Machine Vision and Applications (2022) 33:45
https://doi.org/10.1007/s00138-022-01298-7

ORIGINAL PAPER

Paired-D++ GAN for image manipulation with text

Duc Minh Vo[1,2] · **Akihiro Sugimoto**[3]

Received: 16 September 2021 / Revised: 9 March 2022 / Accepted: 17 March 2022

Abstract
Image manipulation with text is to semantically modify the appearance of an object in a source image based on the given text describing the novel visual attributes while retaining other irrelevant information in the image, such as the background. This has a wide range of applications, such as intelligent image manipulation, and is helpful to those who are not good at painting. We propose a generative adversarial network having a pair of discriminators with different architectures, namely *Paired-D++ GAN*, for image manipulation with text where the two discriminators make different judgments: one for foreground synthesis and the other for background synthesis. The generator of Paired-D++ GAN has the encoder–decoder architecture with skip-connections and synthesizes an object's appearance matching the given text description while preserving other parts of the source image. The two discriminators judge the foreground and background of the synthesized image separately to meet the given input text description and the given source image. The Paired-D++ GAN is trained using the effectively unconditional and conditional adversarial learning process in a simultaneous three-player minimax game. Our comprehensively experimental results on the Caltech-200 bird dataset and the Oxford-102 flower dataset show that Paired-D++ GAN can semantically synthesize images to match an input text description while retaining the background in a source image against the state-of-the-art methods.

Keywords Image manipulation · Image manipulation with text · Generative adversarial network · Image synthesis · Paired-discriminator

1 Introduction

Image manipulation with text [1–3] is to manipulate the visual attributes of an object in a given source image semantically with given text descriptions while still retaining features that are irrelevant to what text descriptions. Since text descriptions [4] are easier and more natural for us than image descriptions such as attributes [5], textures [6] or styles [7], image manipulation with text is promising to widen the range of applications of image synthesis such as intelligent image manipulation, computer-aided design and video game [2,3]. In fact, most text descriptions emphasize foreground characterization (i.e., the main object), while the background is implicitly conditioned. For instance, the text 'this small bird has a blue crown and white belly' only describes the bird's appearance without any additional background information. As a result, the task can be referred to as rendering foreground given as a text description into a given source image.

A straightforward approach is learning a segmentation mask to automatically separate foreground and background information. The foreground is then manipulated concerning the given text. Finally, a new image is created by combining the manipulated foreground and the original background. Nonetheless, due to the fact that the well-annotated masks are not available in the training dataset, learning segmentation mask is not reasonable, limiting the effectiveness of this approach in practice. Therefore, we approach our problem by automatically matching foreground and text description and retaining other (background) information simultaneously [1–3].

Generative adversarial network (GAN) [8] is capable of synthesizing images, and works [4,9–12] have been proposed that condition GAN on either text descriptions [4,9]

✉ Duc Minh Vo
vmduc@nlab.ci.i.u-tokyo.ac.jp

Akihiro Sugimoto
sugimoto@nii.ac.jp

[1] Department of Informatics, SOKENDAI (The Graduate University for Advanced Studies), Tokyo, Japan

[2] Present Address: The University of Tokyo, Tokyo, Japan

[3] National Institute of Informatics, Tokyo, Japan

Published online: 08 April 2022

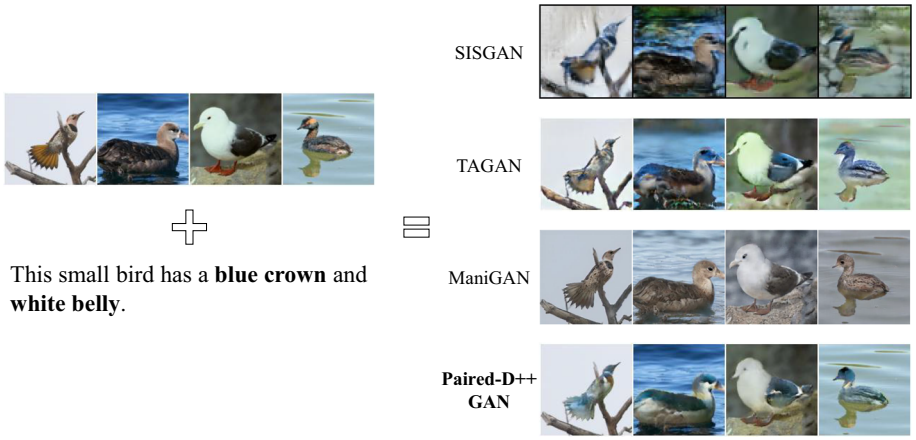

Fig. 1 Examples of synthesized images. Our results (Paired-D++ GAN) match the text description more precisely than SISGAN [1], TAGAN [2] and ManiGAN [3] while successfully retaining background of the source image

or images [10–12] to synthesize images for various tasks. Almost all work on image synthesis [1–4,9,13,14] follows idea of the original GAN architecture where a single discriminator judges whether or not a synthesized image is realistic.

Image manipulation with text requires disentangling the semantics contained in image and text information and then combining the disentangled semantics to synthesize realistic images. This suggests separately dealing with text descriptions and images with different semantic levels. Previous work [1–3], however, judges the foreground and background jointly through a single discriminator. As a consequence, their methods suffer from several potential shortcomings. Firstly, the foreground in the generated image cannot faithfully match the text description. Secondly, the generated image cannot retain the background in the source image well. For instance, as shown in Fig. 1, SISGAN [1] and TAGAN [2] reasonably match the foreground and the text description, whereas their background becomes worse than that of the source image. Meanwhile, MainGAN [3] cannot handle the task properly.

To advance this task, we design a GAN with a pair of discriminators, namely *Paired-D++ GAN*, to separately deal with text descriptions and source images. Indeed, dual discriminator GAN [15] showed that having two discriminators is more effective than GANs with one discriminator for image synthesis. While dual discriminator GAN [15] uses the same architecture for the two discriminators in unconditional GANs to judge real/fake images, we design different architectures for two discriminators to deal with different levels of the semantics of text descriptions and images. The two discriminators separately judge the foreground and background of the synthesized image to meet an input text description and a source image. More precisely, one discriminator is designed for the classification task while the other performs the verification task. Furthermore, we employ the skip-connection in the generator to more precisely retain background information in the source image. We also introduce a training process for adversarial learning in the three-player minimax game of the generator and two discriminators. In this way, Paired-D++ GAN improves the quality of synthesized images to meet the requirements of the text-guided image manipulation problem. We see that Paired-D++ GAN not only matches the foreground more precisely than the other methods but also is able to retain the background of the source image successfully (see Fig. 1). Experiments on the Caltech-200 bird dataset [16] and the Oxford-102 flower dataset [17] demonstrate outperformances of Paired-D++ GAN against SOTAs [1–3,14,18].

The rest of this paper is organized as follows. We briefly review and analyze related work in Sect. 2. Next, we present the details of our proposed method in Sect. 3. Then, Sect. 4 and Sect. 5 discuss our experiments. Section 6 draws the conclusion. We remark that this paper extends the work reported in [19]. Our main extensions in this paper are building a new network where the network is trained with both unconditional and conditional adversarial learning processes and adding more experiments. More precisely, we replace the fixed pre-trained modules used in both discriminators with the trainable ones and use the text-adaptive discriminator [2] to improve foreground-text matching.

2 Related work

With the rapid development of deep learning, many models for image synthesis have been proposed to achieve highly realistic images. They include variational auto-encoder [20,21], auto-regressive models [22,23] and GAN [1–4,8,9,13,14,18,24,25]. Among them, GAN and its variants show remarkably realistic results.

GAN [8] consists of a generator and a discriminator. The generator maps the latent variable into the data space while the discriminator judges whether the output of the generator is real or fake. The generator and the discriminator are simultaneously trained in a minimax game. Interestingly, GAN can be constrained on various conditions not only to gener-

ate plausible images but also to meet the conditions. Some work condition GANs on the attribute label [5,26] or images [12,27–30] for image super-resolution [12], domain transfer [27,28], photo editing [29] and style transfer [30].

