DragDiffusion: Harnessing Diffusion Models for Interactive Point-based Image Editing

Yujun Shi[1]* Chuhui Xue[2] Jun Hao Liew[2] Jiachun Pan[1]
Hanshu Yan[2] Wenqing Zhang[2] Vincent Y. F. Tan[1] Song Bai[2]
[1]National University of Singapore [2] ByteDance Inc.
shi.yujun@u.nus.edu vtan@nus.edu.sg songbai.site@gmail.com

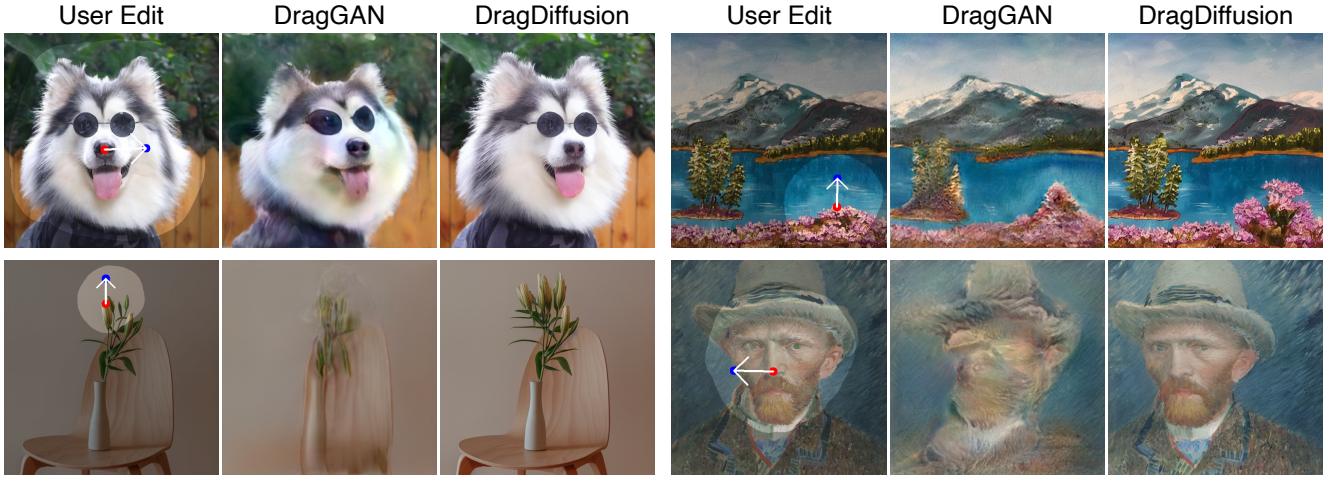

Figure 1. DragDiffusion **greatly improves the applicability of interactive point-based editing.** Given an input image, the user clicks handle points (red), target points (blue), and draws a mask specifying the editable region (brighter area). All results are obtained under the same user edit for fair comparisons. Project page: https://yujun-shi.github.io/projects/dragdiffusion.html

Abstract

Accurate and controllable image editing is a challenging task that has attracted significant attention recently. Notably, DragGAN *developed by Pan et al. (2023) [33] is an interactive point-based image editing framework that achieves impressive editing results with pixel-level precision. However, due to its reliance on generative adversarial networks (GANs), its generality is limited by the capacity of pretrained GAN models. In this work, we extend this editing framework to diffusion models and propose a novel approach* DragDiffusion. *By harnessing large-scale pretrained diffusion models, we greatly enhance the applicability of interactive point-based editing on both real and diffusion-generated images. Unlike other diffusion-based editing methods that provide guidance on diffusion latents of multiple time steps, our approach achieves efficient yet accurate spatial control by optimizing the latent of only one time step. This novel design is motivated by our observations that UNet features at a specific time step provides sufficient semantic and geometric information to support the drag-based editing. Moreover, we introduce two additional techniques, namely identity-preserving fine-tuning and reference-latent-control, to further preserve the identity of the original image. Lastly, we present a challenging benchmark dataset called* DragBench*—the first benchmark to evaluate the performance of interactive point-based image editing methods. Experiments across a wide range of challenging cases (e.g., images with multiple objects, diverse object categories, various styles, etc.) demonstrate the versatility and generality of* DragDiffusion. *Code and the* DragBench *dataset: https://github.com/Yujun-Shi/DragDiffusion.*

1. Introduction

Image editing with generative models [9, 15, 22, 31, 34, 37] has attracted extensive attention recently. One landmark work is DragGAN [33], which enables interactive point-

*Work done when interning with Song Bai.

based image editing, *i.e.*, drag-based editing. Under this framework, the user first clicks several pairs of handle and target points on an image. Then, the model performs semantically coherent editing on the image that moves the contents of the handle points to the corresponding target points. In addition, users can draw a mask to specify which region of the image is editable while the rest remains unchanged.

Despite DRAGGAN's impressive editing results with pixel-level spatial control, the applicability of this method is being limited by the inherent model capacity of generative adversarial networks (GANs) [12, 20, 21]. On the other hand, although large-scale text-to-image diffusion models [38, 42] have demonstrated strong capabilities to synthesize high quality images across various domains, there are not many diffusion-based editing methods that can achieve precise spatial control. This is because most diffusion-based methods [15, 22, 31, 34] conduct editing by controlling the text embeddings, which restricts their applicability to editing high-level semantic contents or styles.

To bridge this gap, we propose DRAGDIFFUSION, the first interactive point-based image editing method with diffusion models [17, 38, 42, 46]. Empowered by large-scale pre-trained diffusion models [38, 42], DRAGDIFFUSION achieves accurate spatial control in image editing with significantly better generalizability (see Fig. 1).

