BK-Editer: Body-Keeping Text-Conditioned Real Image Editing*

Jiancheng Huang[1,2], Yifan Liu[1,3], Linxiao Shi[1,3], Jin Qin[1], and Shifeng Chen[1]

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] Southern University of Science and Technology
shifeng.chen@siat.ac.cn

Abstract. With the firestorm of generative macromodelling, text-conditional image editing is a recently emerged and highly useful task with an unlimited future. Although a lot of research progress has been made, most of the methods still fail to achieve editing under body-shape preservation, i.e., they cannot generate results that conform to the semantics of the editing prompt while preserving the body-shape of the original image subject. To address this great challenge, we propose BK-Editer, a method that achieves satisfactory body-shape preservation and accomplishes editing under body shape preservation, which solves two major problems: 1) the edited result matches the corresponding editing prompt, and 2) the edited subject's body shape is largely the same as the original subject's body shape. In addition, our method does not require time-consuming training on a large-scale dataset and is a self-supervised method.

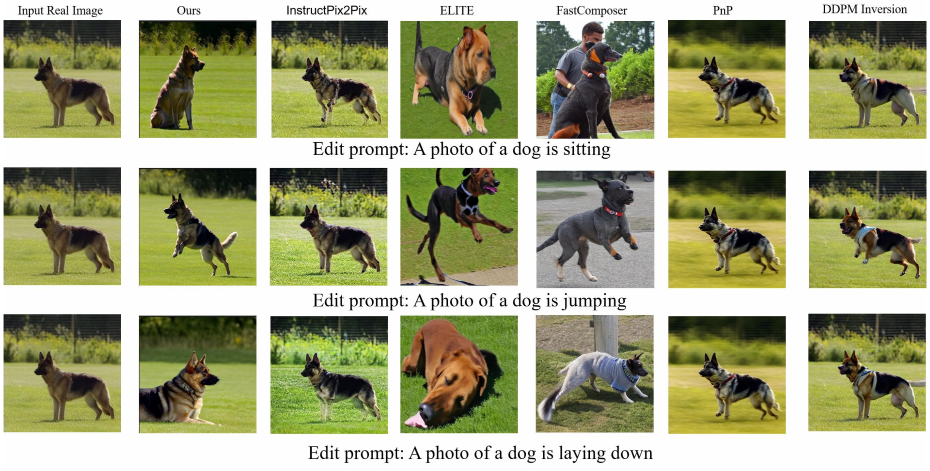
Keywords: Real image editing· Diffusion model· Text-to-image generation· AIGC· Generative model.

1 Introduction

With the rapid development of diffusion models and multi modal generative models, there have been significant advances in text-to-image techniques in recent months, which have enormous commercial value and have found their way into AI painting, commercial design, film animation production and many other areas. [33, 26, 48, 32, 35]. For example, Stable Diffusion [35] is capable of generating a wide variety of high quality images based on text prompts provided by the user. However, for some commercial purposes, the generation of a completely new image is not sufficient to satisfy the user's needs, and the user often wants to edit an existing image, giving rise to the task of text-conditional image editing. There are many approaches to implementing text-based image editing using pre-trained large-scale text-image models [26, 12, 43, 28]. Text-image editing is a very new task, it has only been around for a few months, but the prospect of its application in comic production, video editing, advertising material production, etc. is amazingly valuable.

* S. Chen -Corresponding author.

This work is supported by Shenzhen Science and Technology Innovation Commission (JSGG20220831105002004, JCYJ20200109114835623).



Edit prompt: A photo of a dog is sitting

Edit prompt: A photo of a dog is jumping

Edit prompt: A photo of a dog is laying down

Fig. 1: Comparing with different concurrent image editing methods on the real natural world images, it is obvious that the object in the editing result of our BK-Editer can meet the edit prompt, while keeping the body-shape of the original subject.

It is important for us to emphasise here that synthetic image editing and real image editing are two very different tasks, and it is easy for others to confuse the two tasks. Synthetic image editing means that when an image is synthesised using a source prompt, the edit prompt can be used to create another image that matches it, where the user cannot provide a real image. Real image editing, on the other hand, means that the user can provide a real image and then use the edit prompts to edit that real image.

A number of existing text-conditioned image editing methods [12, 43, 28] are able to perform tasks such as object replacement and style transfer while keeping the overall structure and layout unchanged, and achieve satisfactory results. However, these methods are unable to perform editing while keeping the body-shape of the original subject as shown in Fig. 1. In this context, "body-shape" encompasses not only the appearance but also the form, size, and outline of the object. Some works named subject-driven generation aim to address this problem, such as Textual Inversion [10], Imagic [18], DreamBooth [38], Custorm Diffusion [20], ELITE [44], FastComposer [46] and MasaCtrl [5] are proposed to subject-driven generation, which can also be a kind of editing. They can generate new images matching the edit prompt, but the body-shape problem is still challenge for them. Specially, Textual Inversion [10], Imagic [18], DreamBooth [38] and Custorm Diffusion [20] require finetune on some images of the same object and tend to be over-fitting. ELITE [44] requires training on a large dataset, which is more expensive on time and GPU. MasaCtrl does not require training and finetuning, but its reconstruction performance on real images is unsatisfactory and body-shape keeping is also hard for MasaCtrl.

Our approach focuses on the editing of the original subject under body-shape preservation, i.e., it emphasises that the body-shape of the subject in the edited result should be consistent with that of the original image. Our setting is that we do not need to use multiple images of the same object as input, nor do we need to train on a large dataset for a long time (in this field we usually call it **Training-free**).

A major problem under this setup is how to preserve the body-shape of the original subject during editing. In order to solve this problem, the fundamental difference between our BK-Editer and other concurrent works is that BK-Editer is designed by injecting the information of the original image into the proposed BK-attn in U-Net through the injecting network, and at the same time training these learnable parameters on a single original image by denoising loss. Since in finetuning we use human matting or pretrained segment model such as Segment Anything (SAM) [19] to segment out the subject part, the network can only perceive the subject information and memorise the body-shape of the subject in their learnable parameters. Then, in the edit stage, we use the edit prompt to introduce the edit semantic, and use the learned parameters of the Injecting network and BK-attn for retaining the body-shape.

