Source Prompt Disentangled Inversion for Boosting Image Editability with Diffusion Models

Ruibin Li[1], Ruihuang Li[1], Song Guo[2], and Lei Zhang[1]*

[1] The Hong Kong Polytechnic University
[2] The Hong Kong University of Science and Technology
{ruibin.li,21039075r,cslzhang}@connect.polyu.hk,songguo@cse.ust.hk

Abstract. Text-driven diffusion models have significantly advanced the image editing performance by using text prompts as inputs. One crucial step in text-driven image editing is to invert the original image into a latent noise code conditioned on the source prompt. While previous methods have achieved promising results by refactoring the image synthesizing process, the inverted latent noise code is tightly coupled with the source prompt, limiting the image editability by target text prompts. To address this issue, we propose a novel method called **S**ource **P**rompt **D**isentangled **Inv**ersion (SPDInv), which aims at reducing the impact of source prompt, thereby enhancing the text-driven image editing performance by employing diffusion models. To make the inverted noise code be independent of the given source prompt as much as possible, we indicate that the iterative inversion process should satisfy a fixed-point constraint. Consequently, we transform the inversion problem into a searching problem to find the fixed-point solution, and utilize the pre-trained diffusion models to facilitate the searching process. The experimental results show that our proposed SPDInv method can effectively mitigate the conflicts between the target editing prompt and the source prompt, leading to a significant decrease in editing artifacts. In addition to text-driven image editing, with SPDInv we can easily adapt customized image generation models to localized editing tasks and produce promising performance. The source code are available at https://github.com/leeruibin/SPDInv.

Keywords: Image Editing · Image Inversion · Diffusion Models

1 Introduction

The emergence of diffusion models [15, 47], especially the Latent Diffusion Models (LDMs) [42], has revolutionized the field of image generation. Leveraging the exceptional semantic understanding ability of pre-trained LDMs, researchers have successfully applied them to numerous downstream tasks, such as text-to-image [37, 38, 41, 60], style transfer [54, 57, 62], text-to-video [2, 6, 12], text-to-3D [27,28,39], as well as text-driven image editing [1,3,22,34]. It has been demonstrated [4, 14, 50] that by delicately controlling the attention layer in LDMs, we can achieve complex image editing by modifying only the text prompts.

* Corresponding author.

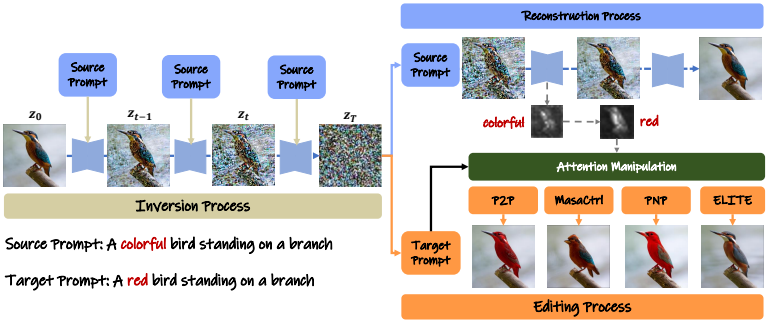

Fig. 1: Illustration of text-driven image editing pipeline.

Originally, the scope of text-driven image editing was limited to images generated by the LDM. However, it was soon found that we can first convert the real image into latent noise through slight modification of the DDIM [46] inference pipeline along with the source prompt so that high-quality reconstruction of the original image can be obtained. The inverted latent noise can then be used to perform sophisticated image editing by interacting with the target prompts, as illustrated in Fig. 1. The inversion process (please refer to Fig. 2(a)), which is crucial for achieving real-image editing, significantly impacts the editing results. Since the advanced LDMs are mostly driven by the Classifier-Free Guidance (CFG) technology [16], reconstruction failures often occur when the CFG parameter needs to be set higher. To address this issue, Mokady *et al.* [32] proposed the Null-Text Inversion (NTI) to optimize the null-text embedding during the reconstruction process without cumbersome tuning of model weights. The Negative-Prompt Inversion (NPI) was further developed to reduce the optimization time [13, 31]. Both NTI and NPI narrow the gap of reconstruction in inversion-based editing, as depicted in Fig. 2(b). Ju *et al.* [21] challenged the optimization-based inversion methods and proposed Direct Inversion (DirectInv) by recording the differences between the inverted and the reconstructed features. The differences are then merged into the inference process to ensure high-quality reconstruction, as shown in Fig. 2(c).

In addition to NTI, NPI and DirectInv, there are many other image inversion methods [7, 10, 13, 20, 49, 56, 61] that have been proposed in recent years. While they can successfully reconstruct the source image by refactoring the image synthesis process, they all rely on DDIM inversion to provide the latent noise code. However, DDIM inversion assumes that the Ordinary Differential Equation (ODE) derived from the DDIM sampling process can be reversed with infinitesimally small steps. This assumption and the corresponding inversion equation exhibit instability during the inversion process. Consequently, conditioned on the given source prompt, the inverted latent noise code $\hat{z}_T$ of the source image will be closely coupled with the source prompt (please refer to Sec. 3.1 for our detail analysis), leading to a significant divergence D_{noi} with the ideal latent

