DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-image Synthesis

Minfeng Zhu[1,3*] Pingbo Pan[3] Wei Chen[1] Yi Yang[2,3†]
[1] State Key Lab of CAD&CG, Zhejiang University [2] Baidu Research
[3] Centre for Artificial Intelligence, University of Technology Sydney
{minfeng_zhu@, chenwei@cad}zju.edu.cn {pingbo.pan@student,Yi.Yang@}uts.edu.au

Abstract

In this paper, we focus on generating realistic images from text descriptions. Current methods first generate an initial image with rough shape and color, and then refine the initial image to a high-resolution one. Most existing text-to-image synthesis methods have two main problems. (1) These methods depend heavily on the quality of the initial images. If the initial image is not well initialized, the following processes can hardly refine the image to a satisfactory quality. (2) Each word contributes a different level of importance when depicting different image content, however, unchanged text representation is used in existing image refinement processes. In this paper, we propose the Dynamic Memory Generative Adversarial Network (DM-GAN) to generate high-quality images. The proposed method introduces a dynamic memory module to refine fuzzy image contents, when the initial images are not well generated. A memory writing gate is designed to select the important text information based on the initial image content, which enables our method to accurately generate images from the text description. We also utilize a response gate to adaptively fuse the information read from the memories and the image features. We evaluate the DM-GAN model on the Caltech-UCSD Birds 200 dataset and the Microsoft Common Objects in Context dataset. Experimental results demonstrate that our DM-GAN model performs favorably against the state-of-the-art approaches.

1. Introduction

The last few years have seen remarkable growth in the use of Generative Adversarial Networks (GANs) [4] for image and video generation. Recently, GANs have been widely used to generate photo-realistic images according to certain text descriptions (see Figure 1). Fully understanding the relationship between visual contents and natural languages is an essential step towards artificial intelligence, *e.g.*, image search and video understanding [33]. Multi-stage methods [28, 30, 31] first generate low-resolution initial images and then refine the initial images to high-resolution ones.

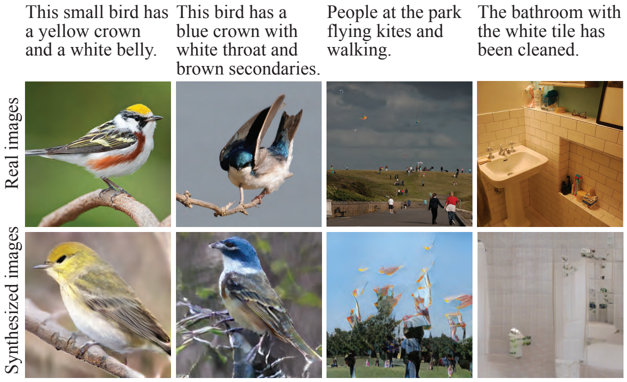

This small bird has a yellow crown and a white belly.
This bird has a blue crown with white throat and brown secondaries.
People at the park flying kites and walking.
The bathroom with the white tile has been cleaned.
Real images
Synthesized images

Figure 1. Examples of text-to-image synthesis by our DM-GAN.

Although these multi-stage methods achieve remarkable progress, there remain two problems. First, the generation result depends heavily on the quality of the initial image. The image refinement process cannot generate high-quality images, if the initial images are badly generated. Second, each word in an input sentence has a different level of information depicting the image content. Current models utilize the same word representations in different image refinement processes, which makes the refinement process ineffective. The image information should be taken into account to determine the importance of every word for refinement.

In this paper, we introduce a novel Dynamic Memory Generative Adversarial Network (DM-GAN) to address the aforementioned issues. For the first issue, we propose to add a memory mechanism to cope with badly-generated initial images. Recent work [27] has shown the memory net-

[*] This work was done when Minfeng Zhu was visiting the University of Technology Sydney.

[†] Part of this work was done when Yi Yang was visiting Baidu Research during his Professional Experience Program.

work's ability to encode knowledge sources. Inspired by this work, we propose to add the key-value memory structure [13] to the GAN framework. The fuzzy image features of initial images are treated as queries to read features from the memory module. The reads of the memory are used to refine the initial images. To solve the second issue, we introduce a memory writing gate to dynamically select the words that are relevant to the generated image. This makes our generated image well conditioned on the text description. Therefore, the memory component is written and read dynamically at each image refinement process according to the initial image and text information. In addition, instead of directly concatenating image and memory, a response gate is used to adaptively receive information from image and memory.