The core distinction lies in our approach's integration of the original image as an additional input, alleviating the training burden on the network. This fundamental divergence from approaches like DreamBooth, Custom Diffusion, and Imagic [18] is a key factor in mitigating over-fitting tendencies.

Our main contributions are summarised as follows:

- We propose a text-conditional image editing method, BK-Editer, which does not need to be trained on top of a dataset to solve the body-keeping problem of the original subject in image editing.
- We design BK-attn, an attention layer that can input features of the original image, and embed it into U-Net, which can be make full use of the original image information during editing.
- Comprehensive experiments show that BK-Editer can obtain satisfactory performance in real image editing as shown in Fig. 1 and Fig. 7.

2 Related Work

Text-to-image generation models [41, 39, 32, 35, 49] have experienced an unprecedented surge in popularity, achieving impressive diversity in the generated images. Initially, high-fidelity image synthesis methods heavily relied on GANs [34, 51, 52, 47, 21, 54, 50, 42], often conditioned on text descriptions, owing to the impressive capabilities of GANs. By leveraging multi-modal vision-language learning, these models establish a connection between text descriptions and synthesized image contents. Recently, various text-to-image diffusion models [41, 13, 27, 8, 29, 10] have been developed, leveraging conditioning of the text prompt within the diffusion model. This approach has gained prominence due to the exceptional generative power and cutting-edge results in image quality and diversity achieved by diffusion models. Notably, recent advancements include DALL·E 2 [32], LDM [35], VQ-Diffusion [11], InstructPix2Pix [4], and GLIDE[26], which further enhance the synthesis process. Diffusion-based models [29, 10] have demonstrated the potential to manipulate images without human intervention, generating high-quality images that align with text descriptions.

Table 1: Comparison of Inversion and Editing Methods for Real Image Editing. (”Given Images” means the number of images required for tuning, and experiments are conducted on a single GTX 3090 GPU.)

Method	Type	Training Time	Tuning Time	Given Images
DDIM Inversion	Inversion	\	\	1
Null-text inversion	Inversion	\	3min	1
Textual Inversion	Inversion	\	50min	> 4
DDPM Inversion	Inversion	\	\	1
Negative-Prompt Inversion	Inversion	\	\	1
Proximal Inversion	Inversion	\	\	1
SDEdit	Editing	\	\	1
P2P	Editing	\	\	1
ELITE	Editing	3 days	\	1
FastComposer	Editing	3 days	\	1
Dreambooth	Editing	\	50min	> 20
LoRA	Editing	\	60min	> 30
Custom Diffusion	Editing	\	16min	> 20
InstructPix2Pix	Editing	4 days	\	1
MasaCtrl	Editing	\	\	1
BK-Editer	Editing	\	3min	1

Manipulating images based on natural language descriptions is a complex task in text-conditioned image editing. The revolution of image editing through text-conditioned editing using GANs [25, 22, 45, 29, 3, 16, 17, 1, 2] has sparked extensive research, while diffusion models, with their strong text feature extraction capabilities using CLIP [30], offer inherent capabilities for precise and diverse image editing. One innovative approach, VQGAN-CLIP [7], combines VQGAN [9] and CLIP [31] in an auto-regressive model, enabling the production of high-quality images and precise edits with controllable outcomes. Additionally, textual inversion provides an alternative image editing approach where models associate specific words in the textual embedding space with subjects in the corresponding images. Through training, the diffusion model can generate images of the specific subject in different scenes described by a sentence containing the designated word. This technique opens up new possibilities for generating images that align precisely with desired textual descriptions.

As shown in Tab. 1, we list many state-of-the-art methods for real image editing task and compare their difference between ours. Our BK-Editer is a training-free method, which means that we don’t need to be trained on a large paired dataset. But we still need to finetune our model on the given real image. So, our BK-Editer focuses on the setting of tuning on a single real image.

3 Background

3.1 Diffusion Model Training

In the training of diffusion model in latent space [36], it starts by encoding a natural image x_0 to a latent clean sample z_0. Then, it defines a diffusion process by adding noise to z_0:

$$q(z_t|z_0) = \mathcal{N}(\sqrt{\alpha_t} z_0, (1-\alpha_t)I), \tag{1}$$

where α_t is the diffusion schedule and $t \in \{1, ..., T\}$ is the time-step. The detail setting of α_t is introduced in [40]. We call $z_t, t \in \{1, ..., T\}$ the noisy latent, where $z_T \sim \mathcal{N}(0, I)$ is the end point of sampling. In DDPM [13], the common setting of T is 1000 and $t \in \{1, ..., 1000\}$. The optimization of the diffusion model is simplified to train a network $\boldsymbol{\epsilon}_\theta(z_t, t)$ to predict the Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$:

$$L_{simple} = \mathbb{E}_{z_0,t}\Big[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(z_t, t)||^2\Big]. \tag{2}$$

3.2 DDIM Sampling and Inversion

If we need to reconstruct a given real image, we should use the deterministic DDIM sampling [40] instead of other stochastic sampling. DDIM sampling is as follows:

$$z_{t-1} = \sqrt{\alpha_{t-1}}\frac{z_t - \sqrt{1-\alpha_t}\cdot\epsilon_\theta(z_t, t)}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}}\cdot\epsilon_\theta(z_t, t). \tag{3}$$

However, only DDIM sampling can't perform real image editing, as it only generates a completely new image from a random start point z_T. To perform real image editing, we need to invert a real image into z_T, and then use this inverted z_T as a starting point of DDIM sampling. If we want to perform real image editing, we can use editing methods such as SDEdit [23] and P2P [12] with z_T as a starting point of DDIM sampling.

