IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 3, 2020, accepted February 13, 2020, date of publication February 24, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975841

Instance Mask Embedding and Attribute-Adaptive Generative Adversarial Network for Text-to-Image Synthesis

JIANCHENG NI, SUSU ZHANG, ZILI ZHOU, JIE HOU, AND FENG GAO
School of Software, Qufu Normal University, Qufu 273165, China

Corresponding author: Susu Zhang (zhangss9719@163.com)

This work was supported in part by the Youth Program of National Science Foundation of China under Grant 61601261, in part by the Plan Project of Graduate Education Quality Improvement of Shandong Province under Grant SDYY17136, and in part by the Interdisciplinary Research Project of Qufu Normal University under Grant QFNUSKC291809120.

ABSTRACT Existing image generation models have achieved the synthesis of reasonable individuals and complex but low-resolution images. Directly from complicated text to high-resolution image generation still remains a challenge. To this end, we propose the instance mask embedding and attribute-adaptive generative adversarial network (IMEAA-GAN). Firstly, we use the box regression network to compute a global layout containing the class labels and locations for each instance. Then the global generator encodes the layout, combines the whole text embedding and noise to preliminarily generate a low-resolution image; the instance embedding mechanism is used firstly to guide local refinement generators obtain fine-grained local features and generate a more realistic image. Finally, in order to synthesize the exact visual attributes, we introduce the multi-scale attribute-adaptive discriminator, which provides local refinement generators with the specific training signals to explicitly generate instance-level features. Extensive experiments based on the MS-COCO dataset and the Caltech-UCSD Birds-200-2011 dataset show that our model can obtain globally consistent attributes and generate complex images with local texture details.

INDEX TERMS Generative adversarial network, global generator, local refinement generator, instance mask embedding, attribute-adaptive discriminator.

I. INTRODUCTION

Conditional deep generative models have realized quite exciting progress in text-to-image generation. The widely used Generative Adversarial Networks (GANs) [1], which jointly learn generators and discriminators, have generated promising individual images on simple datasets. However, once there are heterogeneous objects and scenes in the text, the quality of the generated image becomes drastically worse [2]. This is mainly because most existing approaches only focus on global sentence embedding without considering that each word has a different level of information related to the image. Besides, the ambiguity of text and the unknown shapes of instances make the generation process more difficult to constrain [3]. As a result, those images generated by current models usually have lower resolution and blurred texture. Moreover, instance attributes represent important image feature information [4], but existing methods use the sentence-conditional discriminator which only provides the coarse-grained training feedback, making it hard for generators to disentangle different regions and learn fine-grained attributes.

To address these three limitations, our proposed IMEAA-GAN harnesses a pre-trained box regression network [5] to obtain a global layout which contains class labels and bounding boxes, then generates complex images from this layout through a coarse-to-fine process, where the global generator initially generates a low-resolution image and two local refinement generators hierarchically synthesize high-resolution images by combining the instance-wise attention and the instance mask embedding. Additionally, our model adopts the word-level and attribute-adaptive discriminators to provide fine-grained feedback, thus, the local refinement generators can be instructed to synthesize specific visual attributes.

The contributions of this paper can be listed as follows:

The associate editor coordinating the review of this manuscript and approving it for publication was Guitao Cao.

1) To overcome the complexity and ambiguity of a whole sentence, we explicitly utilize the word-level embedding as input and use box regression network to obtain the global layout that contains spatial positions, object sizes, and class labels.

2) In order to make local refinement generators learn instance-level and fine-grained features, we propose the instance mask embedding mechanism to add pixel-level mask constrains. Therefore, our generators can get more details and semantic information for high-resolution image generation.

3) Two word-level and attribute-adaptive discriminators instead of commonly used sentence-conditional discriminator are employed to classify each attribute independently and generate exact signals for generators to synthesize certain visual attributes.

II. RELATED WORK

As one of the most commonly used image generation models, GANs include generators and discriminators. The generator is mainly used to learn pixel distributions and generate realistic images, while the discriminator should distinguish the received images as real or fake. They continually update in order to achieve dynamic equilibrium [6].

Many methods based on GANs have been proposed to improve image quality, and there are many input types. Zhu *et al.* [7] showed using sketches to modify images. Based on this, Lu *et al.* [8] adopted contextual GAN to synthesize images from sketch constraints. Similarly, Huang *et al.* [9] proposed an image-to-image translation model. In order to synthesize images from category labels, Brocket *et al.* [10] introduced a class-conditional model. Sharma *et al.* [11] improved the text-to-image generation by using dialogue. However, due to the complexity of the input text, Johnson *et al.* [12] proposed the sg2im method to convert the input text into scene graphs for image generation.

Among these various inputs, the text is the easiest and the most convenient type to perform manipulation. An increasing number of researchers have shown interest in text-to-image generation, and there are mainly two manifolds in the research community.

A. SINGLE-STAGE TEXT-TO-IMAGE GENERATION

Many approaches directly generate images from text without intermediate representations. For example, Reed *et al.* [13] have achieved simple image synthesis directly from captions without reasoning any semantic layouts. By contrast, Dong *et al.* [14] input both the image and text into conditional GAN (CGAN) to generate manipulated contents. Based on CGAN, Li *et al.* [15] proposed the Triple-GAN, which contains an extra classifier to label the generated image with its matching text for data augmentation, the labeled image-text pairs then can be used as the training data. Similarly, Dash *et al.* [16] proposed the TAC-GAN to generate diverse images by distinguishing real images from generated images and classifying real images into true classes. Nguyen *et al.* [17] introduced the PPGN, which is similar to TAC-GAN and contains a conditional network, to generate images from captions. Furthermore, based on conditional GANs, Cha *et al.* [18] improved the adversarial training process by forming positive-negative label pairs and employing an auxiliary classifier to predict the semantic consistency of a given image-caption pair.

All of these models produce diverse images directly from descriptions and their main focus isn't in synthesizing high-resolution images, so they only use single-stage generation.

B. MULTI-STAGE TEXT-TO-IMAGE GENERATION

It's difficult to directly generate high-quality images from complex text, Denton *et al.* [19] adopted the LapGAN to generate images by constructing a Laplacian pyramid framework. However, this model still has limitations, the most obvious one is that its deep networks increase the training difficulty, resulting in model collapse. To solve this problem, Zhang *et al.* [20] employed StackGAN which contains two generators to synthesize images within two stages. Afterward, they improved the previous architecture by proposing the StackGAN++ [21] which is designed as a tree structure. But these two models only encode text into a single sentence vector for image generation. Similar to scene graphs, Hong *et al.* [22] introduced the text2img method, they utilized the inferred layouts to generate images, Li *et al.* [23] also obtained graphic layouts with wireframe discriminators. Given a coarse layout, Zhao *et al.* [24] generated images by disentangling each instance into a certain label part and uncertain appearance part. Hinz *et al.* [25] evaluated the detecting frequency of objects and synthesized multiple instances at various spatial locations based on an object pathway. Likewise, Li *et al.* [26] improved the grid-based attention mechanism by coupling attention with the layout. In order to minimize the differences between real and fake images, Yuan and Peng [27] showed symmetrical distillation networks. Then Sun and Wu [28] put forward a new feature normalization approach to synthesize visually different images from given layouts. Xu *et al.* [29] introduced the AttnGAN which aggregates the attention mechanism [30] and the DAMSM loss into text-to-image generation.

However, AttnGAN only leverages a global sentence vector and takes all instances equally, thus it may miss the detailed instance-level information. Our local refinement generators are able to uncover such difference by applying the instance mask embedding. Moreover, the proposed word-level attribute-adaptive discriminators have the capacity to disentangle each attribute independently in order to instruct two local refinement generators to synthesize certain visual attributes.

