DTGAN: Dual Attention Generative Adversarial Networks for Text-to-Image Generation

Zhenxing Zhang, Lambert Schomaker
Bernoulli Institute, University of Groningen
Groningen, The Netherlands
z.zhang@rug.nl, l.r.schomaker@rug.nl

Abstract—Most existing text-to-image generation methods adopt a multi-stage modular architecture which has three significant problems: 1) Training multiple networks increases the run time and affects the convergence and stability of the generative model; 2) These approaches ignore the quality of early-stage generator images; 3) Many discriminators need to be trained. To this end, we propose the Dual Attention Generative Adversarial Network (DTGAN) which can synthesize high-quality and semantically consistent images only employing a single generator/discriminator pair. The proposed model introduces channel-aware and pixel-aware attention modules that can guide the generator to focus on text-relevant channels and pixels based on the global sentence vector and to fine-tune original feature maps using attention weights. Also, Conditional Adaptive Instance-Layer Normalization (CAdaILN) is presented to help feature maps using attention weights. Furthermore, a new type of visual loss is utilized to enhance the image resolution by ensuring vivid shape and perceptually uniform color distributions of generated images. Experimental results on benchmark datasets demonstrate the superiority of our proposed method compared to the state-of-the-art models with a multi-stage framework.

Index Terms—text-to-image synthesis, attention mechanism, conditional normalization, visual loss

I. INTRODUCTION

Generating high-resolution realistic images conditioned on given text descriptions has become an attractive and challenging task in computer vision (CV) and natural language processing (NLP). It has various potential applications, such as art generation, photo-editing and video games. Recent work has achieved crucial improvements in the quality of generated samples through generative adversarial networks (GAN) [1]–[3], while also boosting the semantic consistency between generated visually realistic images and given natural-language descriptions.

However, most state-of-the-art approaches in text-to-image generation [1]–[7] are based on a multi-stage modular architecture as shown in Fig. 1a. Specifically, the network comprises multiple generators which have corresponding discriminators. Furthermore, the generator of the next stage takes the result of the previous stage as the input. This framework has proven to be useful for the task of text-to-image synthesis, but there still exist three significant problems. Firstly, training many networks increases the computation time compared to a unified model and affects the convergence and stability of the generative model [8]. Even worse, the final generator network cannot be improved if the previous generators do not converge to a global optimum. Secondly, this framework ignores the quality of early-stage generator images which plays a vital role in the resolution of finally-generated images [2]. The generator networks for precursor images ($G_0$ in Fig. 1a) are composed of up-sampling layers and convolution layers, lacking the image integration and refinement process with the input natural-language descriptions. Thirdly, multiple discriminators need to be trained.

To address the issues mentioned above, we propose a novel Dual Attention Generative Adversarial Network (DTGAN) which can fine-tune the feature maps for each scale according to the given text descriptions, and synthesize high-quality images only using a single generator/discriminator pair. The overall architecture of the DTGAN is illustrated in Fig. 1b. Our DTGAN consists of four new components, including two new types of attention modules, a new normalization layer, and a new type of visual loss. The first two components in the DTGAN are our designed channel-aware and pixel-aware attention modules which can guide the generator network to focus more on important channels and pixels, and to ignore text-irrelevant channels and pixels by computing attention weights between the global sentence vector and two aforementioned factors. Different from earlier attention models [3], [5], we employ global average pooling and global max pooling to obtain the discriminative regions of image feature maps. In addition,
we apply the attention scores to fine-tune original feature maps rather than adopt the weighted sum of converted word features as new feature maps. We expect that our proposed attention method will significantly bridge the semantic gap between generated images and text descriptions. In the third ingredient, inspired by Adaptive Layer-Instance Normalization (AdaLIN) [9], we present Conditional Adaptive Instance-Layer Normalization (CAdaILN), where the ratio between Instance Normalization [10] and Layer Normalization [11] is adaptively learned during training and the global sentence vector is employed to scale and shift the normalized result. The CAdaILN function is complementary to the attention modules and helps with controlling the amount of change in shape and texture. As a result, armed with the attention modules and CAdaILN, our network can generate photo-realistic images only exploiting a single generator/discriminator pair. The last proposed component is a new variant for computing the visual loss. It is introduced to ensure that generated images and real images have similar color distributions and shape. We expect that the choice of this novel visual loss has a considerable impact on the quality of generated results.