Our approach focuses on optimizing diffusion latents to achieve drag-based editing, which is inspired by the fact that diffusion latents can accurately determine the spatial layout of the generated images [29]. In contrast to previous methods [3, 10, 34, 53], which apply gradient descent on latents of *multiple* diffusion steps, our approach focuses on optimizing the latent of *one appropriately selected step* to conveniently achieve the desired editing results. This novel design is motivated by the empirical observations presented in Fig. 2. Specifically, given two frames from a video simulating the original and the "dragged" images, we visualize the UNet feature maps of different diffusion steps using principal component analysis (PCA). Via this visualization, we find that there exists a single diffusion step (*e.g.*, $t = 35$ in this case) such that the UNet feature maps at this step alone contains sufficient semantic and geometric information to support structure-oriented spatial control such as drag-based editing. Besides optimizing the diffusion latents, we further introduce two additional techniques to enhance the identity preserving during the editing process, namely identity-preserving fine-tuning and reference-latent-control. An overview of our method is given in Fig. 3.

It would be ideal to immediately evaluate our method on well-established benchmark datasets. However, due to a lack of evaluation benchmarks for interactive point-based editing, it is difficult to rigorously study and corroborate the efficacy of our proposed approach. Therefore, to facilitate such evaluation, we present DRAGBENCH—the first benchmark dataset for drag-based editing. DRAGBENCH is a diverse collection comprising images spanning various object categories, indoor and outdoor scenes, realistic and aesthetic styles, *etc*. Each image in our dataset is accompanied with a set of "drag" instructions, which consists of one or more pairs of handle and target points as well as a mask specifying the editable region.

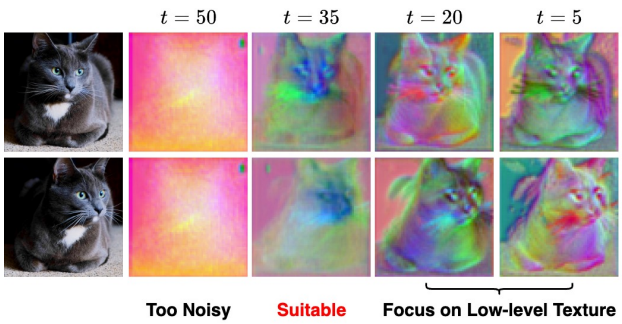

Figure 2. **PCA visualization of UNet feature maps at different diffusion steps for two video frames.** $t = 50$ implies the full DDIM inversion, while $t = 0$ implies the clean image. Notably, UNet features at one specific step (*e.g.*, $t = 35$) provides sufficient semantic and geometric information (*e.g.*, shape and pose of the cat, *etc*.) for the drag-based editing.

Through extensive qualitative and quantitative experiments on a variety of examples (including those on DRAGBENCH), we demonstrate the versatility and generality of our approach. In addition, our empirical findings corroborate the crucial role played by identity-preserving fine-tuning and reference-latent-control. Furthermore, a comprehensive ablation study is conducted to meticulously explore the influence of key factors, including the number of inversion steps of the latent, the number of identity-preserving fine-tuning steps, and the UNet feature maps.

Our contributions are summarized as follows: 1) we present a novel image editing method DRAGDIFFUSION, the first to achieve interactive point-based editing with diffusion models; 2) we introduce DRAGBENCH, the first benchmark dataset to evaluate interactive point-based image editing methods; 3) Comprehensive qualitative and quantitative evaluation demonstrate the versatility and generality of our DRAGDIFFUSION.

2. Related Work

Generative Image Editing. Given the initial successes of generative adversarial networks (GANs) in image generation [12, 20, 21], many previous image editing methods have been based on the GAN paradigm [2, 9, 14, 25, 33, 35, 44, 45, 49, 55, 56]. However, due to the limited model capacity of GANs and the difficulty of inverting the real images into GAN latents [1, 8, 28, 37], the generality of these methods would inevitably be constrained.

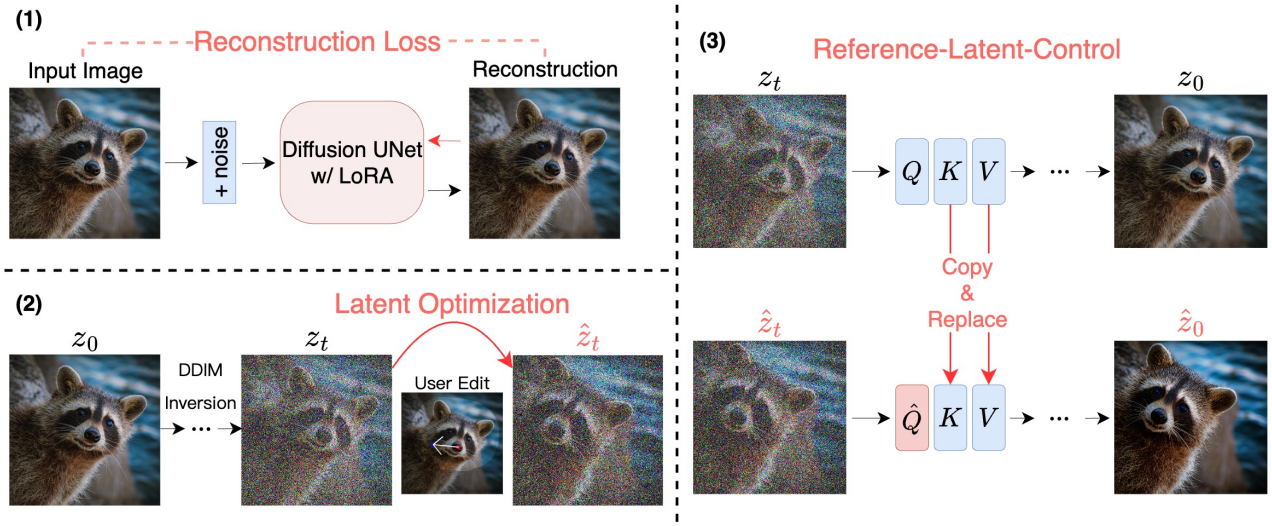

Figure 3. **Overview of** DRAGDIFFUSION**.** Our approach constitutes three steps: firstly, we conduct identity-preserving fine-tuning on the UNet of the diffusion model given the input image. Secondly, according to the user's dragging instruction, we optimize the latent obtained from DDIM inversion on the input image. Thirdly, we apply DDIM denoising guided by our reference-latent-control on $\hat{z}_t$ to obtain the final editing result $\hat{z}_0$. Figure best viewed in color.

