StableDrag: Stable Dragging for Point-based Image Editing

Yutao Cui[1,2], Xiaotong Zhao[2], Guozhen Zhang[1,2], Shengming Cao[2], Kai Ma[2], and Limin Wang[1,3] ⋆

[1] State Key Laboratory for Novel Software Technology, Nanjing University
[2] Platform and Content Group (PCG), Tencent [3] Shanghai AI Lab
https://stabledrag.github.io/

Abstract. Point-based image editing has attracted remarkable attention since the emergence of DragGAN. Recently, DragDiffusion further pushes forward the generative quality via adapting this dragging technique to diffusion models. Despite these great success, this dragging scheme exhibits two major drawbacks, namely inaccurate point tracking and incomplete motion supervision, which may result in unsatisfactory dragging outcomes. To tackle these issues, we build a stable and precise drag-based editing framework, coined as ***StableDrag***, by designing a discriminative point tracking method and a confidence-based latent enhancement strategy for motion supervision. The former allows us to precisely locate the updated handle points, thereby boosting the stability of long-range manipulation, while the latter is responsible for guaranteeing the optimized latent as high-quality as possible across all the manipulation steps. Thanks to these unique designs, we instantiate two types of image editing models including StableDrag-GAN and StableDrag-Diff, which attains more stable dragging performance, through extensive qualitative experiments and quantitative assessment on DragBench.

Keywords: Stable dragging · Image editing · Discriminative tracking · Confident motion supervision

1 Introduction

Controllable image editing with generative models [19, 25, 27, 30, 36, 40] has attained remarkable achievements in the past few years, which can customize the generative results for further refinement purposes. Recently the pioneering DragGAN [43] has largely pushed forward accurate image editing with interactive point-based manipulation, that is, driving semantic objects based on user-input handle points toward the corresponding target points. DragGAN formulates a novel dragging technique, primarily contains motion supervision and point tracking, where the former supervises the local patches around the handle points to

⋆ Corresponding author, E-mail: lmwang@nju.edu.cn

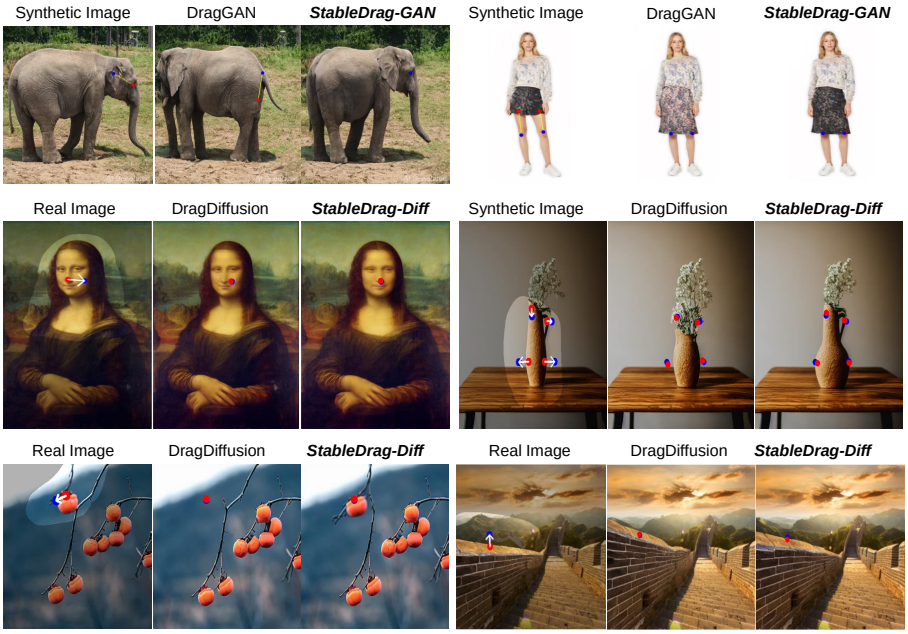

Fig. 1: The comparison between DragGAN/DragDiffusion [50] and our proposed StableDrag. StableDrag-GAN and StableDrag-Diff are our proposed methods constructed upon GAN and Diffusion models respectively. Given an image input (synthetic image by GAN/Diffusion model, or real image), users can assign handle points (red points) and target points (blue points) to drive the semantic positions of the handle points to reach corresponding target points. The example of the Mona Lisa portrait and examples in the last row are the real-image inputs, while the others are synthetic from StyleGAN2 or Stable Diffusion-V1.5 [47] models. The examples demonstrate that our method achieves more precise point-level manipulation and generates higher-quality editing image than DragGAN and DragDiffusion.

move towards the target points step by step, while the latter is responsible for locating the updated handle points at each step.

Despite the great success of DragGAN, its editing ability is still constrained by the inherent model capacity and generality of generative adversarial networks. Therefore, recent works [41, 50] resort to diffusion models [13, 16, 17, 20, 23, 28, 31, 42, 46, 48, 49] for high-quality drag-style image editing. A representative work DragDiffusion [50] explores to adapt the dragging scheme to diffusion models, i.e., first fine-tuning a LoRA, then optimizing the latent at a single diffusion step, finally denoising the optimized latent based on MasaCtrl [7]. For the key component of diffusion latent optimization, it directly follows the DragGAN's convention of iteratively conducting motion supervision and point tracking. We analyze that the current dragging scheme still suffers from the following issues. i) ***Inaccurate point tracking.*** These methods leverage the feature difference as the similarity measurement to track the updated handle points, which is insufficient to precisely locate the right ones from the distractors (i.e., the around

misleading points with similar content). Especially in diffusion models, since the features are sampled from the intermediate diffusion process with much noise injection, the updated points become increasingly challenging to be distinguished from their local surroundings. This may lead to unsatisfactory dragging outcomes, as showcased by the examples of the Mona Lisa portrait and the vase in Fig. 1. ii) ***Incomplete motion supervision.*** During the motion supervision process, the latent may not be adequately optimized at certain steps, resulting in a deterioration of the manipulation quality (see examples of the elephant and the woman in Fig. 1) as well as the point tracking drift. In diffusion models, the latent is more stable and harder to manipulate than GAN's [50], especially when fine-tuning the LoRA on a specific image, which may aggravate the problem.