DDIM inversion is used to invert a latent z_0 to a deterministic noisy latent z_T:

$$\begin{aligned} z_t =& \sqrt{\alpha_t}\frac{z_{t-1} - \sqrt{1-\alpha_{t-1}}\cdot\epsilon_\theta(z_{t-1}, t)}{\sqrt{\alpha_{t-1}}} \\ &+ \sqrt{1-\alpha_t}\cdot\epsilon_\theta(z_{t-1}, t). \end{aligned} \tag{4}$$

3.3 Text Condition and Classifier-Free Guidance

Text-condition diffusion models aim to generate a result latent z_0 from a random noise latent z_T with text prompt P. During the sampling process at inference, the noise estimation network $\epsilon_\theta(z_t, t, C)$ is used to predict the noise in each z_t, where $C = \psi(P)$ is the text embedding. The noise is gradually predicted and removed by $\epsilon_\theta(z_t, t, C)$ for T steps until we obtain z_0.

In text-conditioned image generation, it is necessary to give the textual condition enough control and influence over the generation. Ho et al. [14] propose classifier-free guidance, where the conditional and unconditional predictions are combined. Specifically, let $\varnothing = \psi(\text{``''})$ be the null text embedding and let w be the guidance scale, then the classifier-free guidance prediction is defined by:

$$\epsilon_\theta(z_t, t, C, \varnothing) = w\cdot\epsilon_\theta(z_t, t, C) + (1-w)\ \cdot\epsilon_\theta(z_t, t, \varnothing), \tag{5}$$

where $\epsilon_\theta(z_t, t, C, \varnothing)$ is used to replace $\epsilon_\theta(z_t, t)$ in the sampling Eq. 3, and w is usually in $[1, 7.5]$ in Stable Diffusion. The higher w means the stronger control by the text.

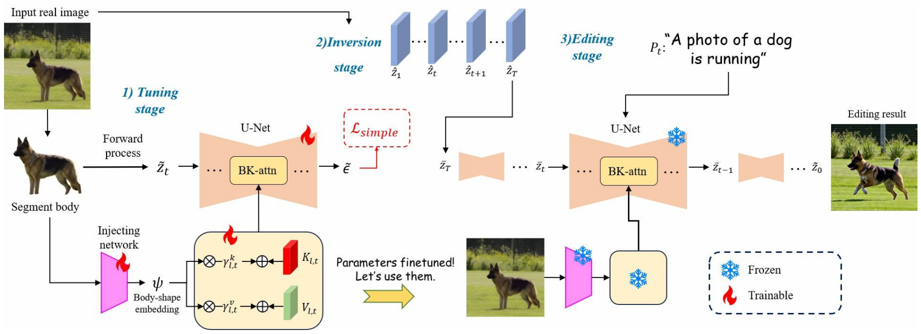


Fig. 2: Pipeline of the proposed BK-Editer, which can be divided into 3 stages, tuning stage, inversion stage and editing stage. 1) Tuning stage is for finetuning our new parameters to remember the body-shape of our subject. 2) Inversion stage is for getting a good start point in the sampling process. 3) Editing stage is using the body-shape to edit.

3.4 Stable Diffusion Model

Diffusion models [13, 40, 27] employ a dual process, involving the gradual addition of Gaussian noise to the training data followed by a subsequent reverse process that restores the original data distribution. The progression unfolds along the Markov chain during the forward process that transforms a data sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ into a sequence of noisy samples $\mathbf{x}_{1:T} = \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T$ in T steps. A neural network can be utilized to implement $p_\theta(\mathbf{x}_{0:T})$, with learnable parameters θ, to reverse the aforementioned process. By training a network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ to predict the Gaussian noise vector added to $\mathbf{x}_t$ [13], the optimization process can be transformed.

In Stable Diffusion (SD), an autoencoder network is employed for encoding $\mathbf{x}_0$ to $\mathbf{z}_0$ and decoding $\mathbf{z}_0$ to $\mathbf{x}_0$, while the U-Net [37] serves as the noise prediction network, denoted as ϵ_θ, responsible for generating latent noises $\hat{\mathbf{z}}_t$. The U-Net ϵ_θ comprises multiple CNN blocks, self-attention layers, and cross-attention layers. The self-attention layer facilitates the flow of information from the text prompt, whereas the cross-attention layer enables the flow of information from the features themselves. Specifically, query features Q are derived from the image features, while key and value features K and V are obtained from either the text embedding (cross-attention) or the image itself (self-attention). The generation of images is influenced by various factors: Firstly, the attention map of cross-attention plays a crucial role in determining the image structure [12, 43, 6], specifically the positions of the objects present. Secondly, the alteration of texture and detail in the generated image is attributed to the manipulation of K and V in the cross-attention layer. Lastly, both K, V, and the attention map in self-attention significantly impact the generated result in terms of content preservation.

3.5 Task Setting and the Body-Keeping Problem

Given a real rather than a synthesised source image I_s and a corresponding text prompt P_s (which can be semantic or empty), and given a target edit prompt P_t. Our task is to generate a new image I_t by a pre-trained stable diffusion model, and this edited result I_t should satisfy the following two requirements 1) I_t semantically matches the target prompt of P_t. 2) The objects in I_t should be consistent with I_s in terms of IDs, and in particular, the body shape of the object should be consistent with that in I_s.

This task has been a great challenge so far, especially when editing on real images, the problem is more serious and the result is unsatisfactory, most of the current image editing methods based on stable diffusion cannot maintain good reconstruction performance when editing actions on real images [12, 43], especially it is difficult to preserve the body-shape of the original subject. For example, if P_t is used directly to synthesise a new image $\bar{I}_t$, although the generated $\bar{I}_t$ can match the semantics of the actions in P_t, even if the $\hat{\mathbf{z}}_T$ obtained from DDIM inversion is used as the starting point of sampling, the subject in $\bar{I}_t$ tends to be different from the original I_s [12].