Recently, due to the impressive generation results from large-scale text-to-image diffusion models [38, 42], many diffusion-based image editing methods have been proposed [4, 6, 7, 15, 22, 26, 29, 30, 32, 34, 50]. Most of these methods aim to edit the images by manipulating the prompts of the image. However, as many editing attempts are difficult to convey through text, the prompt-based paradigm usually alters the image's high-level semantics or styles, lacking the capability of achieving precise pixel-level spatial control. [10] is one of the early efforts in exploring better controllability on diffusion models beyond the prompt-based image editing. In our work, we aim at enabling a even more versatile paradigm than the one studied in [10] with diffusion models—interactive point-based image editing.

Point-based editing. The framework of point-based editing [5, 19, 43] aims at manipulating images in a fine-grained level. Recently, to enable such editing, several GAN-based methods have been proposed [9, 33, 52]. Specifically, DRAGGAN achieves impressive dragging-based manipulation with two simple ingredients: 1) optimizing latent codes to move the handle points towards their target locations and 2) a point tracking mechanism that keep tracks of the handle points. However, its generality is constrained due to the limited capacity of GAN. FreeDrag [27] improves DRAGGAN by introducing a point-tracking-free paradigm. In this work, we extend the editing framework of DRAGGAN to diffusion models and showcase its generality over different domains. There is a work [32] concurrent to ours that also studies drag-based editing with diffusion models. Differently, they rely on classifier guidance to transforms the editing signal into gradients.

LoRA in Diffusion Models. Low Rank Adaptation (*i.e.*, LoRA) [18] is a general technique to conduct parameter-efficient fine-tuning on large and deep networks. During LoRA fine-tuning, the original weights of the model are frozen, while trainable rank decomposition matrices are injected into each layer. The core assumption of this strategy is that the model weights will primarily be adapted within a low rank subspace during fine-tuning. While LoRA was initially introduced for adapting language models to downstream tasks, recent efforts have illustrated its effectiveness when applied in conjunction with diffusion models [13, 41]. In this work, inspired by the promising results of using LoRA for image generation and editing [22, 40], we also implement our identity-preserving fine-tuning with LoRA.

3. Methodology

In this section, we formally present the proposed DRAGDIFFUSION approach. To commence, we introduce the preliminaries on diffusion models. Then, we elaborate on the three key stages of our approach as depicted in Fig. 3: 1) identity-preserving fine-tuning; 2) latent optimization according to the user-provided dragging instructions; 3) denoising the optimized latents guided by our reference-latent-control.

3.1. Preliminaries on Diffusion Models

Denoising diffusion probabilistic models (DDPM) [17, 46] constitute a family of latent generative models. Concerning a data distribution $q(z)$, DDPM approximates $q(z)$ as the marginal $p_\theta(z_0)$ of the joint distribution between Z_0 and a collection of latent random variables $Z_{1:T}$. Specifically,

$$p_\theta(z_0) = \int p_\theta(z_{0:T})\, \mathrm{d}z_{1:T}, \tag{1}$$

where $p_\theta(z_T)$ is a standard normal distribution and the transition kernels $p_\theta(z_{t-1}|z_t)$ of this Markov chain are all Gaussian conditioned on z_t. In our context, Z_0 corresponds to image samples given by users, and Z_t corresponds to the latent after t steps of the diffusion process.

[38] proposed the latent diffusion model (LDM), which maps data into a lower-dimensional space via a variational auto-encoder (VAE) [24] and models the distribution of the latent embeddings instead. Based on the framework of LDM, several powerful pretrained diffusion models have been released publicly, including the Stable Diffusion (SD) model (https://huggingface.co/stabilityai). In SD, the network responsible for modeling $p_\theta(z_{t-1}|z_t)$ is implemented as a UNet [39] that comprises multiple self-attention and cross-attention modules [51]. Applications in this paper are implemented based on the public Stable Diffusion model.

3.2. Identity-preserving Fine-tuning

Before editing a real image, we first conduct identity-preserving fine-tuning [18] on the diffusion models' UNet (see panel (1) of Fig. 3). This stage aims to ensure that the diffusion UNet encodes the features of input image more accurately (than in the absence of this procedure), thus facilitating the consistency of the identity of the image throughout the editing process. This fine-tuning process is implemented with LoRA [18]. More formally, the objective function of the LoRA is

$$\mathcal{L}_{\text{ft}}(z, \Delta\theta) = \mathbb{E}_{\epsilon,t}\left[\|\epsilon - \epsilon_{\theta+\Delta\theta}(\alpha_t z + \sigma_t \epsilon)\|_2^2\right], \quad (2)$$

where θ and $\Delta\theta$ represent the UNet and LoRA parameters respectively, z is the real image, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the randomly sampled noise map, $\epsilon_{\theta+\Delta\theta}(\cdot)$ is the noise map predicted by the LoRA-integrated UNet, and α_t and σ_t are parameters of the diffusion noise schedule at diffusion step t. The fine-tuning objective is optimized via gradient descent on $\Delta\theta$.

