t is a specific intermediate latent in DDIM inversion. Loop of optimizations is performed on z_t to make features of the target points similar enough to that of the handle points, with novel stable motion supervision and automatic mask generation introduced in this paper. Then, the result of optimization z_t^* is fed into the denoising process with reference guidance to achieve identity consistency with $\mathcal{I}$. We present the pipeline

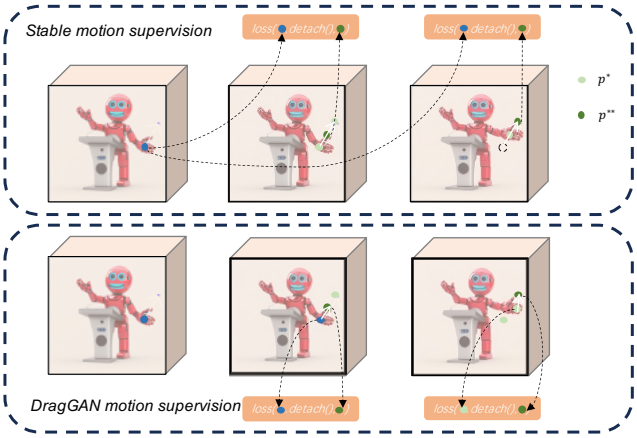


Figure 3. Stable motion supervision. When the next point $\boldsymbol{p}^{**}$ is more similar with the handle point $\boldsymbol{p}$ than $\boldsymbol{p}^{*}$, the handle point has moved to the position of $\boldsymbol{p}^{**}$. Our approach differs significantly from DragGAN in that our method always learns features from the original image, while DragGAN learns features that are continually changing.

in Fig. 2, and elaborate three key components in the following subsections.

3.3. Stable motion supervision

There exists misleading in DragGAN due to the proposed motion supervision. In contrast to DragGAN, we introduce an efficient drag method that is more suitable for diffusion models. Inspired by DIFT [28], we utilize the feature maps of U-Net denoiser as the semantic information of the input image. Specially, we consider the second and third upper layers of U-Net with $(z_t, t, \mathcal{C})$ as inputs, and concatenate the two layers as our feature maps $\mathbf{F}$ used for stable motion supervision. $\mathbf{F}$ is resized to have the same resolution as the input image in order to get precise manipulation.

As shown in Fig. 3, $\boldsymbol{p}^{*}$ represents current position of the handle point, and $\boldsymbol{p}^{**}$ is the next position that we want to drag to. The method for calculating $\boldsymbol{p}^{**}$ is presented in Algorithm 1. We start from $\boldsymbol{p}^{*}$, move along the direction of $\boldsymbol{q}$ by K units, and then find the nearest integer coordinates to determine $\boldsymbol{p}^{**}$. In contrast to DragGAN, we employ $\boldsymbol{p}^{*}$ to learn features of the original image at the $\boldsymbol{p}$ position, ensuring that the learned features are consistently accurate. Inspired by DragonDiffusion [19], we define the similarity loss function of two feature maps as follows:

$$\mathcal{L}_{cos}(\mathbf{F}, \mathbf{F}_0) = \frac{1}{1 + \frac{1+cos(\mathbf{F}, sg(\mathbf{F}_0))}{2}}, \tag{5}$$

where $\mathbf{F}_0$ represents the feature maps corresponding to the input image and $sg(\cdot)$ is the stop gradient operator. Therefore, the stable motion supervision loss is defined as:

Algorithm 1: Next point

Input: Current point $\boldsymbol{p}^{*}$, Target point $\boldsymbol{q}$
Output: Next point $\boldsymbol{p}^{**}$

```
if ‖q − p*‖_2 < K then
    p** ← q;
else
    d ← (q − p*) / ‖q − p*‖_2;
    p** ← p* + round(d * K);
end
return p**
```

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_{cos}(\mathbf{F}(\boldsymbol{p}_i^{**}), \mathbf{F}_0(\boldsymbol{p}_i)) \\ + \lambda \mathcal{L}_{cos}(\mathbf{F} \odot (1 - \mathbf{M}), \mathbf{F}_0 \odot (1 - \mathbf{M})),\end{aligned} \tag{6}$$

where $\mathbf{F}(\boldsymbol{p}_i^{**})$ denotes the feature values of $\mathbf{F}$ at pixel $\boldsymbol{p}_i^{**}$.