III. BACKGROUND

A. BOX REGRESSION NETWORK

Box regression network can effectively reason scene layouts from descriptions or scene graphs [31]. This network takes a sentence embedding or final object embedding as

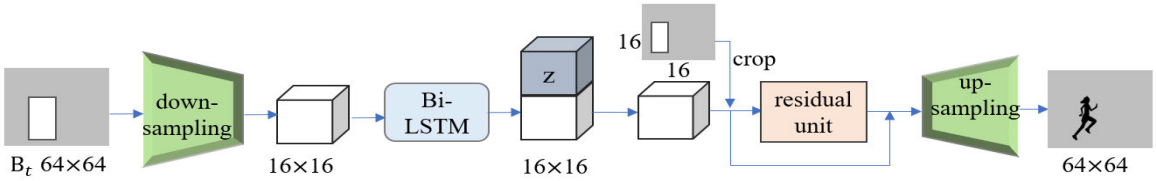

FIGURE 1. Overall framework of the mask regression network.

input and outputs the predicted bounding boxes $B_{1:T} = \{B_1, B_2, \cdots, B_T\}$. The t-th bounding box is parameterized as $B_t = (b_t, l_t)$ where $b_t = (b_t^x, b_t^y, b_t^w, b_t^h) \in \mathbb{R}^4$ indicates the location (x, y) and size $(w \times h)$ of the related object and $l_t \in \{0, 1\}^{L+1}$ represents the one-hot class label of the t-th box. We define L as the number of real object categories and the $(L + 1)$-th label as an end-of-text indicator. The joint probability is calculated as:

$$p(b_t^x, b_t^y, b_t^w, b_t^h, l_t) = p(l_t) \cdot p(b_t^x, b_t^y, b_t^w, b_t^h | l_t) \quad (1)$$

where $p(b_t^x, b_t^y, b_t^w, b_t^h | l_t)$ is the box coordinate probability and $p(l_t)$ represents the label distribution. It is hard to directly model the joint probability since it contains various parameters. Therefore, the coordinate probability of the t-th box is decomposed as:

$$p(b_t^x, b_t^y, b_t^w, b_t^h, l_t) = p(l_t) \cdot p(b_t^x, b_t^y | l_t) \cdot p(b_t^w, b_t^h | b_t^x, b_t^y, l_t) \quad (2)$$

where the probability $p(b_t^x, b_t^y | l_t)$ and $p(b_t^w, b_t^h | b_t^x, b_t^y, l_t)$ are implemented by two bivariate Gaussian mixtures:

$$p(b_t^x, b_t^y | l_t) = \sum_{k=1}^{K} \pi_{t,k}^{xy} \mathcal{N}(b_t^x, b_t^y; \mu_{t,k}^{xy}, \sum\nolimits_{t,k}^{xy}) \quad (3)$$

$$p(b_t^w, b_t^h | b_t^x, b_t^y, l_t) = \sum_{i=k}^{K} \pi_{t,k}^{wh} \mathcal{N}(b_t^w, b_t^h; \mu_{t,k}^{wh}, \sum\nolimits_{t,k}^{wh}) \quad (4)$$

where k indicates the number of mixture components, the label of the t-th object l_t and $\pi_{t,k}^{xy}, \pi_{t,k}^{wh} \in \mathbb{R}$, $\mu_{t,k}^{xy}, \mu_{t,k}^{wh} \in \mathbb{R}^4$, $\sum_{t,k}^{xy}, \sum_{t,k}^{wh} \in \mathbb{R}^{4\times4}$ are parameters of the Gaussian Mixture Model (GMM) [32], [33]. These parameters are calculated by the outputs of LSTM at each step:

$$[h_t, c_t] = LSTM(B_{t-1}; [h_{t-1}, c_{t-1}]) \quad (5)$$

$$l_t = W^l h_t + b^l \quad (6)$$

where h_t is the hidden state, c_t is the t-th cell state. Similarly, $\pi_{t,k}^{xy}, \pi_{t,k}^{wh}, \mu_{t,k}^{xy}, \mu_{t,k}^{wh}$ and $\sum_{t,k}^{xy}, \sum_{t,k}^{wh}$ are computes as:

$$\theta_t^{xy} = W^{xy}[h_t, l_t] + b^{xy} \quad (7)$$

$$\theta_t^{wh} = W^{wh}[h_t, l_t, b_x, b_y] + b^{wh} \quad (8)$$

where $\theta_t^{xy} = [\pi_{t,1:k}^{xy}, \mu_{t,1:k}^{xy}, \sum_{t,1:k}^{xy}]$ and $\theta_t^{wh} = [\pi_{t,1:k}^{wh}, \mu_{t,1:k}^{wh}, \sum_{t,1:k}^{wh}]$ are respectively concatenated to a single vector θ_t^{xy} and θ_t^{wh}, and $[\cdot, \cdot]$ represents the concatenation.

Inspired by the recent progress of the box regression network, we explicitly use it to predict locations for various instances. Different from sg2im [12] and text2img [22], we use word embedding instead of final vectors computed by graph convolutional network [34] or a sentence vector as input to obtain bounding boxes. Each box in our model not only predicts the location but also indicates the size and class label of each instance, which greatly differs from sg2im [12], the global layout is thus synthesized for the further multi-stage generation.

B. MASK REGRESSION NETWORK

Mask regression network [35] has been used for mask segmentation in many computer vision tasks. And Hong *et al.* [22] have constructed shape masks from captions for image generation. As shown in Fig. 1, the mask regression network encodes the bounding box tensor B_t into a binary one $B_t \in \{0, 1\}^{h\times w\times l}$ where $h \times w$ represents the instance size and l is the category label. After a down-sampling block, the encoded features are fed into Bi-LSTM and concatenated with noise z. If and only if the bounding box contains the related class label, the binary tensor B_t is set to 1, other parts outside the box are all set to 0. After applying this mask operation, these masked features are then fed into a residual unit which allows the network to possess a deeper encoding ability by applying the skip connection [36]. Afterward, the predicted segmentation mask $p_t \in \mathbb{R}^{h\times w}$ with all elements in the range (0,1) is obtained through several up-sampling layers for image generation.

Contrary to previous methods that use segmentation mask annotations for both low-resolution and high-resolution image synthesis, our approach employs the predicted pixel-level instance masks only as constraints to two identical local refinement generators so that its up-sampling path can preserve the capacity to refine local texture details. Hence, the synthesized instances are coherent with inferred masks while discarding ambiguous features and containing pixel-level details.

IV. IMEAA-GAN

The proposed IMEAA-GAN performs text-to-image synthesis in three steps: the box regression network infers global layouts to obtain categories, sizes, and locations of objects. Then the global generator generates relatively low-resolution global images from these layouts. Two local refinement generators finally synthesize high-resolution and photographic images.

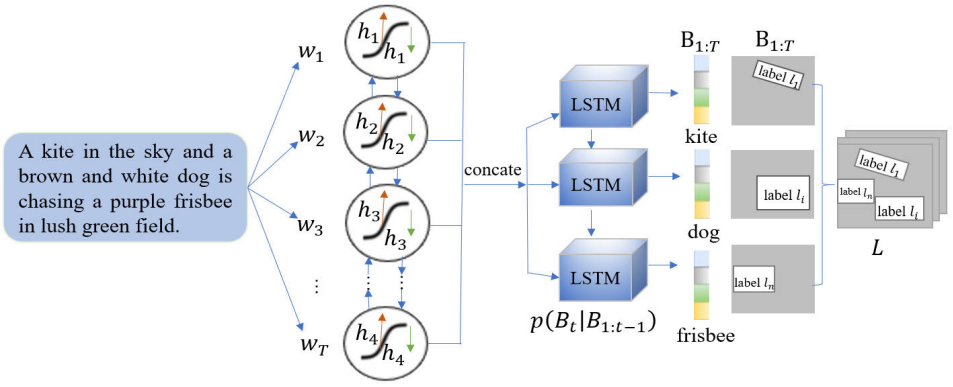

FIGURE 2. Global layout inferred by box regression network.

A. GLOBAL LAYOUT GENERATION

We employ the box regression network to initially infer a global layout L from word-level embedding vectors. The global layout, as an intermediate representation, contains the corresponding bounding boxes for the related instances. The generation process of a global layout is illustrated in Fig. 2.