Remarkably, we find that fine-tuning LoRA for merely 80 steps proves sufficient for our approach, which is in stark contrast to the 1000 steps required by tasks such as subject-driven image generation [13, 40]. This ensures that our identity-preserving fine-tuning process is extremely efficient, and only takes around 25 seconds to complete on an A100 GPU. We posit this efficiency is because our approach operates on the inverted noisy latent, which inherently preserve some information about the input real image. Consequently, our approach does not require lengthy fine-tuning to preserve the identity of the original image.

3.3. Diffusion Latent Optimization

After identity-preserving fine-tuning, we optimize the diffusion latent according to the user instruction (*i.e.*, the handle and target points, and optionally a mask specifying the editable region) to achieve the desired interactive point-based editing (see panel (2) of Fig. 3).

To commence, we first apply a DDIM inversion [47] on the given real image to obtain a diffusion latent at a certain step t (*i.e.*, z_t). This diffusion latent serves as the initial value for our latent optimization process. Then, following along the similar spirit of [33], the latent optimization process consists of two steps to be implemented consecutively. These two steps, namely motion supervision and point tracking, are executed repeatedly until either all handle points have moved to the targets or the maximum number of iterations has been reached. Next, we describe these two steps in detail.

Motion Supervision: We denote the n handle points at the k-th motion supervision iteration as $\{h_i^k = (x_i^k, y_i^k) : i = 1, \ldots, n\}$ and their corresponding target points as $\{g_i = (\tilde{x}_i, \tilde{y}_i) : i = 1, \ldots, n\}$. The input image is denoted as z_0; the t-th step latent (*i.e.*, result of t-th step DDIM inversion) is denoted as z_t. We denote the UNet output feature maps used for motion supervision as $F(z_t)$, and the feature vector at pixel location h_i^k as $F_{h_i^k}(z_t)$. Also, we denote the square patch centered around h_i^k as $\Omega(h_i^k, r_1) = \{(x, y) : |x - x_i^k| \le r_1, |y - y_i^k| \le r_1\}$. Then, the motion supervision loss at the k-th iteration is defined as:

$$\mathcal{L}_{\text{ms}}(\hat{z}_t^k) = \sum_{i=1}^{n} \sum_{q \in \Omega(h_i^k, r_1)} \left\| F_{q+d_i}(\hat{z}_t^k) - \text{sg}(F_q(\hat{z}_t^k)) \right\|_1 + \lambda \left\| (\hat{z}_{t-1}^k - \text{sg}(\hat{z}_{t-1}^0)) \odot (\mathbb{1} - M) \right\|_1, \quad (3)$$

where $\hat{z}_t^k$ is the t-th step latent after the k-th update, $\text{sg}(\cdot)$ is the stop gradient operator (*i.e.*, the gradient will not be back-propagated for the term $\text{sg}(F_q(\hat{z}_t^k))$), $d_i = (g_i - h_i^k)/\|g_i - h_i^k\|_2$ is the normalized vector pointing from h_i^k to g_i, M is the binary mask specified by the user, $F_{q+d_i}(\hat{z}_t^k)$ is obtained via bilinear interpolation as the elements of $q + d_i$ may not be integers. In each iteration, $\hat{z}_t^k$ is updated by taking one gradient descent step to minimize $\mathcal{L}_{\text{ms}}$:

$$\hat{z}_t^{k+1} = \hat{z}_t^k - \eta \cdot \frac{\partial \mathcal{L}_{\text{ms}}(\hat{z}_t^k)}{\partial \hat{z}_t^k}, \quad (4)$$

where η is the learning rate for latent optimization.

Note that for the second term in Eqn. (3), which encourages the unmasked area to remain unchanged, we are working with the diffusion latent instead of the UNet features. Specifically, given $\hat{z}_t^k$, we first apply one step of DDIM denoising to obtain $\hat{z}_{t-1}^k$, then we regularize the unmasked region of $\hat{z}_{t-1}^k$ to be the same as $\hat{z}_{t-1}^0$ (*i.e.*, z_{t-1}).

Point Tracking: Since the motion supervision updates $\hat{z}_t^k$, the positions of the handle points may also change. Therefore, we need to perform point tracking to update the handle points after each motion supervision step. To achieve this goal, we use UNet feature maps $F(\hat{z}_t^{k+1})$ and

$F(z_t)$ to track the new handle points. Specifically, we update each of the handle points h_i^k with a nearest neighbor search within the square patch $\Omega(h_i^k, r_2) = \{(x, y) : |x - x_i^k| \leq r_2, |y - y_i^k| \leq r_2\}$ as follows:

$$h_i^{k+1} = \underset{q \in \Omega(h_i^k, r_2)}{\arg\min} \left\| F_q(\hat{z}_t^{k+1}) - F_{h_i^0}(z_t) \right\|_1. \quad (5)$$

3.4. Reference-latent-control

After we have completed the optimization of the diffusion latents, we then denoise the optimized latents to obtain the final editing results. However, we find that naïvely applying DDIM denoising on the optimized latents still occasionally leads to undesired identity shift or degradation in quality comparing to the original image. We posit that this issue arises due to the absence of adequate guidance from the original image during the denoising process.

To mitigate this problem, we draw inspiration from [7] and propose to leverage the property of self-attention modules to steer the denoising process, thereby boosting coherence between the original image and the editing results. In particular, as illustrated in panel (3) of Fig. 3, given the denoising process of both the original latent z_t and the optimized latent $\hat{z}_t$, we use the process of z_t to guide the process of $\hat{z}_t$. More specifically, during the forward propagation of the UNet's self-attention modules in the denoising process, we replace the key and value vectors generated from $\hat{z}_t$ with the ones generated from z_t. With this simple replacement technique, the query vectors generated from $\hat{z}_t$ will be directed to query the correlated contents and texture of z_t. This leads to the denoising results of $\hat{z}_t$ (*i.e.*, $\hat{z}_0$) being more coherent with the denoising results of z_t (*i.e.*, z_0). In this way, reference-latent-control substantially improves the consistency between the original and the edited images.

