Exploring Global and Local Linguistic Representations for Text-to-Image Synthesis

Ruifan Li, *Member, IEEE*, Ning Wang, Fangxiang Feng, Guangwei Zhang, and Xiaojie Wang

***Abstract*—The task of text-to-image synthesis is to generate photographic images conditioned on given textual descriptions. This challenging task has recently attracted considerable attention from the multimedia community due to its potential applications. Most of the up-to-date approaches are built based on generative adversarial network (GAN) models, and they synthesize images conditioned on the global linguistic representation. However, the sparsity of the global representation results in training difficulties on GANs and a shortage of fine-grained information in the generated images. To address this problem, we propose cross-modal global and local linguistic representations-based generative adversarial networks (CGL-GAN) by incorporating the local linguistic representation into the GAN. In our CGL-GAN, we construct a generator to synthesize the target images and a discriminator to judge whether the generated images conform with the text description. In the discriminator, we construct the cross-modal correlation by projecting the image representations at high and low levels onto the global and local linguistic representations, respectively. We design the hinge loss function to train our CGL-GAN model. We evaluate the proposed CGL-GAN on two publicly available datasets, the CUB and the MS-COCO. The extensive experiments demonstrate that incorporating fine-grained local linguistic information with cross-modal correlation can greatly improve the performance of text-to-image synthesis, even when generating high-resolution images.**

***Index Terms*—Text-to-image synthesis, generative adversarial network (GAN), linguistic representation, cross-modal.**

Manuscript received July 23, 2019; revised December 18, 2019; accepted February 2, 2020. Date of publication February 10, 2020; date of current version November 18, 2020. This work was supported in part by the National Key R&D Program of China under Grant 2019YFF0303300 and Subject II under Grant 2019YFF0303302, in part by the National Natural Science Foundation of China under Grants 61802026 and 61906018, and in part by the Science and Technology Program of the Headquarters of State Grid Corporation of China under Grant 5200-201918255A-0-0-00, and in part by the 111 Project under Grant B08004. The associate editor coordinating the review of this manuscript and approving it for publication was Professor Chi-Chun Lee. *(Corresponding author: Ruifan Li.)*

Ruifan Li and Xiaojie Wang are with the School of Computer Science, and the Engineering Research Center of Information Networks, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: rfli@bupt.edu.cn; xjwang@bupt.edu.cn).

Ning Wang and Fangxiang Feng are with the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: niwang@bupt.edu.cn; f.fangxiang@gmail.com).

Guangwei Zhang is with the Institute of Network Technology, and the Engineering Research Center of Information Networks, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: gwzhang@bupt.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2020.2972856

I. Introduction

IMAGINE that a person describes a picture that he or she wants to draw with a few sentences and—all of a sudden—that imaginary drawing comes into being. This type of illusory daydream has become an attractive topic in the multimedia community today. The topic of text-to-image synthesis focuses on automatically generating photographic images conditioned on specific unstructured textual descriptions [1]–[8]. From a novel research point of view, text-to-image synthesis has broad application prospects, including opportunities in the photo-editing and computer-aided design. Similar to other emerging topics in the multimedia community, such as cross-modal retrieval [9]–[12] and automatic image (video) captioning [13]–[15], the text-to-image synthesis task also emphasizes the cross-modal correlations between textual and image data. Specifically, the text-to-image synthesis is a challenging problem primarily because the distribution of images conditioned on a text description is highly flexible. In other words, there exist a large number of plausible configurations of pixels for an image that correctly illustrate the given text description.

Approaches [1]–[8] based on generative adversarial networks (GAN) [16] have recently achieved promising results on the text-to-image synthesis task. The basic idea behind a GAN is essentially a minimax game mechanism between two players, one player is a generator and the other is a discriminator. The generator attempts to synthesize artificial samples sufficiently good to fool the discriminator. In contrast, the discriminator attempts to distinguish whether a sample is drawn from the real data distribution or is artificially generated by the generator. To solve the task of text-to-image synthesis, a type of conditional GANs [17] is utilized to learn a mapping directly from the linguistic representations to the visual image pixels. Specifically, the text description is encoded into a global linguistic representation using a recurrent neural network (RNN), such as an LSTM [18]. Then, the global linguistic encoding is incorporated into the discriminator as conditional information. Moreover, after being trained against the discriminator, the generator is able to generate pictures corresponding to textual descriptions.

However, generating high-resolution images (over 128 × 128 pixels) under the aforementioned idea of constructing GANs conditions solely on global linguistic representations proves to be unsatisfactory. Typically, the pioneering work of Reed *et al.* [1] can generate only fake images of 64 × 64 pixels, and those fake images usually lack details and vivid object parts. When generating higher-resolution images, GANs-based models often result in unstable training and produce nonsensical results. This result occurs mainly because the two distributions of the natural images and the implicit model may not overlap in the high-dimensional pixel space [19]. This problem becomes even more severe when the natural image distribution is conditioned on a sparse manifold, which is very common when encoding the entire descriptive sentences into a global sentence representation.

In this article, we explore the idea of leveraging both global and local linguistic representations to augment the task of text-to-image synthesis. In particular, we propose cross-modal global and local linguistic representations-based generative adversarial networks (CGL-GAN). A generator is constructed to synthesize target images, and a discriminator is trained to judge whether the generated images are from the real data distribution and conform with the textual descriptions. Notably, in our discriminator, the image is encoded into low-level and high-level representations. The low-level image representation is projected onto the local linguistic representation, and the high-level image representation is projected onto the global linguistic representation. Thus, the discriminator captures the semantic consistency between the generated images and given textual descriptions while maintaining the high visual quality of images. We design a hinge loss function to train the CGL-GAN model.

We evaluate our proposed approach on two publicly available datasets, the CUB and MS-COCO datasets. We conduct extensive experiments by exploring the global and local linguistic representations in the task of text-to-image synthesis. Extensive experiments of quantitative and qualitative analyses are given. Compared with other state-of-the-art methods, our CGL-GAN model achieves comparable performances with fewer trainable parameters. Furthermore, when compared with the single-GAN models, our model achieves higher scores. In addition, we conduct experiments on four variants of the cross-modal projection block. The results demonstrate that the appropriate correlation between visual and linguistic representations can effectively improve the text-to-image task performance. Overall, the CGL-GAN model can generate high-resolution images based on fewer trainable parameters while achieving higher evaluation scores comparable to the leading models, which are powered by multiple GANs. To advance the knowledge sharing, we make our source code publicly available.[1]

The remainder of this article is organized as follows. We first provide a review of related works in Section II. Then, we propose our model for text-to-image synthesis in detail in Section III, including an overview of the model, the text encoder, generator, discriminator and loss function. In Section IV, we introduce the datasets, report on our experiments and analyze their results. Finally, in Section V we draw conclusions and suggest future work directions.

II. Related Work

In this section, we review the related work from two main viewpoints: related works on GANs and advances on the text-to-image synthesis task.

A. Generative Adversarial Networks

GANs, a family of generative network models, have recently attracted considerable interest from the AI community. Notably, GANs have attracted high interest from researchers who strive to synthesize images automatically. The basic idea of a GAN is to train the generator network module by playing a two-player minimax game mechanism. Specifically, a plain-vanilla GAN consists of a generator G, and a discriminator D. Those two players are alternatively trained to compete with each other. The discriminator D is optimized to distinguish synthesized images from real images, while the generator G is trained to synthesize fake images that can fool the discriminator D. Mathematically, the primary training objective can be formulated as a minimax optimization problem. Optimizing the value function V for the generator and the discriminator is defined as follows:

$$\min_G \max_D V(G, D) = \mathbb{E}_{p_{data}(\boldsymbol{x})}\left[\log D(\boldsymbol{x})\right] + \mathbb{E}_{p_z(\boldsymbol{z})}\left[\log(1 - D(G(\boldsymbol{z})))\right], \quad (1)$$

where the symbol $\mathbb{E}[\cdot]$ denotes the expectation operator, and the subscript represents the corresponding integral distribution. The noise vector $\boldsymbol{z}$ is sampled from the Gaussian or uniform distributions. The optimal generator and discriminator are obtained through a minimax optimization problem. Mathematically, Goodfellow *et al.* [16] have shown that when the generator and the discriminator have sufficient capacity, the generated distribution converges to the real data distribution.

The seminal work of Goodfellow *et al.* [16] have provided a theoretical framework for GANs using deep neural networks to generate images without incorporating the supervised information. Subsequently, some studies on generating virtual images have been conducted using the GAN framework [17], [20]–[26]. In general, two main GAN directions exist according to whether they can be trained with or without the supervised information. The first group of methods does not rely on the supervised information. Radford *et al.* [20] have proposed a deep convolutional GAN for representation learning. Those methods have produced somewhat noisy and blurry images but have shown that GANs form a promising direction for synthesizing images. To achieve better synthesized image quality, Denton *et al.* [21] have introduced the Laplacian pyramid into GANs to produce high quality images. Furthermore, Zhao *et al.* [22] and Berthelot *et al.* [23] have stabilized the GAN training from an energy point of view.

