Chapter 2 proposed the Dual-Attention Generative-Adversarial Network (DTGAN), a single-stage text-to-image GAN model, to deal with the problems of a more complicated multi-stage modular architecture. However, DTGAN, as well as several other single-stage methods from literature, suffer from the lack-of-diversity issue, yielding similar outputs given a single textual sequence. To this end, we present an efficient and effective single-stage framework (DiverGAN) to generate diverse, perceptually plausible and semantically consistent images according to a natural-language description. DiverGAN adopts two novel word-level attention modules, i.e., a channel-attention module and a pixel-attention module, which model the importance of each word in the given sentence while allowing the network to assign larger weights to the significant channels and pixels semantically aligning with the salient words. After that, a dual-residual structure is developed to preserve more original visual features while allowing for deeper networks, resulting in faster convergence speed and more vivid details. Furthermore, we propose to plug a fully-connected layer into the pipeline to address the lack-of-diversity problem, since we observe that a dense layer will remarkably enhance the generative capability of the network, balancing the trade-off between a low-dimensional random latent code contributing to variants and modulation modules that use high-dimensional and textual contexts to strengthen feature maps. Inserting a linear layer after the second residual block achieves the best variety and quality. Both qualitative and quantitative results on benchmark data sets suggest the effectiveness of DiverGAN for realizing diversity, without harming quality and semantic consistency.

This chapter is based on


3.1 INTRODUCTION

To address the issues of a multi-stage modular architecture \cite{55, 83, 118, 122, 126, 127, 140}, single-stage text-to-image synthesis methods have been recently studied in DTGAN proposed in Chapter 2 and DF-GAN \cite{105}, merely leveraging a generator/discriminator pair to produce photo-realistic images which semantically align with given textual descriptions. Thanks to the power of deep generative neural networks, the feature fusion and generation process can be integrated into one single-stage procedure.

Although current single-stage text-to-image generation models achieve structural simplicity and superior performance, there still exist two significant issues. Firstly, these approaches suffer from the mode-collapse problem \cite{4}, in which the generator derails and synthesizes an inappropriate image for the text input. In less unfortunate cases of mode collapse, the cGAN generates a set of very similar output images, conditioned on the single natural-language description at the input. As shown in Fig. 3.1, DF-GAN \cite{105} fails to produce diverse samples, although noise is present on the input. This serious obstacle will drastically degrade the diversity of generated images, limiting their applicability in practice. For instance in data augmentation, robust classification can only be achieved if a wide range of shapes were generated. Secondly, existing single-stage text-to-image approaches modulate the feature map by just adopting a global sentence vector, which lacks detailed and fine-grained information at the word level and prevents the model from manipulating parts of the image generation according to natural-language descriptions and qualifications \cite{55, 118}.

One possible explanation for the lack-of-diversity issue is that the single-stage generator focuses more on the textual-context information that is high-dimensional and structured but ignores the random latent code which is responsible for variations \cite{68}. Considering the fact that single-stage text-to-image methods usually utilize modulation modules to reinforce the visual feature map for each scale in order to ensure image quality and semantic consistency, the conditional contexts are likely to provide stronger control over the output image than the latent code, and thus, the generator yields nearly identical instances from a single text description.

In this chapter, we propose to develop an efficient and effective single-stage framework (DiverGAN) for yielding diverse and visually plausible instances that correspond well to the textual contexts. DiverGAN consists of three novel components discussed as follows.

Firstly, we design two new types of word-level attention modules, i.e., a channel-attention module (CAM) and a pixel-attention module (PAM). These modules capture the semantic affinities between word-context vectors and feature maps in the channels and in the (2D) spatial dimensions. CAM and PAM not only guide the model to focus more on the crucial channels and pixels that are semantically
correlated with the prominent words (e.g., adjectives and nouns) in the given textual description, but also alleviate the impact of semantically irrelevant and redundant information. More importantly, with CAM and PAM, DiverGAN can effectively disentangle the attributes of the text description while also accurately controlling the regions of synthesized images.

Secondly, we present a dual-residual block constructed with two residual modules, each of which contains convolutional layers, CAdaILN discussed in Section 2.4.4, exploiting a ReLU activation function followed by a modulation module. The dual-residual block not only benefits the cGAN convergence by retaining more original visual features, but also efficiently improves network capacity, resulting in high-quality images, with more details.

The text-to-image pipeline built upon the above two ingredients and CAdaILN is enough to be capable of producing perceptually realistic images semantically matching with the natural-language descriptions but still suffers from the lack-of-diversity problem.

Therefore, as the third remedy we propose, on the basis of a variety of experiments, to use a linear layer which significantly boosts the generative ability of the network. This dense layer forces the generator to explore a wider range of modes in the distribution of original data. In other words, plugging a linear layer into the single-stage architecture will improve the control of a random latent code over the visual feature map, balancing the trade-off between a random latent code contributing to diversity and modulation modules that modulate the feature map based on word-context vectors. Simultaneously, experimental results will indicate that inserting a linear layer after the second dual-residual block of the architecture achieves the best performance on visual quality and image diversity. As illustrated in Fig. 3.1, DiverGAN equipped with a fully-connected layer is capable of generating birds with different visual appearances of footholds, background colors, orientations and shapes on the CUB bird data set [108].

**Figure 3.1:** Diversity comparison between DF-GAN [105] and DiverGAN based on a single text description at the input. DF-GAN (Left) tends to suppress the random latent code, synthesizing slight variations of a bird. We propose DiverGAN (Right), an efficient and effective framework that is able to avoid the lack-of-diversity issue and yield diverse and high-quality pictures.
We perform comprehensive experiments on three benchmark data sets, i.e., Oxford-102 [75], CUB bird [108] and MSCOCO [63]. Both quantitative and qualitative results demonstrate that the proposed DiverGAN has the capacity to synthesize impressively better images than current single-stage and multi-stage text-to-image models, e.g., StackGAN++ [127], MSGAN [68], SDGAN [122], DMGAN [140], DF-GAN [105], DTGAN. The contributions of this work can be summarized as follows:

- We establish a novel single-stage architecture for the task of text-to-image synthesis. Our framework mitigates the lack-of-diversity issue, producing diverse and high-resolution samples that are semantically correlated with textural descriptions.
- Two new types of word-level attention modules are designed to reinforce visual feature maps with word-context vectors.
- A dual-residual structure is developed to further improve image quality and stabilize the learning of cGANs.
- Finally, as an essential step, we introduce a flattening, fully-connected layer into the single-stage pipeline to deal with the lack-of-diversity issue in text-to-image generation.

The remainder of this chapter is organized as follows. In Section 3.2, we review related work. In Section 3.3, we describe the proposed DiverGAN in detail. Section 3.4 reports the experimental results and this chapter is summarized in Section 3.5.

### 3.2 RELATED WORK

#### 3.2.1 Avoiding mode collapse

Mode collapse is a common but crucial obstacle existing in a generative adversarial network (GAN) [30]. What happens in mode collapse is that the generator network of a GAN sticks to a particular output pattern without further improvement. Mode collapse can present itself in different ways, as will be explained in this section. There are two major directions to mitigate the lack-of-diversity and mode-collapse issues in a GAN. Some publications suggest adapting the objective strategy to optimize the discriminator and the training process, e.g., by employing a mini-batch discriminator [89], spectral regularization [65] or unrolled optimization [70]. Metz et al. [70] proposed Unrolled GAN introducing a new objective to make the generator be updated through an unrolled optimization of the discriminator. Salimans et al. [89] developed a mini-batch discriminator to check multiple output samples in a mini-batch and a novel generator objective to tackle the overtraining on the discriminator. Liu et al. [65] recently proposed the spectral regularization for handling mode collapse. They suggest that the
3.2 Related Work

Optimization of the discriminator is associated with the spectral distributions of the weight matrix.