After motion supervision, the updated latent code z_t^{*} may change the position of handle points. Consequently, we need to determine whether the handle point has reached the next target point. To do this, we compare which one, $\mathbf{F}(\boldsymbol{p}_i^{**})$ or $\mathbf{F}(\boldsymbol{p}_i^{*})$, resembles $\mathbf{F}_0(\boldsymbol{p}_i)$ more. If $cos(\mathbf{F}(\boldsymbol{p}_i^{**}), \mathbf{F}_0(\boldsymbol{p}_i))$ is higher, the handle point $\boldsymbol{p}_i^{*}$ is updated to $\boldsymbol{p}_i^{**}$ and $\boldsymbol{p}_i^{**}$ is further updated to $next(\boldsymbol{p}_i^{*}, \boldsymbol{q}_i)$ according to Algorithm 1.

3.4. Auto mask generation

For user-friendly interaction, image editing should avoid burdening users with the task of drawing masks. Existing drag methods based on diffusion models [19, 26] always rely on mask inputs, which are provided by users. This requires users to have a deep understanding of the drag method because an incorrectly drawn mask can lead to unpredictable outcomes. Consequently, we propose a method that can automatically generate masks, which are able to change adaptively during the training process.

In the process of motion supervision, the parts with larger gradients of z_t^{*} have a greater impact on the results, while those with smaller gradients are generally the parts that are expected to remain stationary. However, if the regions with relatively small gradients keep changing, the final result becomes unpredictable. Due to the self-attention layer in U-Net, minor changes made at step t are very likely to have an unacceptable impact on the final image. Therefore, we still need control over these low-impact areas. We achieve this by using regions with normalized gradients greater than a threshold g as the drag mask $\mathbf{M}$.

Every time $\boldsymbol{p}_i^{*}$ is updated, the hot-zone of z_t also changes due to the position shifting of the drag point. We update our dynamic mask $\mathbf{M}$ each time $\boldsymbol{p}_i^{*}$ changes, but there is no need to change the mask when $\boldsymbol{p}_i^{*}$ remains the same. This is because for the same $\boldsymbol{p}_i^{*}$ and $\boldsymbol{p}_i^{**}$, the hot-zone is consistent.