The second group of methods uses certain supervised information. For example, Mirza and Osindero [17] have proposed conditional GANs by inputting class labels into networks. Later, Odena *et al.* [24] have expanded the GAN discriminator by adding an auxiliary classifier for generating synthetic images. The auxiliary classifier shares a part of the discriminator layers, and incorporates the label information with likelihood scores into the objective function of the discriminator. Very recently, Song *et al.* [25] have proposed binary GANs using an autoencoder and neighbor structure to produce images. GAN-based methods that include supervised information can produce virtual images with global coherence and high diversity. In these methods, the supervised information is generally input into GANs with noise or feature maps in a concatenated manner. The

[1] [Online]. Available: http://github.com/bupt-mmai/txt2im

concatenation methods are arbitrary and it has proven difficult to find their underlying logical basis. Very recently, Miyato and Koyama [26] have proposed an alternative method that uses a projection discriminator, which greatly increased the number of generated image categories.

Another interesting GAN direction involves how to address the problem of training instability. Notably, Arjovsky *et al.* [27] have provided a novel loss function based on the Wasserstein distance. During GAN training, one requirement for this loss function is that the discriminator mapping function must satisfy the Lipschitz continuity. Specifically, the Lipschitz continuity is a smoother condition than are the other commonly used continuities. For a function $f : \mathbb{R} \rightarrow \mathbb{R}$, it requires a positive real constant C that ensures that any two elements x_1 and x_2 in the domain match the following condition, i.e., $|f(x_1) - f(x_2)| \leq C|x_1 - x_2|$. Moreover, in [27], the authors have shown how to ensure the Lipschitz continuity by clipping the gradient, which is not desirable. Gulrajani *et al.* [28] have proposed a type of gradient penalty to reduce the adverse effects of clipping the gradient. Very recently, Miyato *et al.* [29] have proposed an efficient method that uses spectral normalization to ensure the Lipschitz continuity. This work has demonstrated that the Lipschitz continuity can be satisfied by dividing the parameters of each layer by the spectral norm of the layer parameter matrix.

B. Text-to-Image Synthesis

Recently, the task of text-to-image synthesis has attracted considerable attention. Conventional image synthesis tasks are performed with label information based on GANs [30]–[33]. However, in the text-to-image synthesis task, the supervised information is no longer the label information but textual descriptions. In the original work, Reed *et al.* [1] have constructed a conditional GAN-based network to generate images with textual descriptions. Generally, there are two interesting problems in this task.

The first challenge is how to generate higher quality images. Notably, Zhang *et al.* [3] have decomposed this task into two manageable subproblems through a sketch-refinement process with multiple GANs. One subproblem involves sketching the primitive shapes and colors; the other involves generating high-resolution images with photo-realistic details. This multiple-GAN architecture has become a classical architectural form for solving high-resolution text-to-image generation tasks. However, this approach has some drawbacks. For example, the generated images are relatively small and the network architecture is not end-to-end. To address those problems, Zhang *et al.* [4] have proposed an end-to-end model with a tree architecture. Moreover, Zhang *et al.* [5] have proposed using hierarchical-nested adversarial objectives inside the network hierarchies to generate high-resolution photographic images. Xu *et al.* [6] have focused on the fine-grained text-to-image generation and proposed an attention-driven method to improve the fine-grained details of the generated images. Most recently, Qiao *et al.* [7] have proposed MirrorGAN by leveraging the image-to-text task to enhance the semantic consistency for generated images.

The second challenge involves how to effectively construct plausible spatial relationships among objects. The generation of reasonable images through incorporating additional information such as bounding boxes is worth considering. Reed *et al.* [2] have proposed using additional location information for the text-to-image generation task through a type of "what-where" GANs. Hong *et al.* [30] have proposed a hierarchical approach to generate object bounding boxes and have refined each box by estimating the potential object shapes inside the box. Johnson *et al.* [31] have introduced the semantic graph concept into the text-to-image synthesis task. GANs conditioned on the semantic graph rather than the sentence embedding vectors can generate more complex images even when the textual descriptions are quite long. Although a semantic graph has some advantages when used as the conditional information for a GAN, obtaining such graphical information is both more complex and more difficult. Recently, Li *et al.* [8] have proposed a novel object-driven attention mechanism by incorporating a pre-generated semantic layout to train the model, learning the reasonable relationship among multiple objects in the scene.

III. Model

A. Overview

Fig. 1 shows an overview of our proposed model CGL-GAN. Our model consists of three main components, a text encoder, a generator and a discriminator. The text encoder transforms textual descriptions into global and local linguistic representations. We discuss the text encoder in detail in Section III-B. The generator takes the global linguistic information and random Gaussian noises as input and outputs corresponding images. We discuss our generator in detail in Section III-C. The discriminator determines whether input images are both from the real dataset and conforming with the corresponding textual descriptions. To this end, we propose a cross-modal projection block in our discriminator to project the image representations onto the linguistic representations. We illustrate our discriminator and the cross-modal projection block in detail in Section III-D. Finally, we adopt a hinge loss function to train our CGL-GAN model. This hinge loss function will be discussed in detail in Section III-E. After training against the discriminator, the generator can generate corresponding images from textual descriptions.

B. Text Encoder

To capture the dependencies, we feed the words in a sentence into a bidirectional long short term memory (LSTM) model in sequence. Thus, each word corresponds to two hidden states of the bidirectional LSTM, one for each direction. We combine these hidden states into a matrix $\boldsymbol{y}^{\mathcal{L}}$, forming our local representation. Each row of the matrix is copied from the hidden states of the corresponding word in the sentence. Suppose that a sentence has n words, and the hidden states dimensionality of the bidirectional LSTM is d. Then, the local representation resides in a space, i.e., $\boldsymbol{y}^{\mathcal{L}} \in \mathbb{R}^{n \times d}$. In practice, we add padding to fix all the sentences to a constant length of n. After all n words have been fed into the bidirectional LSTM, we use the last hidden

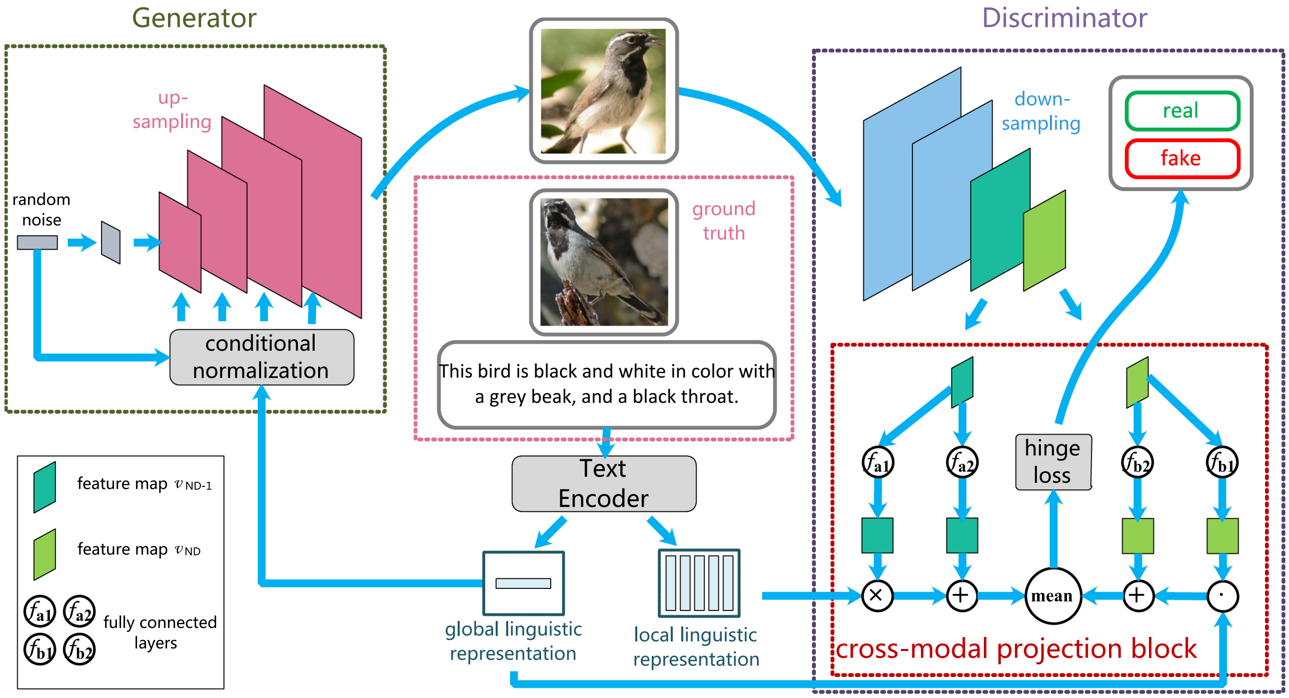
Generator
up-sampling
random noise
conditional normalization
ground truth
This bird is black and white in color with a grey beak, and a black throat.
Text Encoder
global linguistic representation
local linguistic representation
Discriminator
down-sampling
real
fake
f_{a1}
f_{a2}
hinge loss
f_{b2}
f_{b1}
mean
cross-modal projection block
feature map v_{ND-1}
feature map v_{ND}
f_{a1} f_{a2} f_{b1} f_{b2} fully connected layers

Fig. 1. The architecture of our proposed CGL-GAN model. The top left is the generator module. The text description is encoded through the text encoder module as global and local representations in the middle. The right is the discriminator, in which the cross-modal projection block is depicted. The legend is provided at the bottom left.

states as our global linguistic representation $\boldsymbol{y}^{\mathcal{G}}$. Thus, the global linguistic representation $\boldsymbol{y}^{\mathcal{G}}$ has a dimensionality of d, namely, $\boldsymbol{y}^{\mathcal{G}} \in \mathbb{R}^{1\times d}$. The text encoder we adopted is a pre-trained bidirectional LSTM, as used in the AttnGAN model [6]. In our setting, the text encoder is frozen with pre-trained parameters during the entire training procedure.