Considering the aforementioned issues, we argue that there are two primary principles for designing a more stable dragging framework. First, *a robust yet efficient point tracking* method is required, to avoid locating the incorrect points and increasing much latency, thus enabling the point-based drag to be precise. Second, we should guarantee *the motion supervision to be complete* at each optimization step, so as to keep the editing content as high-quality as possible across all the manipulation process, and fully unleash the strong restoring power of generative models. In addition, complete motion supervision can enhance the similarity between the content of the given handle points and the updated points, preventing the accumulation of tracking errors.

Driven by the above analysis, we re-formulate the dragging scheme of point tracking and motion supervision in DragGAN and DragDiffusion, and present a more stable dragging framework for point-based image editing, coined as **StableDrag**. Specifically, inspired by the success in visual object tracking [4,10,12], we try to derive a simple yet powerful point tracking model, in the form of a convolution filter, from a discriminative learning loss. This model is capable of suppressing the tracking confidence score of the distractor points as well as enhancing that of the handle points. At the beginning of the manipulation steps, we update the tracking model weights under the supervision of a tailored similarity learning function. Once the tracking model is prepared, we employ it, in conjunction with the original feature difference method for robust and precise point tracking. Notably, this approach scarcely increases inference latency, since we only need to optimize the simple tracking model (i.e., a single convolution filter) at the initial manipulation step. Furthermore, we design a confidence-based latent enhancement strategy, to make motion supervision complete enough at each step. In detail, we utilize the tracking confidence score of the handle points to assess the quality of the current manipulation process. Normally, we use the same manner of motion supervision as DragDiffusion. Nevertheless, when the quality score falls below an acceptable threshold, we employ the template features (i.e., the initial features of the given start handle points) to supervise that of the current handle points' content, until its confidence score is satisfactory. Thanks to the unique designs for dragging scheme, we instantiate two types of image editing models including StableDrag-GAN and StableDrag-Diff, built on

GAN and Diffusion models respectively, which attains more stable and precise drag performance. Our contributions are summarized as follows:

- We propose a discriminative point tracking method, which allows the model to accurately distinguish the updated handle points from the distractor ones, hence promoting the stability of dragging.
- We devise a confidence-based latent enhancement strategy for motion supervision, which can improve the optimization quality at each manipulation step.
- Under these designs, we build **StableDrag**, a point-based image editing framework, upon different generative models including GAN and Stable Diffusion. Through extensive qualitative experiments on a variety of examples and quantitative assessment on DragBench [50], we demonstrate the effectiveness of our StableDrag-GAN and StableDrag-Diff.

2 Related Work

2.1 Image Editing

Image editing is a hot topic with a wide range of applications. Generative Adversarial Networks (GANs) have made significant strides in the field of image generation [18, 26], leading to numerous prior image editing techniques [2, 14, 32, 43, 45, 53] being founded upon the GAN framework. Nonetheless, the model capacity of GANs remains somewhat constrained, as well as the challenge of effectively transforming real images into GAN latent spaces [1, 9, 37], the practicality of these approaches was inevitably constrained. Recently, large-scale text-to-image diffusion models have produced remarkably realistic generation results [13, 16, 17, 20, 23, 28, 31, 42, 46, 48, 49], which have given rise to numerous diffusion-based image editing methods [3, 6, 7, 15, 22, 29, 35, 38, 39, 41, 44, 51]. These techniques primarily strive to edit images by adjusting the prompts associated with the image. Nevertheless, as many editing endeavors prove challenging to convey through text, the prompt-based strategy frequently modifies the image's high-level semantics or styles, thereby lacking the capability to achieve precise pixel-level spatial manipulation.

In order to facilitate fine-grained editing, a number of studies have been proposed to execute point-based modifications, such as [14, 43, 52]. In particular, DragGAN has exhibited remarkable dragging-based manipulation through two straightforward components: the optimization of latent codes to shift the handle points towards their desired destination points and a point tracking mechanism to locate the updated handle points. However, its generality is constrained due to the limited capacity of GAN. DragDiffusion [50] and DragonDiffusion [41] further extend the dragging scheme to diffusion models to leverage its excellent generative capacity. FreeDrag [36] has proposed to improve DragGAN by introducing a point-tracking-free paradigm. In this work, we explore a new dragging scheme with re-formulating a confident motion supervision module and a discriminative point tracking module, enabling stable point-based image editing.