Among various conditions on GAN, text descriptions make image synthesis easier and more friendly to us. [4] proposed an end-to-end GAN using the text condition. They employed a pre-trained text encoder [31] to extract text features from an input text and then combined text features with a vector representing random noise to produce the input of the generator. They also employed the combination of text features and image features in the discriminator to discriminate real images and generated images. Their proposed model [4] became the baseline of the GAN framework for generating images from text descriptions.

As an extension, a model conditioned on texts and location information was proposed [32]. Models with two stages of GAN, Stack-GAN [9] (and Stack-GAN++ [13]), were also proposed, showing successfully generated higher resolution images (256×256), compared to [4] (64×64). [14] proposed AttnGAN, where an attention mechanism is incorporated into GAN for a fine-grained text-to-image generation. Their model generated image details by paying attention to the relevance of words in text description and image features. These models [4,9,13,14,18] condition on GAN only texts or a pair of texts and location information [32] and focus on generating a new image. In addition, since the text descriptions are usually about the foreground, these mentioned methods generate background in a random manner and struggle to generate faithful foreground and background simultaneously. Different from aforementioned methods [4,9,13,14,18,32], our method is set to manipulate a part of image (foreground) according to text descriptions while retaining irrelevant information (background). Moreover, we do not aim at generating high-resolution images as [9,13,14].

Addressing the background problem in image synthesis, [18] proposed to decompose the image synthesis into two phases using foreground and background generators. They fed random noise vectors to a long short-term memory (LSTM) network to obtain hidden states for the foreground generator and used the first hidden state to generate the background. They then combined foreground and background by a compositor operator. However, decomposing foreground and background may cause less realistic images.

Recent work [33] proposed image manipulation using open-vocabulary instructions. They first train an image–caption joint embedding space. Then, the manipulation is performed by vector arithmetic operations between the image features and the textual features. Finally, an image decoder is used to reconstruct the image from manipulated features. Though their method is capable of handling open-domain, it requires the manipulation instructions should clearly define the source object/attribute to be edited and the target object/attribute to be added (i.e., the inputs consist of an image and two (vocabulary) instructions) which may cause difficulties for human in practice. Moreover, their method is able to manipulate objects/attributes one by one.

The models proposed by [1–3] are most related to ours. They also condition text and source image on GAN. The architecture of the model used in [1] is, however, similar to [4] and has a single discriminator: the noise vector in [4] is replaced by image features from the image encoder. To enhance the matching between text description and the foreground, [2] proposed a text-adaptive discriminator. Their discriminator splits a text description into word-level so that the discriminator is able to match each word to each visual attribute more precisely. [3], on the other hand, attempted to generate attributes matching text description and to reconstruct text-irrelevant contents of the source image at the same time. They thus proposed the text-image affine combination module (ACM) and detail correction module (DCM). The ACM seeks the text-relevant regions in the source image to generate new attributes matching given text descriptions, while the DCM rectifies text-irrelevant regions and completes missing contents. Though they [1–3] generate images that match the semantic meaning of the input text description while maintaining other parts of a source image, they do not preserve background precisely because the discriminator is used only for the foreground. To some extent, the model proposed by [3] cannot handle the task properly because their ACM does not understand the input text well while their DCM tends to reconstruct the whole source image. Table 1 summarizes our closely related work.

Our main difference from the aforementioned models is to fully take into account each role of foreground and background in synthesized images. More precisely, our proposed Paired-D++ GAN is conditioned on both text description and source images. It has skip-connections in its generator to preserve background information as much as possible and two discriminators with different architectures for synthesizing realistic images. Paired-D++ GAN generates foreground and background simultaneously.

3 Proposed method

3.1 Network design

Our network follows the GAN architecture [8] for image synthesis [1,4,9,13]. Like [1–3], we condition GAN on a text descriptions and a source image. As investigated in [7,19], the VGG-16 [34] pre-trained on ImageNet dataset [35] weights background in the early layers and foreground at the latter layers. More precisely, the 1st to the 3rd ReLU layers capture background while the 5th to the 7th ReLU layers do

Table 1 A summary of closely related work

Method	Methodology	Advantages	Disadvantages	Dataset
SISGAN [1]	–Combine image and text as the input of the generator	–Learn the shared representation of image and text	–Do not match image and text well	–Caltech–200
	–An adversarial training strategy using image and text	–Able to semantically synthesize image with text	–Do not preserve background precisely	–Oxford–102
TAGAN [2]	–Split a text description into word–level	Better matching between text description and the foreground	Do not preserve background precisely	–Caltech–200
	–Match each word to each visual attribute by using text–adaptive discriminator			–Oxford–102
ManiGAN [3]	–Text–image affine combination module (ACM)	–ACM seeks the text–relevant regions in source image to generate new attributes matching given text descriptions	–Do not preserve background precisely	–Caltech–200
	–Detail correction module (DCM)	–DCM rectifies text–irrelevant regions and completes missing contents	–ACM does not understand the input text well	–COCO
			–DCM tends to reconstruct whole source image	
LR–GAN [18]	–Generate foreground and background separately	Solve the background problem in image synthesis	The generated images are less realistic	–MNIST–ONE (one digit)
	–Combine generated foreground and background by a compositor operator			–MNIST–TWO (two digits)
				–CIFAR–10
				–Caltech–200
AttnGAN [14]	Incorporate attention mechanism into GAN	Generate image details by paying attention to the relevance of words in text description and image features	–Generate background in random manner	–Caltech–200
			–Cannot generate faithful foreground and background at the same time	–COCO

foreground, and the 4th ReLU layer seems to be in-between as a transition. We thus use different semantic levels of features depending on foreground and background. Namely, we design the network in which a text description on the foreground matches features in the latter layers while features of a source image in the early layers are preserved as much background information as possible. This appropriate-level selection allows our model to synthesize realistic images that meet both the text description and the source image.

[15] argued that dual discriminators in GAN generate better images in quality than a single discriminator, though the two discriminators have the same architecture. To deal with foreground and background separately and more precisely, we employ a pair of discriminators where each of them independently judges the foreground/background of synthesized images. For different semantic levels of foreground and background, we design our discriminators with different architectures and make each play a different role. Namely, we design one discriminator to evaluate matching foreground between a text description and a synthesized image following previous work [1,2,4,9] and the other discriminator to evaluate whether the background of a source image is retained in