C. Generator

The generator in our CGL-GAN component is responsible for producing a corresponding image driven by the global linguistic representation. Our generator consists mainly of several up-sampling blocks [34] and conditional normalization layers [35]. We next describe these two modules as follows.

Specifically, each up-sampling block is built with ResNet [36] based convolutional layers followed by nearest-neighbor up-sampling layers. Suppose that we have N_G up-sampling layers. We denote all the up-sampling blocks of our generator as follows: $\boldsymbol{U} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N_G}\}$. Each up-sampling block receives the feature maps from the previous up-sampling block and outputs double-sized feature maps. However, before being input to the up-sampling blocks, these feature maps are first incorporated with conditional information through a conditional normalization operation. Conditional normalization evolves from batch normalization, which is usually calculated by the following equation:

$$N_{\mathcal{B}}(\boldsymbol{x}) = \frac{\boldsymbol{x} - \mathbb{E}(\boldsymbol{x})}{\sqrt{\mathbb{V}(\boldsymbol{x}) + \epsilon}} \times \boldsymbol{\gamma} + \boldsymbol{\alpha}. \tag{2}$$

The input feature map $\boldsymbol{x}$ is first subtracted from its mean value $\mathbb{E}(\boldsymbol{x})$, and then is divided by its standard deviation $\sqrt{\mathbb{V}(\boldsymbol{x})}$. A small constant ϵ is added to prevent division by zero. The parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ are trainable and used to scale and translate the final feature map, respectively. The idea behind conditional normalization is to use the conditional information to control the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$. In this article, we first batch-normalize the feature map $\boldsymbol{x}$ through equation 2, and then scale and translate the output $N_{\mathcal{B}}(\boldsymbol{x})$ using the parameters $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$ as follows:

$$N_{\mathcal{C}}(\boldsymbol{x}) = N_{\mathcal{B}}(\boldsymbol{x}) \times \boldsymbol{\eta} + \boldsymbol{\beta}. \tag{3}$$

In order to obtain the trainable parameters $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$, we concatenate the global linguistic representation $\boldsymbol{y}^{\mathcal{G}}$ with random noise. Then we feed them into a fully-connected layer. Subsequently, the output of the fully-connected layer is divided into two parts: one part is $\boldsymbol{\eta}$ and the other part is $\boldsymbol{\beta}$.

To generate the pixel distribution of the target image, the generator needs to decode from a specific probability distribution. Therefore, we input a set of random noise $\boldsymbol{Z} = \{\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{N_G}\}$ with a standard multi-dimensional Gaussian distribution (i.e., $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{E})$) into the generator. Here, we have a zero mean vector $\boldsymbol{0}$ and an identity covariance matrix $\boldsymbol{E}$. We reshape the random noise $\boldsymbol{z}_0$ as the initial feature map and feed it into the first up-sampling block $\boldsymbol{u}_1$. Next, we concatenate each of the remaining random noises $\{\boldsymbol{z}_i\}_{i=1}^{N_G}$ with the global linguistic representation $\boldsymbol{y}^{\mathcal{G}}$. Then, the concatenated representation is incorporated into the corresponding up-sampling block $\boldsymbol{u}_i$ through the conditional normalization process [35]. Finally, we use a convolutional layer to decode the high-dimensional feature maps from $\boldsymbol{u}_{N_G}$ into a three-dimensional RGB image as the final output of our generator. To regularize the Lipschitz constant, we apply spectral normalization [29] to all the weights in our generator.

D. Discriminator

Our discriminator plays two important roles. On one hand, it is responsible for determining whether an image is real or fake. On the other hand, it decides whether the image and the textual description are semantically correlated. Our discriminator consists mainly of several down-sampling blocks [34] and a cross-modal projection block (CPB).

These down-sampling blocks can be regarded as image encoders; they encode the input image into a high-dimensional feature map. We use ResNet [36] based convolutional layers to implement our down-sampling blocks. A down-sampling block consists of convolutional layers and average pooling layers. Suppose we have N_D down-sampling blocks. We denote all the down-sampling feature maps as $\boldsymbol{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_{N_D}\}$.

We propose a cross-modal projection block to capture the semantic correlations between visual and linguistic representations. In addition, we consider the last two layer of feature maps most likely form the semantic concepts. Specifically, the visual feature map $\boldsymbol{v}_{N_D}$ is projected onto the global linguistic representation $\boldsymbol{y}^{\mathcal{G}}$, and the visual feature map $\boldsymbol{v}_{N_D-1}$ is projected onto the local linguistic representation $\boldsymbol{y}^{\mathcal{L}}$. The basic idea behind our CPB is that the feature map $\boldsymbol{v}_{N_D}$ is a high-level image representation that is semantically closer to the global linguistic representation $\boldsymbol{y}^{\mathcal{G}}$. Similarly, the feature map $\boldsymbol{v}_{N_D-1}$ is a low-level image representation that it is semantically closer to the local linguistic representation $\boldsymbol{y}^{\mathcal{L}}$. We devise the cross-modal projection operation to correlate visual representations and linguistic representations.

Specifically, the cross-modal projection operation first copies the feature map and then feeds the original feature map and the copy into two different fully-connected layers. Then, one output of the two fully-connected layers is multiplied by the linguistic representations. Finally, the CPB outputs the sum of the element-wise average of those two products. Since the global and local linguistic representations have two different formats, i.e., one vector and one matrix, we adopt different projection methods for each of them. The local linguistic representation $\boldsymbol{y}^{\mathcal{L}}$ is projected onto the low-level image representation $\boldsymbol{v}_{N_D-1}$ through matrix multiplication, denoted as $\times$. The global linguistic representation $\boldsymbol{y}^{\mathcal{G}}$ is projected onto the high-level image representation $\boldsymbol{v}_{N_D}$ through element-wise multiplication, denoted as $\odot$. The cross-modal projection operations are expressed by the following equations:

$$P(\boldsymbol{v}_{N_D-1}, \boldsymbol{y}^{\mathcal{L}}) = f_{a1}(\boldsymbol{v}_{N_D-1}) \times \boldsymbol{y}^{\mathcal{L}} + f_{a2}(\boldsymbol{v}_{N_D-1}), \tag{4}$$

and

$$Q(\boldsymbol{v}_{N_D}, \boldsymbol{y}^{\mathcal{G}}) = f_{b1}(\boldsymbol{v}_{N_D}) \odot \boldsymbol{y}^{\mathcal{G}} + f_{b2}(\boldsymbol{v}_{N_D}), \tag{5}$$

in which, $f_{a1}(\cdot)$ and $f_{a2}(\cdot)$ are the two fully-connected layers for feature map $\boldsymbol{v}_{N_D-1}$, while $f_{b1}(\cdot)$ and $f_{b2}(\cdot)$ are the two fully-connected layers for feature map $\boldsymbol{v}_{N_D}$. The output of our discriminator can be expressed as follows,

$$D(\boldsymbol{I}, \boldsymbol{y}^{\mathcal{L}}, \boldsymbol{y}^{\mathcal{G}}) = \frac{1}{N_P}\sum_{i=1}^{N_P} P_i + \frac{1}{N_Q}\sum_{j=1}^{N_Q} Q_j, \tag{6}$$

where the dimensionality of the two projection vectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ are denoted as N_P and N_Q, respectively. The subscripts i and j are the indexes of the dimensionality. $\boldsymbol{I}$ denotes the image from the repository containing real and fake images.

Our CPB feeds both the global and local linguistic representations into the GAN discriminator as the conditional information. This approach provides the gradients for training our entire text-to-image synthesis model. Moreover, by adjusting the CPB, we can control the correlations between the visual and language representations. Four types of correlations among the four types of visual and language representations are described as follows: 1) conditioned only on global linguistic information (CGL-GAN-OG); 2) conditioned only on local linguistic information (CGL-GAN-OL); 3) projecting the low-level image representation onto the global linguistic representation and projecting the high-level image representation onto the local linguistic representation (CGL-GAN-GL); and 4) projecting the high-level image representation onto the global linguistic representation and projecting the low-level image representation onto the local linguistic representation (CGL-GAN-LG). All four are depicted in Fig. 2. Note that among these four variants, the CGL-GAN-LG version is adopted in our model. We will evaluate these four different CPB forms and report their performances in detail in Section IV.