Still other papers suggest the use of auxiliary structures into a GAN to encourage the generator to explore more modes of the distribution of true data, including an autoencoder [7], [13], [52], [135], multiple generators [15], [28], a conditional augmentation technique [126], etcetera. Che et al. [13] presented a mode regularized GAN (ModeGAN) incorporating an autoencoder into a standard GAN to avoid the mode missing problem. Motivated by ModeGAN, VEEGAN [100] built a reconstruction net to map the distribution of true data to the latent codes so that the generator network is able to synthesize all the data modes. EBGAN [135], BEGAN [7] and VAEGAN [52] also made efforts towards combining a GAN with an autoencoder. To combat the mode-collapse issue, Ghosh et al. [28] provided a multi-agent GAN architecture (MAD-GAN), where multiple generators are employed to capture different modes and one discriminator is designed to identify generated samples.

There are also several studies focusing on mitigation of the lack-of-diversity problem in a cGAN. What mode collapse and lack-of-diversity have in common is that the generator sticks to a subset of possible patterns. In the lack-of-diversity problem, however, the output image appearance is not drastically distorted as in regular mode collapse. Still there is a problem in that different instances generated from the same input text probe all look too similar, as if the randomness of the latent code is ignored. Mao et al. [68] presented a regularization term to encourage the generator to explore more minor modes and force the discriminator to concentrate on the instances from the minor modes. Zhang et al. [126] developed a conditional augmentation technique (CA) to produce diverse instances from a single textual description. CA concatenates a latent code and a textual variable sampled from the conditional Gaussian distribution as the inputs of the generator. Furthermore, CA adds the Kullback-Leibler divergence between the Gaussian distribution and the conditional Gaussian distribution to the generator objective in order to encourage smoothness in the conditional manifold.

Nonetheless, these approaches either require exorbitant computational cost or are invalid for a single-stage text-to-image GAN architecture. Here, we take the multi-generator model [28] as an example. Assuming that various single-stage text-to-image generators are exploited to learn all the modes of the distribution of true data to overcome mode collapse, we expect that different generators are able to fit diverse modes. However, in practice, all generators are still prone to falling into a singular mode for text-to-image synthesis, since each generator only pays attention to the same conditional context. We would like to propose to distinguish three levels of performance for a GAN:
Level 1: Traditional mode collapse \([4, 13, 50, 70, 89, 100, 135]\) - strange, inappropriate patterns become a point attractor in the non-linear cyclic process \([10]\) between the generator and the discriminator (cf. Fig. 6 in \([89]\));

Level 2: Light mode collapse \([5, 54, 64, 68, 126]\) - patterns from the training set become wholly or partially an attractor for the generator. This is akin to lookup-table behavior made possible by the high number of parameters in a GAN. This constitutes the lack-of-diversity problem (cf. Fig. 3.1, left, this chapter);

Level 3: Desired functionality - generation of diverse patterns that are semantically consistent with the text probe as regards the foreground and that provide a believable, natural pattern in the background.

This chapter focuses on solving the lack-of-diversity issue, i.e., the light mode collapse, not the severe, traditional mode collapse.

### 3.3 PROPOSED METHODOLOGY

In this section, we analyze the reason for the lack-of-diversity issue. Subsequently, we discuss the overall architecture of DiverGAN, depicted in Fig. 3.2. After that, two novel types of word-level attention models, i.e., a channel-attention module (CAM) and a pixel-attention module (PAM), are introduced to modulate the visual feature map with word-context embeddings.

#### 3.3.1 Preliminary

Mode collapse is a phenomenon where the model fails to learn all the modes of the distribution of true data, and thus generated samples lack diversity \([4]\). In addition, for a cGAN, the produced instances from a single-condition context seem identical \([68]\). One of the main reasons for the lack-of-diversity issue is that the generator merely visits a part of the distribution of real data and misses a few modes, limited by the generative capability of the network \([41]\). Although the generator tries to map random latent codes into the distribution of original data, such a map is not surjective \([57]\). When the designed model is not powerful, the network can only capture a few modes of the distribution of real data and tends to map different inputs into these modes.

The lack-of-diversity issue becomes worse in the single-stage text-to-image pipeline. Generally, in the single-stage framework, a random latent code is taken as the input which is responsible for variations, while textual features serve as side information to modulate the visual feature map and determine the main visual content. To boost the semantic consistency of generated instances, the generator usually leverages modulation modules for the per-scale feature map, which may make the network tend to concentrate on the textual-context features
3.3 Proposed methodology

The bird is small with a pointy thin beak, yellow and gray colors, and a small head.

(b) Generator Module
(a) Pipeline
(c) Residual Block
(d) Block

Figure 3.2: The overall architecture of the proposed DiverGAN. $z$ is an input latent code, FC is a fully-connected layer, Conv denotes a convolutional layer and Norm represents Conditional Adaptive Instance-Layer Normalization (CAdaILN) discussed in Section 2.4.4. Additionally, CAM and PAM refer to the channel-attention module and pixel-attention module, respectively, discussed in Section 3.3.3. Moreover, the FC in (b) is employed to improve the generative ability of the generator, strengthening the control of $z$ over the visual feature map and boosting diversity. The Block in (c) is used to improve the image quality, and the CAM and PAM in (d) are presented to strengthen the visual feature map. Note that we omit the up-sample operation between residual blocks in (b).

that are high-dimensional and structured, and ignore a low-dimensional latent code. In this chapter, we aim to enhance the generative ability of the network to reinforce the control of a random latent code over the visual feature map for the purpose of improving diversity.

3.3.2 Overall architecture

Fig. 3.2 shows the overall architecture of DiverGAN. The first layer exploits a fully-connected layer to process a latent code $z \in \mathbb{R}^{100}$ and reshapes the result as the initial feature map $F_0 \in \mathbb{R}^{4 \times 4 \times 256}$. After that, $F_0$ is taken into the basic generator module (see Fig. 3.2(b)) that mainly comprises seven dual-residual blocks (see Fig. 3.2(c)) receiving textual embeddings derived from a text encoder as extra conditional contexts to strengthen the visual feature map.

Fig. 3.2(c) shows the details of our modified dual-residual module. Our designed dual-residual block consists of two residual modules (see Fig. 3.2(d)), each of which is constructed with a set of convolutional layers, CAdaILN, ReLU
activation functions followed with modulation modules, i.e., a channel-attention module (CAM) and a pixel-attention module (PAM)), taking the word-context features as side information to modulate the visual feature map. Furthermore, we plug a stack of a convolutional layer and CAdaILN activated by Leaky-ReLU between two residual modules to relieve the control of contextual information. The dual-residual block not only accelerates convergence speed by retaining more original visual features than cascade convolutions, but also enables us to easily increase the depth of the network, efficiently improving network capacity and resulting in high-quality images with more details.

To deal with the lack-of-diversity issue, we conduct a series of experiments on structure design and try to reinforce the control of the random latent code over the visual feature map to enhance variants. We observe that a dense layer can significantly boost the generative ability of the network, encouraging the model to explore minor modes of the distribution of true data. For this reason, we attempt to insert one linear layer into our pipeline. Experimental results demonstrate that embedding a fully-connected layer after the second dual-residual block achieves appealing performance on visual quality and image diversity. More specifically, the output feature map \( F_2 \in \mathbb{R}^{8 \times 8 \times 256} \) of the second dual-residual block is first resized to \( \mathbb{R}^{16384} \) and put into a linear layer that maintains the dimension of the input features. Afterwards, \( F_2 \) is reshaped to \( \mathbb{R}^{8 \times 8 \times 256} \) again and passed to the next dual-residual block. Then, the output of the basic generator module (see Fig. 3.2(b)) is sent to one convolutional layer activated by tahn function to generate a final sample.

**Why does inserting a dense layer address the lack-of-diversity issue?** By inserting a fully-connected layer, the network cannot exploit the spatial 2D layout of the preceding feature maps and needs to encode all the necessary information in a single 1D vector (embedding) as the basis for an unfolding in 2D by the later layers. As a result, we will have a representation at this point in the architecture that lends itself for injection of random noise with a subsequently increased diversity in the generated patterns: Because of the 1D bottleneck, the network cannot easily replicate (partial) 2D patterns from the early feature maps. This avoids the ‘lookup-table’ property that many GAN architectures have.