2.2 Visual Tracking

Since the proposed discriminative point tracking takes inspiration from the visual tracking research, we give a brief overview for these methods. We divide the works into three categories. First, correlation-filter-based trackers [5, 12, 21] learned an online target-dependent discriminative model for tracking. [5, 21] employed online correlation filters to distinguish targets from background and obtains good performance with a high running speed, which is very practical until now. Second, Siamese-based trackers [4, 34] attract a lot of attention due to its simplicity and efficiency. These methods combined a correlation operation with the Siamese network, modeling the appearance similarity and correlation between the target and search. SiamFC [4] employed a Siamese network to measure the similarity between the template and the search area with a high tracking speed. SiamRPN++ [33] improved cross correlation to depth-wise cross correlation, which can increase both the performance and efficiency. Finally, some recent trackers [8, 10, 11] introduced a transformer-based integration module to capture the similarity between the target and search region. Inspired by these findings, we devise a robust point tracking model via discriminative learning. Different from these works, we build the tracking model on top of the intermediate feature of GAN or diffusion models to leverage their discriminativeness and only optimize the tracking model, which is effective yet efficient.

3 Method

3.1 Preliminary on Point-based Dragging

Firstly, we briefly review the recent literature on the point-based dragging framework behind GAN and diffusion models, which are the basics of our work.

DragGAN. Given an image generated by GAN models [24, 27], in conjunction with the user-input handle points $\{p_i = (x_{pi}, y_{pi}), i = 1, 2, ..., n\}$ and the target points $\{t_i = (x_{ti}, y_{ti}), i = 1, 2, ..., n\}$, DragGAN aims to drive the content at every handle point p_i move towards their corresponding target point t_i. In this sense, the primary concern lies in how to precisely control the point-level editing while maintaining high image fidelity. To achieve the goal, DragGAN tailors a novel paradigm, which involves repeated motion supervision and point tracking. Considering the generator's characteristic that the intermediate features are very discriminative, they leverage a simple online motion supervision loss to optimize the latent code. When denoting the local region around p_i as $\Theta(p_i)$, i.e, the pixels whose distance to p_i is less than the radius r_i, the loss can be defined as:

$$\mathcal{L}_1 = \sum_{i=0}^{n} \|\mathbf{F}(\Theta(p_i)) - \mathbf{F}(\Theta(p_i + d_i)\|_1 + \eta\|(\mathbf{F} - \mathbf{F}^0) \cdot (1 - \mathbf{M})\|_1, \tag{1}$$

where $\mathbf{F}$ represents for the intermediate feature at current optimization step, $\mathbf{F}^0$ is the feature at initial step, n is the number of handle points, $d_i = \frac{t_i - p_i}{\|t_i - p_i\|_2}$ is

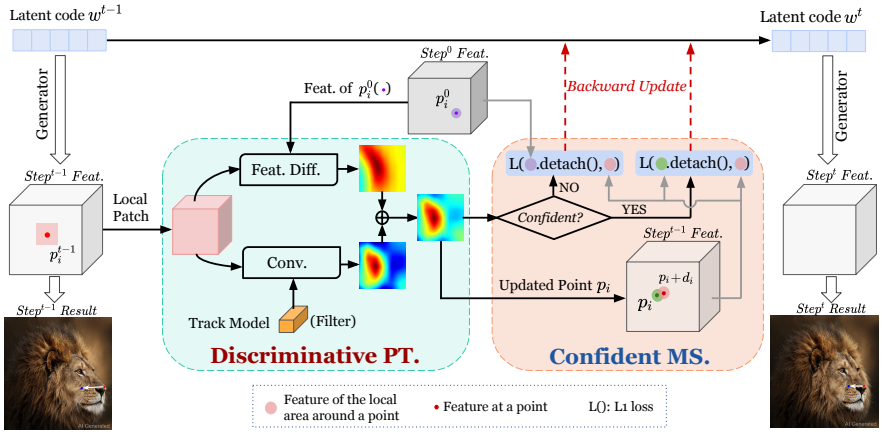

Fig. 2: Illustration of our dragging scheme for an intermediate single-step optimization. The core of the dragging pipeline illustrated herein is based on GAN, whereas the one based on diffusion models remains the same. 'Discriminative PT.' denotes for discriminative point tracking module and 'Confident MS.' represents for confident motion supervision process. p_i means the current handle point at i^{th} step optimization. Notably, the tracking model, in the form of a convolution filter, is only learned at the first optimization step and can be just employed in the subsequent steps. Details about its learning process at the first step are described in Fig. 3. The latent code w is supposed to be optimized via the backward updating across all steps.

a deviation vector and $\mathbf{M}$ is the pre-defined mask to control the changing area. Particularly, since the $\mathbf{F}(\Theta(p_i))$ gets detached, the content of current p_i will be motivated to t_i by a small step. However, due to the inherent indeterminacy of optimization, it is hard to guarantee p_i to approach $p_i + d_i$. Consequently, they utilize a simple feature difference method as point tracking to determine the updated state of pi. The above optimization process iterates until each of the handle points p_i converges to their respective target points t_i.

DragDiffusion. DragDiffusion [50] extends the point-based editing framework to diffusion models, such as Stable Diffusion (SD-V1.5 [47]), so as to unleash its strong power of high stability and superior generation quality. This editing method involves three sub-processes, i.e., finetuning a LoRA on the real image, optimizing the latent on a certain diffusion step and denoising the updated latent to generate the edited image. Specifically, they adopt the same dragging formulation of repeated motion supervision and point tracking on a single intermediate diffusion step to manipulate the latent. Besides, a LoRA finetuing strategy is employed to preserve the image identity through the whole manipulation process. Finally, a self-attention control mechanism MasaCtrl [7] is used to enhance the consistency between the original image and the edited image.