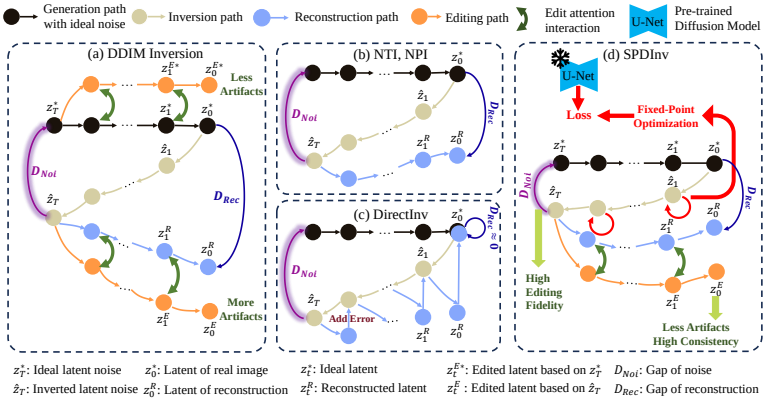

Fig. 2: Pipelines of different inversion methods in text-driven editing. (a) DDIM inversion inverts a real image to a latent noise code, but the inverted noise code often results in large gap of reconstruction D_{Rec} with higher CFG parameters. (b) NTI optimizes the null-text embedding to narrow the gap of reconstruction D_{Rec}, while NPI further optimizes the speed of NTI. (c) DirectInv records the differences between the inversion feature and the reconstruction feature, and merges them back to achieve high-quality reconstruction. (d) Our SPDInv aims to minimize the gap of noise D_{Noi}, instead of D_{Rec}, which can reduce the impact of source prompt on the editing process and thus reduce the artifacts and inconsistent details encountered by the previous methods.

noise code z_T^* (as depicted in Fig. 2(a)), which is supposed to be independent to the source prompt. The dependency on the source prompt brings obstacles for editing $\hat{z}_T$ with the target prompt, resulting in artifacts and inconsistent details in the edited image.

To address the above problem and disentangle the inverted latent noise code $\hat{z}_T$ from the source prompt, we revisit the DDIM sampling process, which iteratively denoises latent feature z_t to z_{t-1} with latent noise z_T^* as the starting point. By reversing the order of z_{t-1} and z_t in the ODE formula of DDIM, we can derive the inversion formula to obtain the ideal latent noise z_T^*, which does not have an analytical solution but can be solved as a fixed-point problem, as discussed in the work of accelerated iterative diffusion inversion (AIDI) [35]. This finding implies that the inversion process should adhere to a fixed-point constraint in order to disentangle the inverted noise code from the given source prompt. Unfortunately, as shown in Fig. 2(a-c), previous efforts [21, 31, 32, 46] have mostly focused on designing elaborate manipulations to reduce the gap of reconstruction, denoted by D_{Rec}, in the synthesizing process, without considering the adverse impact of the source prompt on the inverted noise code and how to reduce the gap of noise, denoted by D_{Noi}.

Based on the above analysis, we propose a **S**ource **P**rompt **D**isentangled **Inv**ersion method, termed as **SPDInv**. As illustrated in Fig. 2(d), SPDInv aims to minimize D_{Noi}, instead of D_{Rec}, so that the inverted noise code $\hat{z}_T$ can be independent to the source prompt as much as possible, reducing its po-

tential conflicts with the target prompt during the editing process. To achieve this goal, we transform each inversion step into a search problem with a fixed-point constraint. Different from AIDI [35] which searches the solution by direct iteration, we reformulate the fixed-point constraint to a loss function and leverage the powerful pre-trained diffusion models to perform the searching, largely narrowing the gap between the inverted noise code and the ideal noise without any source prompt prior. Our proposed SPDInv can be easily integrated into the inversion-based text-driven editing pipelines, such as P2P [14], MasaCtrl [4] and PNP [50], with just 10 lines of codes, significantly alleviating the artifacts and inconsistencies in the edited images. Furthermore, SPDInv can be easily applied to those customized text-to-image generation methods such as ELITE [55], adapting them to text-driven localized editing tasks with minor modifications. The main contributions of this paper are summarized as follows:

- We present SPDInv, a plug-and-play inversion method designed for text-driven image editing. It harnesses the power of pre-trained diffusion models to perform fixed-point searching in the inversion process, disentangling the inverted noise code from the source prompt as much as possible.
- We show that SPDInv can also be integrated with existing customized image generation methods, expanding their applications from customized T2I generation to text-driven localized editing.
- Our experimental results demonstrate that SPDInv effectively mitigates the dependency of inverted noise code on source prompt, significantly reducing the artifacts and inconsistent details in the editing outputs.

2 Related Work

2.1 Generative Models for Image Editing

The rapid advancement of diffusion-based generative models has significantly impacted the field of image and video generation. Large-scale pre-trained models such as Stable Diffusion (SD) [42], GLIDE [34], Imagen [44], and DALL·E2 [41] have demonstrated powerful image synthesis capability, and have been serving as foundational models for many downstream tasks. Prompt-to-Prompt (P2P) [14] is among the first to utilize SD for complex image editing through text interaction, achieving remarkable localized editing results by controlling attention maps. Subsequent works like pix2pix-zero [36] and plug-and-play (PNP) [50] offer fine-grained control over textual embedding and spatial features. MasaCtrl [4] manipulates self-attention features for consistent image generation and non-rigid editing simultaneously. By coupling with DDIM inversion, these methods can be used to edit real images, yet they often encounter editing failures due to the instability of DDIM inversion process.

In addition to localized editing, customized image generation has recently garnered much attention, which aim to generate images with the identity of user-provided object (*e.g.*, dog) unchanged. This can be achieved by fine-tuning parts of a pre-trained diffusion model (*e.g.*, the entire diffusion model [18, 43],

cross-attention [17, 23] module, or text embedding space [11]) or training an auxiliary module to translate visual content into the text embedding space [25, 33, 53, 55, 58]. While these methods have shown promising results in customized image generation, achieving fine-grained localized editing over customized images remains challenging. Our proposed SPDInv can be integrated into the existing customized generation methods to empower them with localized editing capacity.

2.2 Inversion for Real Image Editing

Inversion is a crucial step for editing real images in order to achieve reconstruction fidelity and editability. There are mainly four categories of inversions methods. The first category is DDPM [15] based methods, which directly use the DDPM forward equation to obtain the noise code [20, 30, 49]. However, the obtained noise code by these methods cannot guarantee the reconstruction of the original image. The second category is DDIM [46] optimization-based methods. While DDIM inversion [9] can be directly used to obtain the latent noise code, it may fail to reconstruct the image when the CFG parameters are set higher. To enhance reconstruction consistency, methods such as NTI [32], NPI [31] and ProxEdit [13] have been proposed to optimize the text embedding space. Some other methods have also been proposed to optimize the image latent space [7, 10] or the final noise space [61]. The third category is DDIM optimization-free methods, which aim to address the time-consuming issue of optimization-based methods. EDICT [52] and its variants [59] employ auxiliary invertible neural network to compute the inversion path. DirectInv [21, 56] records the differences between inversion and reconstruction, and then merges the differences into the inference process to ensure high-quality reconstruction. The last category is fine-tuning-based methods, which improve the reconstruction by overfitting the neural network to the given image [11, 24, 45] or training auxiliary networks [19, 55].