The box regression network is designed as an encoder-decoder architecture. For each instance, the network infers the box $B_t = (b_t, l_t)$, and $b_t = (b_t^x, b_t^y, b_t^w, b_t^h) \in \mathbb{R}^4$. Firstly, our IMEAA-GAN takes the text as input, with a pre-trained Bi-LSTM which is used as a text encoder, the whole text is encoded into word embedding vectors and also a global text embedding φ. Every word is related to two hidden states, we concatenate the two states to indicate the semantic information of a word. Thus a feature matrix of all the words is obtained, each column of this matrix represents a word feature vector. At the same time, we concatenate the last hidden states of two directions to get the global text embedding φ. Then we take LSTM [37] as the decoder to approximate the class label l_t and the coordinates b_t, these GMM parameters are mentioned in function (1). To achieve this, we decompose the conditional joint probability as:

$$p(B_{1:T}|\varphi) = \prod_{t=1}^{T} p(B_t|B_{1:t-1}, \varphi) \tag{9}$$

where T is the number of instances. We firstly predict the category l_t for the t-th object, then compute the b_t based on l_t:

$$p(B_t|\varphi) = p(b_t, l_t|\varphi) = p(l_t|\varphi)\, p\,(b_t|l_t, \varphi) \tag{10}$$

here, the class label l_l is calculated by softmax and the coordinates b_t are modeled by GMM:

$$p(l_t|B_{1:t-1}, \varphi) = soft\max(e_t) \tag{11}$$

$$p(b_t|l_t, B_{1:t-1}, \varphi) = \prod_{k=1}^{K} \pi_{t,k} \mathcal{N}(b_t; \mu_{t,k}, \sum\nolimits_{t,k}) \tag{12}$$

where e_t is the softmax logit calculated by the t-th step outputs of each LSTM unit. Similarly, these parameters $\pi_{t,k} \in \mathbb{R}$, $\mu_{t,k} \in \mathbb{R}^4$, and $\sum_{t,k} \in \mathbb{R}^{4\times4}$ that have been mentioned in function (3) and (4) are also computed in this way, k indicates the number of mixture elements. Finally, a global layout L that includes box coordinates and class labels for all entities is generated.

B. IMAGE GENERATION

Our IMEAA-GAN takes advantage of the multi-stage text-to-image generation strategy [38]. Despite there are many methods, such as Obj-GAN [26], and our IMEAA-GAN both use the multi-stage generation, other methods are not robust to complex and ambiguous descriptions and the pixel-level features are not sufficiently used for image synthesis. Obj-GAN has achieved image-level semantical consistency. However, during the generation process, Obj-GAN implements segmentation mask annotations for both low-resolution and high-resolution image synthesis, it is labor-intensive to collect these annotations. In addition, applying them in low-resolution image generation cannot efficiently improve the image quality, since these images are not finely synthesized and the image features are more tend to be random vectors. By contrast, our approach calculates the pixel-level instance mask embedding instead of collecting mask annotations. More importantly, we adopt the instance mask embedding only in two local refinement generators. In this way, our IMEAA-GAN can obtain the capability of capturing visual features and the flexibility of generating fine-grained instances.

Given a coarse layout L_0, the global generator G_{img_0} initially generates an image I_0 with 64×64 resolution. Then the local refinement generator G_{img_1} employs the instance-wise attention and instance mask embedding to refine different regions of the first generated image in order to synthesize a high-quality image. Here, two local refinement generators that have the same architecture are utilized for generating higher resolution images. For the sake of brevity, we will not show the generation process of the 256×256 image because it is the same as the 128×128 image.

1) GLOBAL GENERATOR

The global layout provides the semantic structure of the corresponding text. Fig. 3 shows that given a pre-generated

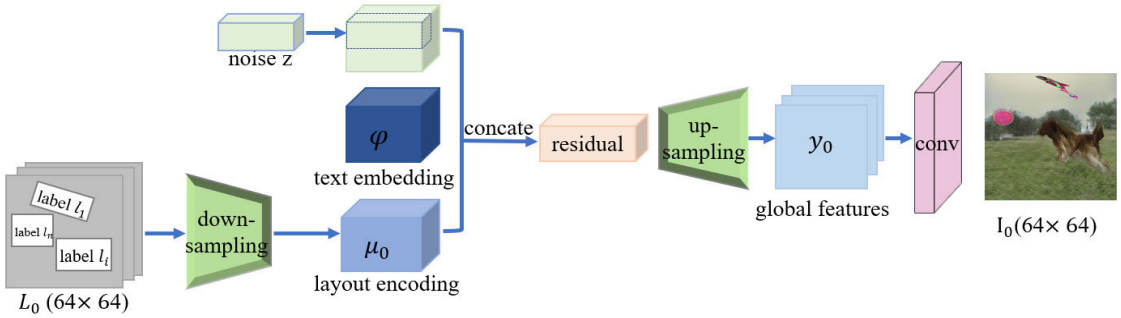

FIGURE 3. Architecture of the global generator.

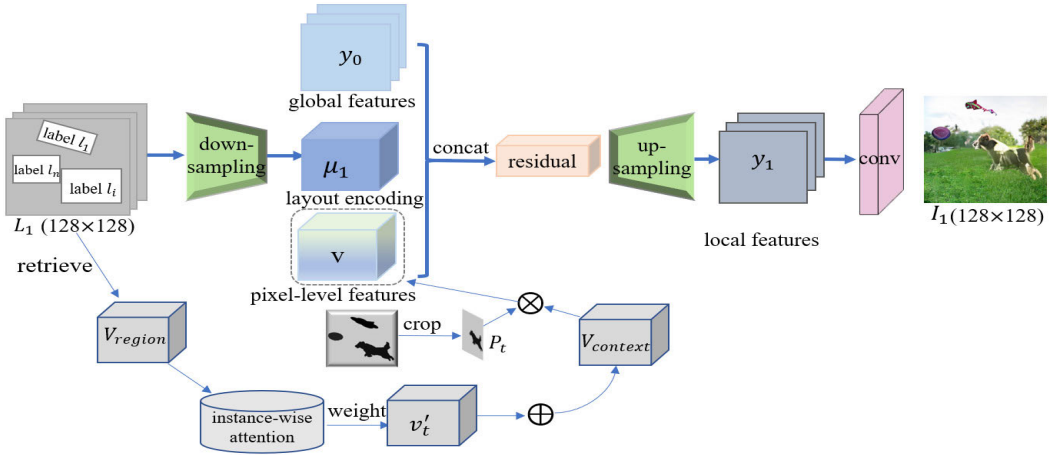

FIGURE 4. Architecture of the local refinement generator.

layout L_0, the global generator G_{img_0} is designed to produce an image that conforms to both the layout and text.

We first compute the global layout embedding vector $\mu_0 \in \mathbb{R}^{h\times w\times d}$ by down-sampling the global layout L_0 and add noise z by spatial replication and depth concatenation. The text embedding φ calculated by the pre-trained LSTM in the box regression network, the layout encoding μ_0, and noise z are concatenated and fed into a residual unit implemented by several residual layers. Our model jointly aggregates the bounding box and text information into a latent feature representation, and we further apply one up-sampling layer to generate the global hidden feature vector y_0 from the latent representations. After the final 3×3 convolution layers, the global image with 64×64 resolution is initially generated. Specifically:

$$y_0 = F_0(\mu_0, \varphi, z) \tag{13}$$

$$I_0 = G_{img_0}(y_0) \tag{14}$$

where F_0 is modeled as neural networks, y_0 is the global hidden layer feature vector. Conditioned on y_0 , the global generator G_{img_0} then generates the low-resolution image I_0.

2) LOCAL REFINEMENT GENERATOR

In the first stage of generation, local details are not explicitly utilized for instance-level image generation, most of the synthesized images lack fine-grained features, resulting in overly smooth textures. To generate high-resolution images, we further employ the local refinement generator, and the overall architecture is illustrated in Fig. 4. During the refinement process, we only repeat two times due to the memory limitation of GPU. With two identical local refinement generators G_{img_1} and G_{img_2}, we first generate the 128×128 images then synthesize 256×256 images.

a: INSTANCE-WISE ATTENTION

Our local refinement generator is designed as the encoder-decoder structure. It first encodes the global layout L_1 by several down-sampling layers to obtain the layout encoding vector $\mu_1 \in \mathbb{R}^{h\times w\times d}$ (d indicts the layout feature dimension). Considering that traditional grid attentionhas been successfully used for image captioning [39], image-to-image translation [40], and visual questioning and answer-ing [41]; the attention-based generative adversarial network AttnGAN uses attention mechanism for image generation; our two local refinement generators need to encode various context information of L_1 along the channel dimension. Hence, as shown in the bottom of Fig. 5, we employ the instance-wise attention to select the context relevant features.