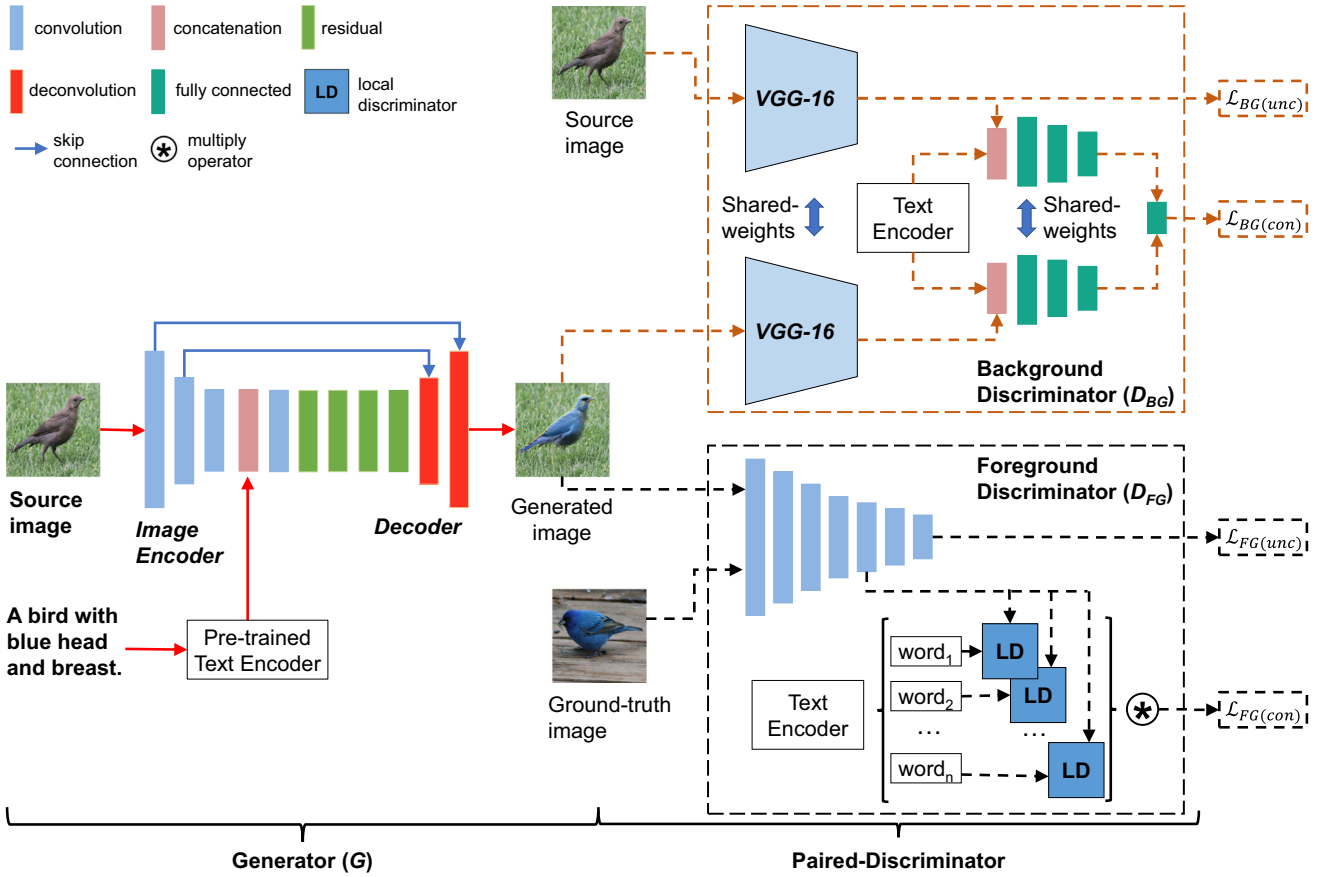

Fig. 2 Framework of our proposed Paired-D++ GAN. The generator G with skip-connection synthesizes an image that adaptively changes the foreground to match the text description while retaining the background of the source images. On the other hand, the foreground discriminator D_{FG} judges whether or not the foreground agrees with the text description. Meanwhile, the background discriminator D_{BG} judges whether or not the background is retained as that of the source images. Three networks (G, D_{FG} and D_{BG}) are simultaneously trained in a three-player minimax game

the synthesized image. We also introduce an effective training strategy for adversarial learning in a three-player minimax game.

3.2 Network architecture

We build our network, Paired-D++ GAN, upon the GAN architecture with one generator G and a pair of discriminators, i.e., foreground discriminator D_{FG} and background discriminator D_{BG} (Fig. 2).

We employ the end-to-end encoder–decoder architecture for our generator G following [1]. The generator G receives a source image and a text description where the source image is with the size of $128 \times 128 \times 3$ and the text description is with a maximum of 50 words. G synthesizes an image of $128 \times 128 \times 3$ that adaptively changes the foreground to match the text description while retaining the background of the source image.

Two discriminators D_{FG} and D_{BG} evaluate: (i) whether the synthesized image is real/fake, and (ii) whether the synthesized image is consistent with the condition. To this end, we employ two kinds of loss functions in training either D_{FG} or D_{BG}: an unconditional loss and a conditional loss. D_{FG} receives either the generated image or the ground-truth foreground image with the text description to evaluate real/fake (using the unconditional loss) and foreground-text matching (using the conditional loss). D_{BG}, on the other hand, receives either the generated image or the source image with the text description to evaluate real/fake (using the unconditional loss) and background preservation (using the conditional loss). We remark that the two discriminators do not share their parameters.

We train G, D_{FG} and D_{BG} simultaneously in a three-player minimax game using unconditional and conditional adversarial losses. This adversarial learning process enables our generator G to generate plausible images that match text

descriptions while preserving the background information of the source image.

3.2.1 Generator

The goal of the generator is to manipulate a part of a source image according to the given text. Therefore, it should be able to understand the image as well as the text. Like previous work [1,2], we employ two encoders to extract features from image and text respectively before feeding their (feature) combination into a decoder to generate manipulated image. Our main difference is that we combine image feature and text feature at high-level layer while retaining background information of the image using skip-connection. Our usage of different semantic levels of features is motivated by the findings of previous work [7,19].

Our generator G consists of an image encoder, a text encoder and a decoder.

The image encoder is a stack of three convolution layers that receives the source image size of $128 \times 128 \times 3$ to produce an image feature with the size of $32 \times 32 \times 512$ at the top. We adopt the pre-trained text encoder [31] for our text encoder and use text embedding augmentation [9] to produce a text feature with the size of 1×128. The channel of the text feature is replicated to the size of $32 \times 32 \times 128$ to be consistent with that of the image feature. We remark that when we replace pre-trained text encoder [31] with a trainable bidirectional RNN [36], our performance does not improve much. Hence, we do not use a trainable text encoder in our generator.

The image feature and the text feature are then concatenated to produce an image-text feature as the input of the decoder.

The decoder in our generator consists of one convolution layer, four residual blocks [37] and two deconvolution layers. The convolution layer reduces the channel of the image-text feature, and the four residual blocks enrich feature maps. On the other hand, the two deconvolution layers upscale the feature maps.

We remark that each of the convolution and deconvolution layers in the image encoder and the decoder is followed by a batch normalization (BN) layer [38] and a ReLU layer. The only exception is the last deconvolution layer in the decoder, where it uses the tanh activation to guarantee that the output range can be normalized to be $[0, 255]$ (in the test step). We note that we use images with the range $[-1, 1]$ in the training step.

To reflect the features at early layers weighting background information into a synthesized image, we employ the skip-connection from the image encoder to the decoder. More precisely, the first layer in the image encoder is connected to the last layer in the decoder, while the second layer in the image encoder is paired with the second last layer in the decoder.

3.2.2 Foreground discriminator

The foreground discriminator should discriminate the foreground of real images and that of generated images. We employ two losses in the foreground discriminator: unconditional loss (for real/fake discrimination) and conditional loss (for foreground-text matching evaluation). Following previous work [1,2,4,9,13,14], we design our foreground discriminator D_{FG} as a classification task that rewards high probability scores to real images and low ones to generated images in the adversarial learning phase. More precisely, the unconditional loss follows work [13,14], while the conditional loss is identical to text-adaptive discriminator [2].

Our D_{FG} consists of an image encoder and a text encoder.

The image encoder is a stack of seven convolution layers. Each of the first five convolution layers uses the filter size of 4×4, the reflection-padding size of 1×1 and the stride size of 2×2, producing 64, 128, 256, 512 and 512 output channels, respectively. These convolution layers encode an input to produce high-level semantic image features containing mostly foreground information [7,19]. These image features are then combined with word features obtained from the text encoder to produce conditional loss (see Sect. 3.3). On the other hand, the image features are continuously fed into the last two convolution layers, each of which is with the filter size of 1×1, and 4×4, respectively, no padding, the stride size of 1×1, outputting 256, 4 channels respectively. The output of the last convolution layer indicates how realistic the image input to D_{FG} is (unconditional loss). We remark that each of all convolution layers except for the last one is followed by a BN layer and a ReLU layer.

Like [2], our text encoder is a bidirectional RNN [36], outputting a vector with the size of 1×512 for each word in the text.

For each word in given text description, we use word-level local discriminator [2] (i.e., LD in Fig. 2) to compute a score $LD_{\boldsymbol{w}_i}$ by using a sigmoid function: $LD_{\boldsymbol{w}_i} = \sigma(\boldsymbol{W}_i(\boldsymbol{w}_i) \cdot \boldsymbol{v} + \boldsymbol{b}_i(\boldsymbol{w}_i))$, where $\boldsymbol{w}_i$ is the i-th word vector from the text encoder, $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ are the weight and the bias corresponding to $\boldsymbol{w}_i$, and $\boldsymbol{v}$ is the global average pooling over the image features. Then, the multiplication of all the scores obtained by the word-level discriminator, i.e., $\prod_{i=1}^{T} \left[LD_{\boldsymbol{w}_i}\right]^{\alpha_i}$ (T is the number of words in the text, α_i is the attention score of the i-th word across the text), is used as conditional loss of D_{FG}. The conditional loss indicates the degree of foreground-text matching. We remark that our conditional loss is identical with that used in the text-adaptive discriminator [2]. Please refer [2] for more details. Different from [2] which computes foreground-text matching using image features extracted from three convolution layers, we use image features from only one convolution layer. This is because lower-level layers may contain background information. We follow [4] to train conditional loss in D_{FG} (cf. Eq. 1).