3.2 Overview

As illustrated in Fig. 1, DragGAN and DragDiffusion may result in deteriorated editing images due to the imprecise point tracking and incomplete motion supervision. Therefore, in this work, we cast attention on the current dragging technique to achieve more stable and precise image manipulation. The developed dragging pipeline is illustrated in Fig. 2, which comprises a discriminative point tracking module and a confident motion supervision module. Specifically, we design a new point tracking approach that integrates the original feature difference with the tracking score yielded from a learned discriminative tracking model, thereby boosting the point tracking accuracy as well as the drag precision. Based on the tracking score, we then explore a confidence-based latent enhancement strategy to achieve complete enough motion supervision. We also observe that DragGAN masters large deformation and creative content (e.g., transforming a lion with its mouth closed into a roaring state) within a short run-time. While DragDiffusion is good at generating superior-quality and higher-fidelity editing outcomes. To enable the dragging model to accommodate a wide range of scenarios, we build StableDrag upon both DragGAN and DragDiffusion with the designed dragging scheme. In this section, we will introduce the proposed dragging method in details.

3.3 Discriminative Point Tracking

Point tracking serves as a pivotal function in identifying the updated handle points p_i, to circumvent dragging erroneous points and produce unsatisfactory editing results. The prevalent approach employed in DragGAN and DragDiffusion is straightforward, that is, conducting nearest neighbor search by identifying the position with minimal feature difference to the initial feature template of p_0. However, this entirely ignores background appearance information, which is crucial for discriminating the handle points from the similar ones in the complex scene. Particularly, in diffusion models, since the supervision features are extracted from the intermediate diffusion stage, which incorporates substantial noise, it becomes progressively difficult to discern the updated points. For instance, as shown in the case of the Mona Lisa portrait of Fig. 1, the handle point of the nose possesses similar appearance with the adjacent points, which causes the misleading location in DragDiffusion. Therefore, in this work, we explore an alternative method for accomplishing more discriminative yet simple point tracking.

Distinguishing the given handle points from the distractors can be addressed using a learnable discriminative tracking model. In our design, the point tracking model constitutes the weights of a convolutional layer, providing the point classification scores as output. In detail, we propose to learn a function $g(\mathbf{F}(\Theta_2), z_i)$, where g denotes a convolution function, Θ_2 is the local patch around the current handle point pi and z_i is the learned tracking model, which returns a high score if the tracking model z_i matches the content at a certain position and discerns it as the updated handle point p_i, and a low score otherwise. In particular, the

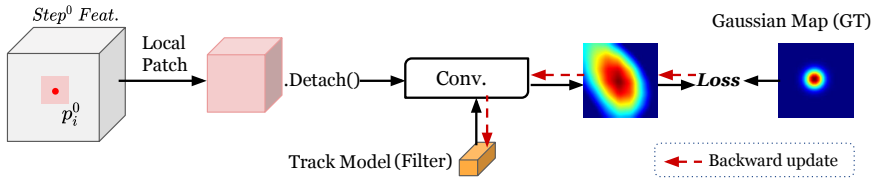

Fig. 3: Learning process of our point tracking model. It is only performed before the manipulation process. The initial feature of the local patch gets detached, indicating that only the tracking model is supposed to be optimized. The tracking model weight is initialized with the the template feature f_i.

tracking model z_i is learned before the latent optimization and keep unchanged across all the manipulation steps. In this sense, this approach scarcely increases the editing runtime. Finally, we merge the classification score yielded by the tracking model with the original feature difference score, so as to achieve both discriminative and precise point location. The detailed procedure of the discriminative point tracking is illustrated in Fig. 2.

Formally, given the local patch $\Theta_2(p_i, r_2) = \{(x, y) \mid |x - x_{pi} < r_2|, |y - y_{pi} < r_2|\}$, the tracked point p_i is updated as:

$$\begin{aligned} S(\Theta_2) &= \lambda * e^{-\|\mathbf{F}(\Theta_2) - f_i\|_1} + (1 - \lambda) * g(\mathbf{F}(\Theta_2), z_i), \\ p_i &:= \underset{q_i \in \Theta_2(p_i, r_2)}{\arg\max} \; S(\Theta_2(p_i, r_2)), \\ s_i &= \underset{q_i \in \Theta_2(p_i, r_2)}{\max} \; S(\Theta_2(p_i, r_2)), \end{aligned} \tag{2}$$

where $S(\Theta_2)$ represents for the tracking confidence score map of the local patch Θ_2, λ is the weighting factor, $f_i = \mathbf{F}^0(p_i^0)$ is the original feature of the initial handle point p_i^0 at the step-0, and s_i is the maximal tracking confidence score at the current step, which is used to guide the motion supervision. In the terms of $S(\Theta_2)$, the former one measures the feature difference the template and the search region. Although it can provide accurate point localization in the majority of instances, it may be misled by the distractor points. Therefore, the second term is responsible to improve the tracking robustness with the discriminative learning, i.e., suppressing the score of surrounding points during the initial optimization process for z_i. Unlike the plain feature difference method, this tracking model is capable of leveraging background information and harnessing distinguishing characteristics of the intermediate feature, thus providing a valuable enhancement to the original approach.