While the above methods can reconstruct well the source image using the inverted noise code, they may generate artifacts and inconsistent details when using the target prompt to edit the image. Some works such as AIDI [35] and FPI [29] have shown the effectiveness of fixed-point constraint in the inversion process. However, they employ the fixed-point iteration to search the solution, which is unstable and sub-optimal. In this work, we reformulate the fixed-point constraint as a loss function and leverage the pre-trained diffusion model to minimize it. Our method significantly reduces the editing artifacts and improves the detail consistency.

3 Our Method

3.1 Analysis on DDIM Inversion

The editing pipeline of most text-driven image editing methods has been depicted in Fig. 1. The given image z_0 is first inverted into latent noise z_T, which serves as the initial point for reconstruction and editing. DDIM inversion is commonly

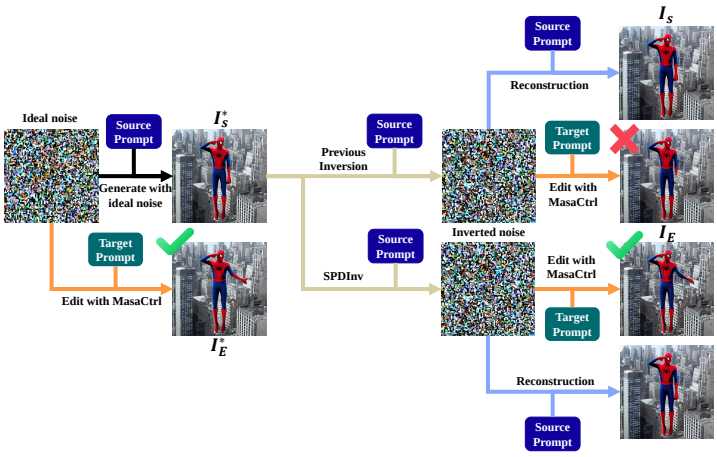

Fig. 3: An example of image editing with ideal noise code (left) and inverted noise code (right). Source prompt: A spiderman in the city. Target prompt: A spiderman in the city with his left hand up.

employed in the existing methods to obtain z_T. While DDIM inversion and its subsequent methods [13, 21, 31, 32] have shown promising results in editing real images, their editing fidelity and flexibility are far from the case if the ideal noise code can be used as the starting point. An example is illustrated in Fig. 3. From a random noise code without any prior knowledge, which can be regarded as an ideal noise, we generate a Spiderman image I_S^* with source prompt *"A spiderman in the city"*. With the same ideal noise and by using the MasaCtrl editing engine [4], we can successfully change the pose of Spiderman with target prompt *"A spiderman in the city with his left hand up"*. However, if we take the inverted noise code of I_S^* as the initial point, the edited result, denoted by I_E, will fail in editing Spiderman's pose (the left hand disappears). This is because during the inversion process, the prior information of source prompt is remained in the inverted noise code. While this prior information facilitates the reconstruction process, it impedes the editing fidelity and flexibility based on the target prompt, resulting in unintended artifacts or content inconsistency.

Let's make more analyses on the DDIM inversion process. Starting from the pure Gaussian noise $z_T \sim N(0,1)$, DDIM employs a deterministic sampling process [46] $z_{t-1} = \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} z_t + \sqrt{\alpha_{t-1}} \left(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1} \right) \epsilon_\theta(z_t, t, c)$ to generate the image latent code z_0. From this sampling equation, we can obtain the ideal inversion equation by using z_t and z_{t-1} with the following equation:

$$z_t = C_{t,1} * z_{t-1} + C_{t,2} * \epsilon_\theta(z_t, t, c), \tag{1}$$

where $C_{t,1} = \frac{\sqrt{\alpha_t}}{\sqrt{\alpha_{t-1}}}, C_{t,2} = \sqrt{\alpha_t}\left(\sqrt{\frac{1}{\alpha_t} - 1} - \sqrt{\frac{1}{\alpha_{t-1}} - 1}\right)$. The detailed derivation of Eq. (1) can be found in the **supplementary materials**. The inputs of the

neural network are supposed to be z_t and t. However, based on the assumption that the ODE formula can be reversed within infinitesimally small steps and due to the practical constraint that z_t is unavailable during one-step inversion at timestep $t-1$, DDIM inversion uses $(z_{t-1}, t-1, c)$, instead of (z_t, t, c), as the input of the neural network, resulting in the following formula:

$$z_t = C_{t,1} * z_{t-1} + C_{t,2} * \epsilon_\theta(z_{t-1}, t-1, c). \tag{2}$$

The coupling of DDIM inversion with source prompt is rooted in the use of Eq. (2). In the ideal inversion (*i.e.*, Eq. (1)), the input should be z_t and t, whereas previous methods have utilized z_{t-1} and $t-1$ as the input (*i.e.*, Eq. (2)). However, z_{t-1} is obtained by denoising z_t conditioned with source prompt. Therefore, Eq. (2) actually introduces source prompt prior into the update. Taking the last inversion step as an example, the neural network should receive a latent z_T a pure noise without any prior information as the input to obtain the ideal inversion code as $z_T = C_{T,1} * z_{T-1} + C_{T,2} * \epsilon_\theta(z_T, T, c)$, while most previous methods utilize the latent feature z_{T-1}, which is obtained through one-step denoising conditioned on the source prompt, as the input, *i.e.*, $\hat{z}_T = C_{T,1} * z_{T-1} + C_{T,2} * \epsilon_\theta(z_{T-1}, T-1, c)$. The coupling of the inverted noise code and source prompt is not limited to the final inversion step, as the multi-step generation nature of LDM accumulates the divergence between the inverted noise and the ideal noise. This eventually results in the inclusion of source prompt prior in the inverted noise, causing a notable deviation from the ideal noise, *i.e.*, gap of noise D_{Noi} as illustrated in Fig. 2(a). This deviation ultimately affects the editing stability and fidelity of the given real image.