**Why does embedding a fully-connected layer after the second residual block achieve the best variety and quality?** If the dense layer is too early in the network, it obtains early, crude featural information that is insufficient to generate semantically consistent output patterns. If you have an early feature map representing small bird components, the network will not be able to assemble a bird. On the other hand, if the dense layer is too late in the network, it will be fed by almost-complete patterns, and there will be a lack of diversity, since the system will operate as a lookup-table for the patterns in the training set that can then only be modified marginally by the last few layers. Additionally, it is extremely difficult to insert a dense layer after the third or later residual blocks due to the limited memory on the GPU. This can only be determined empirically.
3.3 Proposed methodology

To the best of our knowledge, we are the first to propose this kind of text-to-image architecture introducing one linear layer to improve the power of a random latent code to produce images of diverse modes. We expect that DiverGAN can provide a strong basis for the future developments of text-to-image generation.

3.3.3 Dual-attention mechanism

It is well known that the semantic affinities between conditional contexts and visual feature maps are particularly critical for image synthesis. However, this relation will become more complicated for text-to-image generation, since the given sentence contains a suite of words which have different contributions to the synthesized samples. For instance, the adjective in the input text description will attend more to the produced picture than the definite article “the”. Moreover, although our dual-residual structure is beneficial for model capacity and training stability, it may bring noise and redundant information. For these reasons, two new types of word-level attention modules, termed as a channel-attention module (CAM) and a pixel-attention module (PAM), are designed to explore the latent interplay between word-context features and visual feature maps. CAM and PAM have the capacity to identify the significant words (e.g., adjectives and nouns) in the given text description and make the network assign more weights to the crucial channels and pixels semantically associated with these words. In addition, they can alleviate the effects of semantically irrelevant and redundant features from both channel and spatial perspectives.

3.3.3.1 Channel-attention module (CAM)

Channel maps have different responses to the words in the given sentence, and thus, the channels that respond to the prominent words in the natural-language description deserve more attention from the network. Here, we propose CAM to model the importance of each word while assigning larger weights to more useful channels semantically matching with the salient words.

Fig. 3.3 illustrates the detailed structure of CAM. Given a feature map $F_c \in \mathbb{R}^{H \times W \times C}$ (where $H$, $W$ and $C$ denote the height, the width and the channel number of $F_c$, respectively), we first adopt the global average pooling and max pooling to process it to aggregate holistic and discriminative information, thereby deriving two channel feature vectors $F_{ca} \in \mathbb{R}^{1 \times 1 \times C}$ and $F_{cm} \in \mathbb{R}^{1 \times 1 \times C}$. After that, $F_{ca}$ and $F_{cm}$ are fed into two different $1 \times 1$ convolution layers rather than one $1 \times 1$ convolution operation, since average pooling and max pooling acquire different globally spatial statistics [56]. The outputs are then converted to $\mathbb{R}^{1 \times C}$ and repeated $C$ times along dimension 1 to obtain two features: the average-
Figure 3.3: Overview of the proposed channel-attention module, which aims to assign larger weights to more useful channels semantically matching with the salient words. AvgPool and MaxPool refer to the global average pooling and max pooling, respectively. \( H, W \) and \( C \) denote the height, the width and the channel number of the visual feature map, respectively. \( M \) is the dimension of the word embeddings and \( T \) is the number of the words in the given text description. \( 1 \times 1 \) conv indicates the \( 1 \times 1 \) convolution operation.

Mathematically,

\[
F_{caq} = f_{re}(f_{caq}(\text{Avg}(F_c)))
\]

(3.1)

\[
F_{cmq} = f_{re}(f_{cmq}(\text{Max}(F_c)))
\]

(3.2)

where \( \text{Avg} \) and \( \text{Max} \) represent the global average pooling and max pooling, respectively. \( f_{caq} \) and \( f_{cmq} \) denote \( 1 \times 1 \) convolution layers and \( f_{re} \) refers to the reshape and repeat operations.

For word-context vectors \( E \in \mathbb{R}^{M \times T} \) (where \( M \) denotes the dimension of the word embeddings and \( T \) denotes the number of the words in the given text description), we flow them into two different \( 1 \times 1 \) convolution operations followed by ReLU activation to produce two contextual vectors: the key \( F_{ck} \in \mathbb{R}^{C \times T} \) and the value \( F_{cv} \in \mathbb{R}^{C \times T} \), which are in the common semantic space of the visual features. Next, we compute the mean of the value \( F_{cv} \) along the dimension 2 and resize it into \( \mathbb{R}^{1 \times C} \). Meanwhile, we multiply the result and the value \( F_{cv} \) to gain the contextual attention map \( E_{ci} \in \mathbb{R}^{1 \times T} \) that indicates the importance of each word in the sentence. Intuitively, a larger value in the attention map
means that the corresponding word attends more to the synthesized image. The acquisition of the contextual attention map is formulated as:

\[
F_{ck} = f_{ck}(E) \tag{3.3}
\]
\[
F_{cv} = f_{cv}(E) \tag{3.4}
\]
\[
E_{ci} = f_{cmean}(F_{cv} \ast F_{cv}) \tag{3.5}
\]

where \(f_{ck}\) and \(f_{cv}\) refer to \(1 \times 1\) convolution layers followed with ReLU activation. \(f_{cmean}\) represents the average and reshape operations.

Afterwards, to model the semantic affinities between word-context features and channels, we conduct a dot-product operation between the queries and the key \(F_{ck} \in \mathbb{R}^{C \times T}\), and apply a softmax function to get the contextually channel-wise attention matrix \(CA \in \mathbb{R}^{C \times T}\) indicating the similarity weights between channels and words in the textual description. At last, the channel-attention weights \(W_c\) are calculated through a softmax function on the dot products of \(CA\) with the transpose of \(E_{ci}\), and resized into \(\mathbb{R}^{1 \times 1 \times C}\). We formulate a series of operations as:

\[
CA_a = \text{Softmax}(D(F_{caq}, F_{ck})) \tag{3.6}
\]
\[
CA_m = \text{Softmax}(D(F_{cmq}, F_{ck})) \tag{3.7}
\]
\[
W_{ca} = f_{re}(\text{Softmax}(D(CA_a, E_{ci}^T))) \tag{3.8}
\]
\[
W_{cm} = f_{re}(\text{Softmax}(D(CA_m, E_{ci}^T))) \tag{3.9}
\]

where \(D(\cdot)\) denotes the dot-product operation and \(\text{Softmax}\) represents the softmax function. \(CA_a \in \mathbb{R}^{C \times T}\) and \(CA_m \in \mathbb{R}^{C \times T}\) refer to the contextually channel-wise attention matrices of \(F_{caq}\) and \(F_{cmq}\), respectively. \(f_{re}\) indicates the reshape operation. \(W_{ca} \in \mathbb{R}^{1 \times 1 \times C}\) and \(W_{cm} \in \mathbb{R}^{1 \times 1 \times C}\) are the channel-attention weights of \(CA_a\) and \(CA_m\), respectively.

After acquiring the attention scores of channels, we multiply them and the original feature map to re-weight the visual feature map. By doing so, the network will focus more on the useful channels of the feature map and assign larger weights on them. At the same time, we design an adaptive gating method to dynamically merge the output feature maps of the global average pooling and max pooling. The process of getting the fused feature map \(F_{cw} \in \mathbb{R}^{H \times W \times C}\) is described as:

\[
F_{caw} = W_{ca} \odot F_c \tag{3.10}
\]
\[
F_{cmw} = W_{cm} \odot F_c \tag{3.11}
\]
\[
g_{ca} = \sigma(f_{cga}(f_{re}(F_c)) + f_{cgm}(f_{re}(F_{cm}))) \tag{3.12}
\]
\[
F_{cw} = g_{ca} * F_{caw} + (1 - g_{ca}) * F_{cmw} \tag{3.13}
\]
Figure 3.4: Overview of the proposed pixel-attention module, which aims to capture the spatial relationships between pixels and word embeddings, and allow significant pixels to acquire more weights. AvgPool and MaxPool denote the average pooling and max pooling in the spatial dimension, respectively. \( H \), \( W \) and \( C \) denote the height, the width and the channel number of the visual feature map, respectively. \( M \) is the dimension of the word embeddings and \( T \) is the number of the words in the given text description. \( 1 \times 1 \) conv indicates the \( 1 \times 1 \) convolution operation.

where \( \odot \) is the element-wise multiplication. \( F_{caw} \) and \( F_{cmw} \) denote the rescaled feature maps. \( g_{ca} \) represents the response gate for the fusion of visual feature maps. \( f_{re} \) refers to the reshape operation that resizes \( F_{ca} \) and \( F_{cm} \) into \( \mathbb{R}^{1 \times C} \). \( f_{cg} \) and \( f_{cgm} \) represent the fully-connected layers that reduce the number of channels to 1 and \( \sigma \) denotes the sigmoid function.