4. Experiments

4.1. Implementation Details

In all our experiments, unless stated otherwise, we adopt the Stable Diffusion 1.5 [38] as our diffusion model. During the identity-preserving fine-tuning, we inject LoRA into the projection matrices of query, key and value in all of the attention modules. We set the rank of the LoRA to 16. We fine-tune the LoRA using the AdamW [23] optimizer with a learning rate of 5×10^{-4} and a batch size of 4 for 80 steps.

During the latent optimization stage, we schedule 50 steps for DDIM and optimize the diffusion latent at the 35-th step unless specified otherwise. When editing real images, we *do not* apply classifier-free guidance (CFG) [16] in both DDIM inversion and DDIM denoising process. This is because CFG tends to amplify numerical errors, which is not ideal in performing the DDIM inversion [31]. We use the Adam optimizer with a learning rate of 0.01 to optimize the latent. The maximum optimization step is set to be 80. The hyperparameter r_1 in Eqn. 3 and r_2 in Eqn. 5 are tuned to be 1 and 3, respectively. λ in Eqn. 3 is set to 0.1 by default, but the user may increase λ if the unmasked region has changed to be more than what was desired.

Finally, we apply our reference-latent-control in the upsampling blocks of the diffusion UNet at all denoising steps when generating the editing results. The execution time for each component is detailed in Appendix D.

4.2. DragBench and Evaluation Metrics

Since interactive point-based image editing is a recently introduced paradigm, there is an absence of dedicated evaluation benchmarks for this task, making it challenging to comprehensively study the effectiveness of our proposed approach. To address the need for systematic evaluation, we introduce DragBench, the first benchmark dataset tailored for drag-based editing. DragBench is a diverse compilation encompassing various types of images. Details and examples of our dataset are given in Appendix A. Each image within our dataset is accompanied by a set of dragging instructions, comprising one or more pairs of handle and target points, along with a mask indicating the editable region. We hope future research on this task can benefit from DragBench.

In this work, we utilize the following two metrics for quantitative evaluation: *Image Fidelity* (IF) [22] and *Mean Distance* (MD) [33]. IF, the first metric, quantifies the similarity between the original and edited images. It is calculated by subtracting the mean LPIPS [54] over all pairs of original and edited images from 1. The second metric MD assesses how well the approach moves the semantic contents to the target points. To compute the MD, we first employ DIFT [48] to identify points in the edited images corresponding to the handle points in the original image. These identified points are considered to be the final handle points post-editing. MD is subsequently computed as the mean Euclidean distance between positions of all target points and their corresponding final handle points. MD is averaged over all pairs of handle and target points in the dataset. An optimal "drag" approach ideally achieves both a low MD (indicating effective "dragging") and a high IF (reflecting robust identity preservation).

4.3. Qualitative Evaluation

In this section, we first compare our approach with DragGAN on real images. We employ SD-1.5 for our approach when editing real images. All input images and the user edit instructions are from our DragBench dataset. Results are given in Fig. 4. As illustrated in the figure, when dealing with the real images from a variety of domains, DragGAN often struggles due to GAN models' limited capacity. On the other hand, our DragDiffusion can convincingly de-