We conducted experiments to evaluate the DM-GAN model on the Caltech-UCSD Birds 200 (CUB) dataset and the Microsoft Common Objects in Context (COCO) dataset. The quality of generated images is measured using the Inception Score (IS), the Fréchet Inception Distance (FID) and the R-precision. The experiments illustrate that our DM-GAN model outperforms the previous image-to-text synthesis methods, quantitatively and qualitatively. Our model improves the IS from 4.36 to 4.75 and decreases the FID from 23.98 to 16.09 on the CUB dataset. The R-precision is improved by 4.49% and 3.09% on the above two datasets. The qualitative evaluation proves that our model generates more photo-realistic images.

This paper makes the following key contributions:

- We propose a novel GAN model combined with a dynamic memory component to generate high-quality images even if the initial image is not well generated.
- We introduce a memory writing gate that is capable of selecting relevant word according to the initial image.
- A response gate is proposed to adaptively fuse information from image and memory.
- The experimental results demonstrate that the DM-GAN outperforms the state-of-the-art approaches.

2. Related Work

2.1. Generative Adversarial Networks.

With the recent successes of Variational Autoencoders (VAEs) [9] and GANs [4], a large number of methods have been proposed to handle generation [14, 17, 28, 1] and domain adaptation task [25, 32]. Recently, generating images based on the text descriptions gains interest in the research community nowadays.

Single-stage. The text-to-image synthesis problem is decomposed by Reed *et al.* [20] into two sub-problems: first, the joint embedding is learned to capture the relations between natural language and real-world images; second, a deep convolutional generative adversarial network [19] is trained to synthesize a compelling image. Dong *et al.* [3] adopted the pair-wise ranking loss [10] to project both images and natural languages into a joint embedding space. Since previous generative models failed to add the location information, Reed *et al.* proposed GAWWN [21] to encode localization constraints. To diversify the generated images, the discriminator of TAC-GAN [2] not only distinguishes real images from synthetic images, but also classifies synthetic images into true classes. Similar to TAC-GAN, PPGN [16] includes a conditional network to synthesize images conditioned on a caption.

Multi-stage. StackGAN [30] and StackGAN++ [31] generate photo-realistic high-resolution images with two stages. Yuan *et al.* [29] employed symmetrical distillation networks to minimize the multi-level difference between real and synthetic images. DA-GAN [12] translates each word into a sub-region of an image. Our method considers the interaction between each word and the whole generated image. Conditioning on the global sentence vector may result in low-quality images, AttnGAN [28] refines the images to high-resolution ones by leveraging the attention mechanism. Each word in an input sentence has a different level of information depicting the image content. However, AttnGAN takes all the words equally, it employs an attention module to use the same word representation. Our proposed memory module is able to uncover such difference for image generation, as it dynamically selects the important word information based on the initial image content.

2.2. Memory Networks.

Recently, memory network [5, 27] provides a new architecture to reason answers from memories more effectively using explicit storage and a notion of attention. Memory network first writes information into an external memory and then reads contents from memory slots according to a relevance probability. Weston *et al.* [27] introduced the memory network to produce the output by searching supporting memories one by one. End-to-end memory network [23] is a continues form of memory network, where each memory slot is weighted according to the inner product between the memory and the query. To understand the unstructured documents, the Key-Value Memory Network (KV-MemNN) [13] performs reasoning by utilizing different encodings for key memory and value memory. The key memory is used to infer the weight of the corresponding value memory when predicting the final answer. Inspired by the recent success of the memory network, we introduce DM-GAN, a novel network architecture to generate high-quality images via nontrivial transforms between key and value memories.