To further retain the basic features and stabilize the learning of the cGAN, we apply an adaptive residual connection [125] to synthesize the final result \( F_{cu} \in \mathbb{R}^{H \times W \times C} \). It is defined as follows:

\[
F_{cu} = \gamma_c \odot F_{cw} + F_c
\] (3.14)

where \( \gamma_c \) is a learnable parameter which is initialized as 0.

3.3.3.2 Pixel-attention module (PAM)

As discussed above, CAM re-weights the visual feature map from the perspective of channel. However, in addition to the channels in the image feature map, the pixels are of central importance for the quality and semantic consistency. Hence, PAM is presented to effectively capture the spatial interplay between pixels and word-context vectors, allowing the significant pixels to gain more attentions from the network. Noteworthily, PAM is performed on the output of CAM, since the pixels in the same channel of the feature map still share the same weights.
The framework of PAM is depicted in Fig. 3.4. For a feature map \( F_s \in \mathbb{R}^{H\times W\times C} \), we first perform the average pooling and max pooling in the spatial dimension over it to distill global features, acquiring two spatial feature vectors \( F_{sa} \in \mathbb{R}^{H\times W\times 1} \) and \( F_{sm} \in \mathbb{R}^{H\times W\times 1} \). Then, \( F_{sa} \) and \( F_{sm} \) are resized into \( \mathbb{R}^{(H\times W)^{\times} 1} \) and repeated \( C \) times along dimension 2 to obtain two matrices: the average-pooling query \( F_{saq} \in \mathbb{R}^{(H\times W)^{\times} C} \) and the max-pooling query \( F_{smq} \in \mathbb{R}^{(H\times W)^{\times} C} \), respectively. The process of obtaining the queries is formulated as follows:

\[
F_{saq} = f_{re}(\text{Avg}(F_s))
\]

\[
F_{smq} = f_{re}(\text{Max}(F_s))
\]

where \( \text{Avg} \) and \( \text{Max} \) denote the average pooling and max pooling in the spatial dimension, respectively. \( f_{re} \) refers to the reshape and repeat operations.

Given word-context features \( E \in \mathbb{R}^{M\times T} \), we employ the same way as CAM to process them, producing the key \( F_{sk} \in \mathbb{R}^{C\times T} \), the value \( F_{sv} \in \mathbb{R}^{C\times T} \) and the contextual attention map \( E_{si} \in \mathbb{R}^{1\times T} \). Specifically,

\[
F_{sk} = f_{c_{sk}}(E)
\]

\[
F_{sv} = f_{c_{sv}}(E)
\]

\[
E_{si} = f_{c_{mean}}(F_{sv}) \times F_{sv}
\]

where \( f_{c_{sk}} \) and \( f_{c_{sv}} \) refer to \( 1 \times 1 \) convolution layers followed with ReLU activation. \( f_{c_{mean}} \) represents the average and reshape operations.

After that, the spatial-semantic attention map \( SA \in \mathbb{R}^{(H\times W)^{\times} T} \) is achieved via a softmax function on the dot products of the queries with the key \( F_{sk} \in \mathbb{R}^{C\times T} \), indicating the similarity weights between pixels and words in the text description. Subsequently, we conduct a dot-product operation between \( SA \) and the transpose of \( E_{si} \), and leverage a softmax function to obtain the pixel-wise attention weights that are converted to \( \mathbb{R}^{H\times W\times 1} \). The acquisition of the pixel-wise attention weights \( W_s \in \mathbb{R}^{H\times W\times 1} \) is denoted as follows:

\[
SA_a = \text{Softmax}(D(F_{saq}, F_{sk}))
\]

\[
SA_m = \text{Softmax}(D(F_{smq}, F_{sk}))
\]

\[
W_{sa} = f_{re}(\text{Softmax}(D(SA_a, E_{si}^T)))
\]

\[
W_{sm} = f_{re}(\text{Softmax}(D(SA_m, E_{si}^T)))
\]

where \( D(\cdot) \) denotes the dot-product operation and \( \text{Softmax} \) represents the softmax function. \( SA_a \in \mathbb{R}^{(H\times W)^{\times} T} \) and \( SA_m \in \mathbb{R}^{(H\times W)^{\times} T} \) refer to the contextually spatial-attention maps of \( F_{saq} \) and \( F_{smq} \), respectively. \( W_{sa} \in \mathbb{R}^{H\times W\times 1} \) and \( W_{sm} \in \mathbb{R}^{H\times W\times 1} \) are the pixel-wise attention weights of \( SA_a \) and \( SA_m \), respectively.
Next, same as CAM, we perform a matrix multiplication between the pixel-wise attention scores and the original feature map to facilitate the visual feature map. Meanwhile, in order to maintain the features, we concatenate the rescaled feature maps of the average pooling and max pooling, and put the result into a 1 × 1 convolution layer followed with a ReLU function to generate the merged feature map $F_{sw} \in \mathbb{R}^{H \times W \times C}$. Then, an adaptive residual connection is adopted to get the final output $F_{su} \in \mathbb{R}^{H \times W \times C}$. This process is described as:

$$F_{saw} = W_{sa} \odot F_s$$
$$F_{smw} = W_{sm} \odot F_s$$
$$F_{sw} = f_{con}([F_{saw}; F_{smw}])$$
$$F_{su} = \gamma_s \ast F_{sw} + F_s$$

where $\odot$ is the element-wise multiplication. $F_{saw}$ and $F_{smw}$ denote the rescaled feature maps. $\left[ \right]$ refers to the concatenation operation along the channel dimension and $f_{con}$ represents the 1 × 1 convolutional operation followed with a ReLU function. $\gamma_s$ is a learnable parameter which is initialized as 0.

**Why do our attention modules work better?** Firstly, we model the importance of each word to emphasize the salient words (e.g., adjectives and nouns) in the given text description. This will make our generator concentrate more on the semantically related parts of the generated image. Secondly, in order to achieve better quality, we utilize average pooling and max pooling to acquire different globally spatial and channel information. Thirdly, an adaptive gating method is designed to dynamically merge the output feature maps of average pooling and max pooling, obtaining better performance. Fourthly, we specifically spread our attention weights to all the channels and pixels to enhance feature maps in several (!) layers, while applying an adaptive residual connection to synthesize the final result, in order to retain the basic features and stabilize the learning of a cGAN. We have not seen this in literature. Fifthly, different from the previous methods, we leverage our attention modules to modulate per-scale feature map across both channel and spatial dimensions. Sixthly, our proposed CAdaILN can help with flexibly controlling the amount of change in shape and texture, complementing our attention modules. Notice that our proposed dual-attention mechanisms are easy-to-implement methods, although they seem complicated.