3.2 Source Prompt Disentangled Inversion (SPDInv)

The aforementioned analysis highlights the significance of an ideal noise code, which should be disentangled with the source prompt, in editing a real image with target prompts. In practical applications, however, the corresponding ideal noise for a given image and source prompt is unknown, making it difficult to minimize the divergence between the ideal noise and the inverted noise. Nevertheless, Eq. (1) sheds light on the potential solution since it provides a constraint that z_t and z_{t-1} should satisfy at each inversion step. Consequently, narrowing the gap between the inverted and ideal noise codes can be achieved by optimizing z_t and z_{t-1} to meet the constraint of Eq. (1), thereby circumventing the issue of the unavailability of an ideal noise.

The ideal inversion equation in Eq. (1) is actually a fixed-point problem, which has been discussed in AIDI [35]. At the beginning of the inversion step, we have z_{t-1} available. Subsequently, by taking z_t as the variable, we can convert Eq. (1) into:

$$x = f_\theta(x) \quad where \quad x = z_t, f_\theta(x) = C_{t,2} * \epsilon_\theta(z_t, t, c) + C_{t,1} * z_{t-1}. \tag{3}$$

$C_{t,2}$ and $C_{t,1}z_{t-1}$ can be regarded as constants. By optimizing Eq. (3), the obtained z_t will approach to the ideal latent z_t^* in each inversion step. Eventually,

the inverted noise code z_T will approach to the ideal noise code z_T^*. AIDI [35] performs fixed-round iterations by assigning $f_\theta(z_t)$ to z_t to solve a similar fixed-point problem to Eq. (3). However, as LDM utilizes a neural network to map z_t, t, c to noise ϵ, the exact mathematical form of $\epsilon_\theta(z_t, t, c)$ cannot be obtained. As a result, the fixed-round iteration may not converge to a good fixed-point.

Algorithm 1 Source Prompt Disentangled Inversion (SPDInv)

Input: Source image latent z_0, DDIM steps T, source prompt p_s, maximal optimization round K, threshold δ, learning rate η.
Output: Inversion noise z_T

for $t \leftarrow 1$ to T **do**
  Get z_t from z_{t-1} based on Eq. (2)
  for $i \leftarrow 0$ to K **do**
    Calculate $L = ||f_\theta(z_t) - z_t||_2$ based on Eq. (3)
    Update $z_t := z_t - \eta \nabla L$
    if $L < \delta$ **then** *Break* **end if**
  end for
end for

We thereby propose a Source Prompt Disentangled Inversion (SPDInv) method to mitigate the influence of source prompt on the inverted noise code. We convert each inversion step into a search problem to identify the fixed point. Consequently, we reformulate the search problem into a loss function and leverage a pre-trained diffusion model to facilitate the optimization process. Our approach is straightforward yet effective, requiring only 10 lines of modifications to the existing inversion technique but improving the current state-of-the-arts significantly. Our algorithm is summarized in **Algorithm 1**.

Specifically, we utilize Eq. (2) to perform a single-step inversion from z_{t-1} to obtain an initial approximation to z_t. At this moment, z_{t-1} and z_t do not satisfy the constraint in Eq. (1). We employ a powerful pre-trained network $\epsilon_\theta(z_t, t, c)$ (*i.e.*, the Stable Diffusion 1.4 [42]), which is trained on a vast image-text dataset, and leverage its image-text comprehension capability to guide the optimization of Eq. (1). By incorporating z_t into Eq. (3), we transform the searching of z_t into the optimization of the following loss function:

$$\underset{z_t}{\operatorname{argmin}} L = \| f_\theta(z_t) - z_t \|_2 . \tag{4}$$

Eq. (4) can be minimized by the gradient descent techniques through $z_t := z_t - \eta \nabla L$, where η is the learning rate. The pre-trained diffusion model is fixed throughout the optimization process, with only the latent feature z_t being updated. Our experiments demonstrate that our SPDInv method exhibits superior performance compared to AIDI.

Furthermore, we observed that the loss function in Eq. (4) converges at varying speeds for different inversion steps t. In the early stages of the inversion process, more rounds of optimization are required to meet the fixed-point constraint in Eq. (3). When $t > \frac{T}{2}$, the loss quickly converges within a few rounds. Therefore, in addition to setting a maximal number of rounds K for all inversion steps, we introduce a threshold δ to control the termination of the optimization process to improve the efficiency of inversion process.

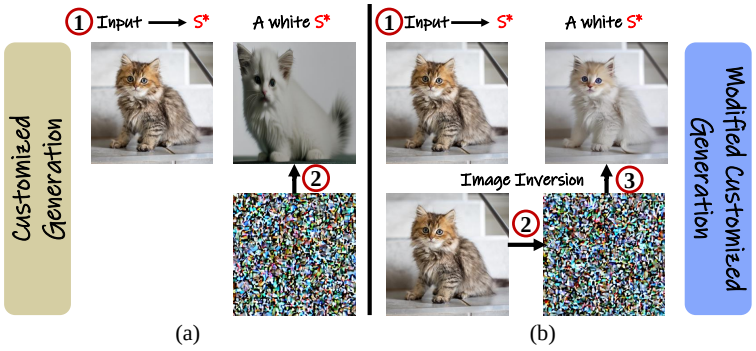

Fig. 4: (a) The pipeline of existing customized image generation methods. (b) The new image generation pipeline by integrating our proposed SPDInv method to generate the latent noise code, which can preserve the background of the input image.