Learning for tracking model $\mathbf{z_i}$. The learning of the point tracking model z_i, which is a convolutional filter with the size of $1 \times C \times 1 \times 1$, is performed before the manipulation process. Overview of the learning process is shown in Fig. 3. We use f_i to initialize z_i and update the weights under the supervision of the following loss:

$$\mathcal{L}_{track} = \|g(\mathbf{F_0}(\Theta_2(p_i, r_2)), z_i) - y_i\|^2. \tag{3}$$

Here, $\mathbf{F_0}$ denotes the initial feature at step-0, y_i represents for the ground-truth label, which is the desired confidence scores at each position, generally set to a Gaussian function centered at p_i. During the learning process, the gradient is not back-propagated through $\mathbf{F_0}(\Theta_2(p_i, r_2))$. In other words, we only need to optimize the tracking model z_i, allowing for rapid convergence. Through the optimization, we highlight the handle points while simultaneously suppressing the confidence score of the background points. Then in the subsequent manipulation steps, the tracking model z_i keeps unchanged for efficiency.

3.4 Confident Motion Supervision

Motion supervision is the core to progressively encourage the points to move towards their intended destination. DragGAN employs an online loss in equation (1) to achieve the goal, however may yielding unsatisfactory results in long-range drag. Alternatively, we devise a confident motion supervision component based on the tenet that, *not only ensuring high-quality and comprehensive supervision at each step but also allowing for suitable modifications to accommodate the novel content creation for the updated states*. For example, the case of a woman wearing a skirt in Fig. 1 demonstrates the significance of complete supervision in maintaining visual coherence.

To attain the above goal, we propose a confidence-based latent enhancement strategy as shown in Fig. 2. Firstly, we introduce the maximal value of the tracking score, i.e. s_i, to represent the current supervision confidence, and the confidence score s_1 at the step-1 to produce the threshold for enhancement strategy. Normally, the original motion supervision as in equation (1) is employed when we discern the current state being confident enough. If the current confidence score falls below the pre-defined threshold, we resort to the initial template for supervision. The concrete enhancement supervision is defined as:

$$\begin{aligned}\mathcal{L}_2 = &\sum_{i=0}^{n} \|\mathbf{F}^0(\Theta(p_i^0)) - \mathbf{F}(\Theta(p_i + d_i)\|_1 \\ &+ \eta\|(\mathbf{F} - \mathbf{F}^0) \cdot (1 - \mathbf{M})\|_1,\end{aligned} \tag{4}$$

where $\mathbf{F}^0(\Theta(p_i^0))$ is the fixed template with no gradient back-propagating, which can enforce the content of updated points to mimic the initial state. Moreover, the choice of whether to use this latent enhancement supervision is determined according to the following guidelines:

$$\mathcal{L}_{motion} = \begin{cases} \mathcal{L}_1, & s_i > \tau * s_1, \\ \mathcal{L}_2, & s_i <= \tau * s_1, \end{cases} \tag{5}$$

where τ is a threshold rate to control the enhancement strength. In this way, we can prevent the current content of handle points from significantly deviate from the original template, thus achieving confident motion supervision. On the other hand, when the confidence score surpasses the threshold, we rely on the dynamic motion supervision $\mathcal{L}_1$ to sustain a high editability.

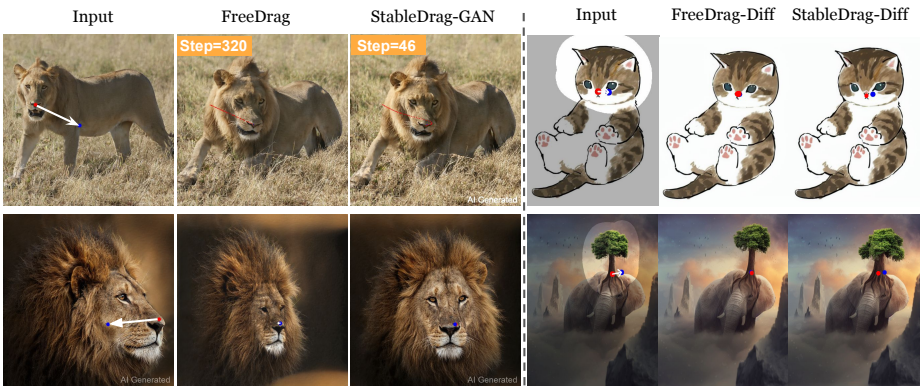

Fig. 4: Comparison between FreeDrag [36] and our StableDrag. For the example in the top left, handle points at each optimization step are visualized to show the difference of the optimization path of FreeDrag and our StableDrag-GAN. The example in the bottom left is to demonstrate our method's strength in creating novel content. And the others are to show that StableDrag can generate more precise dragging outcomes.

Discussion. To better expound the insight of the confident motion supervision, we make a comparison with the method proposed in FreeDrag [36], which employs an adaptive template and a linear search to set free the point tracking module. First, The preset linear search in FreeDrag may impose restrictions on the flexibility of the latent optimization, thereby significantly increasing the difficulty of dragging. As shown in the top-left example of Fig. 4, The handle points of FreeDrag frequently oscillate along the predefined path and necessitate 320 steps of optimization. However, our method allows the handle points to move towards the destination along *a more optimal path*, which is not linear, in only 46 steps. Besides, FreeDrag struggles in generating creative and out-of-distribution content, as demonstrated by the bottom-left example in Fig.4, since it primarily relies on a template feature for supervision, even though an updating strategy is employed. In contrast, our StableDrag-GAN can generate satisfactory creative content given a long-range dragging path, demonstrating better editability.