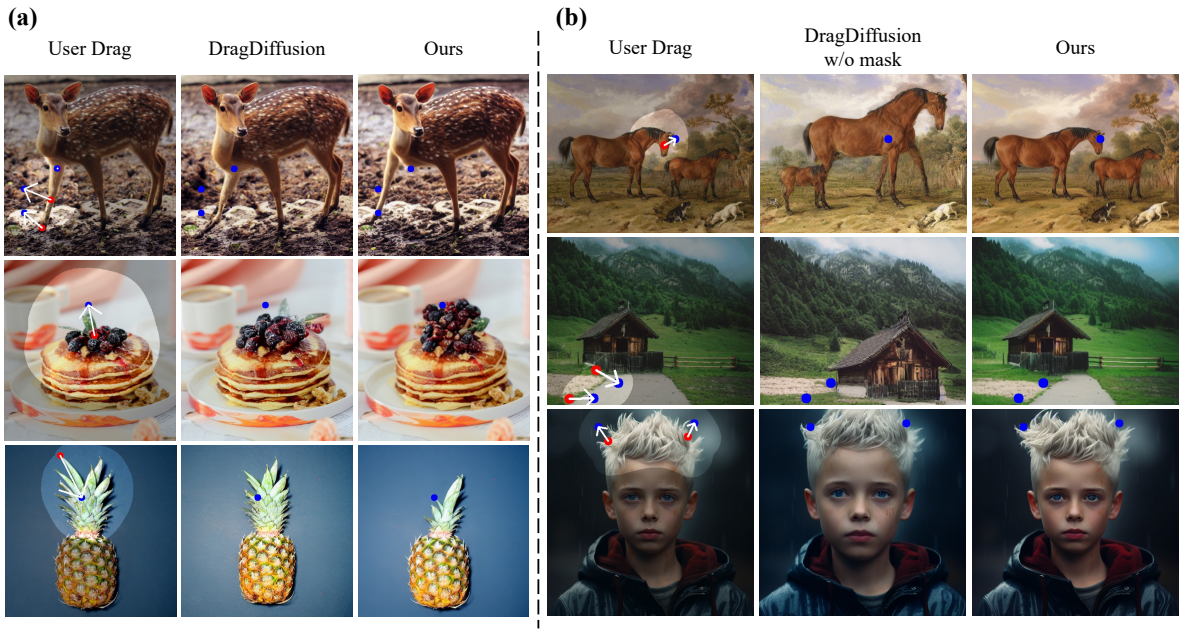


Figure 4. Qualitative comparison of our approach to DragDiffusion. Due to the misleading of motion supervision, DragDiffusion always cannot move the handle points to the target. Furthermore, DragDiffusion often exhibits significant background changes in the absence of the mask. In our method, the handle points reach the target positions precisely even without the mask.

This way, $\mathbf{M}$ adapts to changes of z_t, aligning better with the dragging requirements.

3.5. Reference guidance

With optimized z_i^*, the generated images with direct DDIM denoising often exhibit identity changes and a decrease in image quality. Especially in areas where the pretrained models have limited generation capabilities, naively generating with z_i after DDIM Inversion may not even reproduce the original image. Drawing inspiration from [4, 19, 26], we employ mutual self-attention to ensure that images generated from z_i^* maintain content consistency with the original image.

As illustrated in Fig. 2, we replace the key and value of self-attention layers in backward process of z_i^* with the corresponding key and value in the process of reconstructing the source image $\mathcal{I}$. Due to the feature correlations in lower layers being relatively weak, the mutual self-attention is only applied to the upper layers of U-Net ϵ_θ. We then perform sampling using the linear combination of the reference guidance and normal prediction:

$$\epsilon_i^* \leftarrow w\epsilon_\theta(z_i^*, z_i, i, \mathcal{C}) + (1-w)\epsilon_\theta(z_i^*, i, \mathcal{C}), \qquad (7)$$

where w controls the strength of the reference guidance. Specifically, z_i is the corresponding latent during DDIM inversion, rather than that generated from z_t. When $w = 1$, our reference guidance is the same as mutual self-attention. However, in the cases where the input image is infrequent for the model and z_t undergoes significant changes, using mutual self-attention alone cannot guarantee identity consistency. When we increase the strength of reference guidance, the generated results can better ensure content and texture consistency with the input image.

4. Experiments

In this section, we evaluate the proposed method qualitatively and quantitatively, on both synthetic and real images.

Implementation details. In our experiments, except for generated image manipulations, all tasks are based on Stable Diffusion 1.5 (SD-1.5). Unless otherwise specified, in both DDIM inversion and sampling we set the number of the sampling steps to be 50 via using the stride of 20 over 1000 diffusion steps. We set the latent from the 35-th DDIM step, which contains rich semantic information corresponding to the input image, as z_t. The gradient threshold g is 0.4, the motion step K is 12 and the reference guidance scale w is 4 by default. We use a learning rate of 0.01 for z_t without any decay schedule and train the latent with Adam [14] solver.