### 3.3.4 Network details

The generator network of DiverGAN is composed of seven dual-residual modules which are responsible for modulating different scales of feature maps. Each block in a dual-residual layer comprises convolutional layers, CAdaILN functions, ReLU functions, CAM and PAM. Note that we use Instance-Layer Normalization (ILN) in the last three layers due to the change of channels. The network details of the
<table>
<thead>
<tr>
<th>Layers</th>
<th>Input $\rightarrow$ Output shape</th>
<th>Layer Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td>$(100) \rightarrow (4096) \rightarrow (256, 4, 4)$</td>
<td>FC-$\left(100, 4096\right)$</td>
</tr>
<tr>
<td>Layer I</td>
<td>$(256, 4, 4) \rightarrow (256, 4, 4)$</td>
<td>Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, PAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, Leaky-ReLU, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, PAM</td>
</tr>
<tr>
<td>Up-sample</td>
<td>$(256, 4, 4) \rightarrow (256, 8, 8)$</td>
<td>Up-sampling layer</td>
</tr>
<tr>
<td>Layer II</td>
<td>$(256, 8, 8) \rightarrow (256, 8, 8)$</td>
<td>Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, PAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, Leaky-ReLU, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, PAM</td>
</tr>
<tr>
<td>$F_2$</td>
<td>$(256, 8, 8) \rightarrow (16384) \rightarrow (256, 8, 8)$</td>
<td>FC-$\left(16384, 16384\right)$</td>
</tr>
<tr>
<td>Up-sample</td>
<td>$(256, 8, 8) \rightarrow (256, 16, 16)$</td>
<td>Up-sampling layer</td>
</tr>
<tr>
<td>Layer III</td>
<td>$(256, 16, 16) \rightarrow (256, 16, 16)$</td>
<td>Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, PAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, Leaky-ReLU, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, PAM</td>
</tr>
<tr>
<td>Up-sample</td>
<td>$(256, 16, 16) \rightarrow (256, 32, 32)$</td>
<td>Up-sampling layer</td>
</tr>
<tr>
<td>Layer IV</td>
<td>$(256, 32, 32) \rightarrow (256, 32, 32)$</td>
<td>Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, PAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, Leaky-ReLU, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, CAdaILN, PAM</td>
</tr>
<tr>
<td>Up-sample</td>
<td>$(256, 32, 32) \rightarrow (256, 64, 64)$</td>
<td>Up-sampling layer</td>
</tr>
<tr>
<td>Layer V</td>
<td>$(256, 64, 64) \rightarrow (128, 64, 64)$</td>
<td>Conv-$\left(256, K_3, S_1, P_1\right)$, ILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, ILN, PAM, Conv-$\left(256, K_3, S_1, P_1\right)$, ILN, Leaky-ReLU, Conv-$\left(256, K_3, S_1, P_1\right)$, ILN, ReLU, CAM, Conv-$\left(256, K_3, S_1, P_1\right)$, ILN, PAM</td>
</tr>
<tr>
<td>Up-sample</td>
<td>$(128, 64, 64) \rightarrow (128, 128, 128)$</td>
<td>Up-sampling layer</td>
</tr>
<tr>
<td>Layer VI</td>
<td>$(128, 128, 128) \rightarrow (64, 128, 128)$</td>
<td>Conv-$\left(128, K_3, S_1, P_1\right)$, ILN, ReLU, CAM, Conv-$\left(128, K_3, S_1, P_1\right)$, ILN, PAM, Conv-$\left(128, K_3, S_1, P_1\right)$, ILN, Leaky-ReLU, Conv-$\left(128, K_3, S_1, P_1\right)$, ILN, ReLU, CAM, Conv-$\left(128, K_3, S_1, P_1\right)$, ILN, PAM</td>
</tr>
<tr>
<td>Up-sample</td>
<td>$(64, 128, 128) \rightarrow (64, 256, 256)$</td>
<td>Up-sampling layer</td>
</tr>
<tr>
<td>Layer VII</td>
<td>$(64, 256, 256) \rightarrow (32, 256, 256)$</td>
<td>Conv-$\left(64, K_3, S_1, P_1\right)$, ILN, ReLU, CAM, Conv-$\left(64, K_3, S_1, P_1\right)$, ILN, PAM, Conv-$\left(64, K_3, S_1, P_1\right)$, ILN, Leaky-ReLU, Conv-$\left(64, K_3, S_1, P_1\right)$, ILN, ReLU, CAM, Conv-$\left(64, K_3, S_1, P_1\right)$, ILN, PAM</td>
</tr>
<tr>
<td>Conv</td>
<td>$(32, 256, 256) \rightarrow (3, 256, 256)$</td>
<td>Conv-$\left(N_3, K_3, S_1, P_1\right)$, Tanh</td>
</tr>
</tbody>
</table>

Table 3.1: The details of our generator architecture. $F_1$ and $F_2$ denote dense layers, Layer I-Layer VII denote our proposed dual-residual layers, Conv denotes a convolutional layer, CAdaILN denotes Conditional Adaptive Instance-Layer Normalization, ILN denotes Instance-Layer Normalization, CAM denotes a channel-attention module, PAM denotes a pixel-attention module, ReLU denotes a rectified linear unit and FC denotes a fully-connected layer.
generator are shown in Table 3.1. We utilize the discriminator of DTGAN due to its promising performance.

3.4 EXPERIMENTS

In this section, to prove the effectiveness of the proposed DiverGAN in diversity and producing visually realistic and semantically consistent images, we perform a wealth of quantitative and qualitative evaluations on three benchmark data sets, i.e., Oxford-102 [75], CUB bird [108] and MSCOCO [63]. To be specific, we clarify the details of experimental settings in Section 3.4.1. After that, the proposed DiverGAN is compared to previous cGAN-based approaches for text-to-image generation in Section 3.4.2. Subsequently, we analyze the contributions from different components of DiverGAN in Section 3.4.3.

3.4.1 Experimental settings

Datasets. We evaluate DiverGAN on three extensively employed data sets, i.e., Oxford-102, CUB bird and MSCOCO, which are used by StackGAN [126], AttnGAN [118], MirrorGAN [83], ControlGAN [55], SDGAN [122], DM-GAN [140], DF-GAN [105], etcetera. The Oxford-102 data set includes 5,878 and 2,311 images for training and testing, respectively. Each picture is accompanied by 10 textual descriptions. The statistics of the CUB bird and MSCOCO data sets are described in Chapter 2.

Implementation details. For the text encoder, following the method of DTGAN, we utilize a pretrained bidirectional LSTM [92] to acquire the word embeddings and the global sentence vector, respectively. We adopt the losses in DTGAN owing to its promising results. We set the dimension of the latent code to 100. As for the training, we leverage the Adam optimizer [49] with $\beta = (0.0, 0.9)$ to train our network. We also follow the TTUR [38] and set the learning rates for the generator and the discriminator to 0.0001 and 0.0004, respectively. The batch size is set to 16. DiverGAN is implemented by PyTorch [81]. All the experiments are performed on a single NVIDIA Tesla V100 GPU (32 GB memory).

Evaluation metrics. DiverGAN is evaluated by calculating three widely used metrics including an inception score (IS) [89], a Fréchet inception distance (FID) [103] score and a learned perceptual image patch similarity (LPIPS) [128] score. The first two are to measure the visual quality and the last one is employed to assess the diversity of generated samples. LPIPS measures diversity by computing the average feature distance between synthetic images. The synthesized pictures are meant to be diverse if the LPIPS score is large. The results of LPIPS will be discussed in Section 3.4.3.4. The details of the IS and the FID are introduced in Chapter 2.
Table 3.2: The IS of prior approaches and DiverGAN on the CUB bird and Oxford-102 data sets. The best scores are in bold.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CUB ↑</th>
<th>Oxford-102 ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAN-INT-CLS [87]</td>
<td>2.88±0.04</td>
<td>2.66±0.03</td>
</tr>
<tr>
<td>GAWWN [86]</td>
<td>3.62±0.07</td>
<td>–</td>
</tr>
<tr>
<td>StackGAN [126]</td>
<td>3.70±0.04</td>
<td>3.20±0.01</td>
</tr>
<tr>
<td>StackGAN++ [127]</td>
<td>4.04±0.05</td>
<td>3.26±0.01</td>
</tr>
<tr>
<td>AttnGAN [118]</td>
<td>4.36±0.03</td>
<td>–</td>
</tr>
<tr>
<td>MirrorGAN [83]</td>
<td>4.56±0.05</td>
<td>–</td>
</tr>
<tr>
<td>ControlGAN [55]</td>
<td>4.58±0.09</td>
<td>–</td>
</tr>
<tr>
<td>SDGAN [122]</td>
<td>4.67±0.09</td>
<td>–</td>
</tr>
<tr>
<td>SegAttnGAN [32]</td>
<td>4.82±0.05</td>
<td>3.52±0.09</td>
</tr>
<tr>
<td>DM-GAN [140]</td>
<td>4.75±0.07</td>
<td>–</td>
</tr>
<tr>
<td>DF-GAN [105]</td>
<td>4.86±0.04</td>
<td>3.71±0.06</td>
</tr>
<tr>
<td>DTGAN</td>
<td>4.88±0.03</td>
<td>3.77±0.06</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>4.98 ± 0.06</strong></td>
<td><strong>3.99 ± 0.05</strong></td>
</tr>
</tbody>
</table>

For the MSCOCO data set, the IS fails to evaluate the image quality and can be saturated, even over-fitted, which is observed by DTGAN, DF-GAN [105] and ObjGAN [58]. Therefore, we do not utilize the IS as the evaluation metric on the MSCOCO data set.