3.2.3 Background discriminator

The background discriminator evaluates how real and generated images are different in the background. We, therefore, design the background discriminator as a verification task with a limited number of samples in each category. This is because each image in a dataset has a different background in general, and the number of samples with the identical (very similar) background is limited. To this end, we follow the idea of the Siamese network [39] because it shows the effectiveness of the verification task. Like our foreground discriminator, we also employ two losses for training the background discriminator: unconditional loss (for real/fake discrimination) and conditional loss (for background preservation verification). More precisely, the unconditional loss follows work [13,14], whereas the conditional loss is the contrastive loss [39].

Our D_{BG} consists of an image encoder, a text encoder and a Siamese network.

Our image encoder employs VGG-16 [34] as its backbone, and all fully connected layers are discarded. On the top of VGG-16, we add a convolution layer with the filter size of 4×4, no padding, the stride size of 1×1, outputting 4 channels. We remark that the weights of the image encoder are initialized using the weights of pre-trained VGG-16 [34] on ImageNet dataset [35]. We regard the output of the image encoder as the unconditional loss.

We design a text encoder for learning the text feature. This is because text description is useful to disentangle background and foreground information. Indeed, since the available number of samples with the same background is limited, we consider one image with two different text descriptions (foreground information) as two different images having the same background. Such images thus can be regarded as positive samples for the background verification task. Similarly to the foreground discriminator, the text encoder in our background discriminator is built upon a bidirectional RNN [36]. We take the last state from the text encoder with the size of 1×512 as the text feature. The text feature is then used in the Siamese network.

The Siamese network consists of four fully connected layers in which the first three layers are the shared-parameter layers, and the last one is the joint layer, producing 512, 100, 10 and 1 outputs, respectively. The Siamese network receives two input features (one from the source image with the text description and the other from the generated image with the text description) and passes them to the three shared-parameter layers separately before being jointly trained at the last layer. The output of the Siamese network is the conditional loss which indicates the difference in background between real and generated images.

In order to create the input of the Siamese network, we feed the input image into the image encoder to compute the mean and variance at the first four ReLU layers [19] and then concatenate them with the text feature extracted from the input text description using the text encoder. We remark that the size of the input is 1×1280 where the image feature is with the size of 1×768 and the text feature is with the size of 1×512.

We propose a novel training strategy for D_{BG}, which is based on the contrastive loss function [39] that fully uses a source image and a text description (Eq. 2).

3.3 Adversarial learning for Paired-D++ GAN

Training the generator G, and a pair of discriminators D_{FG} and D_{BG} becomes a three-player minimax game conditioned on images and text descriptions. Using positive/negative training samples, we first update the parameters of D_{FG} while fixing the parameters of D_{BG} and G. Then we update the parameters of D_{BG} while fixing the parameters of D_{FG} and G. Finally, we update the parameters of G while fixing the parameters of the two discriminators. We iterate this adversarial training to minimize each loss function separately.

For the adversarial training for Paired-D++ GAN, we use positive and negative samples whose definitions depend on D_{FG} and D_{BG}. Moreover, we perform both unconditional and conditional losses in training:

- **Unconditional loss** A positive sample of D_{FG} is a ground-truth image, while a positive sample of D_{BG} is a source image. A sample is negative for either D_{FG} or D_{BG} if it is generated from generator G.
- **Conditional loss** A positive sample of D_{FG} is a sample in which foreground is the ground-truth, and its text description is matching. A sample is negative if (1) foreground is the ground-truth, but its text description is mismatching, or (2) foreground is generated even if its text description is matching. A positive sample of D_{BG}, on the other hand, is the one where the background of the source image used in training the generator and discriminators for each iteration is the same regardless of whether text descriptions are matching or mismatching. A sample is negative if the background is generated even if the text descriptions match the foreground.

Let s be an image in a dataset and t be a text description. Then, we let g be an image in the dataset whose foreground is the ground-truth to t (t is thus a matching text description to g). We denote by $\bar{s}$ a randomly selected image (from the dataset) having different background from s, and by $\bar{t}$ a different text description from t (a mismatching text description to g). We also denote $\varphi(\cdot)$ as the text embedding augmentation [9]. Then, positive/negative samples of D_{FG} and D_{BG} can be classified as in Table 2.

Table 2 Types of input pairs used in the adversarial leaning process

	D_{FG} Unconditional	Conditional	D_{BG} Unconditional	Conditional
Positive	x_1={g}	x_3={g, t}	x_6={s}	x_8={(s,t), $(s,\bar{t})$}
Negative	x_2={$G(s,\varphi(t))$}	x_4={$g,\bar{t}$}, x_5= {$G(s,\varphi(t)),t$}	x_7={$G(s,\varphi(t))$}	x_9={$(G(s,\varphi(t)),t)$, $(G(\bar{s},\varphi(t)),t)$}, x_{10}={$(G(s,\varphi(t)),t)$, $(G(s,\varphi(\bar{t})),\bar{t})$}

Let $D(\cdot)$ denote the discriminators (D_{FG} and D_{BG}). At each iteration in training $D(\cdot)$, we randomly select all the types of samples in Table 2 from the training dataset and feed them one by one to $D(\cdot)$ to obtain the probability of whether the sample is positive or negative. We train the two discriminators to reward a high score to a positive sample and a low score to a negative sample. Through the training, we maximize the ability of $D(\cdot)$ to assign relevant scores to the samples. The loss functions for $D(\cdot)$ are defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{FG}} = &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log D_{\text{FG}}(x_1)]}_{\text{unconditional loss}} + \\ &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log(1 - D_{\text{FG}}(x_2))]}_{\text{unconditional loss}} + \\ &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log D_{\text{FG}}(x_3)]}_{\text{conditional loss}} + \\ &\underbrace{\frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\log(1 - D_{\text{FG}}(x_4))]}_{\text{conditional loss}} + \\ &\underbrace{\frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\log(1 - D_{\text{FG}}(x_5))]}_{\text{conditional loss}}, \end{aligned} \tag{1}$$

$$\begin{aligned}\mathcal{L}_{\text{BG}} = &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log D_{\text{BG}}(x_6)]}_{\text{unconditional loss}} + \\ &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log(1 - D_{\text{BG}}(x_7))]}_{\text{unconditional loss}} + \\ &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log D_{\text{BG}}(x_8)]}_{\text{conditional loss}} + \\ &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log(1 - D_{\text{BG}}(x_9))]}_{\text{conditional loss}}, \end{aligned} \tag{2}$$

where p_{data} denotes the all the training data and $\mathbb{E}_{p_{\text{data}}}$ means the expectation over p_{data}. Each term in Eqs. 1 and 2 corresponds to the type of samples: $\log(D(\cdot))$ for positive samples and $\log(1 - D(\cdot))$ for negative samples.