E. Loss Function

We design a hinge loss to train our CGL-GAN. Theoretically, a GAN with a hinge loss converges efficiently toward a Nash equilibrium between the generator and discriminator. Moreover, this type of GAN models has shown promising results [29], [37]. We describe our hinge loss as follows:

$$\begin{aligned} L\left(\widehat{G}, D\right) = {} & \max\left(0, 1 - D\left(\boldsymbol{I}^{\mathcal{R}}, \boldsymbol{y}^{\mathcal{G}}, \boldsymbol{y}^{\mathcal{L}}\right)\right) \\ & + \max\left(0, 1 + D\left(\widehat{G}(\boldsymbol{z}, \boldsymbol{y}^{\mathcal{G}}), \boldsymbol{y}^{\mathcal{G}}, \boldsymbol{y}^{\mathcal{L}}\right)\right), \end{aligned} \tag{7}$$

and

$$L(G, \widehat{D}) = -\widehat{D}\left(G(\boldsymbol{z}, \boldsymbol{y}^{\mathcal{G}}), \boldsymbol{y}^{\mathcal{G}}, \boldsymbol{y}^{\mathcal{L}}\right). \tag{8}$$

In the two preceding equations, $\boldsymbol{z}$ is a random noise vector with a standard Gaussian distribution. In equation (7), $\boldsymbol{I}^{\mathcal{R}}$ denotes the image from the real data distribution, while $\widehat{G}(\boldsymbol{z}, \boldsymbol{y}^{\mathcal{G}})$ denotes the image generated by the generator G. Here, $\widehat{G}$ means that the parameters of the generator G are frozen while training the discriminator D. The value $D(\boldsymbol{I}^{\mathcal{R}}, \boldsymbol{y}^{\mathcal{G}}, \boldsymbol{y}^{\mathcal{L}})$ is the discriminator output, which is the sum of the element-wise average of the two outputs $\boldsymbol{P}$ and $\boldsymbol{Q}$ from the CPB as discussed in Section III-D. In equation (8), $G(\boldsymbol{z}, \boldsymbol{y}^{\mathcal{G}})$ is the image generated by the generator G, and $\widehat{D}$ means that the parameters of the discriminator D are frozen while training the generator G. Once the entire model, including the generator and the discriminator has been trained, the discriminator is discarded, and only the generator is retained and used to generate images from textual descriptions.

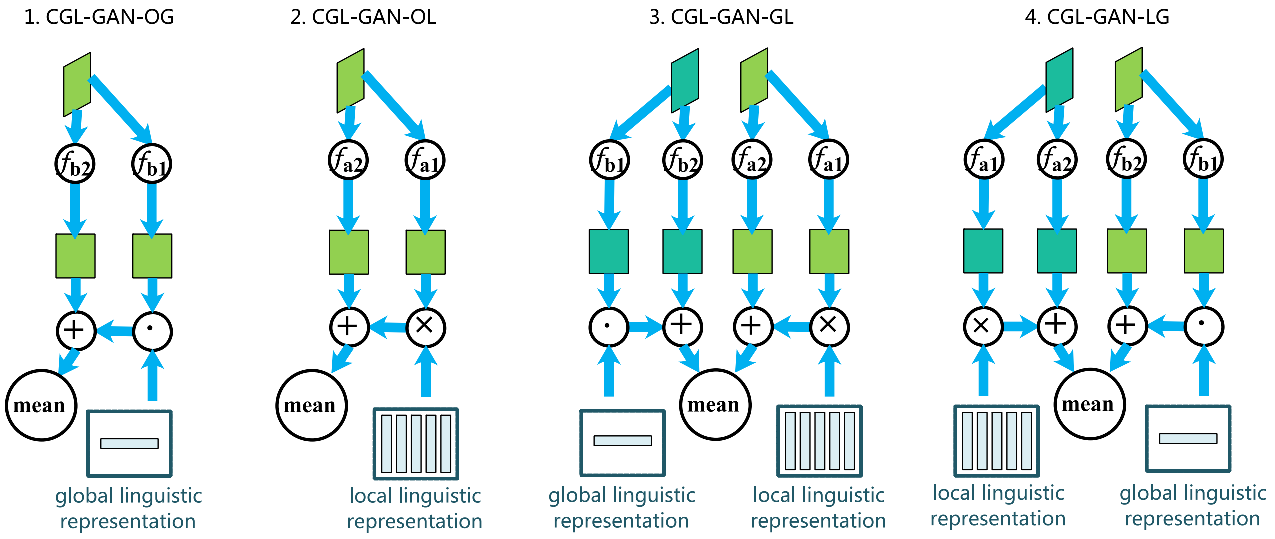

Fig. 2. Comparison on four different versions of CPBs in our discriminator. The left two versions use either global or local linguistic representations. In contrast, the right two versions use those two types of linguistic representations.

IV. Experiment and Analysis

In this section, we evaluate our proposed model by performing the text-to-image synthesis task on two publicly available datasets. In the following, we first describe the adopted CUB and MS-COCO datasets in Section IV-A. Then, we illustrate the evaluation protocol in detail in Section IV-B. Subsequently, we introduce the baseline methods adopted in our experiments and provide the implementation details in Sections IV-C and IV-D, respectively. We report the experimental results in Sections IV-E and IV-F. Finally, we provide an analysis on CPB module in Section IV-G.

A. Dataset

In order to evaluate our proposed approach, we adopt two publicly available datasets, the CUB [38] and the MS-COCO [39]. The details and usage of these two datasets are described as follows. The CUB dataset, i.e., the Caltech-UCSD Birds-200-2011 version, is a bird-species photo dataset. This dataset is built by the California Institute of Technology and is made publicly available.[2] Specifically, the CUB dataset contains 11,788 images with a bounding box of 200 categories of birds. All the images are officially partitioned into two subsets: one for training with 8,855 images and the other for testing with 2,933 images. No validation set is provided. For the task of text-to-image synthesis, Reed *et al.* [1] have provided ten descriptive sentences for each image based on the colors and other characteristics of the bird within that image. To ensure a fair comparison, we follow Zhang *et al.* [3] performing a preprocessing step. Here, we crop all images to make sure that the object-image size ratios are greater than 0.75 referring to the bounding boxes of birds. Implementation details are available at source code.[3]

The MS-COCO dataset, different from the CUB dataset with only birds, contains images of multiple objects with a variety of backgrounds. The entire dataset has been officially partitioned into two subsets: a training set with 80,000 images and a testing set with 40,000 images. No validation set is provided. Each image in the MS-COCO dataset includes five descriptive sentences, only half of the CUB dataset. This MS-COCO dataset can be freely downloaded.[4] Open application programming interfaces (APIs) that can assist in loading, parsing and visualizing the annotations for the MS-COCO dataset are publicly available.[5] Note that compared with the CUB dataset, it is considerably more challenging to train a generative model effectively on the MS-COCO dataset. Samples from these two datasets are shown in the subsequent experiments.

B. Evaluation Metric

In this article, we use the Inception score [40] and Fréchet Inception Distance (FID) [41] to quantitatively evaluate the quality of the generated images by our proposed model.

The Inception score considers evaluating a generative model of text-to-image synthesis from two aspects: the clarity and the diversity. Intuitively, images with a better clarity should have a sharp posterior distribution $p(y|x)$. Moreover, images with a diversity should have an even marginal distribution $p(y)$ over class labels. Those two assumptions are combined using Kullback-Leibler divergence D_{KL}, which measures how one probability distribution is different from the reference probability distribution. Mathematically, the Inception score S is calculated as follows,

$$S = \exp\left(\mathbb{E}_{x\sim p_G}\left[D_{KL}\left(p(y|x) \parallel p(y)\right)\right]\right). \tag{9}$$

In this equation, $\exp(\cdot)$ is an exponential function. $\mathbb{E}[\cdot]$ denotes the expectation operator, in which the subscript $x \sim p_G$ means

[2] [Online]. Available: http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

[3] [Online]. Available: http://github.com/hanzhanggit/StackGAN-Pytorch

[4] [Online]. Available: http://cocodataset.org/#download

[5] [Online]. Available: http://github.com/cocodataset/cocoapi

drawing images from the generator. The larger expected KL divergence represents better performance. In practice, the posterior distribution $p(y|x)$ and the marginal distribution $p(y)$ are obtained through classifying generated images using Google's Inception Net V3. However, the Inception score highly depends on the training dataset, e.g., ImageNet. For a fair comparison, we use the fine-tuned Inception score proposed in [3] for the task of text-to-image synthesis. This implementation code of Inception score is publicly available.[6]

The FID metric was proposed later than the Inception score for the task of text-to-image synthesis. It evaluates the quality of an image generator by measuring the difference between two distributions of the real images and the generated images. The distributions of those two types of images are taken as multivariate Gaussian distributions with different parameters. The lower FID value of a model means a better performance. Formally, the FID value F is calculated as follows,

$$F = \| \boldsymbol{\mu}_r - \boldsymbol{\mu}_g \|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{1/2}\right). \quad (10)$$

In this equation, $\boldsymbol{\mu}_r$ and $\boldsymbol{\mu}_g$ are the mean vectors of features of the real images and the generated images, respectively. Correspondingly, $\boldsymbol{\Sigma}_r$ and $\boldsymbol{\Sigma}_g$ are the covariance matrices. In addition, $\| \cdot \|_2$ denotes the norm operator on a vector. $\mathrm{Tr}(\cdot)$ denotes the trace operator on a matrix. The image features are extracted using the aforementioned Google's Inception Net V3. Compared with the Inception score, FID is a more reasonable evaluation metric. To emphasize, FID compares the generated image directly with the real images, not with images of the ImageNet in the Inception score. In this article, we use the official implementation on the TensorFlow platform for calculating the values of FID.[7]

C. Baseline Model

In this article, we adopt eight baseline models for comparison. These models are listed and briefly reviewed as follows.