We perform extensive experiments on the CUB bird [12] and MS COCO [13] datasets to evaluate the effectiveness of our proposed DTGAN. Both qualitative and quantitative results demonstrate that our approach outperforms existing state-of-the-art models. The contributions of our work can be summarized as follows:

- To the best of our knowledge, we are the first to propose the fine-tuning on each scale of feature maps using the attention modules and the conditional normalization function, in order to generate high-quality and semantically consistent images only employing a single generator/discriminator pair.
- We design two new types of attention modules to guide the generator to focus on text-relevant channels and pixels, and to refine the feature maps for each scale.
- CAdaILN is presented to help attention modules flexibly control the amount of change in shape and texture.
- We are the first to introduce the visual loss in text-to-image synthesis to enhance the image quality.

II. RELATED WORK

A. Text-to-Image Generation

In recent years, the task of text-to-image synthesis has attracted rapidly growing attention from both computer vision (CV) and natural language processing (NLP) communities. Thanks to the significant improvements in image generation approaches especially generative adversarial networks (GAN), researchers have achieved inspiring advances in text-to-image generation. The conditional GAN [14] was first presented by Reed et al. [15] to generate plausible images from detailed text descriptions. The problem of text-to-image generation was decomposed by Zhang et al. [4], [16] into multiple stages. Each stage accomplished the corresponding task by using different generators and discriminators. We aim to generate high-quality images with photo-realistic details just employing a single generator/discriminator pair. Tao et al. [8] presented a Matching-Aware zero-centered Gradient Penalty (MA-GP) loss and a one-stage framework to overcome the problem of multiple generators and discriminators, but it just utilized a fully-connected layer to connect feature maps and sentence vector, lacking an efficient mechanism to fuse image features and sentence vectors. Qiao et al. [1] introduced an image captioning model to regenerate the text description from a generated image, in order to improve the semantic relevancy between the natural-language description and visual content. Zhu et al. [2] applied a dynamic memory module to refine the image quality of the initial stage.

B. Attention

Attention mechanisms play a vital role in bridging the semantic gap between vision and language. They have been extensively explored in the interdisciplinary fields, such as image captioning [17], visual question answering [18] and visual dialog. Over the past few years, there have been some attention methods in text-to-image generation. Xu et al. [3] utilized a spatial attention mechanism to obtain the relationship between the subregions of a generated image and the words in a given natural-language description. The most relevant subregions to the words were very focused. Li et al. [5] designed a channel-wise attention mechanism on the basis of Xu et al. [3], simultaneously taking spatial and channel information into account. However, the aforementioned attention works adopt the weighted sum of converted word features as a new feature map which is largely different from the original feature map. We propose to perform global average pooling and global max pooling on feature maps to extract significant features and fine-tune the original feature map using channel-aware and pixel-aware attention weights. The experiments conducted on CUB bird and MS COCO datasets show the superiority of our proposed attention modules compared to the aforementioned methods.

III. DTGAN FOR TEXT-TO-IMAGE GENERATION

In this section, we elaborate on our proposed DTGAN which is shown in Fig. 2. Unlike prior works [1]–[7], [16], our goal is to generate a high-quality and visually realistic image which semantically aligns with a given natural-language description only employing a single generator/discriminator pair. To this end, we present four significant components: a channel-aware attention module, a pixel-aware attention module, Conditional Adaptive Instance-Layer Normalization (CAdaILN) and a new type of visual loss. Each of them will be discussed in detail after briefly describing the overall framework of our model.

As shown in Fig. 2, our architecture is composed of a text encoder and a generator/discriminator pair. For text encoder, we adopt a bidirectional Long Short-Term Memory (LSTM) network [19] to learn the semantic representation of a given text description. Specifically, in the bidirectional LSTM layer, two hidden states are employed to capture the semantic meaning of a word and the last hidden states are utilized to represent the sentence features. The generator network
A small colorful bird with black and yellow secondaries, white wing bars, and a yellow throat.

Mathematically,

\[ x_a = \text{GAP}(h) \]  
\[ x_m = \text{GMP}(h) \]

where GAP denotes global average pooling, GMP is global max pooling.

Then, we adopt a query, key and value setting to capture the semantic relevancy between channels and the input text, where \( x_a \) and \( x_m \) are used as the query and \( s \) is selected as the key and the value. It is defined as:

\[ q_{ac} = W_{qa}x_a, q_{mc} = W_{qm}x_m \]
\[ k_c = W_{kc}s, v_c = W_{vc}s \]

where \( W_{qa}, W_{qm}, W_{kc} \) and \( W_{vc} \) are the projection matrices.