Since our adversarial learning process is a three-player minimax game, we also train the generator G in which we minimize the terms of $\log(1 - D(\cdot))$ in Eqs. 1 and 2. In practice, however, maximizing $\log(D(\cdot))$ is known to be better than minimizing $\log(1 - D(\cdot))$ in training G [8]. We also introduce the reconstruction loss to keep the structure of the input source image. Now the loss function for G is:

Algorithm 1 Training Paired-D++ GAN

Require: dataset $data$, Paired-D++ GAN with generator G, foreground discriminator D_{FG} and background discriminator D_{BG}, epoch to train T

Ensure: optimized G^*

for $t \leftarrow 1$ to T **do**
  Get a batch $\{s, t, g, \bar{s}, \bar{t}\}$ from $data$
  $x_i \leftarrow$ GENERATESAMPLES$(G, s, t, g, \bar{s}, \bar{t})$ where $i = 1, .., 10$ (Table 2)
  $\mathcal{L}_{\text{FG}} \leftarrow$ COMPUTEFGLOSS$(D_{\text{FG}}, x_1, x_2\ x_3, x_4, x_5)$ (Eq. 1)
  $D_{\text{FG}} \leftarrow$ UPDATEFOREGROUNDDISCRIMINATOR$(D_{\text{FG}}, \mathcal{L}_{\text{FG}})$
  $\mathcal{L}_{\text{BG}} \leftarrow$ COMPUTEBGLOSS$(D_{\text{BG}}, x_6, x_7\ x_8, x_9, x_{10})$ (Eq. 2)
  $D_{\text{BG}} \leftarrow$ UPDATEBACKGROUNDDISCRIMINATOR$(D_{\text{BG}}, \mathcal{L}_{\text{BG}})$
  $\mathcal{L}_{\text{G}} \leftarrow$ COMPUTEGLOSS$(D_{\text{FG}}, D_{\text{BG}}, x_2, x_5, x_7, x_{10})$ (Eq 3)
  $G \leftarrow$UPDATEGENERATOR$(G, \mathcal{L}_{\text{G}})$
end for
return G^*

$$\begin{aligned}\mathcal{L}_{\text{G}} = &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log(D_{\text{FG}}(x_2))]}_{\text{unconditional loss}} + \\ &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log(D_{\text{FG}}(G(x_5))]}_{\text{conditional loss}} + \\ &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log(D_{\text{BG}}(x_7))]}_{\text{unconditional loss}} + \\ &\underbrace{\mathbb{E}_{p_{\text{data}}}[\log(D_{\text{BG}}(x_{10}))]}_{\text{conditional loss}} + \\ &\lambda\mathbb{E}_{p_{\text{data}}}\|s - G(s,\varphi(t))\|_2, \end{aligned} \tag{3}$$

where λ is the hyperparameter, and $\|.\|_2$ is the Euclidean distance. To train G, we randomly select an image s, and two text descriptions t and $\bar{t}$ to generate the synthesized images. We then feed them to the D_{FG} and D_{BG} to receive feedback signals for updating parameters of G. We remark that since our aim is not to reconstruct the source image, λ can be small (we set $\lambda = 0.0002$ in our experiments).

As discussed in [4], training D_{FG} with matching and mismatching text descriptions enables D_{FG} to feedback stronger image–text matching signals, allowing G to generate plausible images that match text descriptions. Our usage of a pair of an image and a text description in training D_{BG}, on the other hand, enables D_{BG} to generate stronger signals as well, leading to the capability of G of retaining background information (though at the beginning, D_{BG} spends more time to

verify background, D_{BG} gradually need not concern foreground thanks to text descriptions and has the ability of easily judging whether the image is real or generated). In addition, the usage of unconditional loss in both discriminators enables our generator to generate more realistic images. Accordingly, the above adversarial learning brings Paired-D++ GAN the capability of generating realistic images that match text descriptions in the foreground and precisely retain the background of source images. Our adversarial learning procedure is illustrated in Algorithm 1.

4 Experimental setup

4.1 Dataset and compared methods

Dataset. We used the Caltech-200 bird dataset [16] and the Oxford-102 flower dataset [17]. The Caltech-200 bird dataset contains 11,788 images belonging to 200 different bird classes. The Oxford-102 flower dataset has 8189 images with 102 classes of the flower. Each image in the datasets has 10 captions collected by [31]. Following previous work [1,2,4], we split the Caltech-200 dataset into 150 training classes and 50 testing classes, and the Oxford-102 dataset into 82 training classes and 20 testing classes. We remark that we resized the images used in our experiments to ones with 128×128.

Compared methods. We compared our method with other text-guided image manipulation methods, including SISGAN [1], TAGAN [2] and ManiGAN [3]. We also compared our method with LR-GAN [18] that generates image foreground and background separately and recursively from input text descriptions (we chose this though the task is different because it generates realistic images). In order to compare our method with state-of-the-art in text-to-image synthesis, we employed AttnGAN [14]. We carefully adapted compared methods for comparison based on the public implementations provided by the authors. For SISGAN [1], we used the re-implementation by Seonghyeon[1] (as recommended by the authors of SISGAN [1]). For TAGAN [2], we used source code and pre-trained models provided by the authors[2]. For ManiGAN [3], we employed the authors' pre-trained model on Caltech-200 and trained a new model on Oxford-102 using provided source code[3]. For LR-GAN [18], we used the publicly available source codes with the parameters recommended by the authors[4]. We remark that we used the combination of a noise vector and a text feature [4] as input for LR-GAN [18]. For AttnGAN [14], we used a pre-trained model on Caltech-200 and adapted their provided implementation to train a new model on Oxford-102[5].

4.2 Implementation and training details

We implemented our model in PyTorch. We adopted the pre-trained text encoder [31] without any fine-tuning in our generator. Like [1], we also used the image augmentation techniques (i.e., flipping, rotating, zooming and cropping). Note that these augmentation techniques are also employed in the compared methods. We conducted all the experiments using a PC with a CPU 6-cores Xeon 3.7 GHz, 64 GB of RAM and a GTX1080 Titan GPU (11GB of VRAM).

We optimized the adaptive loss functions (Sect. 3.3) using Adam optimizer [40] with the learning rate of 2×10^{-3}, the learning rate decay of 0.5 performed every 100 epochs, the momentum $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and the division from zero parameter $\epsilon = 10^{-8}$. We did not use the weight decay. We trained our model with a batch size of 48 for 600 epochs.

4.3 Evaluation metrics

We use the inception score (IS) [41] and Fréchet inception distance (FID) [42] to evaluate the overall quality of synthesized images. We also use two metrics, foreground score (FGS) and background score (BGS), for evaluating foreground and background of synthesized images separately.

IS is widely used for the generative model evaluation through the output of the Inception-v3 network [43]:

$$IS(G) \approx \exp\left(\frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(p(y|\hat{x}^{(i)})||\hat{p}(y))\right), \tag{4}$$

where $\hat{x}$ is a synthesized image by the generator G, N is the number of generated images, D_{KL} is the Kullback–Leibler divergence, y indicates an instance of all classes given in the dataset, $p(y|\hat{x})$ is the conditional class distribution, and $\hat{p}(y) = \frac{1}{N}\sum_{i=1}^{N} p(y|\hat{x}^{(i)})$ is the empirical marginal-class distribution.

FID measures the similarity of real and generated data using the Fréchet distance [44] between their activation distributions extracted from the $pool3$ layer of the Inception-v3 network [43]:

$$FID = \left\|\mu_{\mathrm{real}} - \mu_{\mathrm{gen}}\right\|^2 + \mathrm{tr}(\Sigma_{\mathrm{real}} + \Sigma_{\mathrm{gen}} - 2(\Sigma_{\mathrm{real}}\Sigma_{\mathrm{gen}})^{1/2}), \tag{5}$$

[1] https://github.com/woozzu/dong_iccv_2017.

[2] https://github.com/woozzu/tagan.

[3] https://github.com/mrlibw/ManiGAN.

[4] https://github.com/jwyang/lr-gan.pytorch.

[5] https://github.com/taoxugit/AttnGAN.

where $\mu_{\text{real}}, \mu_{\text{gen}}$ Σ_{real} and Σ_{gen} are means and covariance matrices of the activation distributions of real and generated data, respectively, and $\text{tr}(\cdot)$ is the trace.

We employ the visual-text shared-space [31] and compute the matching (cosine similarity) between text descriptions and foreground for the foreground evaluation:

$$FGS = \frac{f_{\text{im}} \cdot f_{\text{txt}}}{\| f_{\text{im}} \| \| f_{\text{txt}}, \|}, \tag{6}$$

where f_{im} and f_{txt} are the features from the image encoder and the text encoder.