1) GAN-INT-CLS [1] originally adopts the GAN framework for the task of text-to-image synthesis. This model is built based on a single GAN and can generate images with the size of 64×64 only.
2) GAWWN [2] extends the GAN-INT-CLS model. By using key points and bounding boxes, the GAWWN model can generate images with 128×128 pixels.
3) StackGAN [3] is a progressive model based on multiple GANs. Therefore, the StackGAN model trained with a two-stage process can generate high-resolution images with a size of 256×256.
4) StackGAN-V2 [4] extends the StackGAN model with an end-to-end training. In addition, some problems are fixed in the StackGAN. Thus, this model can achieve better performance.
5) HDGAN [5] utilizes a single-stream generator and hierarchically-nested multiple discriminators. Trained with an end-to-end manner, the HDGAN model can generate high resolution images.
6) AttnGAN [6] is the first attention mechanism driven model. The model synthesizes different subregions of an image by attending the relevant words.
7) MirrorGAN [7] emphasizes the semantic consistency using the idea of re-description. A text generation module is appended to consolidate this consistency.
8) Obj-GAN [8] involves a two-step generation process, i.e., layout followed by image. The model synthesizes salient objects by attending relevant words.

Note that those eight models can be grouped from two viewpoints: the number of discriminators and the attention mechanism. The first two models are single-discriminator methods, and the others are multiple-discriminator methods. The last three methods are constructed with the attention mechanism, and the other methods are not. To emphasize, AttnGAN and MirrorGAN transform the local linguistic representation into spatial weights on feature maps through an attention mechanism. In contrast, our CGL-GAN model use the local linguistic representation as conditional information for the single GAN discriminator without an attention mechanism. Those eight models are representative state-of-art models.

D. Implementation Details

In this section, we report the settings of hyper-parameters in our CGL-GAN model and the details of its training process.

First, we introduce the hyper-parameters settings for the text encoder, generator and discriminator. The text encoder has two hyper-parameters: the number of words in a description sentence n and the dimensionality of the word embedding d. For the CUB dataset we set the number of words n to 18 and the dimensionality of the word embedding d to 256. Likewise, for the MS-COCO dataset we set those two parameters n to 16 and d to 256; the latter value is the same as for the CUB dataset. The hyper-parameter of the generator is the number of up-sampling blocks N_G. We set the parameter N_G to 5 for generating both 256×256 and 128×128 images. Similarly, the hyper-parameter of the discriminator is the number of down-sampling blocks N_D. We set the parameter N_D to 5 to discriminate 256×256 images and N_D to 4 to discriminate 128×128 images. Finally, we use a 3×3 convolutional kernel with a stride of 1 for all the convolutional layers in both the generator and the discriminator.

To ensure a fair comparison, we adopt the pre-trained bidirectional LSTM used in the AttnGAN model [6] for our text encoder implementation.[8] We use the Adam optimizer [42] to train our generator and discriminator. We set the hyper-parameters of this optimizer as follows: the learning rate α at 0.0001 and the coefficients β_1 and β_2 for computing the average running gradient and its square at 0 and 0.9, respectively. Moreover, we update the generator and the discriminator simultaneously. We implement our model on the PyTorch platform with an NVIDIA GeForce GTX 1080Ti GPU. Our source code is publicly available for efficient knowledge sharing.[9]

[6] [Online]. Available: http://github.com/hanzhanggit/StackGAN-inception-model

[7] [Online]. Available: http://github.com/bioinf-jku/TTUR

[8] [Online]. Available: http://github.com/taoxugit/AttnGAN

[9] [Online]. Available: http://github.com/bupt-mmai/txt2im

TABLE I
COMPREHENSIVE COMPARISON OF OUR CGL-GAN MODEL WITH REPRESENTATIVE STATE-OF-ART METHODS. THE HIGHER INCEPTION SCORE AND THE LOWER FID VALUE ARE BETTER. SINGLE-GAN BASED MODELS ARE STAR-MARKED

METHOD	IMAGE SIZE	ATTENTION	CUB	MS-COCO		PARMs (M)
			INCEPTION	INCEPTION	FID	
GAN-INT-CLS⋆ [1]	64 × 64	NO	2.88 (±0.04)	7.88 (±0.07)	N/A	N/A
GAWWN⋆ [2]	128 × 128		3.62 (±0.07)	N/A		N/A
StackGAN [3]	256 × 256		3.70 (±0.04)	8.45 (±0.03)		107.8
StackGAN-V2 [4]	256 × 256		4.04 (±0.05)	N/A		103.4
HDGAN [5]	256 × 256		4.15 (±0.50)	11.86 (±0.18)		N/A
AttnGAN [6]	256 × 256	YES	4.36 (±0.30)	25.89 (±0.47)	28.76	169.4
MirrorGAN [7]			4.56 (±0.05)	26.47 (±0.41)	N/A	169.7
Obj-GAN [8]			N/A	27.37 (±0.22)	25.85	194.4
Our CGL-GAN⋆	256 × 256	NO	3.67 (±0.04)	13.62 (±0.02)	37.12	23.3

TABLE II
INCEPTION SCORES COMPARISON OF SINGLE-GAN MODELS ON THE CUB DATASET. ORIGINAL SINGLE-GAN BASED MODELS ARE STAR-MARKED

METHOD	INCEPTION SCORE
GAN-INT-CLS⋆ [1]	2.88 (±0.04)
StackGAN (w/o CA)† [3]	2.48 (±0.00)
StackGAN† [3]	3.02 (±0.01)
StackGAN-V2† [4]	3.49 (±0.04)
HDGAN† [5]	3.52 (±0.40)
Our CGL-GAN⋆	3.67 (±0.04)

E. Quantitative Evaluation

To quantitatively evaluate the performance of baseline models and our proposed model, we compare the Inception scores and FIDs of these models on two datasets, the CUB and the MS-COCO datasets. In addition, to evaluate their training costs, we compute their trainable parameters of these GAN-based models. Those results are shown in Table I. Next, we analyze the experimental results in detail from two aspects: 1) model performance; and 2) scale of trainable parameters.

1) Model performance. Generally, models that use attention mechanism achieve better performance on Inception scores and FIDs than those that do not use an attention mechanism. In addition, models that use multiple discriminators show better performance than those that use a single pair of discriminator and generator. Two single-GAN models, GAN-INT-CLS and our CGL-GAN are star-marked in Table I.

Note that without using the attention mechanism and multiple discriminators, our CGL-GAN model still achieves a comparable performance with those baseline models. On the CUB dataset, our CGL-GAN model achieves better performance than do most baseline models without an attention mechanism, except StackGAN-V2 and HDGAN. On the MS-COCO dataset, the Inception score achieved by our CGL-GAN model is significantly higher than those earned by all of the other models that do not use an attention mechanism. For example, the Inception score achieved by our CGL-GAN model is 1.76 times higher than that of the best HDGAN without an attention mechanism. However, the Inception score earned by our model on the CUB dataset is slightly lower than that of HDGAN. These two results seem inconsistent. We notice, however, that images in the MS-COCO dataset have more objects and more complicated backgrounds than those in the CUB dataset. We speculate that the correlations between visual features and language are more difficult to modeling in the MS-COCO dataset.

Compared with models that use an attention mechanism, our model also achieves comparable results on both Inception scores and FIDs. For example, the Inception score of our model is 13.75% lower than the best model Obj-GAN on the COCO dataset. Although Obj-GAN has twice as much Inception scores as our CGL-GAN, the actual difference is not quite significant. That is because the definition of the Inception score is not a linear function but an exponential function on the average KL divergence, see equation (9). Therefore, a slight variation may lead to a large variation on scores. It would be more reasonable to inspect those models based on the FID metric. Thus, our CGL-GAN model achieves comparable values compared with the Obj-GAN model.