4 Experiments

4.1 Implementation Details

We implement the approach, including StableDrag-GAN and StableDrag-Diff, based on PyTorch. During the process of optimizing the latent code w_i, we use Adam optimizer with learning rate of 0.01 for StableDrag-Diff and the 0.001 for StableDrag-GAN, which follows their default settings. In most cases, the hyperparameters of λ and τ are set to 0.3 and 0.4, respectively. For other parameters and model settings, we follow the default ones in DragGAN and DragDiffusion. The experiments are conducted on an NVIDIA V100 GPU.

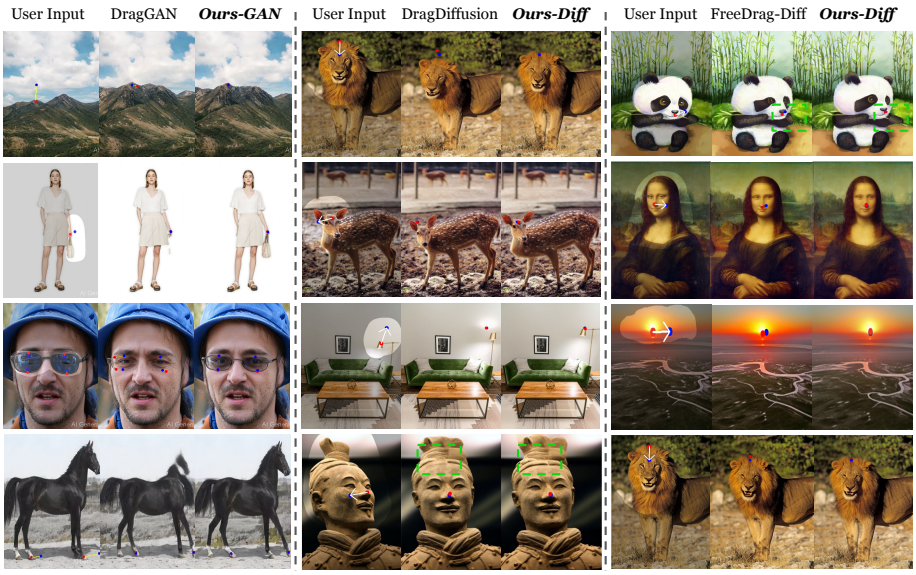

Fig. 5: Comparison between DragGAN [43]/DragDiffusion [50]/FreeDrag [36] and our StableDrag. As in DragGAN, users can optionally draw a mask of the flexible region (brighter area), keeping the rest of the image fixed. The green dashed box in the examples of *the Terra Cotta Warriors Sculpture* and *the Panda* is to show the differences in detail. Best viewed with zooming in.

4.2 Qualitative Comparison

Fig. 5 shows the qualitative results between DragGAN and StableDrag-GAN, DragDiffusion and StableDrag-Diff, FreeDrag-Diff and StableDrag-Diff for fair comparison. To evaluate the method's generality, for the GAN-based models, the input images are generated from StyleGAN2 [27]. While for the Diffusion-based models, we input real images and use DDIM inversion to reconstruct them. It can be seen that our method can more precisely move the handle points to the target points, such as the mountain peak, the lion's chin, the deer's forehead and the little lamp. Besides, our StableDrag can generate higher-quality and higher-fidelity editing results, for example, maintaining the appearance of the bag, the glasses, the horse and the Terra Cotta Warriors sculpture. We also compare our StableDrag-Diff with the FreeDrag [36] based on Diffusion model. We can see that ours-Diff produces more precise results and maintains the details of the initial images. This demonstrates the effectiveness of the proposed discriminative point tracking and confident motion supervision, which can achieve more stable dragging performance.

4.3 Quantitative Results

We quantitatively evaluate our method on DragBench [50], comprising 205 samples with pre-defined drag points and mask. We notice that, in DragBench, there

Table 1: Quantitative comparison on DragBench. 'MD' denotes Mean Distance ↓ and 'IF' is the Image Fidelity (1-LIPIPS) ↑.

Optimization Steps	60	80	100
Metric	MD/IF	MD/IF	MD/IF
DragDiffusion	39.58/0.876	37.98/0.868	38.86/0.863
StableDrag-Diff	**36.36**/**0.893**	**36.98**/**0.884**	**35.92**/**0.869**

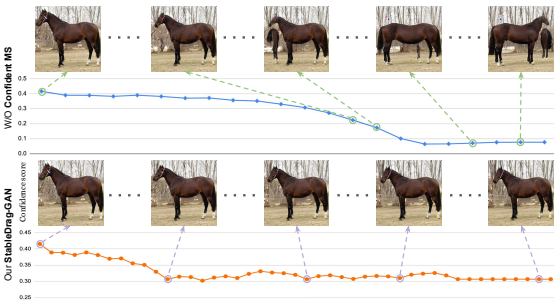

Fig. 6: Effects of the latent enhancement strategy employed in confident motion supervision component. 'W/O' denotes only using the original motion supervision method as in DragGAN.

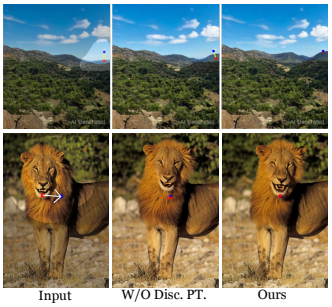

Fig. 7: Effects of the discriminative point tracking. 'W/O' denotes only using the plain feature difference method.

are many examples that are not compatible with proper StyleGAN2 models, so we only conduct the experiments on DragDiffusion and ours-Diff. We compare our StableDrag-Diff to DragDiffusion and use the same LoRA weights and the common hyper-parameters for fair comparison. As shown in Table 1, under three different setting of the optimization steps, StableDrag-Diff consistently outperforms the DragDiffusion, especially surpassing the baseline by 3.22 of Mean Distance score and 0.017 of Image Fidelity score with 60-step optimization. This further indicates that our StableDrag can achieve promising results in editing accuracy and content consistency via the proposed confident motion supervision and discriminative point tracking.