For background evaluation, we use

$$BGS = \frac{\left\| \hat{x} \odot \overline{x_{\text{seg}}} - x \odot \overline{x_{\text{seg}}} \right\|}{N}, \tag{7}$$

where x is the source image and $\odot$ is the element-wise multiplication. $\overline{x_{\text{seg}}}$ is the inverse map of x_{seg} where x_{seg} is the binary segmentation map of x provided from the dataset and N is the number of pixel of the background. We use $\overline{x_{\text{seg}}}$ to mask foreground for x and $\hat{x}$.

5 Experimental results

We validated the effectiveness of Paired-D++ GAN by qualitative evaluation, quantitative evaluation and detailed analysis. Since the user study is expensive and its results are somewhat subjective in some sense, we instead investigated our method using quantitative scores to quantify the effectiveness of image manipulation with text objectively.

5.1 Qualitative evaluation

Figures 3 and 4 illustrate examples of the results obtained by our method and text-guided image manipulation SOTAs (SISGAN [1], TAGAN [2] and ManiGAN [3]) on Caltech-200 bird dataset [16] and Oxford-102 flower dataset [17]. They show that the synthesized images by our method match the text descriptions more precisely than other methods while successfully retaining the background of the source image.

On the Caltech-200 dataset (Fig. 3), we see that the results by SISGAN [1] are capable of (not always though) matching the foreground and the text descriptions; they, however,

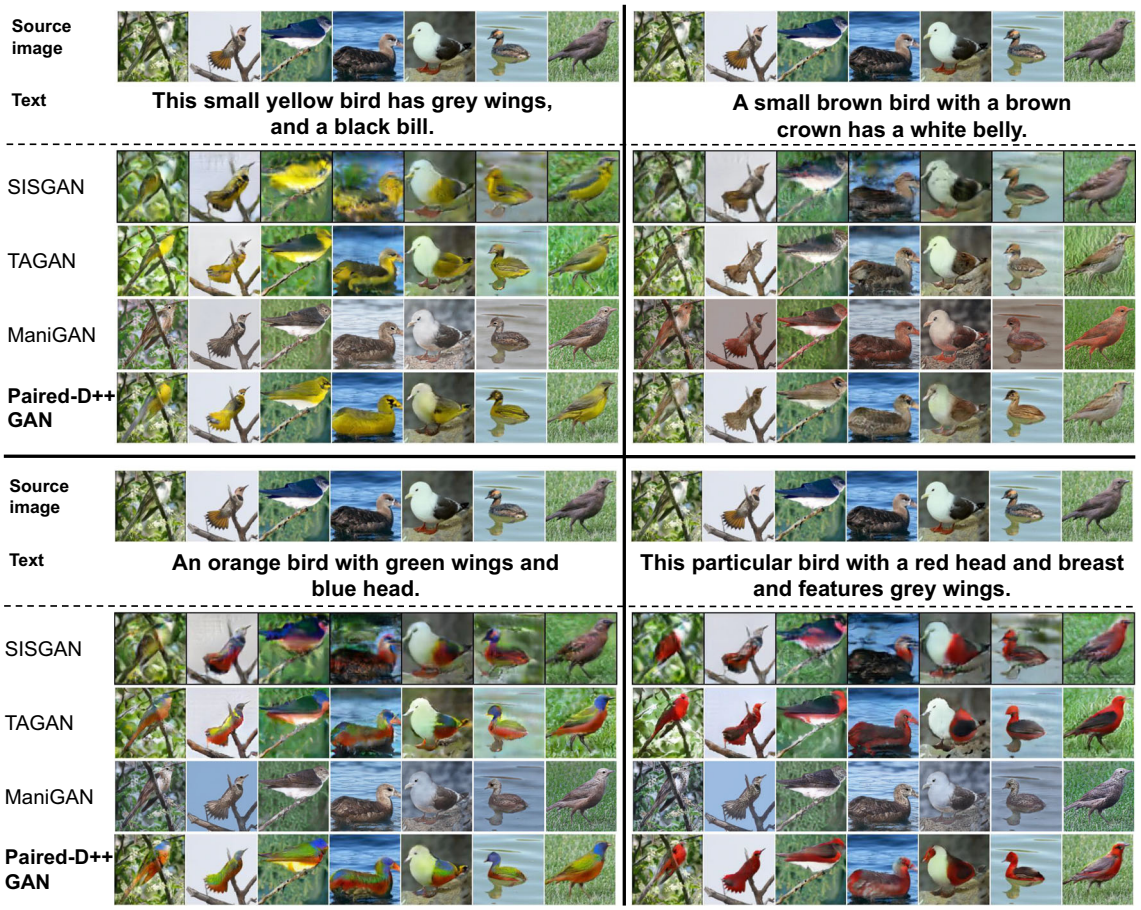

Fig. 3 Visual comparison on the Caltech-200 bird dataset [16]. For each block: source images, text description, results by SISGAN [1], TAGAN [2], ManiGAN [3] and ours. Each image is generated using a source image and a text description

Fig. 4 Visual comparison on Oxford-102 flower dataset [17]. For each block: source images, text description, results by SISGAN [1], TAGAN [2], ManiGAN [3] and ours. Each image is generated using a source image and a text description

do not preserve background well. The results obtained by TAGAN [2] are reasonable, but they are not successful in background preservation. In most cases, though the results by ManiGAN [3] are more realistic, we observe that they cannot understand the text description well and the background in generated image is different from that in the source image. Our method, on the other hand, is clearer in foreground and background.

On the Oxford-102 dataset (Fig. 4), we see that SISGAN [1] have some failures in synthesizing images. We also observe that ManiGAN [3] again cannot handle the task properly because of a lack of text understanding. The results by TAGAN [2] and ours are comparable, but our results are better to some extent. In particular, our method not only matches color attributes described in text description but also is able to handle quantitative attributes while TAGAN [2] is not the case (see the first sample in Fig. 4). This may be because we employ two discriminators separately so that the foreground discriminator has more chance to deal with the text description in more detail. We remark that the number of epochs in training time in our method and that of TAGAN [2] are the same.

5.2 Quantitative evaluation

For the quantitative evaluation, we computed IS, FID, FGS and BGS of the synthesized images, which are shown in Table 3. To compute IS, we iterated 10 times the experiment that we synthesized 8000 images and computed the average and the standard deviation of the resulting scores, as recommended in [41]. To compute FID, we generate 5000 images for each dataset. To compute FGS and BGS, we iterated 5 times the experiment that we synthesized 600 images and computed the average of the resulting scores. Note that we cannot compute BGS for LR-GAN [18] and AttnGAN [14] because these models are set to generate a new image rather than to manipulate a source image (i.e., no ground-truth background). We also remark that we used the visual-text shared-space model [31] pre-trained on the Caltech-200 (or Oxford-102) dataset to compute features for FGS.

Table 3 shows that our method (almost) achieves the best performance in all the metrics, meaning that the images synthesized by our method are superior not only in the overall quality (IS, FID) but also in foreground-text matching (FGS) and in background preservation (BGS). The outper-

Table 3 Quantitative comparison using IS (larger is better), FID (smaller is better), FGS (larger is better) and BGS (smaller is better)

Dataset	Caltech-200				Oxford-102			
Metric	$IS \Uparrow$	$FID \Downarrow$	$FGS \Uparrow$	$BGS \Downarrow$	$IS \Uparrow$	$FID \Downarrow$	$FGS \Uparrow$	$BGS \Downarrow$
Paired-D++ GAN	*7.78±0.35*	*23.76*	**0.137**	**0.0975**	**5.46±0.17**	**16.93**	**0.143**	**0.0537**
SISGAN [1]	5.56±0.14	67.24	0.052	0.1512	4.03±0.11	81.38	0.041	0.1102
TAGAN [2]	7.29±0.21	34.49	*0.095*	*0.1291*	*5.39±0.27*	55.29	*0.108*	*0.1036*
ManiGAN [3]	(**8.47**)	38.27	0.046	0.2367	4.36±0.18	82.32	0.082	0.1729
LR-GAN [18]	5.92±1.04	89.10	0.032	–	3.49±0.04	103.11	0.027	–
AttnGAN [14]	(4.36±0.03)	**22.37**	0.091	–	4.73±0.12	*37.44*	0.068	–

The best results are given in Bold, the second best results are given in Italic. Scores in parentheses indicate those reported in original papers

formance of our method against text-guide image manipulation methods (SISGAN [1], TAGAN [2], ManiGAN [3]) in all the metrics confirms that evaluating foreground and background separately in the training phase is effective. Compared to LR-GAN [18], we see that our methods, SISGAN [1], TAGAN [2] and ManiGAN [3], generate the more realistic image, suggesting that for semantic image synthesis, generating foreground and background at the same time is better than separately and recursively generating foreground and background. In general, text-guided image manipulation methods (ours, SISGAN [1], TAGAN [2], ManiGAN [3]) are better in generating realistic images and in foreground-text matching than AttnGAN [14]. This emphasizes the advantages of text-guide image manipulation methods over directly generating images from text.