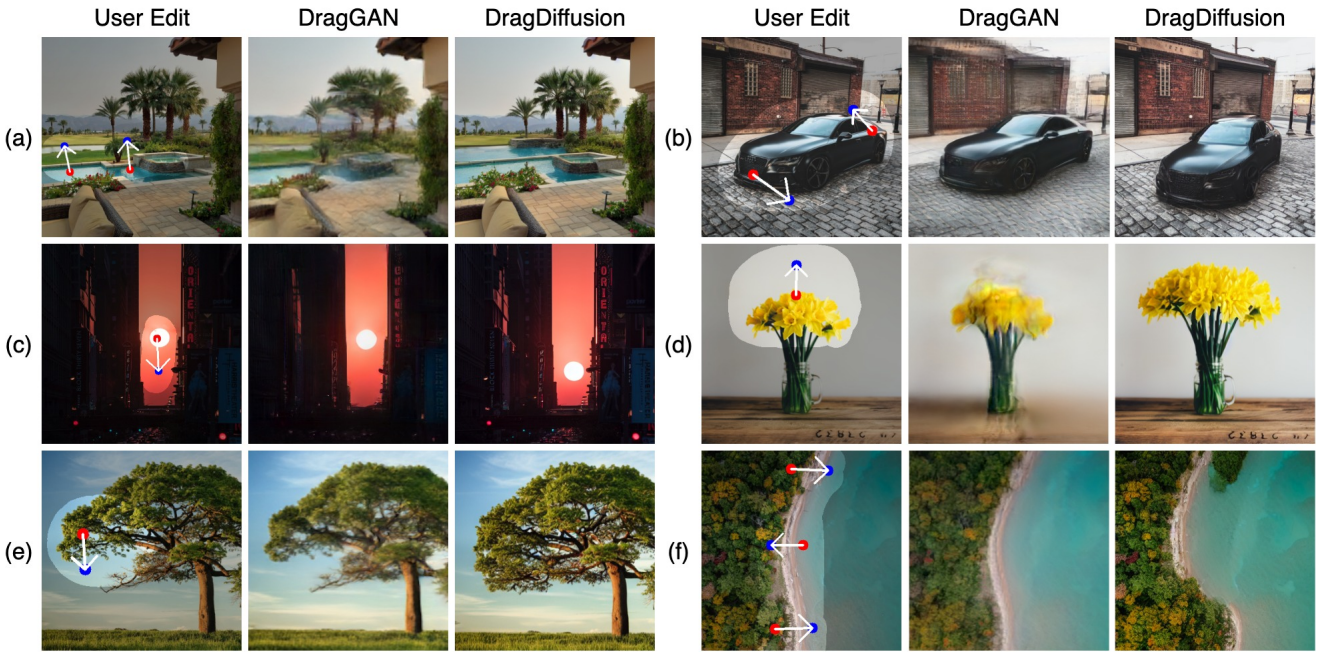

Figure 4. Comparisons between DRAGGAN and DRAGDIFFUSION. All results are obtained under the same user edit for fair comparisons.

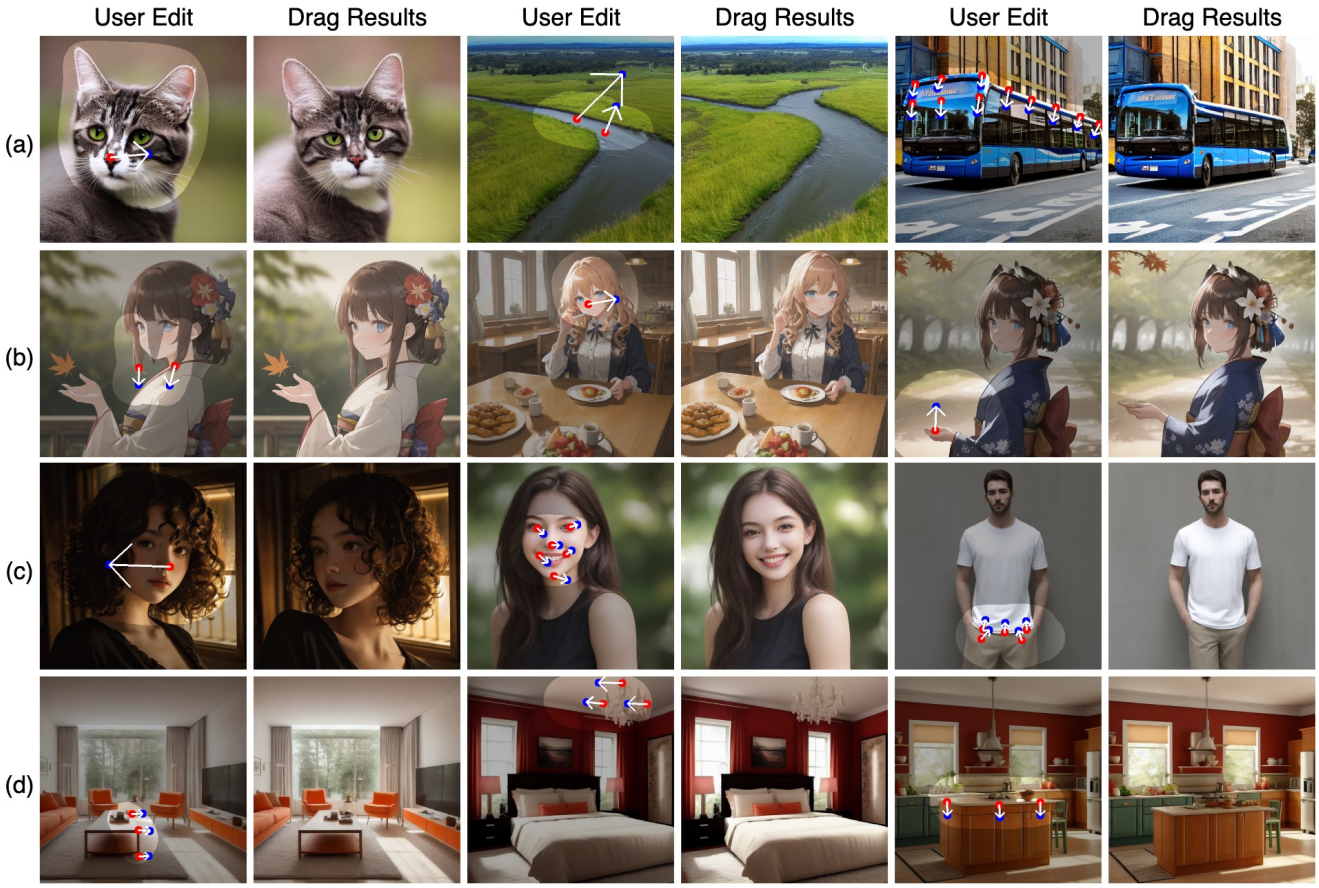

Figure 5. Editing results on diffusion-generated images with **(a)** Stable-Diffusion-1.5, **(b)** Counterfeit-V2.5, **(c)** Majicmix Realistic, **(d)** Interior Design Supermix.

Figure 6. Ablating the number of inversion step t. Effective results are obtained when $t \in [30, 40]$.

liver reasonable editing results. More importantly, besides achieving the similar pose manipulation and local deformation as in DRAGGAN [33], our approach even enables more types of editing such as content filling. An example is given in Fig. 4 (a), where we fill the grassland with the pool using drag-based editing. This further validates the enhanced versatility of our approach. More qualitative comparisons are provided in Appendix F.

Next, to show the generality of our approach, we perform drag-based editing on diffusion-generated images across a spectrum of variants of SD-1.5, including SD-1.5 itself, Counterfeit-V2.5, Majicmix Realistic, Interior Design Supermix. Results are shown in Fig. 5 These results validate our approach's ability to smoothly work with various pre-trained diffusion models. Moreover, these results also illustrate our approach's ability to deal with drag instructions of different magnitudes (*e.g.*, small magnitude edits such as the left-most image in Fig. 5 (d) and large magnitude edits such as the left-most image in Fig. 5 (c)). Additional results with more diffusion models and different resolutions can be found in Appendix F.