The core problem mentioned above is that the body-shape of subject generated in $\bar{I}_t$ is often very different from the original real image I_s. The primary cause of this phenomenon is that during editing, the Key K and Value V of cross-attention in the U-Net are derived from the target editing text prompt P_t, which brings new features different from those of the original real image I_s, so that the editing result has changed dramatically in the body-shape. Therefore, our core idea is to first use some parameters to preserve the body-shape of the source image I_s in the feature space. Then, we utilize these preserved body-shape with the target prompt P_t, and finally synthesize the desired editing image I_t. Thus, our BK-Editer focuses on the setting of tuning on a single real image.

4 Method

To realise the above core idea, we propose BK Editer, which can be divided into 3 stages, tuning stage, inversion stage and editing stage. It is obvious that our BK-Editer focuses on the setting of tuning on a single real image.

4.1 Tuning Stage for Finetuning Network

Tuning stage is the first stage, aiming at learning body-shape information by injecting real image into the network. For better learning the information from I_s, we need some parameters to save the body-shape of I_s. We find that the body-shape can be well learned in two ways. 1)The first way is to finetune the U-Net of Stable Diffusion Model (the nosie estimation network), which actually means saving the information of the body shape of I_s in the finetuned parameters of U-Net. 2) The second way is that we can design an additional network h_θ named injecting network for the input of I_s and train this injecting network, which indeed means saving the information of the body shape of I_s in these trained parameters.

We then combine the above two ways. To begin with, we introduce how the proposed injecting network h_θ works. h_θ is a lightweight network consisting of several base blocks [37]. Our findings suggest that its design does not hold critical importance within the scope of our task. Given the real image I_s, latent representation $\mathbf{z}_0$ is obtained by the encoder of VAE. Then we feed $\mathbf{z}_0$ into the injecting network to extract the

```
Algorithm 1: The 3 Stages of BK-Editer
Require: the latent of the original real image $\mathbf{z}_0$ and the source prompt embeddings $\mathbf{c}_s$.
  The initialization parameters $\epsilon_\theta$ and $h_\theta$.
1) Tuning stage: Set guidance scale $w = 7.5$;
while not converge do
    sampling timestep $t$ and noise $\epsilon$;
    obtain $\mathbf{z}_t$ by forward process with $t$ and $\epsilon$;
    get the output of U-Net $\epsilon_\theta(\mathbf{z}_t, t, \mathbf{c}_s, h_\theta(\mathbf{z}_0))$ with the proposed BK-attn and $h_\theta$;
    update the parameters by $\mathcal{L}_{simple}$.
end
Return trained parameters $\epsilon_\theta$ and $h_\theta$
2) Inversion stage: Set guidance scale $w = 1$ of stable diffusion model, in this stage we
  utilize DDIM inversion to obtain a sequence of latents $\{\hat{\mathbf{z}}_t\}$ including $\hat{\mathbf{z}}_T$. We save the
  BK-attn embeddings $(\bar{K}_{l,t}, \bar{V}_{l,t})$.
Return $\hat{\mathbf{z}}_T$ and BK-attn embeddings
3) Editing stage: Set guidance scale $w = 7.5$ and use $\hat{\mathbf{z}}_T$ as the start point of reverse
  process. Given a editing prompt $P_t$, then get its embedding $\mathbf{c}_t$ and an uncondition
  embedding $\mathbf{c}_u$;
for $t = T, T-1, \ldots, 1$ do
    $\epsilon_c = \epsilon_\theta(\bar{\mathbf{z}}_t, t, \mathbf{c}_t, h_\theta(\mathbf{z}_0))$, using corrsponding $(\bar{K}_{l,t}, \bar{V}_{l,t})$;
    $\epsilon_u = \epsilon_\theta(\bar{\mathbf{z}}_t, t, \mathbf{c}_u, h_\theta(\mathbf{z}_0))$, using corresponding $(\bar{K}_{l,t}, \bar{V}_{l,t})$;
    $\epsilon_t = \epsilon_u + w(\epsilon_c - \epsilon_u)$;
    $\bar{\mathbf{z}}_{t-1} = \text{Reverse}(\bar{\mathbf{z}}_t, \epsilon_t)$;
end
$I_t = \text{Decode}(\bar{\mathbf{z}}_0)$;
Return Editing result $I_t$
```

body-shape embeddings which can mainly saving the body-shape information of the subject in I_s.

$$\psi = h_\theta(\mathbf{z}_0). \tag{6}$$

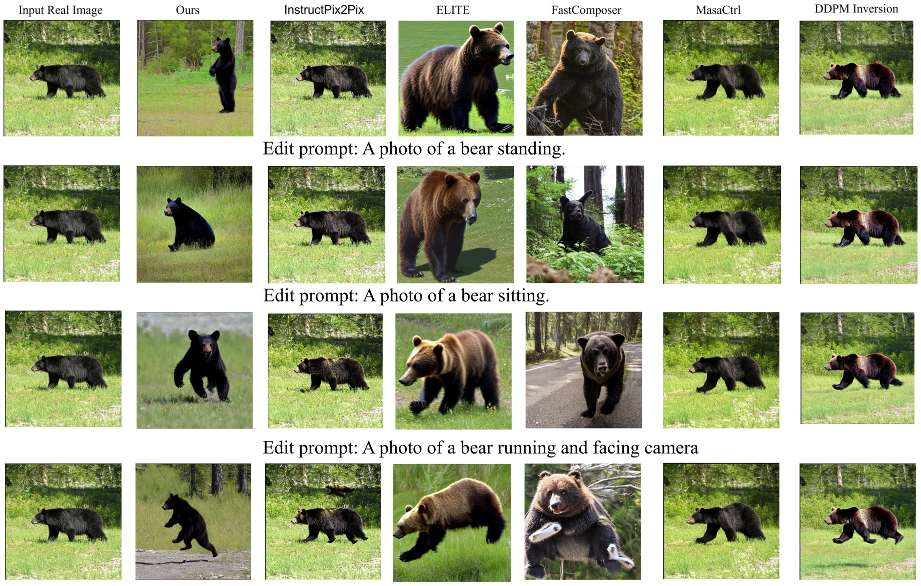
Here we use $\psi \in \mathbb{R}^{C \times H \times W}$ to denote the body-shape embeddings, where C, H and W are the channels and resolution, respeactively.