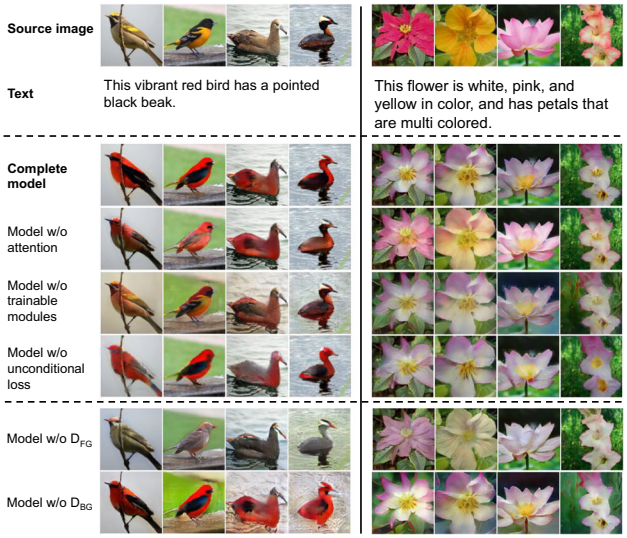

Fig. 5 Examples of results obtained by ablation models

5.3 More detailed analysis

5.3.1 Ablation study

In order to investigate the contribution of each component in the performance, we compare our complete model with several ablation models (see Fig. 5 and Table 4). We classify all the ablation models into two: (A) our complete model, model w/o attention, model w/o trainable modules, model w/o unconditional loss (these employ two discriminators) and (B) model w/o D_{FG} and model w/o D_{BG} (these employ one discriminator only). The details of the ablation models are as follows. The model w/o attention denotes the replacement of the word-level local discriminator (in the foreground discriminator) by just concatenating the image feature and text feature to compute the foreground-text matching. The model w/o trainable modules denotes the replacement of the trainable image encoder (in the background discriminator) and the text encoder (in both the foreground and background discriminators) by the fix VGG-16 [34] pre-trained on ImageNet [35] and the fix pre-trained text encoder [31] respectively (meaning that the parameters of these modules are not updated during training time). The model w/o unconditional loss denotes the dropping of the unconditional loss term in the loss function (Sect, 3.3). The model w/o D_{FG} and the model w/o D_{BG} denote the models dropping the foreground discriminator and the background discriminator, respectively.

Qualitative evaluation We show some examples obtained by all the ablation models in Fig. 5. Overall, we visually observe that the models in (A) generate better images than the models in (B). In particular, the models in (A) are able to generate plausible images in terms of foreground-text matching and background preservation. The models in (B), on the other hand, are manageable in either foreground or background but not in both.

Among the models in (A), we see that foreground-text matching in the model w/o attention is not very impressive. The model w/o trainable modules cannot manipulate the foreground well while the background is not successfully preserved. We also see that the generated images by the model w/o unconditional loss are reasonable but less realistic. Our complete model, on the other hand, is better in both foreground-text matching and background preservation.

Table 4 Quantitative comparisons on ablation models (bold is better)

Dataset	Caltech-200				Oxford-102			
Metric	$IS \Uparrow$	$FID \Downarrow$	$FGS \Uparrow$	$BGS \Downarrow$	$IS \Uparrow$	$FID \Downarrow$	$FGS \Uparrow$	$BGS \Downarrow$
Complete model ($D_{\mathrm{FG}} + D_{\mathrm{BG}}$)	**7.78±0.35**	**23.76**	**0.137**	**0.0975**	**5.46±0.17**	**16.93**	**0.143**	**0.0537**
Model w/o attention	6.56±0.26	28.27	0.093	0.1036	4.89±0.19	38.29	0.121	0.0617
Model w/o trainable modules	6.43±0.17	26.28	0.085	0.1206	4.71±0.17	41.66	0.108	0.0702
Model w/o unconditional loss	6.19±0.13	52.49	0.102	0.1184	4.82±0.23	53.27	0.115	0.0643
Model w/o D_{FG}	5.83±0.19	47.26	0.062	0.1064	4.87±0.23	49.16	0.057	0.0718
Model w/o D_{BG}	6.37±0.31	39.65	0.081	0.1372	5.03±0.26	47.72	0.091	0.1035

When we drop either the foreground discriminator or the background discriminator, the models are unable to work properly. In particular, the model w/o D_{FG} cannot deal with the text description well, while the model w/o D_{BG} cannot retain the background. This means that focusing solely on either the foreground or the background is unable to gain the performance.

Quantitative evaluation To quantitatively evaluate the ablation models, we measure IS, FID, FGS and BGS as shown in Table 4. We note that the settings of this quantitative evaluation are the same as the experiments in Sect. 5.2.

From the fourth and fifth rows in Table 4 we see that the scores obtained by either the model w/o attention or the model w/o trainable modules are worse than those by our complete model. These observations indicate the necessity of the word-level discriminator and the trainable modules. Furthermore, we may regard the model w/o trainable modules as an incremental extension of the work reported in [19]. Through comparing the model w/o trainable modules with our complete model, we may conclude that our main extensions in this paper are sufficiently effective.

Next, we evaluate the plausibility of using the unconditional loss. The sixth row in Table 4 shows that the performance loss of overall quality (IS, FID) is worse than that of FGS and BGS. We may conclude that the usage of the unconditional loss indeed works for improving the quality of generated images while the conditional loss does for foreground-text matching and background preservation. We remark that the model w/o conditional loss is not applicable because our model is a kind of conditional GAN.

Finally, from the last two rows in Table 4, we see that either the model w/o D_{FG} or the model w/o D_{BG} drops the performance of our method drastically. This confirms the necessity of both the discriminators (D_{FG} and D_{BG}) in our method. We also see that the model w/o D_{FG} achieves worse FGS and better BGS than the model w/o D_{BG}, and vice versa. These observations indicate that D_{FG} and D_{BG} properly work for the foreground and the background each.

5.3.2 Interpolation results

We demonstrated the smooth interpolation between the source image and the target image. Fig. 6 shows synthesized images obtained by the linear interpolation between the source and the target images. In Fig. 6 (first two samples), we interpolated two source images with a fixed text description. In contrast, we kept the source image fixed while changing text descriptions in Fig. 6 (last two samples). These results indicate that our method has the capability of independent

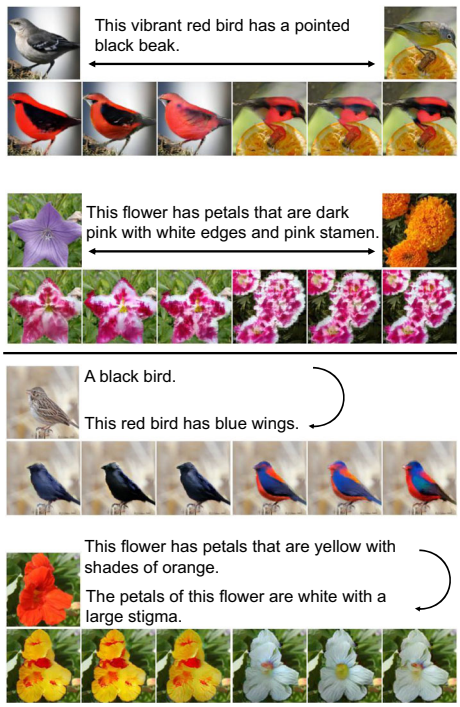

Fig. 6 Examples of interpolation results. Two first samples: interpolation between two source images with the same target text description. Two last samples: interpolation between two target text descriptions for the same source image