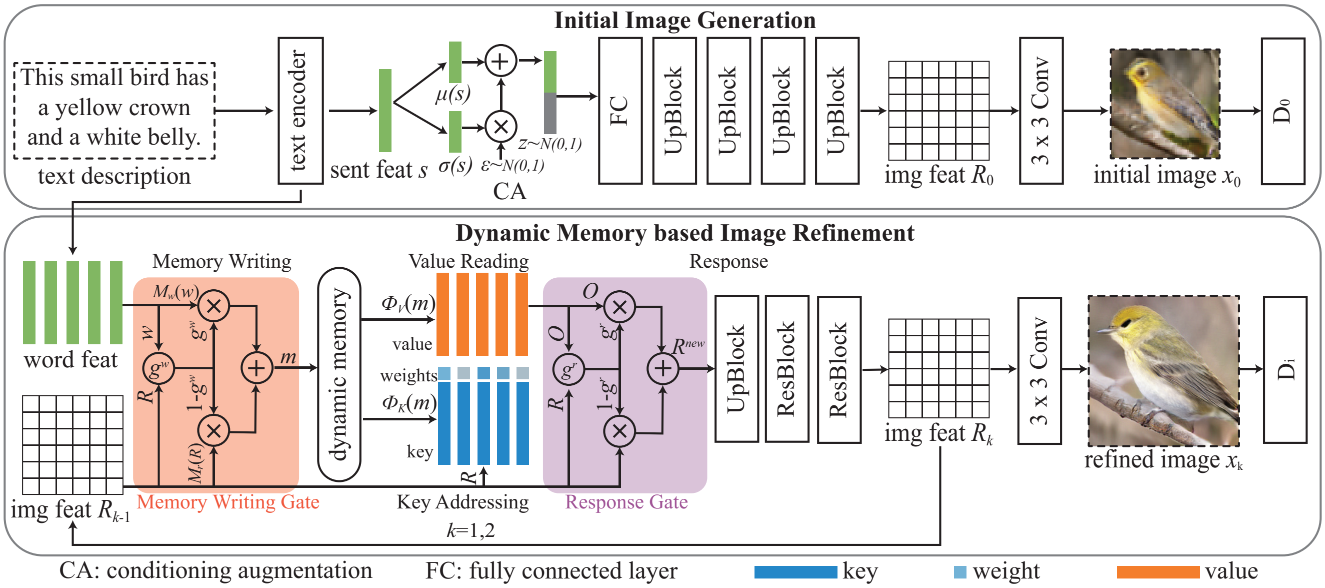


Figure 2. The DM-GAN architecture for text-to-image synthesis. Our DM-GAN model first generates an initial image, and then refines the initial image to generate a high-quality one.

3. DM-GAN

As shown in Figure 2, the architecture of our DM-GAN model is composed of two stages: *initial image generation* and *dynamic memory based image refinement*.

At the *initial image generation* stage, firstly, the input text description is transformed into some internal representation (a sentence feature s and several word features W) by a text encoder. Then, a deep conventional generator predicts an initial image x_0 with a rough shape and few details according to the sentence feature and a random noise vector z: $x_0, R_0 = G_0(z, s)$, where R_0 is the image feature. The noise vector is sampled from a normal distribution.

At the *dynamic memory based image refinement* stage, more fine-grained visual contents are added to the fuzzy initial images to generate a photo-realistic image x_i: $x_i = G_i(R_{i-1}, W)$, where R_{i-1} is the image feature from the last stage. The refinement stage can be repeated multiple times to retrieve more pertinent information and generate a high-resolution image with more fine-grained details.

The dynamic memory based image refinement stage consists of four components: *Memory Writing*, *Key Addressing*, *Value Reading*, and *Response* (Section 3.1). The *Memory Writing* operation stores the text information into a key-value structured memory for further retrieval. Then, *Key Addressing* and *Value Reading* operations are employed to read features from the memory module to refine the visual features of the low-quality images. At last, the *Response* operation is adopted to control the fusion of the image features and the reads of the memory. We propose a memory writing gate to highlight important word information according to the image content in memory writing step (Section 3.2). We also utilize a response gate to adaptively fuse the information read from the memory and the image features (Section 3.3).

3.1. Dynamic Memory

We start with the given input word representations W, image x and image features R_i:

$$W = \{w_1, w_2, ..., w_T\}, w_i \in \mathbb{R}^{N_w}, \quad (1)$$

$$R_i = \{r_1, r_2, ..., r_N\}, r_i \in \mathbb{R}^{N_r}, \quad (2)$$

where T is the number of words, N_w is the dimension of word features, N is the number of image pixels and image pixel feature is a N_r dimensional vector. We are intended to learn a model to refine the image using a more effective way to fuse text and image information via nontrivial transforms between key and value memory. The refinement stage includes the following four steps.