Assuming that the dot products [21] between the sentence-level key \( k_c^T \in R^{1 \times D} \) and the average-pooling query \( q_{ac} \in R^{C \times 1} \), the max-pooling query \( q_{mc} \in R^{C \times 1} \) can capture meaningful features, the attention scores of channel maps are achieved through the following attention mechanism:

\[ \hat{\alpha}_a^c = q_{ac} \cdot k_c^T, \hat{\alpha}_m^c = q_{mc} \cdot k_c^T \]
\[ \alpha_a^c = \text{softmax}(\hat{\alpha}_a^c \cdot v_c) \]
\[ \alpha_m^c = \text{softmax}(\hat{\alpha}_m^c \cdot v_c) \]

where \( \hat{\alpha}_a^c \in R^{C \times D} \) and \( \hat{\alpha}_m^c \in R^{C \times D} \) represent the semantic similarity between channel maps and the global sentence vector, \( \alpha_a^c \in R^{C \times 1} \) and \( \alpha_m^c \in R^{C \times 1} \) denote the final attention weights of channels for global average pooling and global max pooling, respectively, \( \hat{\alpha}_a^c, \hat{\alpha}_m^c, \alpha_a^c \) and \( \alpha_m^c \) are all computed by dot products.

---

**A. Channel-aware Attention Module**

The process of the channel-aware attention module is shown in Fig. 3. The channel-aware attention module takes two inputs: the feature map \( h \) and the global sentence vector \( s \in R^D \). Firstly, we perform global average pooling and global max pooling on \( h \) to extract the channel features:

\[ x_a \in R^{C \times 1 \times 1} \text{ and } x_m \in R^{C \times 1 \times 1}. \]

Global average pooling is to obtain the information of the whole feature map, while global max pooling focuses on the most discriminative part.
After acquiring the attention weights of channels, we multiply them and the original feature maps to update the feature maps. It is denoted as:

\[ o_{ac} = \alpha_c^a \odot h \]
\[ o_{mc} = \alpha_m^c \odot h \]

(8)

(9)

where \( \odot \) is the element-wise multiplication. By doing so, the network will focus on the channels which are more semantically related to the given text description.

Meanwhile, the results of global average pooling and global max pooling are fused through concatenation. Specifically,

\[ o_c = \sigma(W_c[o_{ac}; o_{mc}]) \]

(10)

where \( W_c \) is implemented as \( 1 \times 1 \) convolution, \( \sigma \) is a nonlinear function, such as ReLU.

We further apply an adaptive residual connection [22] to generate the final result. It is defined as follows:

\[ y_c = \gamma_c \ast o_c + h \]

(11)

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

As can be seen from above, our designed channel-aware attention model is a fine-tuning module based on channel information and text features. Moreover, it is applied on each scale of feature maps to improve the semantic consistency of generated samples at the generative stage.

B. Pixel-aware Attention Module

The framework of the pixel-aware attention module is illustrated in Fig. 4. Given the feature map \( h \) and the global sentence vector \( s \), we first exploit average pooling and max pooling to process \( h \). Specifically,

\[ e_a = \text{SAP}(h) \]
\[ e_m = \text{SMP}(h) \]

(12)

(13)

where SAP and SMP represent average pooling and max pooling in the spatial dimension, respectively. \( e_a \in R^{1 \times H \times W} \) and \( e_m \in R^{1 \times H \times W} \) are the new feature maps.

Then, \( s \) is adopted as the key and the value:

\[ k_p = W_{kp}s, v_p = W_{vp}s \]

(14)

where \( W_{kp} \) and \( W_{vp} \) are the learnable matrices.

After that, we compute the dot products of the new feature maps and the key to get the semantic similarity \( \hat{\alpha}_a^p \in R^{(H \times W) \times D} \) and \( \hat{\alpha}_m^p \in R^{(H \times W) \times D} \) between spatial pixels and the global sentence vector. Furthermore, the attention weights are calculated through a softmax function on the dot products of the semantic similarity and the value. It is defined as:

\[ \hat{\alpha}_a^p = e_a \cdot k_p^T, \hat{\alpha}_m^p = e_m \cdot k_p^T \]

(15)

\[ \alpha_a^p = \text{softmax}(\hat{\alpha}_a^p \cdot v_p) \]

(16)

\[ \alpha_m^p = \text{softmax}(\hat{\alpha}_m^p \cdot v_p) \]

(17)

where \( \alpha_a^p \in R^{H \times W \times 1} \) and \( \alpha_m^p \in R^{H \times W \times 1} \) represent the final attention weights of spatial pixels for average pooling and max pooling, respectively.