3.3 Application to Customized Image Generation

Customized image generation [8, 48, 51] aims to generate new images by using the text concept (denoted by "S*") extracted from the given image together with other input text prompt. This can be achieved by fine-tuning parts of a pre-trained diffusion model (*e.g.*, the entire U-Net [18, 43], cross-attention modules [17,23], text embedding space [11,32]) or training an auxiliary module to translate visual contents into the text embedding space [33,55,58,60]. Especially, recent methods like ELITE [55], PhotoMaker [25], InstantID [53] can achieve quick customized generation of animals, portraits and other items. However, one of the limitations of customized image generation is that the generated image usually exhibits poor background and layout preservation. One example is depicted in Fig. 4(a). When we use the state-of-the-art customized image generation method ELITE to change the color of the cat in the given image, a cat with different pose and background can be returned.

To address the above mentioned issue of established customized image generation methods, we can easily integrate our proposed SPDInv into them to augment their localized editing capabilities. As shown in Fig. 4, with the existing methods (*e.g.*, ELITE), we can first transform the given image into the text embedding space aligned with text "S*" (*i.e.*, step 1 in Fig. 4(b)). Then, instead of performing synthesis with a random noise code, we uses SPDInv to invert the image into a noise code (step 2 in Fig. 4(b)), which works as the key to maintain the layout and background of the input image. Finally, with the inverted noise code and the new text prompt such as "a white S*", we can use pre-trained diffusion models (*e.g.*, stable diffusion v1.4 for ELITE) to generate an image with only the color changed, as depicted in the upper right of Fig. 4(b). The new pipeline in Fig. 4(b) extends the capability of existing customized image generation methods to perform high quality localized editing.

4 Experiment

4.1 Experiment Setting

Dataset. We use two datasets to evaluate our proposed SPDInv method. The first is the *PIE-Bench* [21] provided by DirectINV [21], which comprises 700 images with different editing types, including changing object, pose, color, material, background, style, adding and deleting object. In addition, following the setting of NTI [32], we randomly choose 100 images from the *COCO2017* [26] evaluation dataset without cherry-picking to build another test set. We construct the target prompt for each image with the same editing types as *PIE-Bench*. We call this test set as *TDE-Bench* (*T*ext-*D*riven *E*diting *Bench*mark).

Evaluation Metrics. Multiple metrics are employed to evaluate the performance of SPDInv from different aspects, including overall structure distance (assessed by DINO score [5]), background preservation (assessed by PSNR, LPIPS, MSE, SSIM), and prompt-image consistency (assessed by CLIP score [40]). As in previous methods, DINO score and CLIP score are calculated based on the entire image, while PSNR, LPIPS, MSE, and SSIM are calculated based on the region outside the annotated editing mask.

Comparison Methods. In Sec. 4.2, we compare SPDInv with seven representative and state-of-the-art inversion based editing methods, including DDIM inversion [46], Null-text inversion (NTI) [32], Negative prompt inversion (NPI) [31], Direct Inversion (DirectINV) [21], ProxEdit [13], Noise Map Guidance (NMG) [7], and AIDI [35]. When the P2P editing engine is used, all these comparison methods can be evaluated. However, many of these methods are inapplicable to the MasaCtrl and PNP editing engine due to the lack of source code. So we can only compare with DirectINV, NMG and AIDI under the MasaCtrl engine, and compare with DirectINV and AIDI under the PNP engine.

In Sec. 4.3, we choose the state-of-the-art customized image generation methods ELITE [55] as the baselines to demonstrate the improvement on localized editing brought by our SPDInv. We further compare our improved editing methods with two strong non-inversion based editing methods BlendDM [1] and InstructP2P [3] to demonstrate the effectiveness of SPDInv.

Other Settings. In our experiments, we set the DDIM sampling step as 50, the Classifer Free Guidance (CFG) as 7.5, and other parameters as their default values. The base model utilized is Stable Diffusion v1.4. The experiments and time consumption are tested on RTX3090.

4.2 Results on Text-Driven Image Editing

Tab. 1 presents the quantitative results of the competing inversion-based methods on PIE-Bench. We set the maximal optimization round $K = 25$ in SPDInv. One can see that SPDInv shows significant improvement over previous methods. When using the P2P editing engine, compared with the second-best methods (DirectINV or ProxEdit), SPDInv achieves visible improvements in DINO score (24% $\uparrow$), PSNR (5% $\uparrow$), LPIPS (21% $\downarrow$), MSE (13% $\downarrow$), SSIM (1.43 $\uparrow$), and CLIP

Table 1: Performance comparison of different inversion methods under the Prompt-to-Prompt (P2P) [14], Mutual self control (MasaCtrl) [4] and Plug-and-Play (PNP) [50] editing engines on PIE-Bench. Best and second best metrics are highlighted in red and blue colors, respectively.

Inversion	Editing Engine	DINO↓ $\times 10^3$	PSNR↑	LPIPS↓ $\times 10^3$	MSE↓ $\times 10^4$	SSIM↑ $\times 10^2$	CLIP↑	Inversion times(s)
DDIM	P2P	69.43	17.87	208.80	219.88	71.14	25.01	11.55
NTI	P2P	13.44	27.03	60.67	35.86	84.11	24.75	137.54
NPI	P2P	16.17	26.21	69.01	39.73	83.40	24.61	11.75
AIDI	P2P	12.16	27.01	56.39	36.90	84.27	24.92	87.21
NMG	P2P	23.50	25.83	81.58	107.95	82.31	24.05	16.71
DirectINV	P2P	11.65	27.22	54.55	32.86	84.76	25.02	19.94
ProxEdit	P2P	11.87	27.12	45.70	32.16	84.80	24.28	11.75
SPDInv	P2P	8.81	28.60	36.01	24.54	86.23	25.26	27.04
DDIM	MasaCtrl	28.38	22.17	106.62	86.97	79.67	23.96	11.55
NMG	MasaCtrl	40.54	20.35	127.85	135.17	77.52	24.56	16.71
DirectINV	MasaCtrl	24.70	22.64	87.94	81.09	81.33	24.38	19.94
AIDI	MasaCtrl	55.93	19.25	177.57	178.13	75.58	24.01	87.21
SPDInv	MasaCtrl	20.48	24.12	71.74	64.77	82.54	24.61	27.04
DDIM	PNP	28.22	22.28	113.33	83.51	79.00	24.95	11.55
DirectINV	PNP	24.29	22.43	106.09	80.52	79.62	25.02	19.94
AIDI	PNP	25.36	23.11	98.10	78.19	80.57	25.03	87.21
SPDInv	PNP	15.58	26.72	91.55	34.69	82.04	25.14	27.04

(0.24 ↑). The inversion time of SPDInv is longer than DDIM, NPI and ProxEdit, similar to NMG and DirectINV, and much shorter than NTI and AIDI (which also solves a fixed-point problem). With the MasaCtrl and PNP editing engines, SPDInv still shows great improvement over previous methods on most metrics, demonstrating the flexibility and effectiveness of SPDInv.