Memory Writing: Encoding prior knowledge is an important part of the dynamic memory, which enables recovering high-quality images from text. A naive way to write the memory is considering only partial text information.

$$m_i = M(w_i), m_i \in \mathbb{R}^{N_m} \quad (3)$$

where $M(\cdot)$ denotes the 1×1 convolution operation which embeds word features into the memory feature space with N_m dimensions.

Key Addressing: In this step, we retrieve relevant memories using key memory. We compute a weight of each memory slot as a similarity probability between a memory slot m_i and an image feature r_j:

$$\alpha_{i,j} = \frac{exp(\phi_K(m_i)^T r_j)}{\sum\limits_{l=1}^{T} exp(\phi_K(m_l)^T r_j)}, \tag{4}$$

where $\alpha_{i,j}$ is the similarity probability between the i-th memory and the j-th image feature and $\phi_K()$ is the key memory access process which maps memory features into dimension N_r. $\phi_K()$ is implemented as a 1×1 convolution.

Value Reading: The output memory representation is defined as the weighted summation of value memories according to the similarity probability:

$$o_j = \sum_{i=1}^{T} \alpha_{i,j}\phi_V(m_i), \tag{5}$$

where $\phi_V()$ is the value memory access process which maps memory features into dimension N_r. $\phi_V()$ is implemented as a 1×1 convolution.

Response: After receiving the output memory, we combine the current image and the output representation to provide a new image feature. A naive approach will be simply concatenating the image features and the output representation. The new image features are obtained by:

$$r_i^{new} = [o_i, r_i], \tag{6}$$

where $[\cdot,\cdot]$ denotes concatenation operation. Then, we are able to utilize an upsampling block and several residual blocks [6] to upscale the new image features into a high-resolution image. The upsampling block consists of a nearest neighbor upsampling layer and a 3×3 convolution. Finally, the refined image x is obtained from the new image features using a 3×3 convolution.

3.2. Gated Memory Writing

Instead of considering only partial text information using Eq.3, the memory writing gate allows the DM-GAN model to select the relevant word to refine the initial images. The memory writing gate g_i^w combines image features R_i from the last stage with word features W to calculate the importance of a word:

$$g_i^w(R, w_i) = A * w_i + B * \frac{1}{N}\sum_{i=1}^{N} r_i, \tag{7}$$

where A is a $1 \times N_w$ matrix and B is a $1 \times N_r$ matrix. Then, the memory slot $m_i \in R^{N_m}$ is written by combining the image and word features.

$$m_i = M_w(w_i) * g_i^w + M_r(\frac{1}{N}\sum_{i=1}^{N} r_i) * (1 - g_i^w), \tag{8}$$

where $M_w(\cdot)$ and $M_r(\cdot)$ denote the 1x1 convolution operation. $M_w(\cdot)$ and $M_r(\cdot)$ embed image and word features into the same feature space with N_m dimensions.

3.3. Gated Response

We utilize the adaptive gating mechanism to dynamically control the information flow and update image features:

$$\begin{aligned} g_i^r &= \sigma(W[o_i, r_i] + b), \\ r_i^{new} &= o_i * g_i^r + r_i * (1 - g_i^r), \end{aligned} \tag{9}$$

where g_i^r is the response gate for information fusion, σ is the sigmoid function, W and b are the parameter matrix and bias term.

3.4. Objective Function

The objective function of the generator network is defined as:

$$L = \sum_i L_{G_i} + \lambda_1 L_{CA} + \lambda_2 L_{DAMSM}, \tag{10}$$

in which λ_1 and λ_2 are the corresponding weights of conditioning augmentation loss and DAMSM loss. G_0 denotes the generator of the initial generation stage. G_i denotes the generator of the i-th iteration of the image refinement stage.