Dataset. We employ DragBench introduced in DragDiffusion as our evaluation dataset as it encompasses various types of images, and we also follow its dragging instruc-

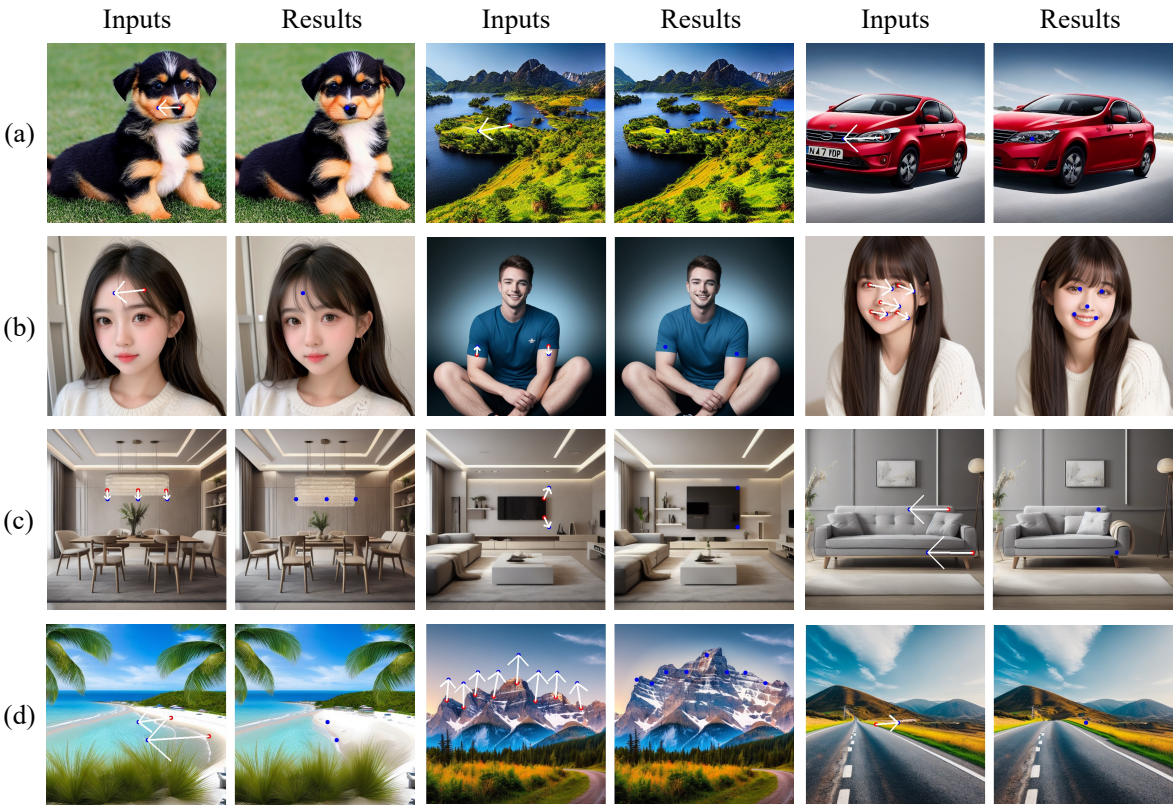


Figure 5. Qualitative results of our approach on generated images with SD-1.5 and its fine-tuned variants. (a) SD-1.5. (b) Majicmix Realistic V7. (c) Interior Design Supermix. (d) Dvarch.

tions. DragBench contains 205 images, with well-annotated prompts, masks, and dragging points.

Evaluation metrics. We evaluate the dragging results from two aspects: the quality of dragged image and the accuracy of position of handle points. We employ Image Fidelity (IF) [13] and Mean Distance (MD) [21] to assess these two aspects, respectively. IF evaluates the similarity between the input and output images, which is calculated by subtracting the mean LPIPS [32] over all pairs of original and edited images from 1. MD quantifies how well the handle points move to the target points. The same as DragDiffusion, we utilize DIFT [28] to identify the points in the output image that correspond to the handle points in the input image. These identified points are then treated as the definitive handle points after the dragging process. Subsequently, MD is calculated as the average Euclidean distance between the positions of all target points and their corresponding final handle points. A higher IF and a lower MD indicate a better quality of the dragging approach.