Next, same as the channel-aware attention module, we perform a matrix multiplication between the attention weights and the original feature maps to derive the new features \( o_{ap} \) and \( o_{mp} \):

\[ o_{ap} = \alpha_a^p \odot h \]
\[ o_{mp} = \alpha_m^p \odot h \]

(18)

(19)

In addition, we concatenate \( o_{ap} \) and \( o_{mp} \), and apply a nonlinear function \( \sigma \) to compute the result \( o_p \). Finally, an adaptive residual connection [22] is utilized to integrate \( h \) and \( o_p \). This process is denoted as:

\[ o_p = \sigma(W_p[o_{ap}; o_{mp}]) \]
\[ y_p = \gamma_p \ast o_p + h \]

(20)

(21)

where \( W_p \) is implemented as \( 1 \times 1 \) convolution, \( \sigma \) is a nonlinear function, such as ReLU, \( \gamma_p \) is a learnable parameter which is initialized as 0.

C. Conditional Adaptive Instance-Layer Normalization

In order to stabilize the training of GAN [23], most existing text-to-image generation models [1]–[3], [5], [24] employ Batch Normalization (BN) [25] which applies the normalization to a whole batch of generated images instead for single ones. However, the convergence of BN heavily depends on the size of a batch [26]. Furthermore, the advantage of BN is not obvious for text-to-image generation since each generated image is more pertinent to the given text description and the global sentence vector \( s \), \( W_1 \) and \( W_2 \) are fully-connected layers, \( \hat{a} \) is the output of CAdaLN. The
ratio of IN and LN is dependent on a learnable parameter $\rho$, whose value is constrained to the range of [0, 1]. Moreover, $\rho$ is updated together with generator parameters.

D. Visual Loss

To ensure that generated images and real images have similar color distributions and shape, we propose a new type of visual loss for the generator which is illustrated in Fig. 2. The visual loss plays a vital role in improving the quality and resolution of finally-generated images. It is based on the image features of the real image $I$ and the generated sample $\hat{I}$, and defined as:

$$L_{vis} = \left| f(I) - f(\hat{I}) \right|_1$$

(24)

where $f(I)$ and $f(\hat{I})$ denote the image features of the real image and the fake image. They are achieved by the feature extractor which consists of 5 downsampling residual blocks, 1 residual block and 1 convolutional layer. We impose a L1 loss to minimize the distance between these two image features. To the best of our knowledge, we are the first to present this type of visual loss and apply it in the task of text-to-image generation.

IV. EXPERIMENTS

In this section, we carry out a set of experiments on the CUB bird [12] and MS COCO [13] datasets, in order to quantitatively and qualitatively evaluate the effectiveness of the proposed DTGAN. The previous state-of-the-art GAN models in text-to-image synthesis are first compared with our approach. Then, we analyze the significant components of our designed architecture.

A. Experimental Setup and Benchmarks

Datasets. Two popular datasets in text-to-image generation, CUB bird and MS COCO datasets, are employed to test our method. The CUB dataset encompasses 11,788 images which are split into 82,783 training images and 40,504 validation images. The MS COCO dataset contains 123,287 images which are split into 82,783 training images and 40,504 validation images. Each image in the CUB dataset and MS COCO dataset has ten corresponding text descriptions and five corresponding text descriptions, respectively. We preprocess the CUB dataset using the method in StackGAN [16].

Training. Apart from the visual loss, we follow the losses in [8] due to its excellent performance. For text encoder, the dimension $D$ is set to 256 and the length of words is set to 18. We implement our model using PyTorch [27]. In the experiments, the network is trained using Adam optimizer [28] with $\beta_1 = 0.0$ and $\beta_2 = 0.9$. We follow the two timescale update rule (TTUR) [29] and set the learning rate of the generator and the discriminator to 0.0001 and 0.0004. The dimension of the noise vector is set to 100. The batch size is set to 24 on the CUB dataset and to 16 on the COCO dataset.

Evaluation Metric. Inception score (IS) [30] and Fréchet inception distance (FID) [31] score are extensively employed in the assessment of text-to-image generation. Generated samples are meant to be diverse and meaningful if the IS is large.

A lower FID means that synthesized images are closer to the corresponding real images. We adopt these two indexes as the quantitative evaluation measure and generate 30,000 images conditioned on unseen text descriptions for each evaluation metric. Meanwhile, we perform a human test to evaluate the quality and semantic consistency of generated images.