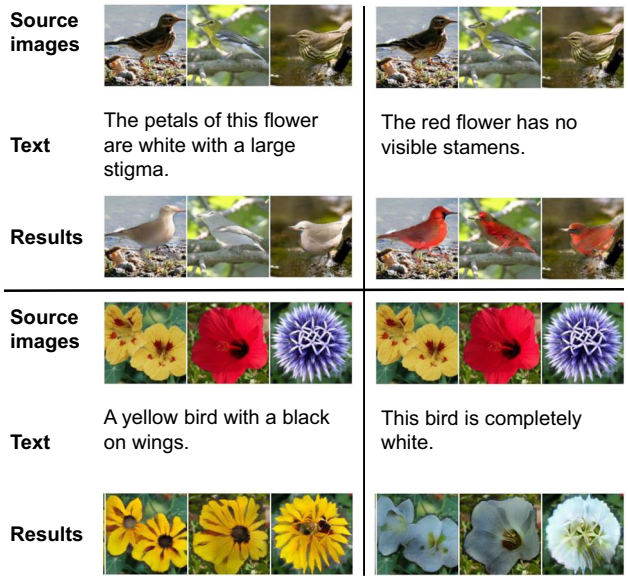

Fig. 7 Zero-shot results from a source image and text descriptions that are not related to each other, showing the effectiveness of foreground and background discriminators

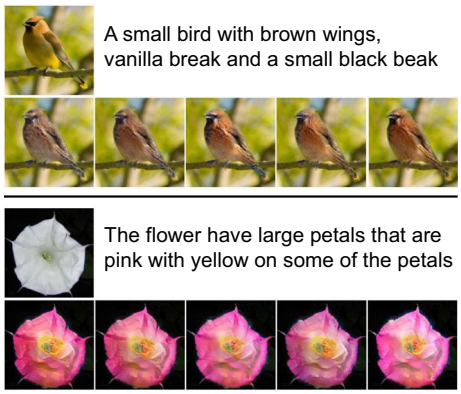

Fig. 8 Zero-shot results from the same source image and text descriptions, showing variety of foregrounds

interpolation between source images and text descriptions. We remark that our method preserves the background well regardless of the interpolation.

Figure 7 shows generated images obtained using source images from the Caltech-200 [16] dataset with text descriptions from the Oxford-102 [17] dataset (not used in the training phase), and vice versa. Fig. 7 shows that our model retains the background of source images and changes only the foreground to match text descriptions (e.g., color) even if they are not used in training (regardless of untrained text descriptions). This illustrates the flexible capability of our model to disentangle the foreground and the background.

We also show in Fig. 8 the effectiveness of text embedding augmentation [9] in our method to synthesize various images using the same source image and text descriptions.

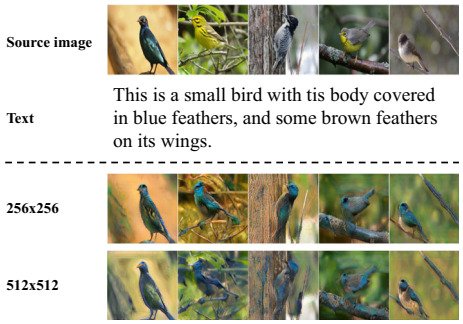

Fig. 9 Examples of failure cases when synthesizing high resolution images

5.4 Limitations

We visually observe that our method ignores the shape of the foreground in the given text. For instance, considering the given text "This **small** yellow bird has gray wings, and a black bill" in Fig. 3, we see that our method is unable to change the shape of the birds to the **small** ones. This is because our method automatically detects foreground and directly edits on the detected foreground rather than generating a new foreground. Moreover, we do not use any extra shape-guide information to instruct the network. We remark that other text-guide image manipulation methods (SISGAN [1], TAGAN [2] and ManiGAN [3]) have the same limitations. Adding extra shape-guide information to manipulate the foreground more precisely is left for future work.

Even we do not aim at generating high-resolution images, we experimentally explore the capability of our model in dealing with higher resolution. To this end, we employ our trained models on the source images with the size of 256×256 and 512×512. As shown in Fig. 9, we see that our model is capable of changing a part of the foreground corresponding to a given text, but it cannot preserve the background well. Developing a method that can handle source images of any size is also left for future work.

6 Conclusion

We proposed Paired-D++ GAN conditioned on both text descriptions and images for image manipulation with text. Our Paired-D++ GAN consists of one generator and two discriminators with different architectures where one discriminator is used for judging the foreground, and the other is for judging the background. Our method is able to synthesize a realistic image where an input text description matches its corresponding part (foreground) of the image while preserving the background of a given source image. Experimental results on the Caltech-200 and the Oxford-102 datasets demonstrate the efficacy of our method.

References

1. Dong, H., Yu, S., Wu, C., Guo, Y.: Semantic image synthesis via adversarial learning. In: ICCV, (2017)
2. Nam, S., Kim, Y., Kim, S.J.: Manipulating images with natural language. In: NeurIPS, Text-Adaptive Generative Adversarial Networks (2018)
3. Li, B., Qi, X., Lukasiewicz, T., Philip H.S.T.: Text-guided image manipulation. In: CVPR, Manigan (2020)
4. Reed, S., Akata, Z., Xinchen Y., Logeswaran L., Bernt S., Honglak L.: Generative adversarial text-to-image synthesis. In: ICML (2016)
5. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: ECCV (2016)
6. Efros, A.A., Freeman, W.T.; Image quilting for texture synthesis and transfer, In: SIGGRAPH (2001)
7. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR (2016)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
9. 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 (2017)
10. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017)
11. Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks. In: ECCV (2016)
12. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017)
13. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: Stackgan++: Realistic image synthesis with stacked generative adversarial networks. In: TPAMI (2019)
14. 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* (2018)
15. Nguyen, T., Le, T., Vu, H., Phung, D.: Dual discriminator generative adversarial nets. In: NIPS (2017)
16. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology (2011)
17. Nilsback, M-E., Zisserman, A.: Automated flower classification over a large number of classes. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (2008)
18. Yang, J., Kannan, A., Batra, D., Parikh, D.: Lr-gan: Layered recursive generative adversarial networks for image generation. In: ICLR (2017)
19. Vo, D.M., Sugimoto, A.: Paired-d gan for semantic image synthesis. In: ACCV (2018)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes, In: ICLR (2014)
21. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models, In: ICML (2014)
22. Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: ICML (2016)
23. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A.: Conditional image generation with pixelcnn decoders. In: NIPS (2016)
24. Jiang, Y., Chang, S., Wang, Z.: Two transformers can make one strong gan. In: NeurIPS, Transgan (2021)
25. Hudson, D.A., Zitnick, C.L.: Generative adversarial transformers. In: ICML (2021)
26. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Interpretable representation learning by information maximizing generative adversarial nets. In: NIPS (2016)
27. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. In: ICLR (2017)
28. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation, In: NIPS (2017)
29. Perarnau, G., Van De Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional GANs for image editing. In: NIPS Workshop on Adversarial Training (2016)
30. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: ECCV (2016)
31. Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: CVPR (2016)
32. Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. In: NIPS (2016)
33. Liu, X., Lin, Z., Zhang, J., Zhao, H., Tran, Q., Wang, X., Hongsheng L.: Open-domain image manipulation with open-vocabulary instructions. In: ECCV, Open-edit (2020)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.-F.: Imagenet large scale visual recognition challenge. In: IJCV (2015)
36. Schuster, M., Paliwal, K. K.: Bidirectional recurrent neural networks. In: IEEE Transactions on Signal Processing (1997)
37. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
38. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
39. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR (2005)
40. Diederik, P.K., Jimmy B.A: A method for stochastic optimization. In: ICLR (2015)
41. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X.: Improved techniques for training gans. In: NIPS (2016)
42. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS (2017)
43. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
44. Dowson, D.C., Landau, B.V.: The fréchet distance between multivariate normal distributions. J. Multiv. Anal. **12**(3), 450–455 (1982)