4.4. Quantitative Analysis

In this section, we conduct a rigorous quantitative evaluation to assess the performance of our approach. We begin by comparing DRAGDIFFUSION with the baseline method DRAGGAN. As each StyleGAN [21] model utilized in [33] is specifically designed for a particular image class, we employ an ensemble strategy to evaluate DRAGGAN. This strategy involves assigning a text description to characterize the images generated by each StyleGAN model. Before editing each image, we compute the CLIP similarity [36] between the image and each of the text descriptions associated with the GAN models. The GAN model that yields the highest CLIP similarity is selected for the editing task.

Furthermore, to validate the effectiveness of each component of our approach, we evaluate DRAGDIFFUSION in the following two configurations: one without identity-preserving fine-tuning and the other without reference-latent-control. We perform our empirical studies on the DRAGBENCH dataset, and Image Fidelity (IF) and Mean Distance (MD) of each configuration mentioned above are reported in Fig. 8. All results are averaged over the DRAGBENCH dataset. In this figure, the x-axis represents MD and the y-axis represents IF, which indicates the method with better results should locate at the upper-left corner of the coordinate plane. The results presented in this figure clearly demonstrate that our DRAGDIFFUSION significantly outperforms the DRAGGAN baseline in terms of both IF and MD. Furthermore, we observe that DRAGDIFFUSION without identity-preserving fine-tuning experiences a catastrophic increase in MD, whereas DRAGDIFFUSION without reference-latent control primarily encounters a decrease in IF. Visualization on the effects of identity-preserving fine-tuning and reference-latent-control are given in Fig. 9, which corroborates with our quantitative results.

4.5. Ablation on the Number of Inversion Step

Next, we conduct an ablation study to elucidate the impact of varying t (*i.e.*, the number of inversion steps) during the latent optimization stage of DRAGDIFFUSION. We set t to be $t = 10, 20, 30, 40, 50$ steps and run our approach on DRAGBENCH to obtain the editing results ($t = 50$ corresponds to the pure noisy latent). We evaluate Image Fidelity (IF) and Mean Distance (MD) for each t value in Fig. 7(a). All metrics are averaged over the DRAGBENCH dataset.

In terms of the IF, we observe a monotonic decrease as t increases. This trend can be attributed to the stronger flexibility of the diffusion latent as more steps are inverted. As for MD, it initially decreases and then increases with higher t values. This behavior highlights the presence of a critical range of t values for effective editing ($t \in [30, 40]$ in our figure). When t is too small, the diffusion latent lacks the necessary flexibility for substantial changes, posing challenges in performing reasonable edits. Conversely, overly large t values result in a diffusion latent that is unstable for editing, leading to difficulties in preserving the original image's identity. Given these results, we chose $t = 35$ as our default setting, as it achieves the lowest MD while maintaining a decent IF. Qualitative visualization that corroborates with our numerical evaluation is provided in Fig. 6.

4.6. Ablation Study on the Number of Identity-preserving Fine-tuning Steps

We run our approach on DRAGBENCH under 0, 20, 40, 60, 80, and 100 identity-preserving fine-tuning steps, respectively (0 being no fine-tuning). The outcomes are assessed using IF and MD, and the results are presented in Fig.7 (b).

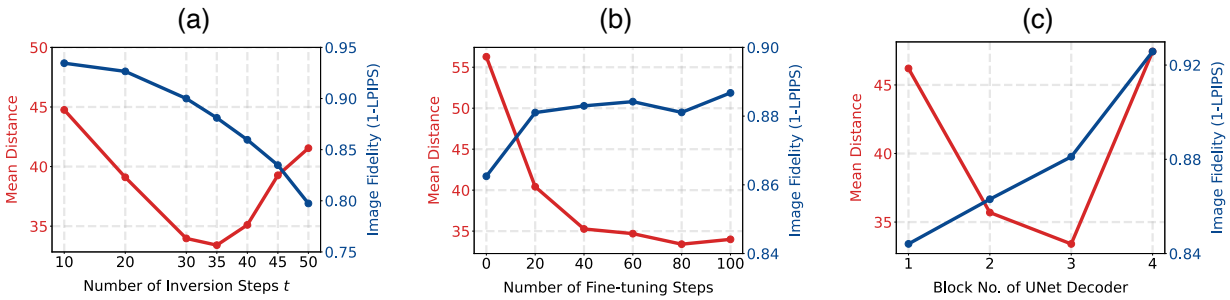

Figure 7. Ablation study on **(a)** the number of inversion step t of the diffusion latent; **(b)** the number of identity-preserving fine-tuning steps; **(c)** Block No. of UNet feature maps. Mean Distance (↓) and Image Fidelity (↑) are reported. Results are produced on DRAGBENCH.

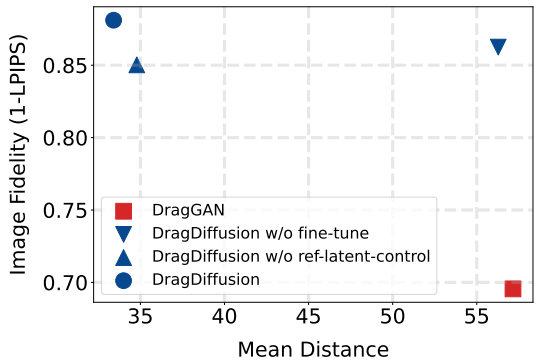

Figure 8. Quantitative analysis on DRAGGAN, DRAGDIFFUSION and DRAGDIFFUSION's variants without certain components. Image Fidelity (↑) and Mean Distance (↓) are reported. Results are produced on DRAGBENCH. The approach with better results should locate at the upper-left corner of the coordinate plane.