After obtaining the above body-shape embeddings ψ, we feed them into a proposed new type of attention layer named Body-Keeping Attention Layer (BK-attn). The BK-attn layer's foundation is built upon the self-attention layer of Stable Diffusion. For instance, given the l-th self-attention layer during step t of reverse process, we have:

$$\bar{K}_{l,t} = K_{l,t} + \gamma^k_{l,t}\text{Resize}(\psi), \quad \bar{V}_{l,t} = V_{l,t} + \gamma^v_{l,t}\text{Resize}(\psi), \tag{7}$$

$$\text{Attention}(Q_{l,t}, \bar{K}_{l,t}, \bar{V}_{l,t}) = \text{Softmax}(\frac{Q_{l,t}\bar{K}^T_{l,t}}{\sqrt{d}})\bar{V}_{l,t}, \tag{8}$$

where $(K_{l,t}, V_{l,t})$ and $(\bar{K}_{l,t}, \bar{V}_{l,t})$ denote the original and the new (Key, Value), respectively. Besides, $(\gamma^k_{l,t}, \gamma^v_{l,t})$ denote the learnable weights for the original Key and Value embeddings. By the above BK-attn, the body-shape information can be easily injected into the attention layer of the U-Net and perceived by the model. We use



Edit prompt: A photo of a bear standing.

Edit prompt: A photo of a bear sitting.

Edit prompt: A photo of a bear running and facing camera

Edit prompt: A photo of a bear jumping

Fig. 3: Real image editing results of different editing methods on bear images.

$\epsilon_\theta(\mathbf{z}_t, t, \mathbf{c}_s, h_\theta(\mathbf{z}_0))$ to represent the U-Net with our BK-attn, where $h_\theta(\mathbf{z}_0)$ is the body-shape embeddings.

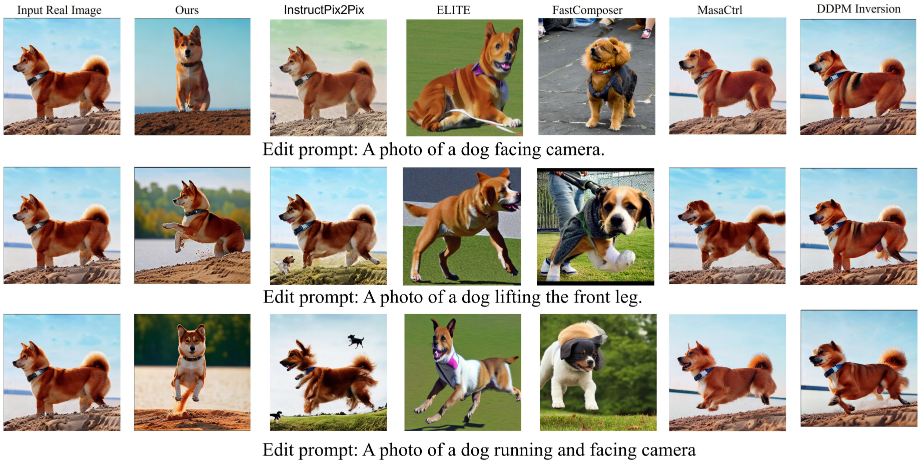
Now we define our proposed Body-Keeping finetuning as follows. For instance, given the U-Net ϵ_θ and the designed Injecting network h_θ, we can train them by the way similar to the training of Stable Diffusion Model. In particular, we sample a random timestep t as well as a standard Gaussian noise ϵ in each optimisation step, and then obtain $\mathbf{z}_t$ according to the forward process formula. Then, we get the output of U-Net $\epsilon_\theta(\mathbf{z}_t, t, \mathbf{c}_s, h_\theta(\mathbf{z}_0))$ to predict the added noise $\widetilde{\epsilon}_t$, where $\mathbf{c}_s$ is the text embedding of source prompt P_s. Since the real noise ϵ is known by us, we can thus use ϵ as supervision to train our learnable parameters h_θ and ϵ_θ to output a more precise noise $\epsilon_\theta(\mathbf{z}_t, t, \mathbf{c}_s, h_\theta(\mathbf{z}_0))$. The loss function is

$$\mathcal{L}_{simple} = \mathbb{E}_{t,\epsilon} \min_{h_\theta, \epsilon_\theta} \left\| \epsilon_\theta(\mathbf{z}_t, t, \mathbf{c}_s, h_\theta(\mathbf{z}_0)) - \epsilon \right\|^2 . \tag{9}$$

To focus the model's parameters on remembering the subject's body shape rather than the background, we use the segmentation model to segment out the subject part of the image, and simply set the background part to blank as in Fig. 2. With the learning procedure above, the well-optimized h_θ and ψ can actually preserve the body-shape of the source real image I_s well, which is important for the editing stage. The algorithm of the tuning stages is provided in Algorithm 1.

4.2 Inversion Stage for Obtaining BK-attn Embeddings

Specifically, as illustrated in Fig. 2, for obtaining the BK-attn embeddings $((\bar{K}_{l,t}, \bar{V}_{l,t}))$ in the diffusion process, we first perform DDIM inversion [8, 40] to generate a sequence



Edit prompt: A photo of a dog facing camera.

Edit prompt: A photo of a dog lifting the front leg.

Edit prompt: A photo of a dog running and facing camera

Fig. 4: Real image editing results of different editing methods on corgi dog images.

of noised latents $\{\hat{\mathbf{z}}_t\}$ including $\hat{\mathbf{z}}_T$. Note that we use the finetuned network with BK-attn for this inversion. We save the BK-attn embeddings $((\bar{K}_{l,t}, \bar{V}_{l,t}))$ for the edit stage.