2) Scale of trainable parameters. As shown in Table I, the scale of trainable parameters of our model is in general one order of magnitude smaller than those of the baseline models. Specifically, the parameters of our model are only 21.6% of StackGAN, 22.5% of StackGAN-v2, 13.8% of AttnGAN and MirrorGAN, and 12.0% of Obj-GAN. Moreover, note that Table I only counts the trainable parameters of GANs, in other words, generators and discriminators. Extra parameters in modules (e.g., DAMSM in AttnGAN and MirrorGAN, Caption net in MirrorGAN, Box generate net and shape generate net in Obj-GAN) are not included. The actual trainable parameters of these three models (AttnGAN, MirrorGAN and Obj-GAN) would be larger than those shown in Table I. We have such a huge parameter reduction because we only use a single pair of generator and discriminator to perform all of the workload. This experiment shows that using both global linguistic representations and local linguistic representation as the conditional information of the discriminator can make a single-discriminator model achieves a comparable performance to those multiple-discriminator models.

To further prove this point, we next remove the discriminators of those previous multiple-discriminator models, keeping only one generator and one discriminator. Then, we evaluate the Inception score of our CGL-GAN model and these single-GAN models on the CUB dataset. The experimental results of generating images with 256×256 pixels are reported in Table II. In this table, the Inception scores of the other models are sourced from

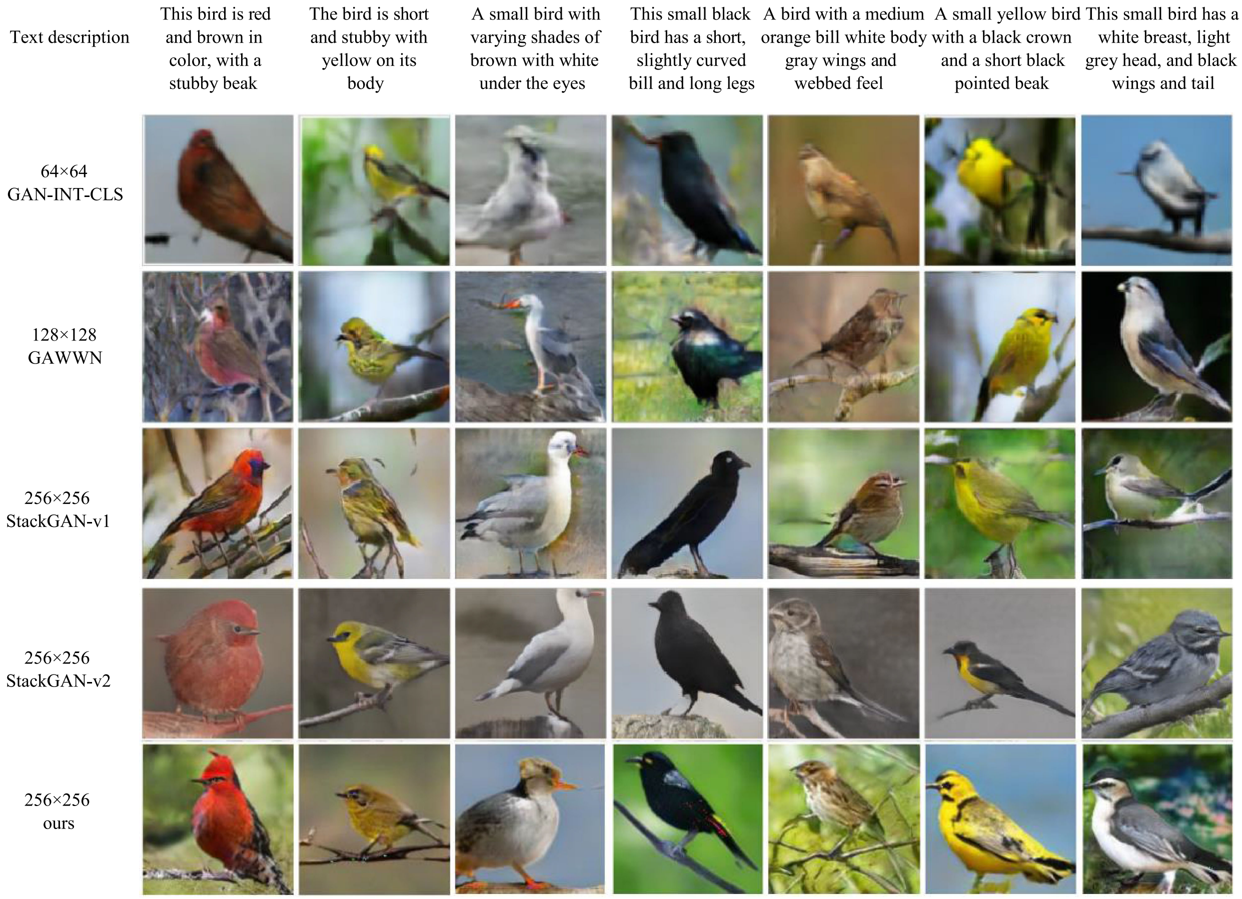
Text description
This bird is red and brown in color, with a stubby beak
The bird is short and stubby with yellow on its body
A small bird with varying shades of brown with white under the eyes
This small black bird has a short, slightly curved bill and long legs
A bird with a medium orange bill white body gray wings and webbed feel
A small yellow bird with a black crown and a short black pointed beak
This small bird has a white breast, light grey head, and black wings and tail
64×64 GAN-INT-CLS
128×128 GAWWN
256×256 StackGAN-v1
256×256 StackGAN-v2
256×256 ours

Fig. 3. Image synthesis examples on the CUB dataset using GAN-INT-CLS [1], GAWWN [2], StackGAN [3], StackGAN-V2 [4] and our proposed CGL-GAN model. The top row is the text description. The following five rows are the corresponding images for the text description by different methods. The generative method and image resolution are shown in the leftmost column.

the corresponding published papers [3]–[5]. Note that StackGAN, StackGAN-V2 and HDGAN are multiple-discriminators models in which previous discriminators are removed. The StackGAN (w/o CA) denotes the model removing the Conditioning Augmentation. Those four models are dagger-marked for clarity. However, our model and GAN-INT-CLS are original single-discriminator models that remain the same as in Table I. Those two models are star-marked.

As shown in Table II, our model has a higher Inception scores on the CUB dataset than all of the state-of-the-art methods that are not powered by multiple discriminators. Compared with GAN-INT-CLS, which is the earliest text-to-image model, our proposed model achieves 27.43% improvement in terms of Inception score (from 2.88 to 3.67). This experiment shows that our model has good performance in generating high-resolution images conditioned on text descriptions. In addition, this experiment demonstrates, to some extent, that using both local and global linguistic representations as conditional information of GANs can improve the performance.

F. Qualitative Evaluation

In this section, we qualitatively evaluate our proposed models with different hyper-parameter settings on two publicly available datasets: the CUB and MS-COCO datasets. Some examples are first demonstrated. For the CUB dataset, examples are shown in Fig. 3. In this figure, the top row contains the textual descriptions and the other rows contain the corresponding images. The bottom row shows the images generated by our CGL-GAN model. The images generated by the other four models are sourced from the work [4]. As shown in Fig. 3, GAN-INT-CLS can only generate 64×64 images; these reflect only the general shapes and colors of the birds. GAWWN achieves better performance than GAN-INT-CLS by introducing additional spatial information, such as bounding boxes and key points. However, the images generated by GAWWN are still unsatisfactory. The results of StackGAN and StackGAN-V2 are relatively satisfactory, and StackGAN-V2 is better than StackGAN. The results of our CLG-GAN model are better than those of StackGAN and somewhat better than StackGAN-V2.

For the MS-COCO dataset, some examples are shown in Fig. 4. In this figure, the top row contains the textual descriptions for the corresponding images, the middle row contains images generated by StackGAN, and the bottom row contains images generated by our CGL-GAN model. We note that images generated by our model are intuitively closer to the real images. StackGAN reflects only the general shapes and colors of the objects and fails to capture the fine-grained details. For example,

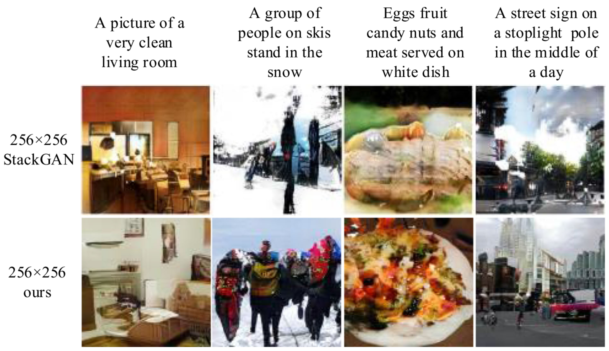
A picture of a very clean living room
A group of people on skis stand in the snow
Eggs fruit candy nuts and meat served on white dish
A street sign on a stoplight pole in the middle of a day
256×256 StackGAN
256×256 ours

Fig. 4. Image synthesis examples on the MS-COCO dataset using StackGAN [3] and our proposed CGL-GAN model. The top row is the text description. Correspondingly, the following two rows are the generated images with the resolution 256×256.