However, it is important to note that the IS on the COCO dataset fails to evaluate the image quality and can be saturated, even over-fitted, which is observed by [32] and [8]. Therefore, we do not utilize the IS as the evaluation metric on the COCO dataset.

B. Comparison with State of the Art

Quantitative Results. We compare our model with prior state-of-the-art GAN approaches in text-to-image synthesis on the CUB and MS COCO datasets. Table I reports the IS of our proposed DTGAN and other compared methods on the CUB dataset. We can observe that our model has the best score, significantly improving the IS from 4.75 to 4.88 on the CUB dataset. The experimental results demonstrate that our DTGAN can generate visually realistic images with higher quality and better diversity than state-of-the-art models.

The comparison between our method, StackGAN++ [4] and AttnGAN [3] with respect to FID on the CUB and COCO datasets is shown in Table II. We can see that our DTGAN achieves a remarkably lower FID than compared approaches on both datasets, which indicates that our generated data distribution is closer to the real data distribution. Specifically, we impressively reduce the FID from 35.49 to 23.61 on the challenging COCO dataset and from 23.98 to 16.35 on the CUB dataset.

Qualitative Results. In addition to quantitative experiments, we perform qualitative comparison with StackGAN++ [4] and AttnGAN [3] on both datasets, which is illustrated in Fig. 5. It can be observed that the details of birds generated by StackGAN++ and AttnGAN are lost ($2^{th}$, $3^{th}$ and $4^{th}$ column), the shape is strange ($1^{th}$, $2^{th}$ and $3^{th}$ column) and
the colors are even wrong (3rd column). Furthermore, the samples synthesized by these two approaches lack text-relevant objects (5th, 6th and 7th column), the backgrounds are unclear and inconsistent with the given text descriptions (5th and 7th column), and the colors are rough (8th column) on the challenging COCO dataset. However, our DTGAN generates more clear and visually plausible images than StackGAN++ and AttnGAN, verifying the superiority of our DTGAN. For instance, as shown in the 1st column, owing to the successful application of the visual loss, a long-wingspan bird with vivid shape is produced by the DTGAN, whereas it is too hard for StackGAN++ and AttnGAN to generate this kind of bird. In the meantime, the birds generated by the DTGAN have more details and richer color distributions compared to StackGAN++ and AttnGAN in the 2nd, 3rd and 4th column, since the DTGAN armed with channel-aware and pixel-aware attention modules is able to generate high-resolution images which semantically align with given text descriptions. More importantly, our method also yields high-quality and visually realistic results on the challenging COCO dataset. For example, the number of the skiers and surfers is correct, the backgrounds are reasonable and people in the images are clear in the 5th and 6th column. Moreover, the beach and the sea are very beautiful in the 7th column and the pizza looks delicious in the 8th column. Generally, these qualitative results confirm the effectiveness of DTGAN.

**Human Evaluation.** In order to evaluate the image quality and semantic consistency of StackGAN++ [4], AttnGAN [3] and DTGAN, we also conduct a human test on the CUB and COCO datasets. We randomly select 100 images from both datasets, respectively. Users are asked to choose the best image from three given images according to the image quality and text description. As shown in Table III, our method outperforms AttnGAN by 61.3% on the CUB dataset and 71.4% on the COCO dataset, confirming the effectiveness of DTGAN.

**C. Component Analysis**

In this section, we perform an extensive ablation study on the CUB dataset, so as to evaluate the contributions from different components of our DTGAN. The novel components in our model include a channel-aware attention module (CAM), a pixel-aware attention module (PAM), CAdaILN and a new type of visual loss (VL). We first quantitatively explore the effectiveness of each component by removing the corresponding part in the DTGAN step by step, i.e., 1) DTGAN, 2) DTGAN without the VL, 3) DTGAN without CAdaILN, 4) DTGAN without the PAM, 5) DTGAN without the CAM, 6) DTGAN without the CAM and PAM. All the results are reported in Table IV.