Specifically, with the sub-region vectors V_{region} of the pre-generated image I_0, our local refinement generator retrieves the relevant instance vectors from the layout L_1. Afterward, it assigns instance-wise attention weights to each instance

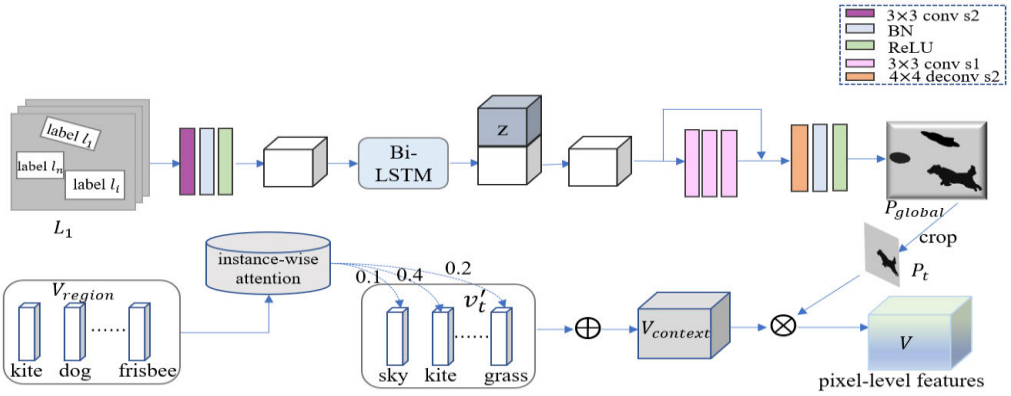

FIGURE 5. Overall framework of the proposed instance mask embedding mechanism.

vector V'_t and then calculates the weighted sum of the input information. The instance-wise context vector of the t-th object is calculated as:

$$V^t_{context} = \sum_{t=1}^{T} w_t V'_t \tag{15}$$

where $t \in (1, 2, \cdots, T)$ denotes the number of objects, V'_t and w_t represent the embedding vector and the attention weight of the t-th instance, respectively.

b: INSTANCE MASK EMBEDDING MECHANISM

Different parts of bounding boxes may overlap during the refinement process, multiple pixels may cover the same pixel, and the output shapes do not always align with the ground truth. These problems can be solved as a space sampling issue where the proposed instance mask embedding can pose spatial and morphological constraints on instance feature projection.

In general, many methods use mask annotations, which are not flexible to obtain, to separately add the shape of each instance. As a result, the generated images as a whole may present poor scene layouts though each instance is correctly rendered. Differently, we employ the predicted pixel-level instance mask embedding for image synthesis, in this way we can avoid consuming too much model capacity and unstable training.

As shown in the top of Fig. 5, given a global layout L_1, we use the mask regression network to obtain the aggregated mask $P_{global} \in \mathbb{R}^{h \times w}$. Our down-sampling block is made up of a 3×3 convolution (stride-2) followed by batch normalization and ReLU activation, the residual unit is implemented with three 3×3 convolution layers and a skip connection, and the up-sampling block consists of a 4×4 deconvolution (stride-2) followed by the batch normalization and ReLU activation. Then the aggregated mask P_{global} is cropped to get the t-th instance mask embedding P_t To clearly represent the overlapping parts and make the generated features comply with the instance mask embedding, the most relevant context vector should be selected by the local refinement generator. Thus, for the t-th instance, we copy the instance-wise context vector $V^t_{context}$ to the instance mask embedding P_t, the pixel-level feature vector V which contains latent pixel details is calculated by:

$$V = \max_{t:1 \le i \le T} P_t \otimes V^t_{context} \tag{16}$$

where $\otimes$ is the vector outer-product, t is the number of instances in the image. When there are several pixels covering a single pixel, we perform max-pooling to select the corresponding pixel that associated with the most related instance-wise context vector $V^t_{context}$, then employ pixel representation at this position.

Meanwhile, in order to integrate the global information from G_{img_0} to G_{img_1}, we inject the global hidden layer feature vector y_0 into the refinement stage (see Fig. 4). y_0, μ_1, and V as input are aggregated by concatenation along the channel dimension and subsequently fed into a residual unit. We further apply one up-sampling layer as the decoder to calculate the local hidden feature vector y_1. As the input of the final 3×3 convolution layers, the hidden layer vector y_1 is subsequently mapped to an image with resolution 128×128. Specifically:

$$y_1 = F_1(y_0 + \mu_1, V, V_{context}) \tag{17}$$

$$I_1 = G_{img_1}(y_1) \tag{18}$$

where F_1 is modeled as neural networks, y_0 is the global hidden feature vector, and μ_1 represents the high-resolution layout encoding. The pixel-level feature vector V and the instance-wise context vector $V_{context}$ are calculated and aggregated into the concatenation of μ_1 and y_0 to get the local hidden feature vector y_1. Then the local refinement generator outputs a high-resolution image I_1 conditioned on the hidden feature vector. Additionally, we also apply another local refinement generator G_{img_2} and finally have synthesized images with the resolution 256×256.

3) ATTRIBUTE-ADAPTIVE DISCRIMINATOR

The discriminator should have a large receptive field to differentiate synthesized and ground truth [42], this requires either

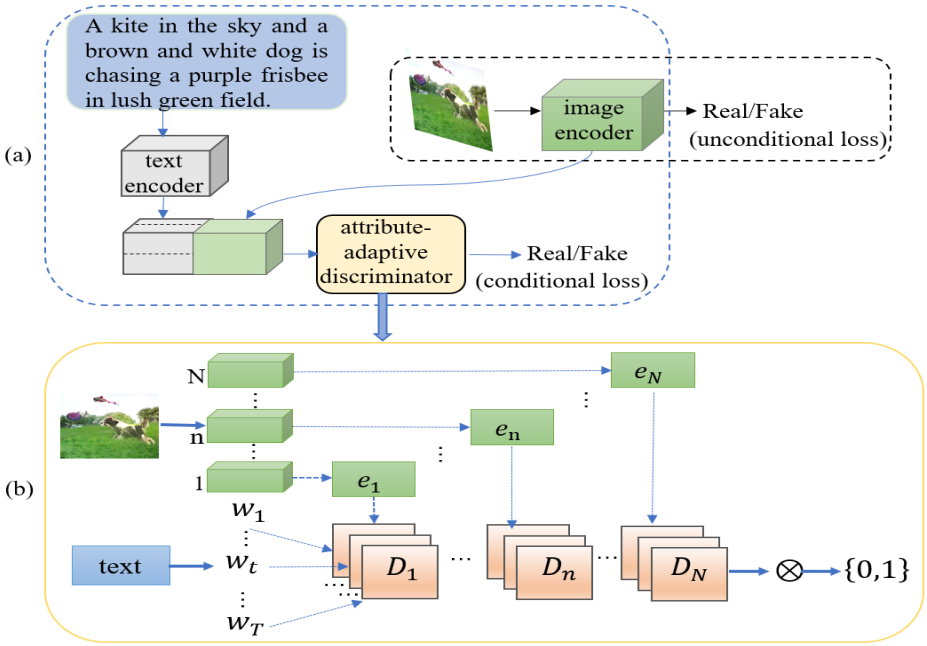

FIGURE 6. Overview of the image discriminator (a) and the proposed attribute-adaptive discriminator (b).

bigger convolution kernels or a considerably deeper network, resulting in an increased model capacity and repeated patterning images. To this end, we employ multi-scale discriminators D_{img_0}, D_{img_1}, and D_{img_2} to separately train different resolution images. The sentence-level discriminator is adopted for D_{img_0}, the identical D_{img_1} and D_{img_2} are designed as word-level attribute-adaptive discriminators.

Generative models tend to synthesize the "average" pattern instead of the related attribute features, this is mainly because the global sentence-wise discriminator cannot be attached to a specific type of visual attribute and only provides the coarse training feedback. Therefore, our attribute-adaptive discriminators D_{img_1} and D_{img_2} are trained to recognize each attribute and discriminate whether it exists in the synthesized image. Each attribute-adaptive discriminator is made up of word-level discriminators to disentangle different attributes with fine-grained training signals. the overall structure of the image discriminator and the proposed word-level attribute-adaptive discriminator is shown in Fig. 6.