4.1. Qualitative evaluation

We conduct extensive experiments for the qualitative comparison between our method and DragDiffusion. To provide a thorough evaluation, we adopt SD-1.5 [24] as our base model for all real images, and also employ other fine-tuned variants of SD-1.5 for generated image manipulation.

We present real image editing results for various object categories and user inputs in Fig. 4. Our approach accurately moves the handle points to reach the target positions, resulting in a wide range of natural and diverse manipulation effects such as altering the poses of animals, rearranging landscapes, modifying human body parts, and reshaping objects, in the absence of the mask. Conversely, DragDiffusion is unable to precisely move the handle points to the target positions, which is mainly attributed to the misleading of point tracking that is employed in DragGAN and DragDiffusion. It is worth noting that DragDiffusion always changes the non-dragged regions when no mask is provided.

We also conduct a series of point manipulations based on the generated images, as shown in Fig. 5. The classifier-free guidance scale is configured at 7.5 for various SD-1.5 variants. Notably, the reference guidance is unnecessary for generative editing, as the prompt of generated images pro-

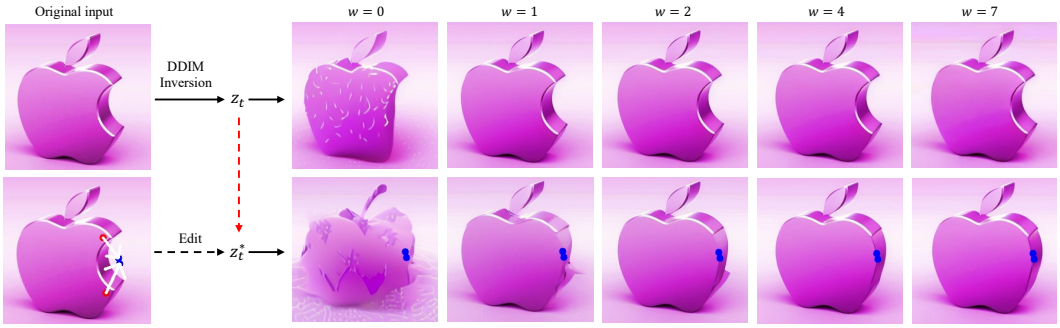


Figure 6. Out-of-distribution manipulations. In some cases, the input image cannot be reconstructed with DDIM Inversion. Utilizing reference guidance, edited image preserves its identity with the original image.

Method	IF(↑)	MD(↓)	Time(s)
DragGAN	0.695	57.3	59.1
DragDiffusion	0.882	36.71	69.4
DragDiffusion w/o LoRA	0.863	53.63	33.7
DragDiffusion w/o mask	0.766	47.03	65.6
SDE-Drag	0.871	45.44	50.1
Ours w/ mask	<u>0.889</u>	**29.07**	**27.3**
Ours	**0.910**	<u>32.55</u>	<u>30.6</u>

Table 1. Quantitive evaluation of different methods with DragBench. Evaluation metrics include Image Fidelity (IF) and Mean Distance (MD).

vides substantial guidance during the denoising process. As shown in Fig. 5, our method achieves point-based editing with high quality and precision on generated images. It can be observed that in some cases, the details of generated images have changed. This is because the self-attention layer in diffusion models may introduce slight variation in other parts when a specific area of the image is edited. If users want to preserve certain areas without any changes, they can employ inpainting during image generation with masks to determine which regions should remain unchanged.

4.2. Quantitative evaluation

We conduct quantitative evaluation by comparing our method to the baseline dragging method on diffusion, DragDiffusion. The evaluation, which incorporates IF and MD metrics, makes use of the DragBench dataset provided by DragDiffusion. The results, as shown in Tab. 1, demonstrate that our approach significantly surpasses DragDiffusion and SDE-Drag on the DragBench dataset. Notably, our method achieves superior dragging results without requiring LoRA fine-tuning, offering substantial time savings during extensive experiments. The performance of DragDiffusion degrades significantly without LoRA nor mask. The execution time in Tab. 1 is calculated on a single A100 GPU.