The visual comparison results are illustrated in Fig. 5 using the P2P engine. We can see that DDIM inversion always exhibits poor content consistency to the input target prompts. While the other competing methods show good editing performance on image cake (row 1), NTI, NPI, AIDI and ProxEdit suffer from artifacts in editing the image cat (row 2), detail inconsistency in editing the images statue (row 3), and fail in editing lipstick (row 5) and lightning (row 6). NMG and DirectInv fail to follow the editing instruction on image lipstick (row 4), lightning (row 5) and mountain (row 6). On the contrary, our SPDInv achieves successful editing in all these cases. More visual comparisons using MasaCtrl and PNP editing engines can be found in the **supplementary material**.

Due to the limited space, we put the experimental results on TDE-Bench in the **supplementary material**. Similar conclusions can be made.

4.3 Results on Localized Editing with Customized Generation

As described in Sec. 3.3, SPDInv can be integrated into the existing customized image editing methods such as ELITE [55] to endow them the localized editing capability. In this section, we integrate SPDInv into ELITE and name the

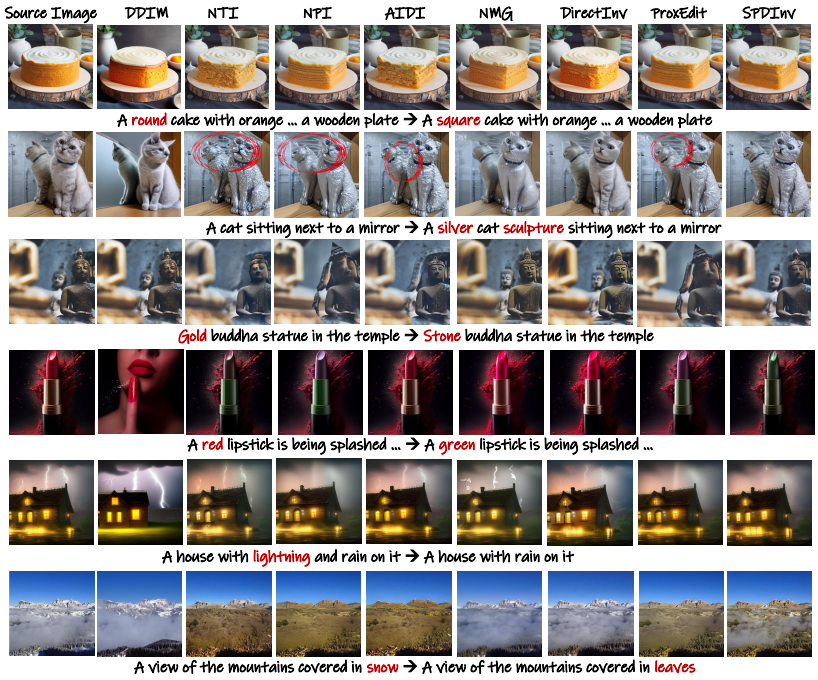

Fig. 5: Visual comparison of different editing methods with P2P on PIEBench.

resulted method as SPDInv-ELITE. We then verify the performance of SPDInv-ELITE on localized and customized image editing by using the test dataset provided by ELITE, which includes 20 subjects and the corresponding masks.

Tab. 2 presents the quantitative results. In addition to ELITE, we also provide the results of another two popular non-inversion editing methods BlendDM [1] and InstructP2P [3]. The results of P2P, PNP and MasaCtrl coupled with SPDInv are also listed. We see that the original ELITE performs worse than BlendDM and InstructP2P because it has poor background preservation capability. However, upon integration with SPDInv, significant improvements are achieved in all metrics, including DINO score (85% ↑), PSNR (62% ↑), LPIPS (63% ↓), MSE (86% ↓), SSIM (21.28 ↑), and CLIP (3.46 ↑). Furthermore, SPDInv-ELITE shows superior performance to SPDInv-PNP and SPDInv-MasaCtrl and comparable performance to SPDInv-P2P across the evaluation metrics.

The visual comparisons are depicted in Fig. 6. We see that the original ELITE can preserve the identity of object in the image, but suffers from the preservation of background and layout. Non-inversion editing method BlendDM achieves good background preservation but compromises the identities of flower (row 1) and teddy bear (row 3). InstructP2P shows good editing results in some cases (rows 2 and 3) but tends to change the identity and background in other cases (row 1). In contrast, SPDInv-ELITE enables localized and customized editing with

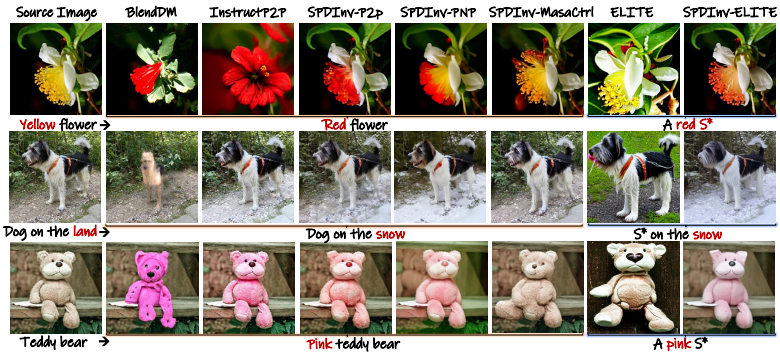

Fig. 6: Visual comparisons of localized image editing by different methods.