The attribute-adaptive discriminator consists of a set of word-level discriminators $\{D_1, D_2, \cdots, D_N\}$. Given an image, the image encoder outputs image features, see Fig. 6 (b), we implement the global average pooling to all feature layers to compute the one-dimensional image feature vector e. Meanwhile, we use the text encoder to get word vectors $\{w_1, w_2, \cdots, w_T\}$, then respectively feed them into word-level discriminators. Take the t-th word vector w_t as an example, the one-dimensional sigmoid word-level discriminator F_{wt} is used to decide whether the synthesized image contains a visual attribute that related to w_t. Specifically, the word-level discriminator F_{w_t} is represented as:

$$F_{w_t}(e_n) = \sigma(W(w_t) \cdot e_n + b(w_t)) \tag{19}$$

where σ is the sigmoid function, e_n represents the one-dimensional image vector of the n-th image feature layer, $W(w_t)$ denotes the weight matrix and $b(w_t)$ is the bias.

We also reduce the influence of less significant words in the discrimination process. For this, we apply the word-level instance-wise attention to indicate the correlation degree between the word and the visual attribute. The attention mechanism mainly has two aspects: calculating attention distributions; computing the average of the weighted sum based on attention distributions. Note that the discriminator should have a multi-scale receptive field to detect multi-scale image features, the attention distribution $\alpha_{t,n}$ is calculated as:

$$\alpha_{t,n} = \frac{\exp(S_{t,n})}{\sum_{t=1}^{T} \exp(S_{t,n})}, \quad S_{t,n} = (\bar{v})^{\mathrm{T}} w_t \tag{20}$$

where $\alpha_{t,n}$ is the attention weight assigned to the t-th word of n-th image feature layer. $S_{t,n}$ is the attention scoring function calculated by the dot product model. $\bar{v}$ denotes the average of word vector w_t.

With the attention distribution, the final score of the word-level discriminator is multiplicatively aggregated as:

$$D(I, x) = \prod_{t=1}^{T} [\sum_{n=1}^{N} \gamma_{tn} F_{w_t}(e_n)]^{\alpha_{t,n}} \tag{21}$$

where I represents the generated image, x denotes the text, T is the total number of input words, $\alpha_{t,n}$ represents the attention distribution. γ_{tn} is the weight of softmax function, and this parameter is used to determine the importance of each word for the layer n.

Hence, compared with the sentence-level discriminator that operates at coarse-level and only determines whether the

synthesized image roughly matches the text, our attribute-adaptive discriminators can provide feedbacks at different stages and identify the existence of related visual attributes.

C. OBJECTIVE FUNCTION

Our final objective function consists of a GAN adversarial loss [1] and a DAMSM loss [29]. The GAN cross-entropy loss function $\mathcal{L}_{\text{GAN}}$ is determined by the adversarial training of image generators and attribute-adaptive discriminators. Both generators and discriminators all consist of an uncon-ditional loss and a conditional loss. The generator objective is defined as:

$$\begin{aligned}&\mathcal{L}_{\text{GAN}}(G)\\&= \sum_{i=0,1,2} \mathcal{L}_{G_{img_i}} = -\frac{1}{2}[\mathbb{E}_{I\sim P_{G_{img_i}}} \log D_{img_i}(I)\\&+\mathbb{E}_{I\sim P_{G_{img_i}}} \log D_{img_i}(I,x)]\end{aligned} \tag{22}$$

where the first item represents the unconditional loss, the second item is the conditional loss, I and x denote the synthesized image and the related text, respectively.

The adversarial loss for each discriminator also consists of an unconditional and a conditional item:

$$\begin{aligned}&\mathcal{L}_{\text{GAN}}(D)\\&= \sum_{i=0,1,2} \mathcal{L}_{D_{img_i}} = -\frac{1}{2}[\mathbb{E}_{I\sim P_{data}} \log D_{img_i}(I)\\&+\mathbb{E}_{I\sim P_{G_{img_i}}} \log(1-D_{img_i}(I)) + \mathbb{E}_{I\sim P_{data}} \log D_{img_i}(I,x)\\&+\mathbb{E}_{I\sim P_{G_{img_i}}} \log(1-D_{img_i}(I,x))]\end{aligned} \tag{23}$$

where P_{data} represents the distribution of the ground truth.

Additionally, we adopt the DAMSM loss introduced in AttnGAN to calculate the fine-grained image-text matching loss. Hence, our final objective loss is obtained by:

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \lambda_1 \mathcal{L}_{DAMSM} \tag{24}$$

where λ_1 is a hyper-parameter. $\mathcal{L}_{DAMSM}$ is the loss of the Deep Attentional Multimodal Similarity Model (DAMSM) pre-trained on ground truth images and related descriptions.

V. EXPERIMENTS

A. EXPERIMENTAL SETUPS

Extensive experiments are conducted to qualitatively and quantitatively evaluate the proposed IMEAA-GAN. We use a single Tesla P100 with 16GB video memory, Linux 4.4.0-135-generic operating system, and PyTorch 0.4.1 framework. Our model is trained for 200 epochs with a batch size of 16 on MS-COCO, and 800 epochs with a batch size of 10 on the Caltech-UCSD Birds-200-2011 (CUB) dataset. We set the learning rates of generators and discriminators all to 0.0002, the hyper-parameter of DAMSM loss is set $\lambda_1 = 50$ on MS-COCO and $\lambda_1 = 5$ on CUB. We use the Adam algorithm [43] to optimize the adversarial training. The exponential decay rates $\beta_1, \beta_2 \in [0, 1)$ for the first and second moment estimates are set to 0.5, 0.999, respectively.

1) DATASETS

We perform experiments on MS-COCO and CUB datasets. The MS-COCO dataset [44] has pixel-level annotations and contains 82,783 training images, 40504 validation, and 40,775 testing images. There are 80 object categories in this dataset, each image has 5 text descriptions and corresponding instance labels.

Derived from the CUB-200 dataset, the CUB dataset [45] includes a total of 11,788 images that provide class labels, bounding boxes, and bird attributes information. It has 200 different bird categories, each image has 10 descriptions describing the bird attributes. We employ 150 bird categories (including 8,855 images) as our training set while those other 50 categories (including 2,933 images) as the testing set.

2) EVALUATION METRICS

We use the Inception Score (IS) [46], Fréchet Inception Distance (FID) [47], and R-precision [29] to quantitatively evaluate the generation performance of IMEAA-GAN.

A pre-trained Inception v3 network [48] is adopted to compute the IS and FID. The IS evaluates the image quality and diversity, namely: this metric measures the uniqueness of synthesized images and the number of object categories [49], while the FID calculates the Wasserstein-2 distance [50] between the ground truth and synthesized images according to final layer activations. A lower FID indicates a shorter distance between the generated image distribution and ground truth image distribution. Therefore, the larger the IS value while the smaller the FID value, the better the model performance. Same as AttnGAN and MirrorGAN [51], we also apply R-precision to measure the matching degree between the image and text. Specifically, we randomly select 99 descriptions from the dataset, then compute cosine distance to indicate the similarity (in feature space) between the generated image and the related text. We sort these 100 (including a ground truth text) descriptions and select the top k most similar descriptions to calculate the R-precision. In practice, we set $k = 1$, meaning that the R-precision indicates whether the ground truth text more closely matches the synthesized image than those 99 randomly sampled text descriptions.

B. QUALITATIVE RESULTS

Our model has produced high-fidelity 256×256 images containing complex scenes and multiple instances, Fig. 7 shows the synthesized results on MS-COCO. Conditioned on the instance mask embedding, IMEAA-GAN is able to separate instances from the background, reduce overlapping pixels. Given the similar input, due to the use of attribute-adaptive discriminators, IMEAA-GAN can also synthesize various detailed attributes. For example, the sheep in the third column of Fig. 7 show that our approach can well distinguish the word-level information and generate diverse images corresponding to various features.