**TABLE III**

<table>
<thead>
<tr>
<th>Datasets</th>
<th>StackGAN++</th>
<th>AttnGAN</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUB</td>
<td>8.1%</td>
<td>15.3%</td>
<td>76.6%</td>
</tr>
<tr>
<td>COCO</td>
<td>6.2%</td>
<td>11.2%</td>
<td>82.6%</td>
</tr>
</tbody>
</table>

**TABLE IV**

<table>
<thead>
<tr>
<th>ID</th>
<th>Components</th>
<th>IS ↑</th>
<th>FID ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>4.88 ± 0.03</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>4.72 ± 0.04</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>4.11 ± 0.04</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>4.71 ± 0.05</td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td>4.60 ± 0.07</td>
</tr>
<tr>
<td>6</td>
<td>✓</td>
<td>✓</td>
<td>4.54 ± 0.04</td>
</tr>
</tbody>
</table>
By comparing Model 1 (DTGAN) with Model 2 (removing the VL), the VL significantly improves the IS from 4.72 to 4.88 and reduces the FID by 2.88 on the CUB dataset, which demonstrates the importance of adopting VL in the DTGAN. By exploiting CAdaILN in our DTGAN, Model 1 performs better than Model 3 (removing CAdaILN) on the IS and FID by 0.77 and 8.89, confirming the effectiveness of the proposed CAdaILN. Both Model 4 (removing the CAM) and Model 5 (removing the CAM and PAM), indicating that these two new types of attention modules can help the generator produce more realistic images. Furthermore, Model 1 achieves better results than both Model 4 and Model 5, which shows the advantage of combining the CAM and PAM.

**Attention Modules.** To better understand what has been learned by the CAM and PAM during training, we visualize the channel-aware and pixel-aware attention maps for different images in Fig. 6. We can see that in the 2th row, the eyes, beaks, legs and wings of birds are highlighted by the channel-aware attention maps. Meanwhile, in the 3th row, the pixel-aware attention maps highlight most important areas of images, including the branches and the whole bodies of birds. This suggests that the CAM helps the generator focus on the crucial parts of birds, while the PAM guides the generator to refine the globally visual contents. Then, the generator can fine-tune the discriminative regions of images obtained by our attention modules. Note that some texture background regions that are related to the contexts might be focused.

**Visual Loss.** In addition, we conduct an ablation study to validate the effectiveness of the VL. The visual comparison between the DTGAN and our model without the VL is shown in Fig. 7. We can see that, in the first two columns, the DTGAN without the VL fails to generate long-wingspan birds with reasonable shape and vivid wings. In the meantime, the proposed model without the VL synthesizes the blue birds which have rough color distributions and lack colorful details in the last two columns. However, the DTGAN produces realistic long-wingspan birds which have semantically consistent shape and colors, while also yielding blue birds with more vivid details and richer color distributions. This indicates that the VL has the ability to potentially ensure the quality of the generated image, including the shape and color distributions of objects in an image.

**CAdaILN.** To further verify the benefits of CAdaILN, we conduct an ablation study on normalization functions. We first design a baseline model by removing CAdaILN from the DTGAN. Then, we compare the variants of normalization layers. Note that BN conditioned on the global sentence vector (BN-sent) and BN conditioned on the word vectors (BN-word) are based on semantic-conditioned Batch Normalization in SDGAN [24] and the CAdaILN function with the word vectors (CAdaILN-word) is achieved through the word-level normalization method in SDGAN. The results of the ablation study are shown in Table V. It can be observed that by comparing Model 2 with Model 4 and Model 3 with Model 5, CAdaILN significantly outperforms the BN layer whether using the sentence-level cues or the word-level cues. Moreover, by comparing Model 4 with Model 5, CAdaILN with the global sentence vector performs better than CAdaILN-word by improving the IS from 4.71 to 4.88 and reducing the FID from 19.08 to 16.35 on the CUB dataset, since sentence-level features are easier to be trained in our generator network than word-level features. The above analysis demonstrates the effectiveness of our designed CAdaILN.

**V. CONCLUSION**

In this paper, we propose the Dual Attention Generative Adversarial Network (DTGAN), a novel framework for text-
to-image generation, to generate high-quality realistic images which semantically align with given text descriptions, only employing a single generator/discriminator pair. DTGAN exploits two new types of attention modules: a channel-aware attention module and a pixel-aware attention module, to guide the generator to focus more on the text-relevant channels and pixels. In addition, to flexibly control the amount of change in shape and texture, Conditional Adaptive Instance-Layer Normalization (CAdaILN) is adopted as a complement to the attention modules. To further enhance the quality of generated images, we design a new type of visual loss which computes the L1 loss between the features of generated images and real images. DTGAN surpasses state-of-the-art results on both CUB and COCO datasets, which confirms the superiority of our proposed method. Our future work might be directed at zero-shot text-to-image synthesis.

ACKNOWLEDGMENT

We would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster.

REFERENCES