4.4 Exploratory Analysis

To verify the effectiveness and give a thorough analysis on our proposed method, we perform a detailed ablation study through qualitative visualization based on both GAN and diffusion models, and quantitative evaluation on DragBench based on diffusion models.

Confident motion supervision. Here we study the effect of our confident motion supervision component. Firstly, we conduct experiments of the horse editing based on StableDrag-GAN. It can be seen from Fig. 6 that, as the confidence score gradually decreases, StableDrag without the confident motion supervision

Table 2: Effects of our discriminative point tracking and confident motion supervision. 'DPT' denotes the discriminative point tracking and 'CMS' is the confident motion supervision.

Metric	Mean Distance ↓	Image Fidelity ↑
DragDiffusion	39.58	0.876
StableDrag-Diff W/O DPT.	38.63	**0.895**
StableDrag-Diff W/O CMS.	37.87	0.875
StableDrag-Diff	**36.36**	0.893

Table 3: Analysis on time consuming of training tracker and the drag process. Evaluation is performed on StableDrag-Diff.

Model	Tracker iters.	Drag steps	Tracker time(s)	Drag time(s)
StableDrag-Diff	1000	60	1.17	29.06
StableDrag-Diff	1000	80	1.08	38.80

module produces low-quality editing image. This indicates the importance of performing confident supervision at each step, and also demonstrates that the tracking score can reflect the quality of motion supervision. As shown in Table 2 ,the image fidelity decrease by 0.018 when substituting the confident motion supervision with original supervision method in DragGAN, which further substantiates the above conclusion.

Discriminative point tracking. In Fig. 7 and Table 2, we evaluate our StableDrag and the one without the discriminative tracking model. We can see that StableDrag without the discriminative tracking model may suffer from misleading by the background distractor points, causing inaccurate results. Especially, StableDrag-Diff without our discriminative tracking model increases the StableDrag-Diff by 2.27 of Mean Distance. From the results, we can derive that the proposed discriminative tracking model helps the dragging model to achieve more accurate point-based image editing.

Practicality of the tracking module. The proposed point tracker is concise in both formulation and implementation. As shown in Table. 3, the training process of the tracker (about only 1 second) costs far less time than the drag process. As for the point tracking before each supervision step, it runs very fast since only a convolution operation should be performed. It is worth noting that, during the point tracking process, we use a local search strategy to avoid discerning two completely similar objects (e.g., two almost identical dog) in global area. Besides, the core code implementation is simple and easy to adapt to other related methods, since only around 60-rows code is added to the baseline. And we will release the code.

Sensitivity analysis on τ and λ. To better understand the robustness of the proposed method, we have conducted sensitivity analysis on τ and λ as in Table 4

Table 4: Sensitivity analysis on τ, where λ is fixed to 0.0.

τ	0.0	0.2	0.4	0.6	0.8	1.0
MD/IF	42.1/0.868	41.6/0.874	**39.8**/0.891	43.3/0.913	47.4/0.939	51.2/**0.955**

Table 5: Sensitivity analysis on λ, where τ is fixed to 0.0.

λ	0.0	0.2	0.4	0.6	0.8	1.0
MD/IF	42.1/0.868	41.6/0.869	41.6/0.87	**37.9**/**0.875**	40.7/0.874	39.0/0.875

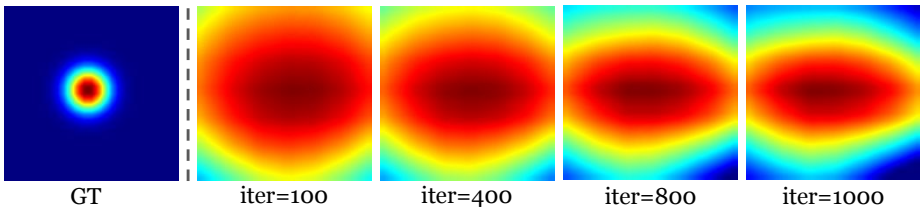

Fig. 8: Visualization of the learning process for the tracking model z_i.

and Table 5. Through the results, we can arrive that, i) the confident motion supervision is critical for stable dragging and a proper threshold is important, ii) merging the proposed tracker with the original feature difference can obtain optimal dragging performance.

4.5 Visualization of learning process for z_i

To give a more comprehensive understanding for the discriminative tracking model, in Fig. 8, we visualize the prediction results of the tracking model during the learning process. It can be seen that, with the training iterations increasing, background points (i.e., points away from the center) are gradually suppressed, resulting in a more robust and discriminative point tracking model, which can help the dragging model to generate more accurate editing results.

5 Conclusion

We have built a stable drag-based editing framework, coined as StableDrag, by designing a discirminative point tracking method and a confidence-based latent enhancement strategy for motion supervision. With the proposed point tracking method, we can precisely locate the updated handle points, thereby boosting the stability of long-range manipulation. While the latter can guarantee the optimized latent as high-quality as possible across all the manipulation steps. Thanks to the unique designs, we have instantiated two types of models including StableDrag-GAN and StableDrag-Diff to demonstrate the generality. Through extensive qualitative and quantitative experiments on a variety of examples, StableDrag has attained stable and precise drag performance. We expect our findings and analysis can facilitate the development of precise image editing.