4.3 Edit Stage with Body-Keeping

The overall architecture of the proposed pipeline to perform editing is shown in Fig. 2. $\hat{\mathbf{z}}_T$ obtained from inversion stage is used as the start point of the edit stage. During each denoising step t of generating the target editing image I_t, we also make full use of the trained U-Net with BK-attn and the injecting network h_θ. Specifically, we use the saved BK-attn embeddings $((\bar{K}_{l,t}, \bar{V}_{l,t}))$ in the corresponding place of the edit stage. Note that the information of $(\bar{K}_{l,t}, \bar{V}_{l,t})$ is from $h_\theta(\mathbf{z}_0)$ which preserves the body-shape of the source image I_s.

5 Experiments

Using publicly available checkpoints, we evaluate the effectiveness of our proposed method on two pretrained models: the state-of-the-art text-to-image Stable Diffusion Model [35] and the anime-style model Anything-V4. Our experimental focus lies in real image editing with body-keeping, where we employ DDIM scheduler [40] with 50 denoising steps during the inversion, tuning, and editing stages. The classifier-free guidance is carefully set to 7.5, while the remaining hyperparameters can be adjusted based on specific model requirements.

5.1 Comparisons with Other Concurrent Works

On Pretrained Stable Diffusion Model. In our evaluation, we compare the proposed BK-Editer with several state-of-the-art text-conditioned image editing methods, including InstructPix2Pix [4], Custorm Diffusion [20], P2P [12], MasaCtrl [5], FastComposer [46], ELITE [44], DDPM Inversion [15] and PnP [43]. Their codes and checkpoints are utilized to generate the editing outcomes.

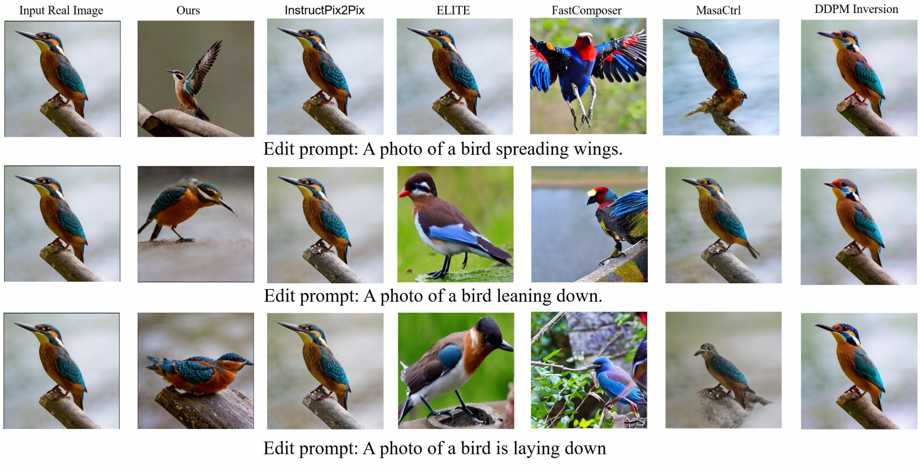

Input Real Image
Ours
InstructPix2Pix
ELITE
FastComposer
MasaCtrl
DDPM Inversion
Edit prompt: A photo of a bird spreading wings.
Edit prompt: A photo of a bird leaning down.
Edit prompt: A photo of a bird is laying down

Fig. 5: Real image editing results of different editing methods on bird shape images.

Fig. 1 demonstrate the editing performance, wherein our BK-Editer method exhibits superior performance in real image editing. Editing of real images remains a difficult task, and the main challenge lies in the difficulty of satisfying two goals simultaneously. First, existing methods either fail to generate editing results with the same body-shape as the original image, or fail to satisfy the corresponding semantics in the editing prompt.

The effectiveness of the proposed method, BK-Editer, is evident in its successful resolution of both of two challenges. Unlike approaches such as Custorm Diffusion [20] that remembers the subject by the parameters in the special token embeddings and cross-attention layers, our method focuses on learning parameters associated with the body-shape with the injecting of I_s. Furthermore, unlike ELITE [44] and FastComposer [46], our approach does not rely on training with large datasets, eliminating the need for dataset collection and the consumption of GPU resources and training time. The limitations of existing methods such as P2P [12] can be attributed to the utilization of the attention map from the source image in generating the edited image, leading to the replication of the original spatial structure. Although MasaCtrl [5] generates editing results with similar body, its reconstruction performance on real images remains inadequate. When these methods are constrained to real image editing without the use of additional control information like ControlNet, their effectiveness diminishes significantly.

On Preatrained Anything-V4 Model. We have conducted extensive tests to evaluate the effectiveness of our approach in the realm of animation image editing, specifically utilizing Anything-V4 dataset. The editing results showcased in Fig. 6 include comparisons with various other methods such as ELITE, InstructPix2Pix, P2P, MasaCtrl, PnP, and FastComposer. It is worth noting that our method focuses on real image input, rather than generating images based on given prompts. Moreover, the proposed BK-Editer demonstrates the versatility of our method by keeping the body-shape of the animated subjects in the source image while incorporating the desired editing prompt.

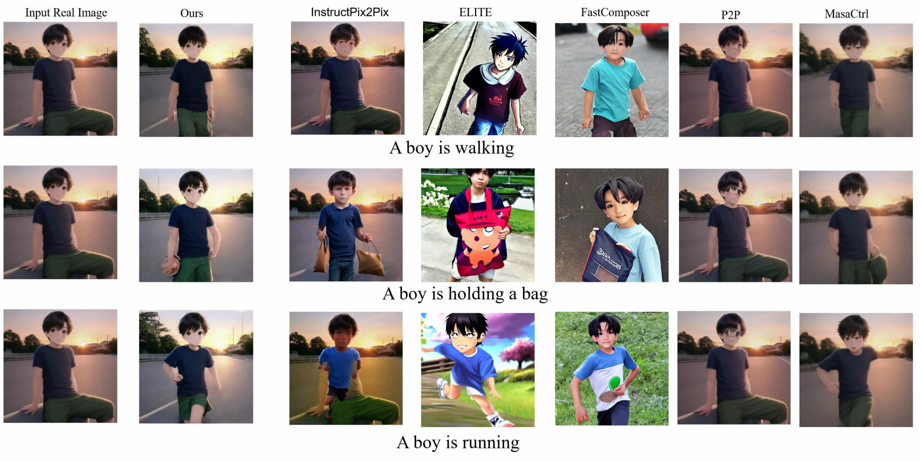


Fig. 6: Real image editing results of different editing methods on anime human images. It is obvious that our BK-Editer can achieve very good body-keeping performance on humanoid objects with matching to the edit prompt.