Adversarial Loss: The adversarial loss for G_i is defined as follows:

$$L_{G_i} = -\frac{1}{2}[\mathbb{E}_{x\sim p_{G_i}} logD_i(x) + \mathbb{E}_{x\sim p_{G_i}} logD_i(x, s)], \tag{11}$$

where the first term is the unconditional loss which makes the generated image real as much as possible and the second term is the conditional loss which makes the image match the input sentence. Alternatively, the adversarial loss for each discriminator D_i is defined as:

$$\begin{aligned} L_{D_i} = & \underbrace{-\frac{1}{2}[\mathbb{E}_{x\sim p_{data}} logD_i(x) + \mathbb{E}_{x\sim p_{G_i}} log(1 - D_i(x))}_{\text{unconditional loss}}, \\ & \underbrace{+\mathbb{E}_{x\sim p_{data}} logD_i(x, s) + \mathbb{E}_{x\sim p_{G_i}} log(1 - D_i(x, s))]}_{\text{conditional loss}}, \end{aligned} \tag{12}$$

where the unconditional loss is designed to distinguish the generated image from real images and the conditional loss determines whether the image and the input sentence match.

Conditioning Augmentation Loss: The Conditioning Augmentation (CA) technique [30] is proposed to augment training data and avoid overfitting by resampling the input sentence vector from an independent Gaussian distribution. Thus, the CA loss is defined as the Kullback-Leibler divergence between the standard Gaussian distribution and the Gaussian distribution of training data.

$$L_{CA} = D_{KL}(\mathcal{N}(\mu(s), \Sigma(s))||\mathcal{N}(0, I)), \tag{13}$$

where $\mu(s)$ and $\Sigma(s)$ are mean and diagonal covariance matrix of the sentence feature. $\mu(s)$ and $\Sigma(s)$ are computed by fully connected layers.

DAMSM Loss: We utilize the DAMSM loss [28] to measure the matching degree between images and text descriptions. The DAMSM loss makes generated images better conditioned on text descriptions.

3.5. Implementation Details

For text embedding, we employ a pre-trained bidirectional LSTM text encoder by Xu *et al.* [28] and fix their parameters during training. Each word feature corresponds to the hidden states of two directions. The sentence feature is generated by concatenating the last hidden states of two directions. The initial image generation stage first synthesizes images with 64x64 resolution. Then, the dynamic memory based image refinement stage refines images to 128x128 and 256x256 resolution. We only repeat the refinement process with dynamic memory module two times due to GPU memory limitation. Introducing dynamic memory to low-resolution images (*i.e.* 16x16, 32x32) can not further improve the performance. Because low-resolution images are not well generated and their features are more like random vectors. For all discriminator networks, we apply spectral normalization [15] after every convolution to avoid unusual gradients to improve text-to-image synthesis performance. By default, we set $N_w = 256$, $N_r = 64$ and $N_m = 128$ to be the dimension of text, image and memory feature vectors respectively. We set the hyperparameter $\lambda_1 = 1$ and $\lambda_2 = 5$ for the CUB dataset and $\lambda_1 = 1$ and $\lambda_2 = 50$ for the COCO dataset. All networks are trained using ADAM optimizer [8] with batch size 10, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is set to be 0.0002. We train the DM-GAN model with 600 epochs on the CUB dataset and 120 epochs on the COCO dataset.

4. Experiments

In this section, we evaluate the DM-GAN model quantitatively and qualitatively. We implemented the DM-GAN model using the open-source Python library PyTorch [18].

Datasets. To demonstrate the capability of our proposed method for text-to-image synthesis, we conducted experiments on the CUB [26] and the COCO [11] datasets. The CUB dataset contains 200 bird categories with 11,788 images, where 150 categories with 8,855 images are employed for training while the remaining 50 categories with 2,933 images for testing. There are ten captions for each image in CUB dataset. The COCO dataset includes a training set with 80k images and a test set with 40k images. Each image in the COCO dataset has five text descriptions.

Evaluation Metric. We quantify the performance of the DM-GAN in terms of Inception Score (IS), Fréchet Inception Distance (FID), and R-precision. Each model generated 30,000 images conditioning on the text descriptions from the unseen test set for evaluation.

The IS [22] uses a pre-trained Inception v3 network [24] to compute the KL-divergence between the conditional class distribution and the marginal class distribution. A large IS means that the generated model outputs a high diversity of images for all classes and each image clearly belongs to a specific class.