Acknowledgements

This work is supported by the National Key R&D Program of China (No. 2022ZD0160900), the National Natural Science Foundation of China (No. 62076119, No. 61921006), the Fundamental Research Funds for the Central Universities (No. 020214380119), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4432–4441 (2019) 4
2. Abdal, R., Zhu, P., Mitra, N.J., Wonka, P.: Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Transactions on Graphics (ToG) **40**(3), 1–21 (2021) 4
3. Bar-Tal, O., Ofri-Amar, D., Fridman, R., Kasten, Y., Dekel, T.: Text2live: Text-driven layered image and video editing. In: European Conference on Computer Vision. pp. 707–723. Springer (2022) 4
4. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: ECCV Workshops (2016) 3, 5
5. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2010) 5
6. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023) 4
7. Cao, M., Wang, X., Qi, Z., Shan, Y., Qie, X., Zheng, Y.: Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. arXiv preprint arXiv:2304.08465 (2023) 2, 4, 6
8. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2021) 5
9. Creswell, A., Bharath, A.A.: Inverting the generator of a generative adversarial network. IEEE transactions on neural networks and learning systems **30**(7), 1967–1974 (2018) 4
10. Cui, Y., Jiang, C., Wang, L., Wu, G.: Mixformer: End-to-end tracking with iterative mixed attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022) 3, 5
11. Cui, Y., Song, T., Wu, G., Wang, L.: Mixformerv2: Efficient fully transformer tracking. In: Advances in Neural Information Processing Systems (2023) 5
12. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ATOM: accurate tracking by overlap maximization. In: CVPR (2019) 3, 5
13. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. In: NeurIPS (2021) 2, 4
14. Endo, Y.: User-controllable latent transformer for stylegan image layout editing. arXiv preprint arXiv:2208.12408 (2022) 4
15. Epstein, D., Jabri, A., Poole, B., Efros, A.A., Holynski, A.: Diffusion self-guidance for controllable image generation. arXiv preprint arXiv:2306.00986 (2023) 4

16. Esser, P., Rombach, R., Blattmann, A., Ommer, B.: ImageBART: Bidirectional context with multinomial diffusion for autoregressive image synthesis. In: NeurIPS (2021) 2, 4
17. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. In: ICLR (2023) 2, 4
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc. (2014), https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf 4
19. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) 1
20. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: CVPR (2022) 2, 4
21. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015) 5
22. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022) 4
23. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020) 2, 4
24. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: NeurIPS (2021) 5
25. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019) 1
26. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019) 4
27. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: CVPR (2020) 1, 5, 11
28. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. arXiv preprint arXiv:2210.09276 (2022) 2, 4
29. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6007–6017 (2023) 4
30. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) 1
31. Kumari, N., Zhang, B., Zhang, R., Shechtman, E., Zhu, J.Y.: Multi-concept customization of text-to-image diffusion. arXiv preprint arXiv:2212.04488 (2022) 2, 4
32. Leimkühler, T., Drettakis, G.: Freestylegan: Free-view editable portrait rendering with the camera manifold. arXiv preprint arXiv:2109.09378 (2021) 4
33. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2019) 5

34. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2018) 5
35. Liew, J.H., Yan, H., Zhou, D., Feng, J.: Magicmix: Semantic mixing with diffusion models. arXiv preprint arXiv:2210.16056 (2022) 4
36. Ling, P., Chen, L., Zhang, P., Chen, H., Jin, Y.: Freedrag: Point tracking is not you need for interactive point-based image editing. arXiv preprint arXiv:2307.04684 (2023) 1, 4, 10, 11
37. Lipton, Z.C., Tripathi, S.: Precise recovery of latent vectors from generative adversarial networks. arXiv preprint arXiv:1702.04782 (2017) 4
38. Mao, J., Wang, X., Aizawa, K.: Guided image synthesis via initial image editing in diffusion model. arXiv preprint arXiv:2305.03382 (2023) 4
39. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021) 4
40. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014) 1
41. Mou, C., Wang, X., Song, J., Shan, Y., Zhang, J.: Dragondiffusion: Enabling drag-style manipulation on diffusion models. arXiv preprint arXiv:2307.02421 (2023) 2, 4
42. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021) 2, 4
43. Pan, X., Tewari, A., Leimkühler, T., Liu, L., Meka, A., Theobalt, C.: Drag your GAN: Interactive point-based manipulation on the generative image manifold. arXiv preprint arXiv:2305.10973 (2023) 1, 4, 11
44. Parmar, G., Singh, K.K., Zhang, R., Li, Y., Lu, J., Zhu, J.Y.: Zero-shot image-to-image translation. arXiv preprint arXiv:2302.03027 (2023) 4
45. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2085–2094 (2021) 4
46. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125 (2022) 2, 4
47. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022) 2, 6
48. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 2, 4
49. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., et al.: Photorealistic text-to-image diffusion models with deep language understanding. arXiv preprint arXiv:2205.11487 (2022) 2, 4
50. Shi, Y., Xue, C., Pan, J., Zhang, W., Tan, V.Y., Bai, S.: Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. arXiv preprint arXiv:2306.14435 (2023) 2, 3, 4, 6, 11
51. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1921–1930 (2023) 4

52. Wang, S.Y., Bau, D., Zhu, J.Y.: Rewriting geometric rules of a gan. ACM Transactions on Graphics (TOG) **41**(4), 1–16 (2022) 4
53. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14. pp. 597–613. Springer (2016) 4