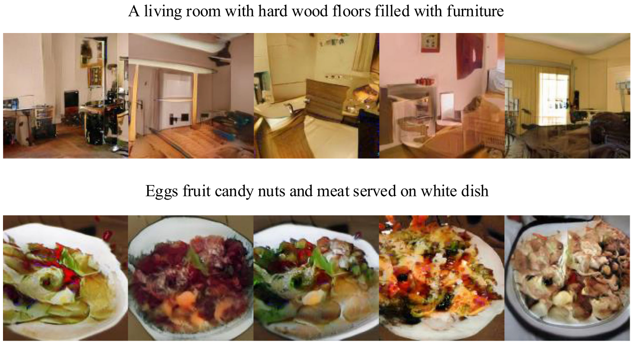
A living room with hard wood floors filled with furniture
Eggs fruit candy nuts and meat served on white dish

Fig. 5. Text-to-image synthesis examples from the MS-COCO dataset conditioned on the same text descriptions. For a given text description, diverse and meaningful images are generated. The images have a resolution 256×256.

TABLE III
PERFORMANCE COMPARISON ON DIFFERENT VARIANTS OF CPB. THE HIGHER INCEPTION SCORE AND THE LOWER FID VALUES ARE BETTER

METHOD	GLOBAL	LOCAL	INCEPTION	FID
CGL-GAN-OL		✓	2.42 (±0.04)	92.06
CGL-GAN-OG	✓		2.48 (±0.07)	85.41
CGL-GAN-GL	✓	✓	3.36 (±0.04)	34.36
CGL-GAN-LG	✓	✓	3.51 (±0.03)	31.92

given the textual description “A group of people on skis stand in the snow,” the image generated by the StackGAN model does not show the semantic of “a group of people”. In another example for the textual description “Eggs fruit candy nuts and meat served on white dish,” the image generated by our CGL-GAN model captures the detail “white dish”. Our proposed model performs better at this point. To summarize, those visualized results both on both the CUB and MS-COCO datasets demonstrate that our CGL-GAN model generates an image with quite satisfactory qualities.

We note another interesting result. As shown in Fig. 3, the images generated by our model are intuitively more realistic than those generated by StackGAN. Nevertheless, referring to the previous Table I, where StackGAN earns a higher Inception score than does our proposed model. Those two results are seemingly contradictory. We conjecture that there are two reasons for this phenomenon. One is that the Inception score does not accurately reflect the human evaluations of images in principle. Therefore, it is possible for a generated image to be quite close to the real image from a human point of view, but for the Inception score to still be relatively low. Salimans *et al.* have discussed the defects of the inception score metric [40]. Another possible reason is that the diversity of our model generated images is not sufficiently high to earn a high Inception score. Fig. 5 shows some diverse samples that are generated conditioned on the same textual description. Although our model generates a variety of results from the same description, the diversity still seems insufficient to earn high inception scores. As discussed in Section IV-B, diversity plays an important role in evaluating the Inception scores performance of generative models. Although our generated images have high quality, their lack of high diversity may lead to the reduced Inception scores.

Finally, in order to show the progression of image quality during the training process, we visualize the training process of the generator by showing images in different iterations in Fig. 6. The leftmost column in each line is the textual description, and the other columns are the corresponding images from different iterations. The generated images in each column come from the same training iterations, which appears at the top of each column. As shown in Fig. 6, the images generated by our model become progressively clearer when the number of iterations increases. At earlier training stages, when the number of iterations is small (less than 17 K), the generated images are nonsense mosaics of color blocks, and our generator ignores some of the information in the textual descriptions. As the iteration number increase from 35 K to 69 K, our generator gradually learns the shapes of the birds and background information. After 104 K iterations, our generator produces photographic-quality images with abundant background detail and fine-grained object detail.

G. CPB Analysis

Finally, we demonstrate the rationality of the CPB module in our CGL-GAN model by comparing its Inception scores and FIDs. We have described four different versions of our CPB in Section III-D and have shown their architectures in Fig. 2. Note that the CPB shown in the overview Fig. 1 is the version of CGL-GAN-LG. The previous experiments having been reported thus far are all adopted this CPB version. Here, we further evaluate these four CPB variants to generate images with 128×128 pixels on the CUB dataset. In our experiments, all the other settings are kept unchanged. The Inception scores and FIDs of these four CPBs are shown in Table III.

As shown in Table III, the Inception scores of variants conditioned only on one type of linguistic representation (i.e., CGL-GAN-OG and CGL-GAN-OL) are notably lower than those conditioned on both global and local linguistic representations (i.e., CGL-GAN-GL and CGL-GAN-LG). The CGL-GAN-OL variant earns the lowest Inception score, while the CGL-GAN-LG variant achieves the highest score. The Inception score of CGL-GAN-LG is 45% higher than that of CGL-GAN-OL. We also note that a trend reversal in the FID of all four variants compared with the Inception scores. The lower FID values are better. Typically, the FID of CGL-GAL-LG is 65% lower than that of CGL-GAN-OL. Actually, images generated by CGL-GAN-OL

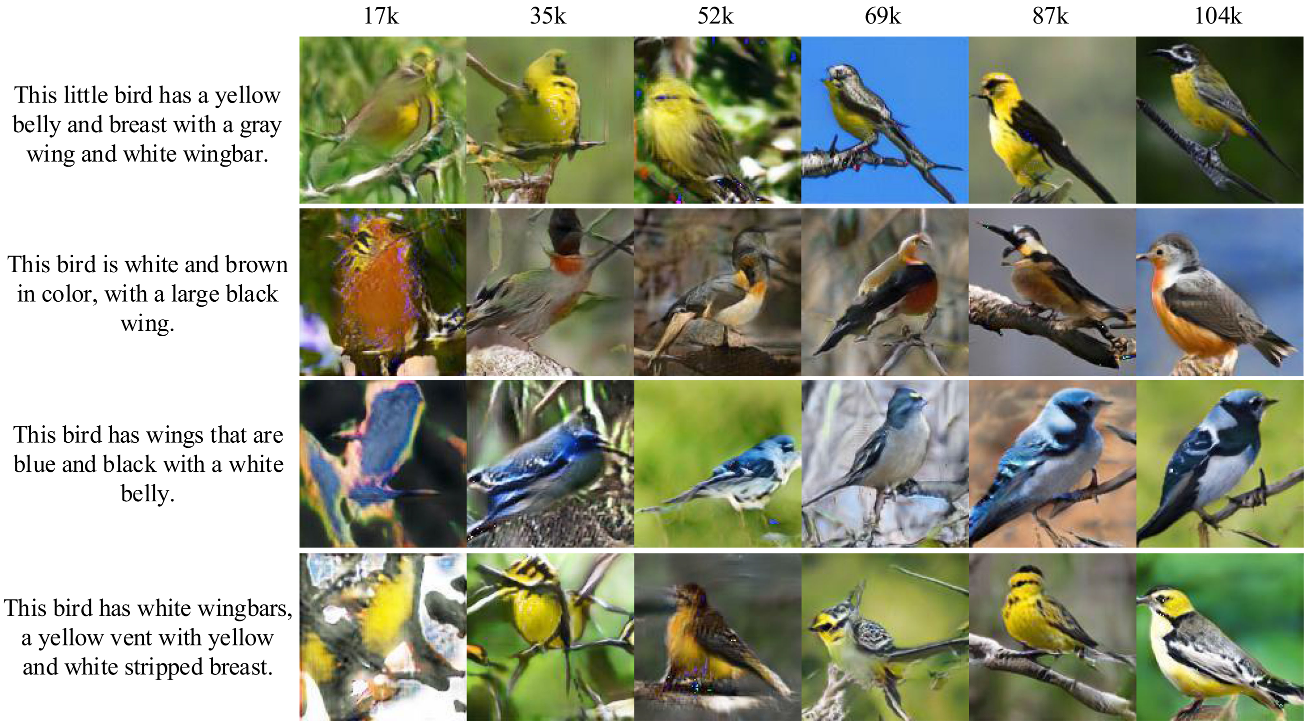
17k
35k
52k
69k
87k
104k
This little bird has a yellow belly and breast with a gray wing and white wingbar.
This bird is white and brown in color, with a large black wing.
This bird has wings that are blue and black with a white belly.
This bird has white wingbars, a yellow vent with yellow and white stripped breast.

Fig. 6. Generating image examples in successive iterations by our model. The top shows the iteration numbers. The following is the text descriptions and their synthesized images. This process demonstrates the generative quality grows with the iteration numbers. The images have the resolution 256×256.

and CGL-GAN-OG are barely recognizable. This means the gradients returned by these two CPB variants cannot stably train their corresponding generators. In contrast, two variants conditioned on both global and local linguistic representations can provide better gradients to train their corresponding generators. This experiment demonstrates that our proposed CPB plays a significant role in improving the performance of text-to-image synthesis models.