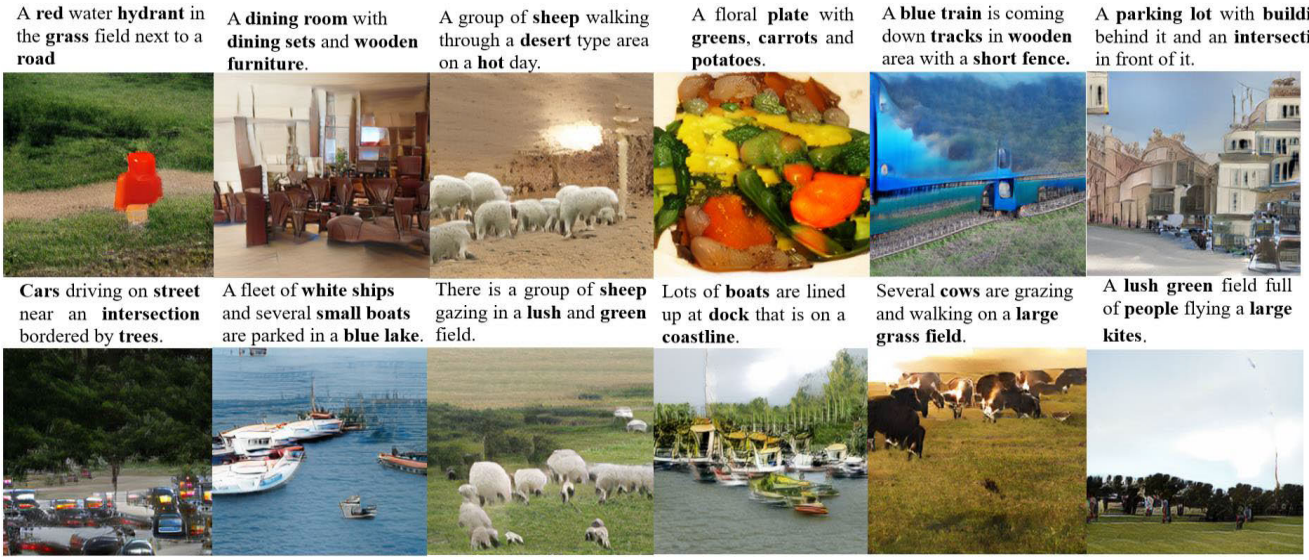

FIGURE 7. Qualitative examples generated by IMEAA-GAN on the MS-COCO dataset.

FIGURE 8. Qualitative examples generated by IMEAA-GAN on the CUB dataset.

To prove the generalization ability of our IMEAA-GAN, we also perform experiments on the CUB dataset. As shown in Fig. 8, the generate high-quality 256×256 images vividly display the color and texture of different birds, there are almost no indistinguishable instances and overlapping parts by using instance mask embedding mechanism, Moreover, with the guidance of attribute-adaptive discriminators, our images present correct and fine-grained attributes.

We adopt the multi-stage generation strategy to synthesize high-resolution images. During the refinement stage, we have attempted to stage up the generator to 4. However, the training process becomes unstable and difficult to control due to the complexity of deep neural networks and the memory limitation of the GPU. Therefore, we only apply one global generator and two local refinement generators for the optimal generation. The intermediate results of different stages on CUB and MS-COCO are illustrated in Fig. 9.

Figure 9 shows that IMEAA-GAN is capable of refining images to match the text. The global generator initially generates coarse-grained 64×64 images (e.g. Fig. 9(a)), but these synthesized images lack fine-grained textures. Then two local refinement generators generate fine-grained images (e.g. Fig. 9(b), Fig. 9(c)). The context-wise instance vectors can be obtained by our generators, so the synthesized images

FIGURE 9. Example results of different stages of the IMEAA-GAN on MS-COCO (left) and CUB (right) datasets.

FIGURE 10. Qualitative comparison of different methods on the MS-COCO dataset.

are further well-improved, contain more accurate texture features and clear backgrounds. For example, in the second right row of Fig. 9, there is no short beak in the initial 64 × 64 image, our local refinement generators are able to encode the "short beak" information and synthesize the missing features.

Further, as illustrated in Fig. 10, we compare the IMEAA-GAN with other methods conditioned on the same text. The sg2im method converts the input text into scene graphs to infer semantic layouts, and this approach has achieved the synthesis of 128 × 128 images. But scene graphs lack core object attributes and spatial information (e.g. positions and sizes), it is difficult to generate details that consistent with semantic layouts. In addition, the information conveyed by scene graphs is very limited, the features of an instance are not only determined by its position and class labels but also interactions with others, so it fails to solve the overlapping pixels and separate different object appearances.

As shown in the second row of Fig. 10, AttnGAN has synthesized 256 × 256 images. Conditioned on a sentence

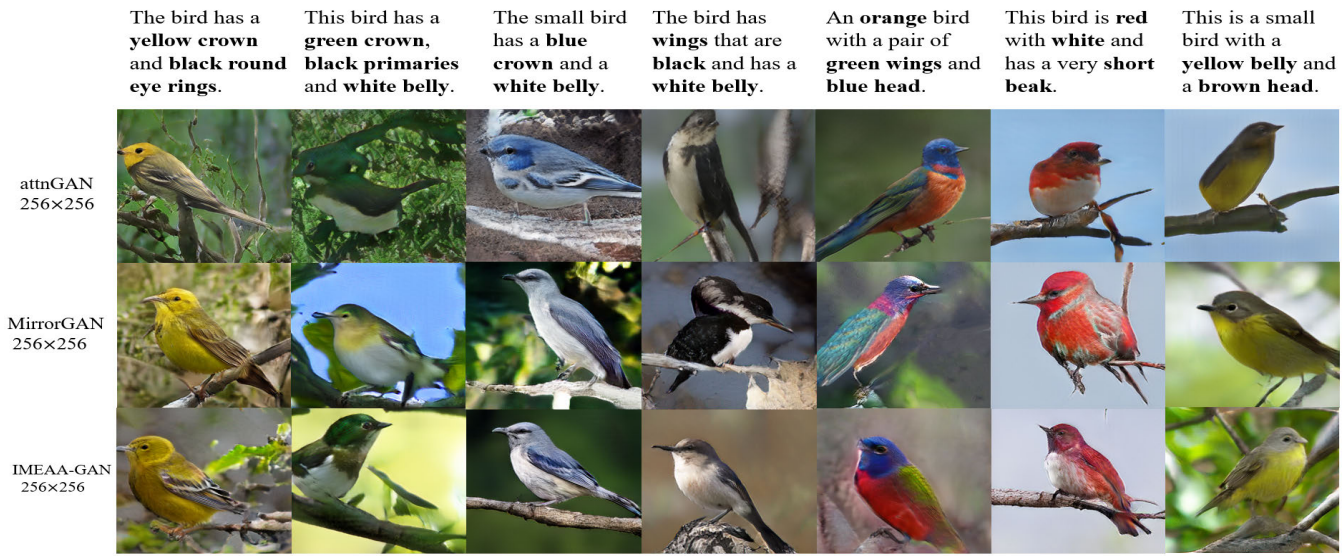

FIGURE 11. Qualitative comparison of different methods on the CUB dataset.

vector, the effect of each word is not fully considered, it assigns all instances with the same weight. Thus, lacking word-level embedding and ignoring interactions between different instances are difficult for it to generate high-quality images. Besides, it uses sentence-level discriminators that only provide coarse-grained feedback, so its generators tend to generate texture associated with the wrong word. This can explain why the synthesized results appear realistic features but lack meaningful layouts and correct attributes.

The recent MirrorGAN [51] has made great progress on complex image generation, the example results are shown in the third row of Fig. 10. This method outperforms the first two models, it guarantees the semantical consistency in multiple object generation, the synthesized images match the text at the image level. Yet, MirrorGAN lacks investigations on uneven instance distribution and feature occlusion, the visual appearance and instance interactions are not finely regulated. For example, the ''cattle'' in the first image of the third row contain reasonable appearance, but the ''green hillside'' is inappropriately shown as the ''dry field''.

Different from these aforementioned methods, IMEAA-GAN adopts word-level attribute-adaptive discriminators. As presented in the last row of Fig. 10, the synthesized instances have correct attributes. Besides, due to the use of instance mask embedding and instance-wise attention mechanism, as well as the maximum pooling of multiple pixels, overlapping pixels between different instances have been solved. So these generated instances, which contain clear shapes and texture features, are more recognizable and semantically meaningful.

We also perform comparative experiments on the CUB dataset as shown in Fig. 11. Since sg2im mainly aims at the positional relationship between different instances, every image in CUB only contains a single object, so we just compare the IMEAA-GAN with AttnGAN and MirrorGAN. Observing the second and third columns of Fig. 11, though these two methods both accurately capture attribute features, IMEAA-GAN can better display the main attributes and differentiate birds from their backgrounds. In general, our approach has the capacity to synthesize individuals with more vivid details as well as more clear shapes.