For the Oxford-102 data set, we do not list the FID due to the lack of compared scores. It should be noted that we synthesize 30,000 pictures from unseen textual descriptions for the IS and FID.

Table 3.3: The FID of StackGAN++ [127], AttnGAN [118], DF-GAN [105] and and DiverGAN on the CUB bird and COCO data sets. The best results are in bold.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CUB ↓</th>
<th>COCO ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>StackGAN++ [127]</td>
<td>26.07</td>
<td>51.62</td>
</tr>
<tr>
<td>AttnGAN [118]</td>
<td>23.98</td>
<td>35.49</td>
</tr>
<tr>
<td>DF-GAN [105]</td>
<td>19.24</td>
<td>28.92</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>15.63</strong></td>
<td><strong>20.52</strong></td>
</tr>
</tbody>
</table>
Figure 3.5: Qualitative comparison of DM-GAN \cite{140}, DF-GAN \cite{105} and DiverGAN conditioned on the text descriptions on (a) the MSCOCO and (b) CUB bird data sets.

3.4.2 Comparison with state of the art

3.4.2.1 Quantitative results

We compare our method with previous single-stage \cite{105, 132} and multi-stage \cite{32, 55, 83, 118, 122, 126, 127, 140} cGAN-based text-to-image approaches on the CUB bird, Oxford-102 and MSCOCO data sets. The IS of DiverGAN and other compared models on the CUB bird and Oxford-102 data sets are reported in Table 3.2. We can see that DiverGAN achieves the best performance, significantly increasing the IS from 4.88 to 4.98 on the CUB bird data set and from 3.77 to 3.99 on the Oxford-102 data set. Experimental results indicate that
3.4 experiments

DiverGAN is capable of producing perceptually plausible pictures, with higher quality than state-of-the-art approaches.

The comparison between DiverGAN, StackGAN++ [127], AttnGAN [118] and DF-GAN [105] with respect to the FID on the CUB bird and MSCOCO data sets is shown in Table 3.3. We can observe that DiverGAN obtains a remarkably lower FID than compared methods on both data sets, which demonstrates that the distribution of our generated data is closer to the distribution of true data. More specifically, we impressively reduce the FID from 28.92 to 20.52 on the challenging MSCOCO data set and from 19.24 to 15.63 on the CUB bird data set.

3.4.2.2 Qualitative results

In addition to quantitative comparison, we conduct the qualitative experiments on the CUB bird, Oxford-102 and MSCOCO data sets, which are illustrated in Fig. 3.5 and Fig. 3.6.

Fig. 3.5 shows the qualitative results of DM-GAN [140], DF-GAN [105] and DiverGAN on the MSCOCO and CUB bird data sets, indicating that DiverGAN has the capacity to synthesize high-quality and semantic-consistency pictures conditioned on the text descriptions. For instance, in terms of the generation of a complex scene, DiverGAN synthesizes a red bus with more vivid details than DF-GAN and DM-GAN in the 1st column of Fig. 3.5 (a). It can also be seen that DiverGAN produces a kitchen with a plausible wooden counter (2nd column), a clear red sign on the road (3rd column), fresh vegetables and fruits with rich colors (4th column), a man surfing on the realistic sea waves (5th column) and a beautiful night street (6th column), whereas both DM-GAN and DF-GAN yield unclear...
objects ($1^{st}$, $2^{nd}$, $3^{rd}$, $7^{th}$ and $8^{th}$ column) and the background with a single color ($4^{th}$, $5^{th}$ and $6^{th}$ column). More importantly, DiverGAN creates an impressive European clock tower in the $7^{th}$ column. The above results demonstrate that DiverGAN equipped with the dual-residual structure is capable of capturing the crucial words in the textual description and highlighting the main objects of the synthesized image, generating a high-quality multi-object scene with vivid details.

As can be observed in Fig. 3.5 (b), DM-GAN, DF-GAN and DiverGAN all yield promising birds with consistent colors and shapes, but our method better concentrates on the semantically related parts of the generated image, synthesizing perceptually realistic birds. In addition, some backgrounds synthesized by DM-GAN ($1^{st}$, $2^{nd}$, $5^{th}$ and $7^{th}$ column) and DF-GAN ($5^{th}$, $7^{th}$ and $8^{th}$ column) are not plausible. It indicates that, with word-level attention modules, our model is able to bridge the semantic gap between visual feature maps and word-context vectors, producing high-quality samples which semantically align with the text descriptions. The qualitative comparison between DF-GAN, DTGAN and DiverGAN on the Oxford-102 data set is depicted in Fig. 3.6. We can observe that our approach synthesizes visually plausible flowers with more vivid details, richer colors and more clear shapes than DF-GAN and DTGAN, which confirms the effectiveness of DiverGAN.
Table 3.4: Ablation study of DiverGAN. CAM, PAM and FC represent the channel-attention module, the pixel-attention module and the insertion of the fully-connected layer between the first and the second residual block, respectively. The best results are in bold.

<table>
<thead>
<tr>
<th>ID</th>
<th>Components</th>
<th>CUB set</th>
<th>Oxford-102 (IS) ↑</th>
<th>COCO (FID) ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CAM PAM CAdaILN FC</td>
<td>IS ↑</td>
<td>FID ↓</td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>✓ ✓ ✓ ✓</td>
<td>4.98 ± 0.06</td>
<td>15.63</td>
<td>3.99 ± 0.05</td>
</tr>
<tr>
<td>M2</td>
<td>✓ ✓ ✓ -</td>
<td>4.91 ± 0.06</td>
<td>16.42</td>
<td>3.87 ± 0.07</td>
</tr>
<tr>
<td>M3</td>
<td>✓ ✓ - ✓</td>
<td>4.36 ± 0.05</td>
<td>24.17</td>
<td>3.46 ± 0.04</td>
</tr>
<tr>
<td>M4</td>
<td>✓ - ✓ ✓</td>
<td>4.59 ± 0.06</td>
<td>20.86</td>
<td>3.77 ± 0.04</td>
</tr>
<tr>
<td>M5</td>
<td>- ✓ ✓ ✓</td>
<td>4.73 ± 0.04</td>
<td>21.19</td>
<td>3.81 ± 0.05</td>
</tr>
</tbody>
</table>

3.4.2.3 Human evaluation

We perform a human test on the CUB bird and MSCOCO data sets, so as to evaluate the image quality and the semantic consistency of DM-GAN [140], DF-GAN [105] and DiverGAN. We randomly select 100 images from both data sets, respectively. Given the same text description, users are asked to choose the best sample from the images synthesized by these three approaches according to the image details and the corresponding natural-language description. In addition, the final scores are computed by two judges for fairness. As illustrated in Fig. 3.7, our method impressively outperforms DM-GAN and DF-GAN on both data sets, especially on the challenging MSCOCO data set, which demonstrates the superiority of our proposed DiverGAN.

3.4.3 Component analysis

In order to evaluate the contributions from different components of DiverGAN, we conduct extensive ablation studies on the CUB bird, Oxford-102 and MSCOCO data sets. The novel components in our model include a channel-attention module (CAM), a pixel-attention module (PAM), CAdaILN proposed in DTGAN, a dual-residual block and the insertion of a fully-connected layer (FC) between the first and the second dual-residual block. We first quantitatively explore the effectiveness of each component by removing the corresponding part in DiverGAN step by step, i.e., M1: DiverGAN, M2: DiverGAN without the FC, M3: DiverGAN without CAdaILN, M4: DiverGAN without PAM, M5: DiverGAN without CAM. It is worth mentioning that we do not delete the dual-residual block in our ablation studies, since it is the basic structure in DiverGAN. All the results are reported in Table 3.4.