Furthermore, we note that the Inception scores and FID of CGL-GAN-LG are slightly better (i.e., higher Inception score and lower FID) than that of CGL-GAN-GL. The only difference between CGL-GAN-LG and CGL-GAN-GL is whether the granularity of the text and image representations is aligned. The CGL-GAN-LG model aligns the text and image representations with the analogous granularity, while CGL-GAN-GL does not. In other words, aligning the text and image representations with the analogous granularity can achieve better results.

Note that in the discriminator, we only evaluate the case in which the down-sampling feature map $\boldsymbol{v}_4$ (the number of down-sampling layers N_D equals 5 in our setting) is aligned with the local linguistic representation $\boldsymbol{y}^{\mathcal{L}}$. We have the following consideration. The feature maps in front layers are visually the edges and angles of image features, and those of the rear layers are parts and objects in images. We consider the last two layers as the most likely to form the semantic concepts. Those semantic concepts are then used for aligning linguistic representations. On the other hand, the size of feature maps extracted from the front layers ($\boldsymbol{v}_1$ to $\boldsymbol{v}_3$) are much larger than that of $\boldsymbol{v}_4$. Therefore, the fully-connected layer in a similar CPB with front feature maps would require more hidden layer parameters. Furthermore, obtaining stability of GANs training would be quite difficult. From the intuitive and practical perspectives, therefore, we adopt the second-to-last feature map to align the local linguistic representation, and the last feature map to align the global linguistic representation. In summary, an effective alignment of visual and linguistic representations can improve the overall performance.

V. Conclusion

In this article, we explore the use of both global and local linguistic representations for text-to-image synthesis by proposing a high-resolution text-to-image synthesis model. Our model takes both global and local linguistic representations as generative conditions. In this model, we propose CPB to incorporate the linguistic information into GANs. The extensive experiments demonstrate that incorporating both the global and local linguistic representations can greatly improve the performance of text-to-image synthesis models when generating high-resolution images. Compared with leading models powered by multiple GANs, our model achieves comparable Inception scores and FIDs. Moreover, our model can generate high-resolution images with fewer trainable parameters.

Nevertheless, some interesting work remains to be investigated in the future. First, we note that our proposed model is not a substitute for but an improvement on the existing multiple GANs-based models. Therefore, one interesting line for future work would be to generate higher-resolution images by stacking a set of our GANs. Second, in this study we do not consider other methods to increase the model's performance, such as the attention mechanism or incorporating spatial information. Therefore, another line of future work would be to introduce such schemes to improve our model's performance.

Acknowledgment

The authors would like to express their gratitude to the editors and anonymous reviewers for their constructive comments that allowed them to improve the final version of this article.

References

[1] S. Reed *et al.*, “Generative adversarial text to image synthesis,” in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1060–1069.

[2] S. E. Reed *et al.*, “Learning what and where to draw,” in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 217–225.

[3] H. Zhang *et al.*, “StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 5908–5916.

[4] H. Zhang, “StackGAN++: Realistic image synthesis with stacked generative adversarial networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.

[5] Z. Zhang, Y. Xie, and L. Yang, “Photographic text-to-image synthesis with a hierarchically-nested adversarial network,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 6199–6208.

[6] T. Xu *et al.*, “AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 1316–1324.

[7] T. Qiao, J. Zhang, D. Xu, and D. Tao, “MirrorGAN: Learning text-to-image generation by redescription.” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 1505–1514.

[8] W. Li *et al.*, “Object-driven text-to-image synthesis via adversarial training.” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 12174–12182.

[9] F. Feng, R. Li, and X. Wang, “Deep correspondence restricted Boltzmann machine for cross-modal retrieval,” *Neurocomput.*, vol. 154, pp. 50–60, 2015.

[10] Z. Qiu, Y. Pan, T. Yao, and T. Mei, “Deep semantic hashing with generative adversarial networks,” in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 225–234.

[11] X. Huang and Y. Peng, “TPCKT: Two-level progressive cross-media knowledge transfer,” *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2850–2862, Nov. 2019.

[12] C. Li, T. Yan, X. Luo, L. Nie, and X. Xu, “Supervised robust discrete multimodal hashing for cross-media retrieval,” *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2863–2877, Nov. 2019.

[13] Y. Mao, C. Zhou, X. Wang, and R. Li, “Show and tell more: Topic-oriented multi-sentence image captioning.” in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 4258–4264.

[14] X. Xiao, L. Wang, K. Ding, S. Xiang, and C. Pan, “Deep hierarchical encoder-decoder network for image captioning,” *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2942–2956, Nov. 2019.

[15] L. Li, S. Tang, Y. Zhang, L. Deng, and Q. Tian, “GLA: Global local attention for image description,” *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 726–737, Mar. 2018.

[16] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. Advances Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[17] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014. [Online]. Available: https://arxiv.org/abs/1411.1784

[18] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[19] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Hk4_qw5xe

[20] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *Proc. Int. Conf. Learn. Representations*, 2016. [Online]. Available: https://iclr.cc/archive/www/doku.php%3Fid=iclr2016:accepted-main.html

[21] E. Denton, S. Chintala, A. Szlam, and R. Fergus, “Deep generative image models using a Laplacian pyramid of adversarial networks,” in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 1486–1494.

[22] J. J. Zhao, M. Mathieu, and Y. Lecun, “Energy-based generative adversarial networks,” in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=ryh9pmcee

[23] D. Berthelot, T. Schumm, and L. Metz, “BEGAN: Boundary equilibrium generative adversarial networks,” 2017. [Online]. Available: https://arxiv.org/abs/1703.10717, Accessed on: May 31, 2017.

[24] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier GANs,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[25] J. Song *et al.*, “Binary generative adversarial networks for image retrieval,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 394–401.

[26] T. Miyato and M. Koyama, “cGANs with projection discriminator,” in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=ByS1VpgRZ

[27] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” 2017. [Online]. Available: https://arxiv.org/abs/1701.07875, Accessed on: Dec. 6, 2017.

[28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[29] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1QRgziT

[30] S. Hong, D. Yang, J. Choi, and H. Lee, “Inferring semantic layout for hierarchical text-to-image synthesis,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 7986–7994.

[31] J. Johnson, A. Gupta, and L. Fei-Fei, “Image generation from scene graphs,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 1219–1228.

[32] W. Xu, S. Keshmiri, and G. R. Wang, “Adversarially approximated autoencoder for image generation and manipulation,” *IEEE Trans. Multimedia*, vol. 21, no. 9, pp. 2387–2396, Sep. 2019.

[33] Y. Guo *et al.*, “Auto-embedding generative adversarial networks for high resolution image synthesis,” *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2726–2737, Nov. 2019.

[34] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=B1xsqj09Fm

[35] H. De Vries *et al.*, “Modulating early visual processing by language,” in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 6594–6604.

[36] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.

[37] J. H. Lim and J. C. Ye, “Geometric GAN,” 2017. [Online]. Available: https://arxiv.org/abs/1705.02894v1, Accessed on: May 9, 2017.

[38] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The caltech-UCSD birds-200-2011 dataset,” California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011. [Online]. Available: http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

[39] T.-Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 740–755.

[40] T. Salimans *et al.*, “Improved techniques for training GANs,” in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[41] M. Heusel *et al.*, “GANs trained by a two time-scale update rule converge to a nash equilibrium,” in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.

[42] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015. [Online]. Available: https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:accepted-main.html

Ruifan Li (Member, IEEE) received the B.S. and M.S. degrees in control systems, and in circuits and systems from the Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001, respectively, and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2006. He is currently an Associate Professor with the School of Computer Science, BUPT and is affiliated with the Engineering Research Center of Information Networks, Ministry of Education. Since 2006, he has been with the School of Computer Science, BUPT. From February 2011, he spent one year as a visiting scholar with the Information Sciences Institute, University of Southern California, Los Angeles, CA, USA. His current research activities include multimedia information processing, neural information processing, and statistical machine learning. He is a member of the China Computer Federation, and Chinese Association of Artificial Intelligence. He was an Active Reviewer for dozens of peer-reviewed journals.

Ning Wang received the B.E. degree from Southwest Jiaotong University, Chengdu, China, in 2015. He is currently working toward the master's degree with the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include multimedia information processing and machine leaning.

Fangxiang Feng received the B.S. and Ph.D. degrees from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2010 and 2015, respectively. He is currently an Assistant Professor with the School of Computer Sciences, BUPT. His research interests include multi-modal deep learning and computer vision.

Guangwei Zhang received the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2009. He is currently with the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China and is affiliated with the Engineering Research Center of Information Networks, Ministry of Education. His current research activities include multimedia in the Internet of Things and artificial intelligence.

Xiaojie Wang received the Ph.D. degree from Beihang University, Beijing, China, in 1996. He is a Full Professor and the Director of the Centre for Intelligence Science and Technology, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include natural language processing and multi-modal cognitive computing. He is an executive member of the Council of Chinese Association of Artificial Intelligence and the Director of the Natural Language Processing Committee. He is a member of the Council of Chinese Information Processing Society and the Chinese Processing Committee of China Computer Federation.