Figure 12 demonstrates that IMEAA-GAN can generate diverse images using the same input. The results contain various shapes and complex scenes, this is mainly owing to word-level attribute-adaptive discriminators which provide specific signals. Therefore, only changing a few words, under the guidance of discriminators, the generators can synthesize images with detailed attributes, and these samples look similar but unique to each other.

TABLE 1. Quantitative comparison of different methods on the MS-COCO dataset.

Methods	IS ↑	FID ↓	R-precision ↑
sg2im	7.3 ± 0.16	81.83	65.6 ± 0.73
AttnGAN	23.79 ± 0.32	35.49	85.47 ± 0.69
MirrorGAN	26.47 ± 0.41	(-)	86.01 ± 0.42
IMEAA-GAN	$\mathbf{30.49 \pm 0.57}$	**32.64**	$\mathbf{88.56 \pm 0.38}$

C. QUANTITATIVE RESULTS

As shown in Table 1 and Table 2, we measure the performance of different methods in terms of IS, FID, and R-precision, the best results are in bold. Based on the MS-COCO and CUB datasets, compared with MirrorGAN, we have almost increased IS by 15.19% and 4.17%, R-precision by 2.96% and 2.63%. Compared with the officially pre-trained AttnGAN, our model decreases the FID by 8.03% on MS-COCO and 32.90% on CUB, which confirms that IMEAA-GAN is able to generate images with more diverse objects and higher quality than other methods.

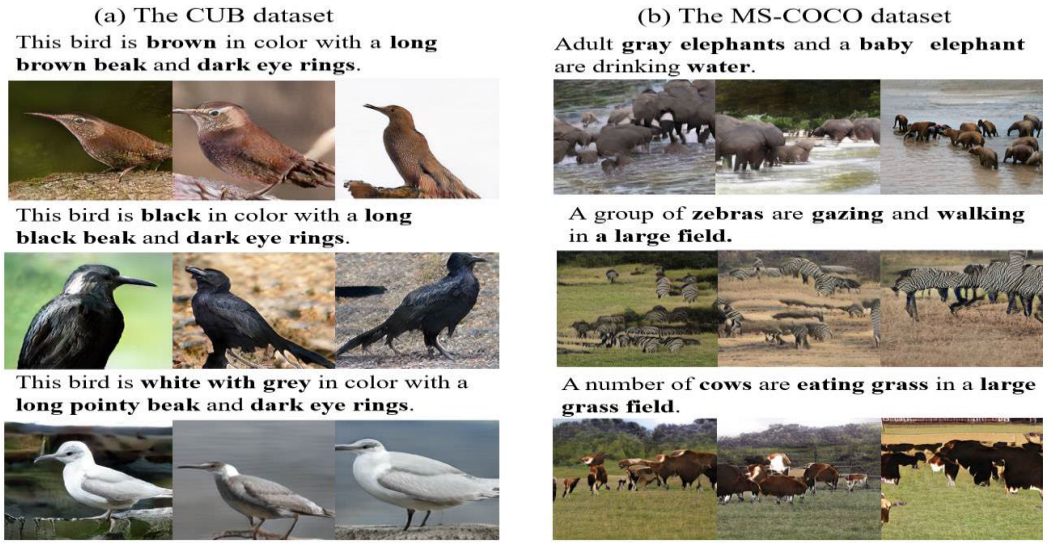

FIGURE 12. Diverse results of attribute-adaptive generation.

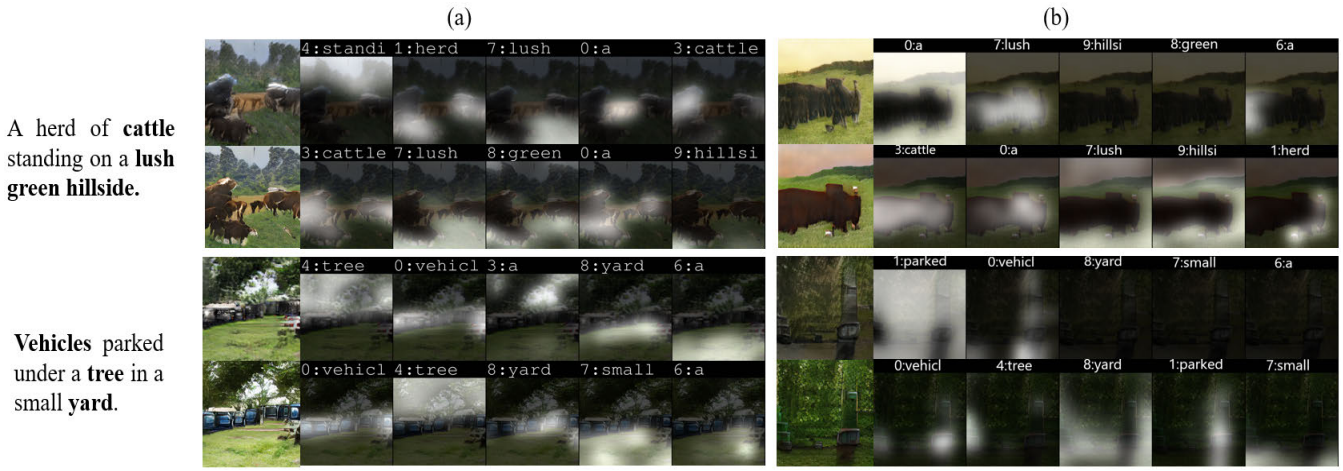

FIGURE 13. Visualization and comparison of attribute-adaptive discriminators (a) and sentence-level discriminators (b).

Our model can obtain the most relevant instance at the position where has overlapping pixels, so these synthesized results are closely consistent with global layouts and ground truth images. Hence, as demonstrated in Table 1 and Table 2, feeding generated results into the pre-trained Inception v3 network, we get better performance of the IS and FID. In addition, we also obtain the highest R-precision, which indicates that the images and attributes generated by our generators are most relevant to descriptions. However, other methods occur lots of overlapping pixels and blurred objects, and the Wasserstein-2 distances between ground truth and generated samples are quite large. So it is hard to adaptively disentangle corresponding visual features under linguistic expression variants. By comparison, IMEAA-GAN greatly improves the quality and diversity of generated images, as well as the text-image matching degree.

Besides, observing that the IS values based on CUB differ significantly from the MS-COCO, this is because all images in CUB are birds and the feature distributions are similar, while the MS-COCO contains different instance categories and complex scenes, the feature distributions among various objects are greatly different. Therefore, the IS values on MS-COCO are generally larger than that of the CUB.

TABLE 2. Quantitative comparison of different methods on CUB.

Methods	IS ↑	FID ↓	R-precision ↑
sg2im	(-)	(-)	(-)
AttnGAN	4.36 ± 0.03	23.98	67.82 ± 4.43
MirrorGAN	4.56 ± 0.05	(-)	70.46 ± 0.54
IMEAA-GAN	$\mathbf{4.75 \pm 0.07}$	**16.09**	$\mathbf{72.31 \pm 0.91}$

D. ABLATION STUDY

To verify the effectiveness of the proposed discriminators, as shown in Fig. 13 (a), we visualize our word-level attribute-adaptive discriminators. Meanwhile, to make a comparison,

FIGURE 14. **Ablation comparison of the instance mask embedding effect.**

we adopt two commonly used sentence-level discriminators which have the same structure as our baseline model, the visualization maps of sentence-level discriminators are presented in Fig. 13 (b). The highlighted regions indicate the feedback information provided by discriminators. With feedbacks, generators are instructed to synthesize related attributes and instances. The discriminators in our baseline model are conditioned on a whole sentence, so it is hard to highlight word-level regions, thus, resulting in an excessively large range of highlighted areas. What's worse, the baseline method even omits highlighting when synthesizing certain attributes, see images in the third row of Fig. 13(b).

All these illustrate that sentence-level discriminators can only provide the coarse-grained information and fail to offer effective feedback signals. In contrast, our attribute-adaptive discriminators are word-level that can provide generators with detailed attribute feedbacks and highlight the related regions. Therefore, our generators can focus on the most relevant regions to perform pixel-level attribute generation.