By comparing M1 (DiverGAN) with M2 (removing the FC), the introduction of the FC remarkably enhances the IS from 4.91 to 4.98 on the CUB data set and from
Table 3.5: Ablation study on attention modules. Attn and ControlAttn indicate the attention modules in AttnGAN [118] and ControlGAN [55], respectively. CAM and PAM denote our channel-attention module and pixel-attention module, respectively. The best results are in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>CUB set</th>
<th>Oxford-102 (IS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IS ↑</td>
<td>FID ↓</td>
</tr>
<tr>
<td>Attn [118]</td>
<td>4.52 ± 0.04</td>
<td>21.33</td>
</tr>
<tr>
<td>ControlAttn [55]</td>
<td>4.63 ± 0.05</td>
<td>20.47</td>
</tr>
<tr>
<td>CAM &amp; PAM</td>
<td>4.91 ± 0.06</td>
<td>16.42</td>
</tr>
</tbody>
</table>

3.87 to 3.99 on the Oxford-102 data set, and reduces the FID from 16.42 to 15.63 on the CUB data set and from 22.53 to 20.52 on the challenging MSCOCO data set. Experimental results demonstrate the importance of adopting a linear layer in DiverGAN. By exploiting CAdaILN in DiverGAN, M1 performs better than M3 (removing CAdaILN), confirming the effectiveness of the proposed CAdaILN. To be specific, the IS is improved by 0.62 on the CUB data set and 0.53 on the Oxford-102 data set, and the FID is decreased by 8.54 on the CUB data set and 6.28 on the MSCOCO data set. Furthermore, M1 achieves better results than both M4 (removing CAM) and M5 (removing PAM), which indicates that these two new types of word-level attention modules can help the generator yield more realistic images.

3.4.3.1 Effectiveness of the attention modules

To prove the effectiveness of our CAM and PAM, we also explore the performance of DiverGAN with other attention modules in text-to-image generation methods on the CUB and Oxford-102 data sets. Concretely, we replace the CAM and PAM in each residual block of M2 (DiverGAN removing the FC) with the attention modules in AttnGAN [118] and ControlGAN [55], respectively. Table 3.5 displays the comparable results. It can be observed that our proposed attention mechanisms outperform AttnGAN and ControlGAN, improving the IS by 0.28 on the CUB data set and 0.14 on the Oxford-102 data set, and reducing the FID from 20.47 to 16.42 on the CUB data set, which verifies the superiority of our proposed CAM and PAM.

The reason behind this result may be that the prior attention modules directly convert semantic vectors to visual feature maps, adopting the weighted sum of converted word features as the new feature map, which is largely different from the original feature map. However, our attention modules aim to strengthen the visual feature map according to the contextual-semantic relevance while also preserving the basic features to some extent. Additionally, CAM and PAM can emphasize the salient words in the given natural-language description instead


of equally treating all words. Noteworthily, with word-level attention modules, DiverGAN also has the ability to manipulate the parts of generated samples, which we detail in Section 3.4.3.4.

### 3.4.3.2 Effectiveness of the proposed CAdaILN

To further evaluate the effectiveness of CAdaILN discussed in Section 2.4.4, we perform an ablation study on the baseline model that removes CAdaILN from DiverGAN (M3). We apply the variants of normalization layers on M3. The results of the ablation study are reported in Table 3.6. We can see that by comparing B2 with B4 and B3 with B5, CAdaILN impressively outperforms the BN whether using the sentence-level linguistic cues or the word-level linguistic cues. Moreover, by comparing B5 with B4, CAdaILN with the global sentence vector performs better than CAdaILN-word by improving the IS from 4.71 to 4.91 on the CUB data set and from 3.73 to 3.87 on the Oxford-102 data set, and reducing the FID from 17.19 to 16.42 on the CUB data set and from 23.79 to 22.53 on the MSCOCO data set. The reason behind this result may be that sentence-level features is easier to train in our generator network than word-level features. The above results indicate the effectiveness of our presented CAdaILN in DTGAN.

### 3.4.3.3 Effect of the number of residual blocks in the residual-structure

To evaluate the impact of different number of residual blocks in the dual-residual structure, we compare the performance of a residual block (single-block) and two residual blocks (dual-block) on the CUB and Oxford-102 data sets, as depicted in Table 3.7. By comparison with the single-residual block, the proposed dual-residual block enhances the IS by 0.25 on the CUB data set and 0.12 on the Oxford-102 data set, and decreases the FID by 2.16 on the CUB data set, which shows the effectiveness of the dual-residual structure.
This bird is brown and white in color, with a brown beak

Figure 3.8: Qualitative results of other compared approaches and DiverGAN (bottom row) when given a single text description, on the CUB data set. DF-GAN + FC(2) and DTGAN + FC(2) refer to DF-GAN and DTGAN that plug a dense layer after the second residual block, respectively. F5, F6, F7, F8 and F9 represent the corresponding models in Table 3.8.
Table 3.7: Effect of the number of residual blocks in the dual-residual structure. Single-block and dual-block indicate the residual structure with a single block and two residual blocks, respectively. The best results are in bold.

<table>
<thead>
<tr>
<th>Metric</th>
<th>single-block</th>
<th>dual-block</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUB IS ↑</td>
<td>4.73±0.06</td>
<td>4.98±0.06</td>
</tr>
<tr>
<td>FID ↓</td>
<td>18.09</td>
<td>15.63</td>
</tr>
<tr>
<td>Oxford-102 (IS) ↑</td>
<td>3.87±0.06</td>
<td>3.99±0.05</td>
</tr>
</tbody>
</table>

Table 3.8: Effectiveness of the fully-connected layer (FC) in image diversity. Baseline (F5) corresponds to model M2 - DiverGAN removing the FC - in Table 3.4. CA and MS indicate conditioning augmentation (CA) and a mode-seeking regularization term (MS), respectively. FC(1) and FC(1+1) represent the insertion of one and two fully-connected layers after the first residual block, respectively. The best results are in bold.

<table>
<thead>
<tr>
<th>ID</th>
<th>Architecture</th>
<th>CUB set IS ↑</th>
<th>FID ↓ LPIPS ↑</th>
<th>Oxford-102 (IS) ↑</th>
<th>COCO (FID) ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>StackGAN++ [127]</td>
<td>4.04±0.05</td>
<td>26.07</td>
<td>0.362</td>
<td>3.26±0.01</td>
</tr>
<tr>
<td>F2</td>
<td>MSGAN [68]</td>
<td>-</td>
<td>25.53</td>
<td>0.373</td>
<td>-</td>
</tr>
<tr>
<td>F3</td>
<td>DF-GAN [105]</td>
<td>4.86±0.04</td>
<td>19.24</td>
<td>0.537</td>
<td>3.71±0.06</td>
</tr>
<tr>
<td>F4</td>
<td>DTGAN</td>
<td>4.88±0.03</td>
<td>16.35</td>
<td>0.544</td>
<td>3.77±0.06</td>
</tr>
<tr>
<td>F5</td>
<td>Baseline</td>
<td>4.91±0.06</td>
<td>16.42</td>
<td>0.549</td>
<td>3.87±0.07</td>
</tr>
<tr>
<td>F6</td>
<td>F5+CA [127]</td>
<td>4.72±0.06</td>
<td>17.61</td>
<td>0.535</td>
<td>3.82±0.06</td>
</tr>
<tr>
<td>F7</td>
<td>F5+MS [68]</td>
<td>4.57±0.05</td>
<td>18.48</td>
<td>0.667</td>
<td>3.82±0.05</td>
</tr>
<tr>
<td>F8</td>
<td>F5+FC(1)</td>
<td>4.76±0.05</td>
<td>18.34</td>
<td>0.579</td>
<td>3.87±0.07</td>
</tr>
<tr>
<td>F9</td>
<td>F5+FC(1+1)</td>
<td>4.73±0.06</td>
<td>18.68</td>
<td>0.655</td>
<td>3.85±0.06</td>
</tr>
<tr>
<td>F10</td>
<td>ours</td>
<td>4.98±0.06</td>
<td>15.63</td>
<td>0.682</td>
<td>3.99±0.05</td>
</tr>
</tbody>
</table>

3.4.3.4 Effectiveness of the fully-connected layer in image diversity

To validate the effectiveness of the dense layer in diversity, we perform quantitative evaluation on the CUB, Oxford-102 and MSCOCO data sets, and qualitative comparison on the CUB data set.