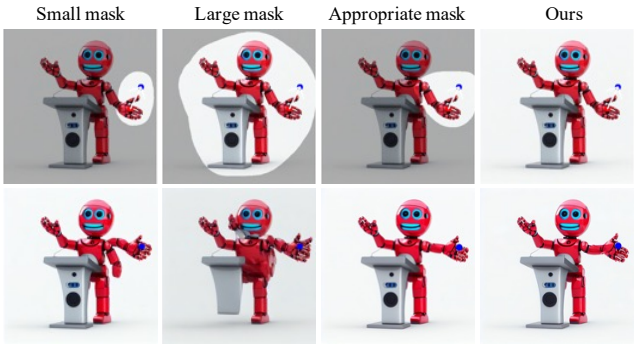


Figure 7. When users do not provide an appropriate mask, the generated results are prone to issues.

4.3. Discussions

Reference Guidance. With our reference guidance, we enable the diffusion models to perform out-of-distribution manipulations. As shown in Fig. 6, the original image cannot be faithfully reconstructed using DDIM Inversion. When we introduce reference guidance, z_t can better reconstruct the input image. It is worth noting that $z_i(i = t, t-1, ..., 1)$ is obtained during the DDIM Inversion process, so it contains richer information compared to z_t. After editing z_t, the image generated from z_t^* is very different from the input image in terms of identity. As we increase the reference guidance scale w, the consistency between the generated image and original image gradually improves. However, if w is too large, the generated image may become oversaturated. In our experiments, a value for w in the range of 3 to 5 is considered a good choice.

Auto mask generation. When users create imprecise masks, the results are unpredictable. As shown in Fig. 7, when the mask is too small, it will restrict the editing of the image. On the other hand, when the mask is too large, changes may occur in the background. Therefore, an appropriate mask is necessary for point-based image editing. Our

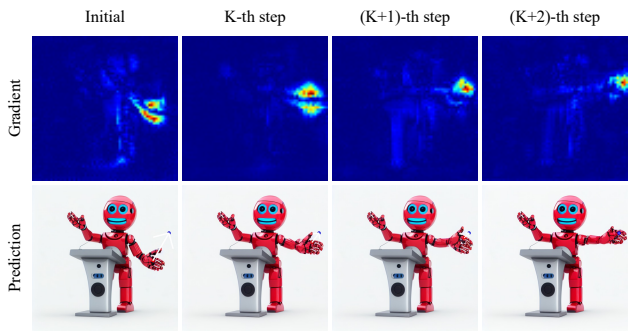
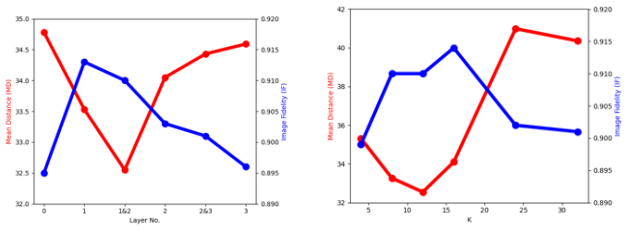


Figure 8. The changes in the gradient and prediction of z_t during optimization. The term of "K-th step" means the K-th update of handle points $\boldsymbol{p}^*$.



(a) Which upper layer to use. (b) Effects of K.

Figure 9. Ablation study.

auto mask generation method liberates users from the need to manually draw masks, which is user-friendly, especially for novice users.

Figure 8 shows the update of gradient and prediction with the optimization of z_t. It can be observed that when the handle points update, the gradient of z_t will change simultaneously. In the K-th step, the whole arm of the robot needs to be lifted, so the gradients for the arm part are relatively large. However, by the time we reach the (K+1)-th step, the robot's arm has already been dragged into the right place, and only the palm part needs to extend outward. Therefore, only the gradients for the palm part are high. Our auto mask effectively captures these details and incorporates this variation into the updates of z_t, thereby providing better control over the direction of z_t. That is the reason why the IF metric of without mask is higher than that with mask in Tab. 1.