Further, we also demonstrate the necessity of instance mask embedding for the local refinement generators. The image quality of the ablated version and our full model are shown in Fig. 14 (a) and Fig. 14 (b), respectively. The ablated model has the same settings as our full IMEAA-GAN except that it does not use the instance mask embedding (Fig. 14 (a) w/o IME). Images in Fig. 14 (a) lack detailed and complete features, for example, the zebras and giraffes in the first row are only synthesized with scattered features. It is difficult for the ablated model to synthesize corresponding instances in the correct locations, so the image accuracy and fidelity are quite low. With the instance mask embedding (Fig. 14 (b) w/ IME), the synthesized images can meet the shape and location constraints. Even for complex scenes, for example, the giraffes in Fig. 14 (b), there are almost no overlapping pixels and indistinguishable instances.

VI. CONCLUSION

In this paper, we present a novel Instance Mask Embedding and Attribute-Adaptive Generative Adversarial Network (IMEAA-GAN) for text-to-image generation. With the instance mask embedding, which provides shape constraints and solves the overlapping problem between different pixels, our two local refinement generators are able to refine the initial image synthesized by a global generator. We also proposed the word-level attribute-adaptive discriminators, which focus on individual attributes and provide effective feedback to discriminate whether the generated instances match the attribute descriptions, so as to guide generators synthesize accurate features. Experimental results illustrate that our model is capable of generating complex images with high-fidelity attributes on different datasets. However, once the text contains various scene settings and instances, the image quality drops drastically. Our future work will focus on using knowledge graphs to infer corresponding semantic layouts and generating multiple high-resolution images from a single semantic layout.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[2] T. Hinz, S. Heinrich, and S. Wermter, "Semantic object accuracy for generative Text-to-Image synthesis," 2019, *arXiv:1910.13321*. [Online]. Available: http://arxiv.org/abs/1910.13321

[3] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Jul. 2017, pp. 1511–1520.
[4] H. Tang, X. Chen, W. Wang, D. Xu, J. J. Corso, N. Sebe, and Y. Yan, "Attribute-guided sketch generation," 2019, *arXiv:1901.09774*. [Online]. Available: http://arxiv.org/abs/1901.09774
[5] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2888–2897.
[6] D. Zhang, J. Shao, G. Hu, and L. Gao, "Sharp and real image super-resolution using generative adversarial network," in *Proc. Int. Conf. Neural Inf. Process.*, Guangzhou, China, Nov. 2017, pp. 217–226.
[7] J.-Y. Zhu, P. Krähenbühl, E. E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Comput. Vision-(ECCV)*, vol. 9909. Cham, Switzerland: Springer, 2018, pp. 597–613.
[8] Y. Lu, S. Wu, Y.-W. Tai, and C.-K. Tang, "Image generation from sketch constraint using contextual GAN," in *Proc. Eur.Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 205–220.
[9] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. Eur.Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 172–189.
[10] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*. [Online]. Available: http://arxiv.org/abs/1809.11096
[11] S. Sharma, D. Suhubdy, V. Michalski, S. E. Kahou, and Y. Bengio, "Chatpainter: Improving text to image generation using dialogue," Feb. 2018, *arXiv:1802.08216*. [Online]. Available: https://arxiv.org/abs/1802.08216
[12] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1219–1228.
[13] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. Int. Conf. Mach. Learn.*, May 2016, pp. 1060–1069.
[14] H. Dong, S. Yu, C. Wu, and Y. Guo, "Semantic image synthesis via adversarial learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5706–5714.
[15] C.-X. Li, T. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4088–4098.
[16] A. Dash, J. Cristian Borges Gamboa, S. Ahmed, M. Liwicki, and M. Zeshan Afzal, "TAC-GAN–text conditioned auxiliary classifier generative adversarial network," 2017, *arXiv:1703.06412*. [Online]. Available: http://arxiv.org/abs/1703.06412
[17] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4467–4477.
[18] M. Cha, Y. L. Gwon, and H. T. Kung, "Adversarial learning of semantic relevance in text to image synthesis," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 3272–3279, Jul. 2019.
[19] E. L. Denton, S. Chintala, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1486–1494.
[20] H. Zhang, T. Xu, and H. Li, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5907–5915.
[21] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.
[22] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical Text-to-Image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7986–7994.
[23] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "LayoutGAN: Generating graphic layouts with wireframe discriminators," 2019, *arXiv:1901.06767*. [Online]. Available: http://arxiv.org/abs/1901.06767
[24] B. Zhao, L. Meng, W. Yin, and L. Sigal, "Image generation from layout," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 8584–8593.
[25] T. Hinz, S. Heinrich, and S. Wermter, "Generating multiple objects at spatially distinct locations," 2019, *arXiv:1901.00686*. [Online]. Available: http://arxiv.org/abs/1901.00686
[26] W. Li, P. Zhang, L. Zhang, Q. Huang, X. He, S. Lyu, and J. Gao, "Object-driven Text-To-Image synthesis via adversarial training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12174–12182.
[27] M. Yuan and Y. Peng, "Text-to-image synthesis via symmetrical distillation networks," 2018, *arXiv:1808.06801*. [Online]. Available: http://arxiv.org/abs/1808.06801
[28] W. Sun and T. Wu, "Image synthesis from reconfigurable layout and style," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 10531–10540.
[29] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 1316–1324.
[30] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 6077–6086.
[31] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[32] C. E. Rasmussen, "The infinite Gaussian mixture model?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 554–560.
[33] P. Paalanen, J.-K. Kamarainen, J. Ilonen, and H. Kälviäinen, "Feature representation and discrimination based on Gaussian mixture model probability densities—Practices and algorithms," *Pattern Recognit.*, vol. 39, no. 7, pp. 1346–1358, Jul. 2006.
[34] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, "Semantic graph convolutional networks for 3D human pose regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3425–3435.
[35] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3150–3158.
[36] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3147–3155.
[37] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[38] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 6199–6208.
[39] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4651–4659.
[40] S. Ma, J. Fu, C. W. Chen, and T. Mei, "DA-GAN: Instance-level image translation by deep attention generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 5657–5666.
[41] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," 2016, *arXiv:1606.00061*. [Online]. Available: http://arxiv.org/abs/1606.00061
[42] G. Seif and D. Androutsos, "Large receptive field networks for high-scale image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Salt Lake City, UT, USA, Jun. 2018, pp. 763–772.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[45] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR2011-001, 2011.
[46] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
[47] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.

[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.
[49] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, "MMD GAN: Towards deeper understanding of moment matching network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2203–2213.
[50] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.
[51] T.-T. Qiao, J. Zhang, D.-Q. Xu, and D.-C. Tao, "MirrorGAN: Learning text-to-image generation by redescription," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 1505–1514.

JIANCHENG NI received the Ph.D. degree in computer science from Sichuan University, Sichuan, China, in 2008. He is currently a Professor and the Deputy Dean of the School of Software, Qufu Normal University. He is also a Senior Engineer of Huawei Cloud Computing and Big Data. He is also the author of five textbooks, six software copyrights, and more than 40 academic articles. His research interests include computer vision, machine learning, and distributed computing.

SUSU ZHANG received the double B.S. degrees in software engineering and English from Qufu Normal University, Qufu, China, in 2018. She is currently pursuing the M.S. degree in software engineering. She is also a member of the China Computer Federation and a Software Engineer of Hewlett Packard Enterprise. In 2018, she received the postgraduate recommendation from Qufu Normal University. Her research interests include image generation, deep learning, and computer vision.

ZILI ZHOU was born in Heze, Shandong, China, in 1973. He received the M.S. and Ph.D. degrees from the Information College of East China Normal University, Shanghai, China, in 2004 and 2009, respectively. He is currently an Associate Professor with Qufu Normal University. He is also the author of more than 30 articles and six inventions. His research interests include natural language understanding, knowledge graph, and ontology.

JIE HOU was born in Jining, Shandong, China, in 1996. She received the B.S. degree in software engineering from Qufu Normal University, in 2018, where she is currently pursuing the M.S. degree. Her current research interests include image processing, handwritten Chinese recognition, and digitization of ancient documents.

FENG GAO was born in Qingdao, Shandong, China, in 1995. He received the B.S. degree from Qufu Normal University, in 2018, where he is currently pursuing the M.S. degree. His major research fields are semantic fusion, the representation and processing of intelligence information. His research interests include the graph neural networks and machine reading comprehension in natural language processing.