Table 2: Evaluation on localized and customized image editing.

Edit Engine	DINO$_{\times 10^3}$ ↓	PSNR↑	LPIPS$_{\times 10^3}$ ↓	MSE$_{\times 10^4}$ ↓	SSIM$_{\times 10^2}$ ↑	CLIP↑
BlendDM	59.21	15.51	244.07	306.75	67.45	20.21
InstructP2P	155.49	18.19	161.20	362.01	78.05	20.06
ELITE	148.37	14.83	201.94	359.58	67.62	15.72
SPDInv-P2P	14.77	26.59	56.30	42.58	91.20	18.63
SPDInv-MasaCtrl	31.78	22.04	91.59	71.79	87.34	16.04
SPDInv-PNP	31.33	23.23	83.63	69.79	86.90	18.83
SPDInv-ELITE	21.23	24.14	74.36	48.73	88.90	19.18

improved identity and background preservation. Additionally, SPDInv-ELITE outperforms MasaCtrl in identity and background preservation, and exhibits competitive visual results with P2P and PNP but superior performance in background preservation (row 2).

4.4 Ablation Study on Hyper-parameter Selection

There are four hyper-parameters in our SPDInv algorithm, *i.e.*, δ, η, K, T. Via extensive experiments, we empirically set them to $\delta = 5e^{-6}, \eta = 0.001, K = 25, T = 50$. Based on these default settings, in this section we conduct ablation experiments to investigate their effects on the final results by altering only one parameter while keeping the others fixed. Specifically, we select $K \in \{5, 25, 50\}, \delta \in \{5e^{-4}, 5e^{-5}, 5e^{-6}, 5e^{-7}\}, \eta \in \{0.005, 0.001, 0.01, 0.1\}$ and $T \in \{10, 50, 75, 100\}$. The first subset of PIE-Bench is employed in the ablation study.

The results are shown in Tab. 3. First, it is not a surprise that a higher value of K leads to improvements in PSNR, LPIPS, MSE, SSIM and DINO metrics because more iterations are used to solve the fixed-point problem. Nonetheless, there is a slight decline in the CLIP metric, and it requires more computational cost. Therefore, we choose $K = 25$ as the default value. Second, for the threshold δ, SPDInv achieves the best performance at $\delta = 5e^{-6}$. Further decrease of δ yields marginal improvements but results in increased inversion time. Third, for

Table 3: Ablation study on the hyper-parameters of SPDInv with PIE-Bench.

Hyper parameter	DINO$_{\times 10^3}$ ↓	PSNR↑	LPIPS$_{\times 10^3}$ ↓	MSE$_{\times 10^4}$ ↓	SSIM$_{\times 10^2}$ ↑	CLIP↑
$K = 5$	8.52	31.49	22.31	10.42	90.21	26.70
$K = 25$	8.43	31.61	21.70	10.12	90.28	26.67
$K = 50$	7.41	32.12	20.55	9.23	90.56	26.32
$\delta = 5e^{-5}$	9.00	29.61	29.24	16.02	89.83	26.85
$\delta = 5e^{-6}$	8.43	31.61	21.70	10.12	90.28	26.67
$\delta = 5e^{-7}$	8.59	31.65	22.30	10.23	90.22	26.67
$\eta = 0.005$	10.39	31.08	23.60	11.33	90.00	26.87
$\eta = 0.001$	8.43	31.55	22.13	10.28	90.10	26.67
$\eta = 0.01$	12.65	30.52	30.20	14.63	89.42	26.72
$\eta = 0.1$	48.91	22.82	130.98	149.87	80.89	23.26
$T = 10$	7.20	31.78	20.65	10.02	90.40	25.88
$T = 50$	8.43	31.61	21.70	10.12	90.28	26.67
$T = 75$	11.26	31.27	22.55	10.74	90.11	27.05
Default	**8.43**	**31.61**	**21.70**	**10.12**	**90.28**	**26.67**

the learning rate η, SPDInv performs the best when $\eta = 0.001$. Though slight enhancements in CLIP metrics can be achieved with $\eta = 0.005$ or $\eta = 0.01$, this comes at the price of deterioration of other metrics. Thus $\eta = 0.001$ is selected as our default setting. Finally, for the DDIM sampling steps T, we select $T = 50$ as the default value due to its balanced performance across all metrics. In addition, $T = 50$ is also the default setting in all the baselines mentioned in Sec. 4.2.

5 Conclusion

We proposed SPDInv, a novel inversion method designed to enhance the editability of text-driven image editing. To make the inverted noise code as independent as possible to the source prompt so that the image can be better edited with the target prompt, we incorporated a fixed-point constraint to the the inversion process, and showed that this fixed-point constraint could be effectively transformed into a loss function. Consequently, we utilized a pre-trained diffusion model to minimize this loss, and successfully disentangled the inverted noise code with source prompt, significantly improving the editing fidelity and flexibility. Additionally, by integrating SPDInv into customized image generation methods, we enhanced their localized editing capabilities. Experimental results demonstrated the superior performance of SPDInv on benchmark datasets.

SPDInv has some limitations. First, it relies on the existing editing engines (*i.e.*, P2P, PND, MasaCtrl) to edit images and thus inherits the limitations of them, such as low successful rate in adding and drop contents. Second, while SPDInv yields promising results in editing animals, foods and routine items, it still faces difficulties in editing portraits. Finally, SPDInv indeed narrows the gap between the inverted noise code and the ideal noise code, but it does not completely eliminate it. In the future, we will on one hand further improve the inversion process of SPDInv, and on the other hand design new editing pipelines to improve the stability and robustness of image editing results.