5.2 User Study

Determining the quality of body-shape preservation, especially amidst changes in viewpoint or pose, can pose challenges in qualitative assessments. As a means to address this, a user study is needed to validate the consistency and quality of body-shape preservation.

Thus, we conduct a user study with 30 participants to evaluate the human perception of our method on our edited results. Altogether, 20 real images are collected in various situations as our test set, including animals, humans, anime humans, anime animals, etc.

In the first part, we invite 30 participants to score the visual quality of only 10 edited results from 1 (worst) to 9 (best). Table 2 reports the average rating scores of different methods, among which our method receives the highest ratings.

In the second part, as shown in Fig. 7, we further conduct large-scale user study on 20 test images to evaluate the human perception of our method and 3 strongest SOTA methods on real image editing. Following the setting in other editing methods [18], we evaluate results via user answers on the six questions shown in Fig. 7 using a Likert scale of 1 (bad) to 3 (good). All methods are tested on the above 20 real images. Fig. 7 reports the rating distributions of different methods, among which our method BK-Editer receives more “good” and less “bad” ratings. Also, we performed a statistical analysis on the ratings using a paired t-test (using the T-Test function in MS Excel) between our approach and each of the other methods. With a significant level of 0.001, all the t-test results are statistically significant.

5.3 Ablation Study

The effectiveness of the proposed BK-Editer method and the use of the injecting network and BK-attn to input information from the original image can be effectively

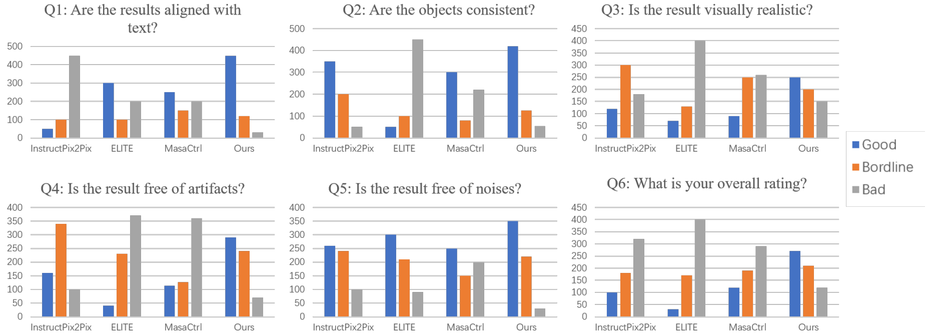


Fig. 7: Comparing with SOTA image editing methods on 30 people and 20 real images for 6 questions user study.

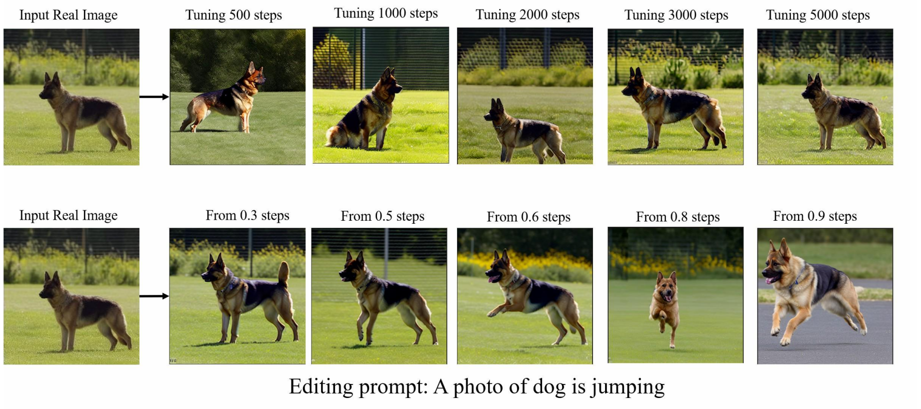


Fig. 8: Ablation study of the number of finetunning iterations (top) and the range of using BK-attn in different reverse timesteps (bottom).

demonstrated by comparing the results of real image editing. In order to increase the interpretability of the method and to explore the reasons for the success of the method, we performed ablation experiments by manipulating specific parameters (e.g., the number of iterations for training the Injecting network and U-Net parameters) and the different effects of using BK-attn at different time steps.

We gained insight into the behaviour of the method by analysing the effects on reconstruction and editing. We also compared the effects of using different numbers of training iterations, as shown in Figure 8 (top), where too many iterations of training can lead to overfitting, and too few iterations of training can lead to body size information that has not yet been learned. At Figure 8 (below), we found that using too few steps of BK-attn (starting from 0.9T steps) leads to a worse body shape retention in the edited results, which leads to a deviation from the original appearance of the edited results. On

Image	InstructPix2Pix	ELITE	FastComposer	PnP	MasaCtrl	Dreambooth	BK-Editer
1	5.56	6.28	7.23	6.55	6.37	6.67	**7.68**
2	6.12	7.12	7.12	6.91	5.91	7.57	**8.44**
3	5.86	6.69	7.01	5.71	4.90	7.24	**8.00**
4	5.13	6.56	5.68	5.61	6.54	7.52	**7.81**
5	5.80	6.96	6.23	5.78	6.15	7.10	**7.84**
6	4.78	6.44	6.73	5.53	5.80	6.89	**8.01**
7	5.65	5.95	6.24	5.40	6.08	6.11	**7.76**
8	4.42	5.33	6.06	4.28	4.79	5.49	**6.58**
9	5.78	6.75	6.02	5.52	5.58	7.00	**7.36**
10	6.44	7.32	6.10	6.34	5.12	5.04	**8.12**
average	5.55	6.54	6.44	5.76	5.72	6.66	**7.76**

Table 2: The scores of 10 editing images for user study.

the contrary, if too many steps of BK-attn are used (starting from 0.3T steps), the edited image is very similar to the source image.