**Quantitative results.** The quantitative comparison is divided into three groups according to the goals. The first group is designed to evaluate the quality and diversity of samples synthesized by single-stage text-to-image methods, i.e., DF-GAN and DTGAN; StackGAN++ and MSGAN, which propose conditioning augmentation (CA) and a mode seeking regularization term (MS) to enhance the image diversity, respectively. The second group aims to compare our proposed fully-connected (FC) method with the CA and MS. For fair comparisons, we take the model M2 (DiverGAN removing the FC) in Table 3.4 as the baseline model.
F5). F6 (F5 + CA) and F7 (F5 + MS) add the CA and MS to F5 for comparison, respectively. In the last group, i.e., F8 (F5 + FC(1)), F9 (F5 + FC(1+1)) and F10 (DiverGAN), we verify the effectiveness of the different ways the linear layer is inserted into F5. F8 and F9 denote the model that plugs one and two dense layers after the first dual-residual block of F5, respectively. The quantitative results are reported in Table 3.8. By comparing F10 (DiverGAN) with F1 (StackGAN++), F2 (MSGAN), F3 (DF-GAN) and F4 (DTGAN), DiverGAN improves the LPIPS from 0.544 to 0.682 on the CUB data set, confirming the effectiveness of DiverGAN in diversity. By comparison with F5, F7 enhances the LPIPS by 0.118 on the CUB data set, whereas it decreases the IS by 0.34 on the CUB data set and 0.05 on the Oxford-102 data set, and increases the FID by 2.06 on the CUB data set and 1.91 on the MSCOCO data set. It demonstrates that although the introduction of the MS boosts image diversity, it may hurt image quality. It can be observed that F5 outperforms F6, which shows that CA may be not effective for current single-stage text-to-image methods in diversity. F8, F9 and F10 all achieve larger LPIPS than F5, which validates the effectiveness of a linear layer in improving diversity. Furthermore, F10 performs better than F8 and F9 by improving the LPIPS from 0.655 to 0.682 and the IS from 4.76 to 4.98 on the CUB data set and 3.87 to 3.99 on the Oxford-102 data set, and reducing the FID by 2.71 on the CUB data set and 1.64 on the MSCOCO data set, which indicates that inserting a linear layer after the second dual-residual block of F5 obtains the best performance on image quality and diversity. In conclusion, DiverGAN is capable of only leveraging a generator/discriminator pair to generate perceptually realistic and diverse samples when given a single text description.

**Qualitative results.** Subjective visual evaluation between DF-GAN, DTGAN, DF-GAN + FC(2), DTGAN + FC(2), F5, F6 (F5 + CA), F7 (F5 + MS), F8 (F5 + FC(1)), F9 (F5 + FC(1+1)) and F10 (DiverGAN) is presented in Fig. 3.8. DF-GAN + FC(2) and DTGAN + FC(2) refer to DF-GAN and DTGAN plugging a linear layer after the second residual block of the architecture, respectively.

We can observe that, although both DF-GAN (1st row) and DTGAN (2nd row) perform well in quality, the shapes of the synthetic birds look similar and the background colors are the same. However, after inserting a dense layer into the framework, DF-GAN (3rd row) and DTGAN (4th row) both yield more diverse birds (e.g., different shapes, orientations and even background colors) than the original frameworks, demonstrating the generalizability of our proposed method. The reason behind similar backgrounds for DF-GAN + FC(2) may be that the background colors in DF-GAN only depend on the textual embeddings due to the introduction of the modulation module. By comparing DiverGAN (9th row) with F6 (F5 + CA) and F7 (F5 + MS), we can see that a linear layer contributes to producing diverse birds with vivid details, whereas CA does not improve the diversity of birds and MS may affect the image quality. For example, in the 9th row, the birds are on the branch or ground, and the orientations of birds, background colors and the visual appearances of footholds are different. Nonetheless, the
A colorful <color> bird has wings with dark stripes and small eyes

Figure 3.9: Generated samples of DiverGAN by changing the color attribute value in the input text description and the random latent codes, respectively.

birds in the 5th row still have similar shapes and the same background color, while the birds in the 6th row look a little blurry. By comparison with F5, we can observe that F8 (F5 + FC(1)), F9 (F5 + FC(1+1)) and DiverGAN all generate diverse samples, further confirming the effectiveness of the FC in diversity. In addition, DiverGAN generates realistic birds with higher quality than F6 and F7, which validate the superiority of DiverGAN.

To validate the sensitivity of DiverGAN, we generate birds by modifying just one word in the given text description. As can be seen in Fig. 3.9, when we change the color attribute in the natural-language description, the proposed DiverGAN further produces semantically consistent birds according to the modified text while retaining visual appearances (e.g., shape, position and texture) correlated with the unmodified parts. Additionally, our method synthesizes a suite of birds with different visual appearances of footholds, background colors, orientations and shapes by changing latent codes. Therefore, with word-level attention modules and the FC method, DiverGAN is able to effectively disentangle attributes of the input text description while accurately controlling regions of the sample without hurting diversity.

3.4.3.5 Interpolation of latent space in DiverGAN

To better understand how DiverGAN utilizes latent codes to achieve diversity, we conduct linear interpolation between two random latent codes and produce
Figure 3.10: Four image synthesis results with linear interpolation between two random latent codes in DiverGAN on the CUB data set. This visual analysis of bipolar contrasts in feature space shows a fluent transition of both foreground and background features.

Figure 3.11: Four image synthesis results with linear interpolation between two random latent codes in DiverGAN on the MSCOCO data set.
corresponding pictures. The interpolation results of DiverGAN on the CUB data set are presented in Fig. 3.10. We can see that both the background and the visual appearances of footholds (1st and 4th row), both the shapes and the positions of birds (2nd row), the orientations and the shapes of birds and the visual appearances of footholds (3rd row) change gradually with the variances of latent codes. Fig. 3.11 visualizes the interpolation samples generated by DiverGAN trained on the MSCOCO data set. It can be observed that the tower appearance (1st row), the beach (2nd row), people (2nd and 3rd row), waves (3rd row), grass and animals (4th row) vary continuously along with latent vectors. The above results show a smooth transition of both foreground objects and background appearances. More importantly, we discover that DiverGAN is likely to yield a series of high-quality images when the samples conditioned on the first and the last latent codes are visually realistic. At the same time, interpolation results often look blurry if DiverGAN is not able to generate plausible samples according to these two latent codes, which we detail in the next chapter. Therefore, we assume that DiverGAN may synthesize high-quality pictures based on the latent codes around a ‘good’ latent code enabling DiverGAN to yield a realistic sample. Our next challenge is to find such ‘good’ latent codes in order to facilitate the synthesization of large numbers of satisfactory results, e.g., for the purpose of data augmentation.

3.5 CONCLUSION

In this chapter, we propose a unified and effective single-stage framework called DiverGAN for yielding diverse and perceptually realistic samples which are semantically related to given textual descriptions. DiverGAN exploits two new types of word-level attention modules, i.e., a channel-attention module (CAM) and a pixel-attention module (PAM), to make the generator concentrate more on useful channels and pixels that semantically match with the salient words in the natural-language description. Afterwards, a dual-residual block is employed to accelerate convergence speed while enhancing image quality. Furthermore, a fully-connected layer is introduced into our architecture to combat the lack-of-diversity problem by enhancing the generative ability of the network. Extensive experiments on three benchmark data sets show that DiverGAN achieves remarkably better performance than existing methods in quality and diversity. Furthermore, our presented components, i.e., CAM and PAM, are general methods and can be readily integrated into current text-to-image architectures to reinforce feature maps with textual-context vectors. More importantly, our proposed pipeline tackles the lack-of-diversity issue existing in the current single-stage methods and can serve as a strong basis for developing better single-stage text-to-image models. One pervasive problem in the evaluation of the performance of image-generation
methods is the degree of subjectivity involved. In case of human evaluations, more effort should be spent in guiding the human attention to aspects of the image: “Is the content semantically consistent with the text probe?” but also: “Is the background pattern believable/natural?”. Therefore, in future research, a focus is needed on extending the questionnaire for the human subjects to ask more precisely what they think about the backgrounds of generated samples.
Part II

EXPLAINABLE CONDITIONAL TEXT-TO-IMAGE TRANSFORMS