Ablation study. We perform ablation studies to analyze the impact of the feature used in stable motion supervision, and we evaluate the performance on DragBench, as shown in Fig. 9a. Notably, the concatenated feature maps from upper layers 1&2 exhibit the best performance, demonstrating an optimal balance between resolution and discriminativeness. Additionally, we investigate the effects of different values of K in Fig. 9b, and it is observed that $K = 12$ yields better results.

Methods	IF(↑)	MD(↓)
DragGAN	0.856	51.32
DragonDiffusion	**0.899**	52.21
Stable motion supervision	0.889	**29.07**

Table 2. Quantitative comparison of motion supervision between our method and previous works. Our approach achieves more accurate dragging results than other methods.

Figure 10. When the manipulated portion in the image is very small or the dragging distance is extensive, it is challenging for the dragging to have a noticeable effect.

A comparison of "drag" method between our approach and previous methods is shown in Tab. 2. We only replace stable motion supervision with the "drag" method in other works. It can be found that stable motion supervision is more suitable for the feature space of diffusion models, although DragonDiffusion achieves slightly higher IF.

Limitations. Despite the outstanding performance, our method has some limitations. Particularly, when the editing area is too small or the dragging distance is too long, the dragging result may be unsatisfactory. As depicted in Fig. 10, when the girl's face occupies only a small part of the image, and we intend to edit the mouth, the desired changes of mouth do not occur as expected.

5. Conclusion

In this work, we attribute an excellent point-based image editing method to three criteria: user-friendly interaction, good performance, and effective generalization. We abandon the LoRA fine-tuning and masks, which is convenient for users. Furthermore, we propose a stable motion supervision method and reference guidance to generate high-quality results based on diffusion models. Leveraging the powerful generative capabilities of the diffusion model, we achieve impressive dragging results across various domains. However, in our experiments, we observe that for very small-scale manipulations or long-distance dragging in images, the results are often not satisfactory. Diffusion models with stronger generative capabilities may be needed to achieve a broader range of dragging manipulations.

Acknowledgements: This work is partially supported by Natural Science Foundation of China Grant 41930110.

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 Transactions on Graphics (ToG)*, 40(3):1–21, 2021. 2

[2] 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

[3] 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. 1

[4] 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. 5

[5] Yuki Endo. User-controllable latent transformer for stylegan image layout editing. In *Computer Graphics Forum*, pages 395–406. Wiley Online Library, 2022. 2

[6] Dave Epstein, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A Efros. Blobgan: Spatially disentangled scene representations. In *European Conference on Computer Vision*, pages 616–635. Springer, 2022. 2

[7] 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. 1, 2

[8] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 3

[10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[11] 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. 1

[12] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1

[13] 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. 1, 6

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[16] 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. 2

[17] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*, 2023. 2

[18] 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. 1, 2

[19] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*, 2023. 2, 4, 5

[20] Shen Nie, Hanzhong Allan Guo, Cheng Lu, Yuhao Zhou, Chenyu Zheng, and Chongxuan Li. The blessing of randomness: Sde beats ode in general diffusion-based image editing. *arXiv preprint arXiv:2311.01410*, 2023. 2

[21] 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. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 1, 2, 6

[22] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2

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

[24] 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 (CVPR)*, pages 10684–10695, 2022. 1, 2, 3, 6

[25] 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. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1

[26] 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, 4, 5

[27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 2, 3

[28] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023. 4, 6

[29] Jianyuan Wang, Ceyuan Yang, Yinghao Xu, Yujun Shen, Hongdong Li, and Bolei Zhou. Improving gan equilibrium by raising spatial awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11285–11293, 2022. 2

[30] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Rewriting geometric rules of a gan. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022. 2

[31] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1

[32] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[33] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6027–6037, 2023. 1

[34] Jiapeng Zhu, Ceyuan Yang, Yujun Shen, Zifan Shi, Bo Dai, Deli Zhao, and Qifeng Chen. Linkgan: Linking gan latents to pixels for controllable image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7656–7666, 2023. 2