Figure 9. Qualitative validation on effectiveness of identity-preserving fine-tuning and reference-latent-control.

Initially, as the number of fine-tuning steps increases, MD exhibits a steep downward trend while IF shows an upward trend. This reflects that identity-preserving fine-tuning can drastically boost both the precision and consistency of drag-based editing. However, as the fine-tuning progresses, both MD and IF subsequently begins to plateau. This phenomenon shows that lengthy fine-tuning of LoRA would no longer significantly improve the performance of our approach. Considering the experimental results, we conduct identity-preserving fine-tuning for 80 steps by default to balance between effectiveness and efficiency. Visualizations that corroborate our quantitative evaluation are presented in the Appendix G.

4.7. Ablation Study on the UNet Feature Maps

Finally, we study the effect of using different blocks of UNet feature maps to supervise our latent optimization. We run our approach on the DRAGBENCH dataset with the feature maps output by 4 different upsampling blocks of the UNet *Decoder*, respectively. The outcomes are assessed with IF and MD, and are shown in Fig. 7(c). As can be seen, with deeper blocks of UNet features, IF consistently increases, while MD first decreases and then increases. This trend is because feature maps of lower blocks contain coarser semantic information, while higher blocks contain lower level texture information [11, 50]. Hence, the feature maps of lower blocks (*e.g.*, block No. of 1) lack fine-grained information for accurate spatial control, whereas those of higher blocks (*e.g.*, block No. of 4) lack semantic and geometric information to drive the drag-based editing. Our results indicate that the feature maps produced by the third block of the UNet decoder demonstrate the best performance, exhibiting the lowest MD and a relatively high IF. Visualizations that corroborate our quantitative evaluation are presented in the Appendix H.

5. Conclusion and Future Works

In this work, we introduce DRAGDIFFUSION, a novel method extending interactive point-based editing to large-scale diffusion models. Additionally, we present the DRAGBENCH dataset for evaluation. Our method demonstrates versatility and generality in both qualitative and quantitative analyses. Limitations are discussed in Appendix E, with future work aimed at enhancing the robustness and reliability of drag-based editing on diffusion models.

Acknowledgement

This work is supported by funding from a Ministry of Education Academic Research Fund (AcRF) Tier 2 under grant number A-8000423-00-00 as well as AcRF Tier 1 under grant numbers A-8000980-00-00 and A-8000189-01-00.

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

[2] 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

[3] 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. 2

[4] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European Conference on Computer Vision*, pages 707–723. Springer, 2022. 3

[5] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 529–536. 2023. 3

[6] 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

[7] 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. 3, 5

[8] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018. 2

[9] Yuki Endo. User-controllable latent transformer for stylegan image layout editing. *arXiv preprint arXiv:2208.12408*, 2022. 1, 2, 3

[10] 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

[11] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arxiv:2307.10373*, 2023. 8

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 2

[13] Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *arXiv preprint arXiv:2305.18292*, 2023. 3, 4

[14] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems*, 33:9841–9850, 2020. 2

[15] 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, 3

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 5

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

[18] 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. In *International Conference on Learning Representations (ICLR)*, 2022. 3, 4

[19] Takeo Igarashi, Tomer Moscovich, and John F Hughes. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)*, 24(3):1134–1141, 2005. 3

[20] 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

[21] 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. 2, 7

[22] 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, 2, 3, 5

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 5

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[25] Thomas Leimkühler and George Drettakis. Freestylegan: Free-view editable portrait rendering with the camera manifold. *arXiv preprint arXiv:2109.09378*, 2021. 2

[26] Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. Magicmix: Semantic mixing with diffusion models. *arXiv preprint arXiv:2210.16056*, 2022. 3

[27] Pengyang Ling, Lin Chen, Pan Zhang, Huaian Chen, and Yi Jin. Freedrag: Point tracking is not you need for interactive point-based image editing. *arXiv preprint arXiv:2307.04684*, 2023. 3

[28] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017. 2

[29] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided image synthesis via initial image editing in diffusion model. *arXiv preprint arXiv:2305.03382*, 2023. 2, 3

[30] 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

[31] 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, 5

[32] 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. 3

[33] 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. 1, 2, 3, 4, 5, 7

[34] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *arXiv preprint arXiv:2302.03027*, 2023. 1, 2, 3

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

[36] 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. PMLR, 2021. 7

[37] 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, 2

[38] 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. 2, 3, 4, 5

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 4

[40] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 3, 4

[41] Simo Ryu. Low-rank adaptation for fast text-to-image diffusion fine-tuning. https://github.com/cloneofsimo/lora, 2022. 3

[42] 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. 2, 3

[43] Scott Schaefer, Travis McPhail, and Joe Warren. Image deformation using moving least squares. In *ACM SIGGRAPH 2006 Papers*, pages 533–540. 2006. 3

[44] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1532–1540, 2021. 2

[45] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9243–9252, 2020. 2

[46] 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

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 4

[48] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023. 5

[49] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151, 2020. 2

[50] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. 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*, pages 1921–1930, 2023. 3, 8

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[52] 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. 3

[53] 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. 2

[54] 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. 5

[55] Jiapeng Zhu, Ceyuan Yang, Yujun Shen, Zifan Shi, Deli Zhao, and Qifeng Chen. Linkgan: Linking gan latents

to pixels for controllable image synthesis. *arXiv preprint arXiv:2301.04604*, 2023. 2

[56] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 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*, pages 597–613. Springer, 2016. 2