References

1. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18208–18218 (2022)
2. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22563–22575 (2023)
3. 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. 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)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
6. Chai, W., Guo, X., Wang, G., Lu, Y.: Stablevideo: Text-driven consistency-aware diffusion video editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23040–23050 (2023)
7. Cho, H., Lee, J., Kim, S.B., Oh, T.H., Jeong, Y.: Noise map guidance: Inversion with spatial context for real image editing. arXiv preprint arXiv:2402.04625 (2024)
8. Choi, J., Choi, Y., Kim, Y., Kim, J., Yoon, S.: Custom-edit: Text-guided image editing with customized diffusion models. arXiv preprint arXiv:2305.15779 (2023)
9. Elarabawy, A., Kamath, H., Denton, S.: Direct inversion: Optimization-free text-driven real image editing with diffusion models. arXiv preprint arXiv:2211.07825 (2022)
10. Epstein, D., Jabri, A., Poole, B., Efros, A., Holynski, A.: Diffusion self-guidance for controllable image generation. Advances in Neural Information Processing Systems **36** (2024)
11. 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. arXiv preprint arXiv:2208.01618 (2022)
12. Guo, Y., Yang, C., Rao, A., Wang, Y., Qiao, Y., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv preprint arXiv:2307.04725 (2023)
13. Han, L., Wen, S., Chen, Q., Zhang, Z., Song, K., Ren, M., Gao, R., Chen, Y., Liu, D., Zhangli, Q., et al.: Improving negative-prompt inversion via proximal guidance. arXiv preprint arXiv:2306.05414 (2023)
14. 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)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
16. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
17. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)

18. Hu, L., Gao, X., Zhang, P., Sun, K., Zhang, B., Bo, L.: Animate anyone: Consistent and controllable image-to-video synthesis for character animation. arXiv preprint arXiv:2311.17117 (2023)
19. Huang, Z., Wu, T., Jiang, Y., Chan, K.C., Liu, Z.: Reversion: Diffusion-based relation inversion from images. arXiv preprint arXiv:2303.13495 (2023)
20. Huberman-Spiegelglas, I., Kulikov, V., Michaeli, T.: An edit friendly ddpm noise space: Inversion and manipulations. arXiv preprint arXiv:2304.06140 (2023)
21. Ju, X., Zeng, A., Bian, Y., Liu, S., Xu, Q.: Direct inversion: Boosting diffusion-based editing with 3 lines of code. arXiv preprint arXiv:2310.01506 (2023)
22. 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)
23. Kumari, N., Zhang, B., Zhang, R., Shechtman, E., Zhu, J.Y.: Multi-concept customization of text-to-image diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1931–1941 (2023)
24. Li, S., van de Weijer, J., Hu, T., Khan, F.S., Hou, Q., Wang, Y., Yang, J.: Stylediffusion: Prompt-embedding inversion for text-based editing. arXiv preprint arXiv:2303.15649 (2023)
25. Li, Z., Cao, M., Wang, X., Qi, Z., Cheng, M.M., Shan, Y.: Photomaker: Customizing realistic human photos via stacked id embedding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
27. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023)
28. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Generating multiview-consistent images from a single-view image. arXiv preprint arXiv:2309.03453 (2023)
29. Meiri, B., Samuel, D., Darshan, N., Chechik, G., Avidan, S., Ben-Ari, R.: Fixed-point inversion for text-to-image diffusion models. arXiv preprint arXiv:2312.12540 (2023)
30. 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)
31. Miyake, D., Iohara, A., Saito, Y., Tanaka, T.: Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. arXiv preprint arXiv:2305.16807 (2023)
32. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6038–6047 (2023)
33. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)
34. 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)

35. Pan, Z., Gherardi, R., Xie, X., Huang, S.: Effective real image editing with accelerated iterative diffusion inversion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15912–15921 (2023)
36. Parmar, G., Kumar Singh, K., Zhang, R., Li, Y., Lu, J., Zhu, J.Y.: Zero-shot image-to-image translation. In: ACM SIGGRAPH 2023 Conference Proceedings. pp. 1–11 (2023)
37. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4195–4205 (2023)
38. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
39. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
40. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
41. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
42. 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)
43. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: 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. pp. 22500–22510 (2023)
44. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)
45. 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)
46. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2020)
47. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
48. Tewel, Y., Gal, R., Chechik, G., Atzmon, Y.: Key-locked rank one editing for text-to-image personalization. In: ACM SIGGRAPH 2023 Conference Proceedings. pp. 1–11 (2023)
49. Tsaban, L., Passos, A.: Ledits: Real image editing with ddpm inversion and semantic guidance. arXiv preprint arXiv:2307.00522 (2023)
50. 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)
51. Voronov, A., Khoroshikh, M., Babenko, A., Ryabinin, M.: Is this loss informative? faster text-to-image customization by tracking objective dynamics. Advances in Neural Information Processing Systems **36** (2024)

52. Wallace, B., Gokul, A., Naik, N.: Edict: Exact diffusion inversion via coupled transformations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22532–22541 (2023)
53. Wang, Q., Bai, X., Wang, H., Qin, Z., Chen, A.: Instantid: Zero-shot identity-preserving generation in seconds. arXiv preprint arXiv:2401.07519 (2024)
54. Wang, Z., Zhao, L., Xing, W.: Stylediffusion: Controllable disentangled style transfer via diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7677–7689 (2023)
55. Wei, Y., Zhang, Y., Ji, Z., Bai, J., Zhang, L., Zuo, W.: Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. arXiv preprint arXiv:2302.13848 (2023)
56. Xu, S., Huang, Y., Pan, J., Ma, Z., Chai, J.: Inversion-free image editing with natural language. arXiv preprint arXiv:2312.04965 (2023)
57. Yang, B., Gu, S., Zhang, B., Zhang, T., Chen, X., Sun, X., Chen, D., Wen, F.: Paint by example: Exemplar-based image editing with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18381–18391 (2023)
58. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721 (2023)
59. Zhang, G., Lewis, J.P., Kleijn, W.B.: Exact diffusion inversion via bi-directional integration approximation. arXiv preprint arXiv:2307.10829 (2023)
60. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)
61. Zhang, Y., Xing, J., Lo, E., Jia, J.: Real-world image variation by aligning diffusion inversion chain. Advances in Neural Information Processing Systems **36** (2024)
62. Zhang, Y., Huang, N., Tang, F., Huang, H., Ma, C., Dong, W., Xu, C.: Inversion-based style transfer with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10146–10156 (2023)