6 Limitations and Conclusion

Overall, we propose BK-Editer that enables body-shape keeping and editing, which solves two major problems: 1) the edited results can be matched with the corresponding edit prompts, and 2) the edited objects can maintain the body-shape of the original real image. In addition, our method does not need to scan large datasets for very time-consuming training, nor does it need to collect full supervised data and labels.

The body-shape keeping performance of our BK-Editer is good enough since we can ensure the body-shape kept during the edit stage. Our method does not need some additional information (e.g., joint maps, depth maps, sketches, segmentation masks, etc.) supplied by the user to strongly perform body-shape control [24, 53], as providing this additional information is very cumbersome for the user. However, our method also has the problem that it still requires several minutes of finetuning the network at the tuning stage, and our next work is to explore faster tuning methods. Besides, the results from our ablation studies can be considered as failure cases. Some instances of failure are also observed in human experiments.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan++: How to edit the embedded images? In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8296–8305 (2020)
2. Abdal, R., Zhu, P., Mitra, N.J., Wonka, P.: Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Trans. Graph. **40**(3), 21:1–21:21 (2021)

3. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019)
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. arXiv preprint arXiv:2211.09800 (2022)
5. 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)
6. Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., Cohen-Or, D.: Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. arXiv preprint arXiv:2301.13826 (2023)
7. Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., Raff, E.: Vqgan-clip: Open domain image generation and editing with natural language guidance. In: ECCV. pp. 88–105. Springer (2022)
8. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. NeurIPS **34**, 8780–8794 (2021)
9. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR. pp. 12873–12883 (2021)
10. 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)
11. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: CVPR. pp. 10696–10706 (2022)
12. 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)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS **33**, 6840–6851 (2020)
14. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021)
15. Huberman-Spiegelglas, I., Kulikov, V., Michaeli, T.: An edit friendly ddpm noise space: Inversion and manipulations. arXiv preprint arXiv:2304.06140 (2023)
16. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. Advances in Neural Information Processing Systems **34**, 852–863 (2021)
17. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2019)
18. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. arXiv preprint arXiv:2210.09276 (2022)
19. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
20. Kumari, N., Zhang, B., Zhang, R., Shechtman, E., Zhu, J.Y.: Multi-concept customization of text-to-image diffusion. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
21. Li, B., Qi, X., Lukasiewicz, T., Torr, P.: Controllable text-to-image generation. NeurIPS **32** (2019)
22. Li, B., Qi, X., Lukasiewicz, T., Torr, P.H.: Manigan: Text-guided image manipulation. In: CVPR. pp. 7880–7889 (2020)
23. Meng, C., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021)

24. Mou, C., Wang, X., Xie, L., 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)
25. Nam, S., Kim, Y., Kim, S.J.: Text-adaptive generative adversarial networks: manipulating images with natural language. NeurIPS **31** (2018)
26. 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)
27. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: ICML. pp. 8162–8171. PMLR (2021)
28. Parmar, G., Singh, K.K., Zhang, R., Li, Y., Lu, J., Zhu, J.Y.: Zero-shot image-to-image translation. arXiv preprint arXiv:2302.03027 (2023)
29. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: ICCV. pp. 2085–2094 (2021)
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
31. 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: ICML. pp. 8748–8763. PMLR (2021)
32. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
33. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: ICML. pp. 8821–8831. PMLR (2021)
34. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: ICML. pp. 1060–1069. PMLR (2016)
35. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2021)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. Springer (2015)
38. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. arXiv preprint arXiv:2208.12242 (2022)
39. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Gontijo-Lopes, R., Ayan, B.K., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. In: Advances in Neural Information Processing Systems (2022)
40. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
41. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. NeurIPS **32** (2019)
42. Tao, M., Tang, H., Wu, F., Jing, X.Y., Bao, B.K., Xu, C.: Df-gan: A simple and effective baseline for text-to-image synthesis. In: CVPR. pp. 16515–16525 (2022)
43. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation. arXiv preprint arXiv:2211.12572 (2022)
44. 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 (2023)
45. Xia, W., Yang, Y., Xue, J.H., Wu, B.: Tedigan: Text-guided diverse face image generation and manipulation. In: CVPR. pp. 2256–2265 (2021)

46. Xiao, G., Yin, T., Freeman, W.T., Durand, F., Han, S.: Fastcomposer: Tuning-free multi-subject image generation with localized attention. arXiv preprint arXiv:2305.10431 (2023)
47. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: CVPR. pp. 1316–1324 (2018)
48. Yu, J., Xu, Y., Koh, J.Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B.K., et al.: Scaling autoregressive models for content-rich text-to-image generation. arXiv preprint arXiv:2206.10789 (2022)
49. Yu, J., Xu, Y., Koh, J.Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B.K., et al.: Scaling autoregressive models for content-rich text-to-image generation. arXiv preprint arXiv:2206.10789 (2022)
50. Zhang, H., Koh, J.Y., Baldridge, J., Lee, H., Yang, Y.: Cross-modal contrastive learning for text-to-image generation. In: CVPR. pp. 833–842 (2021)
51. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV. pp. 5907–5915 (2017)
52. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan++: Realistic image synthesis with stacked generative adversarial networks. IEEE TPAMI **41**(8), 1947–1962 (2018)
53. Zhang, L., Agrawala, M.: Adding conditional control to text-to-image diffusion models. arXiv preprint arXiv:2302.05543 (2023)
54. Zhu, M., Pan, P., Chen, W., Yang, Y.: Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In: CVPR. pp. 5802–5810 (2019)