The FID [7] computes the Fréchet distance between synthetic and real-world images based on the extracted features from a pre-trained Inception v3 network. A lower FID implies a closer distance between generated image distribution and real-world image distribution.

Following Xu *et al.* [28], we use the R-precision to evaluate whether a generated image is well conditioned on the given text description. The R-precision is measured by retrieving relevant text given an image query. We compute the cosine distance between a global image vector and 100 candidate sentence vectors. The candidate text descriptions include R ground truth and 100-R randomly selected mismatching descriptions. For each query, if r results in the top R ranked retrieval descriptions are relevant, then the R-precision is r/R. In practice, we compute the R-precision with R=1. We divide the generated images into ten folds for retrieval and then take the mean and standard deviation of the resulting scores.

4.1. Text-to-image Quality

We compare our DM-GAN model with the state-of-the-art models on the CUB and COCO test datasets. The performance results are reported in Table 1 and 2.

As shown in Table 1, our DM-GAN model achieves 4.75 IS on the CUB dataset, which outperforms other methods by a large margin. Compared with AttnGAN, DM-GAN improves the IS from 4.36 to 4.75 on the CUB dataset (8.94% improvement) and from 25.89 to 30.49 on the COCO dataset (17.77% improvement). The experimental results indicate that our DM-GAN model generates images with higher quality than other approaches.

Table 2 compares the performance between AttnGAN and DM-GAN with respect to the FID on the CUB and COCO datasets. We measure the FID of AttnGAN from the officially pre-trained model. Our DM-GAN decreases the FID from 23.98 to 16.09 on the CUB dataset and from 35.49 to 32.64 on the COCO dataset, which demonstrates that DM-GAN learns a better data distribution.

As shown in Table 2, the DM-GAN improves the R-precision by 4.49% on the CUB dataset and 3.09% on the COCO dataset. Higher R-precision indicates that the generated images by the DM-GAN are better conditioned on the given text description, which further demonstrates the effectiveness of the employed dynamic memory.

In summary, the experimental results indicate that our DM-GAN is superior to the state-of-the-art models.

Dataset	GAN-INT-CLS [20]	GAWWN [21]	StackGAN [30]	PPGN [16]	AttnGAN [28]	DM-GAN
CUB	2.88±0.04	3.62±0.07	3.70±0.04	(-)	4.36±0.03	**4.75±0.07**
COCO	7.88±0.07	(-)	8.45±0.03	9.58±0.21	25.89±0.47	**30.49±0.57**

Table 1. The inception scores (higher is better) of GAN-INT-CLS [20], GAWWN [21], StackGAN [30], PPGN [16], AttnGAN [28] and our DM-GAN on the CUB and COCO datasets. The best results are in bold.

Dataset	Metric	AttnGAN	DM-GAN
CUB	FID↓	23.98	**16.09**
	R-precision↑	67.82±4.43	**72.31±0.91**
COCO	FID↓	35.49	**32.64**
	R-precision↑	85.47±3.69	**88.56±0.28**

Table 2. Performance of FID and R-precision for AttnGAN [28] and our DM-GAN on the CUB and COCO datasets. The FID of AttnGAN is calculated from officially released weights. Lower is better for FID and higher is better for R-precision.

Architecture	IS↑	FID↓	R-Precision↑
baseline	4.51±0.04	23.32	68.60±0.73
+M	4.57±0.05	21.41	70.66±0.69
+M+WG	4.65±0.05	20.83	71.40±0.64
+M+WG+RG	**4.75±0.07**	**16.09**	**72.31±0.91**

Table 3. The performance of different architectures of our DM-GAN on the CUB datasets. M, WG and RG denote dynamic memory, memory writing gate and response gate respectively.

4.2. Visual Quality

For qualitative evaluation, Figure 3 shows text-to-image synthesis examples generated by our DM-GAN and the state-of-the-art models. In general, our DM-GAN approach generates images with more vivid details as well as more clear background in most cases, comparing to the AttnGAN [28], GAN-INT-CLS [20] and StackGAN [30], because it employs a dynamic memory model using varied weighted word information to improve image quality.

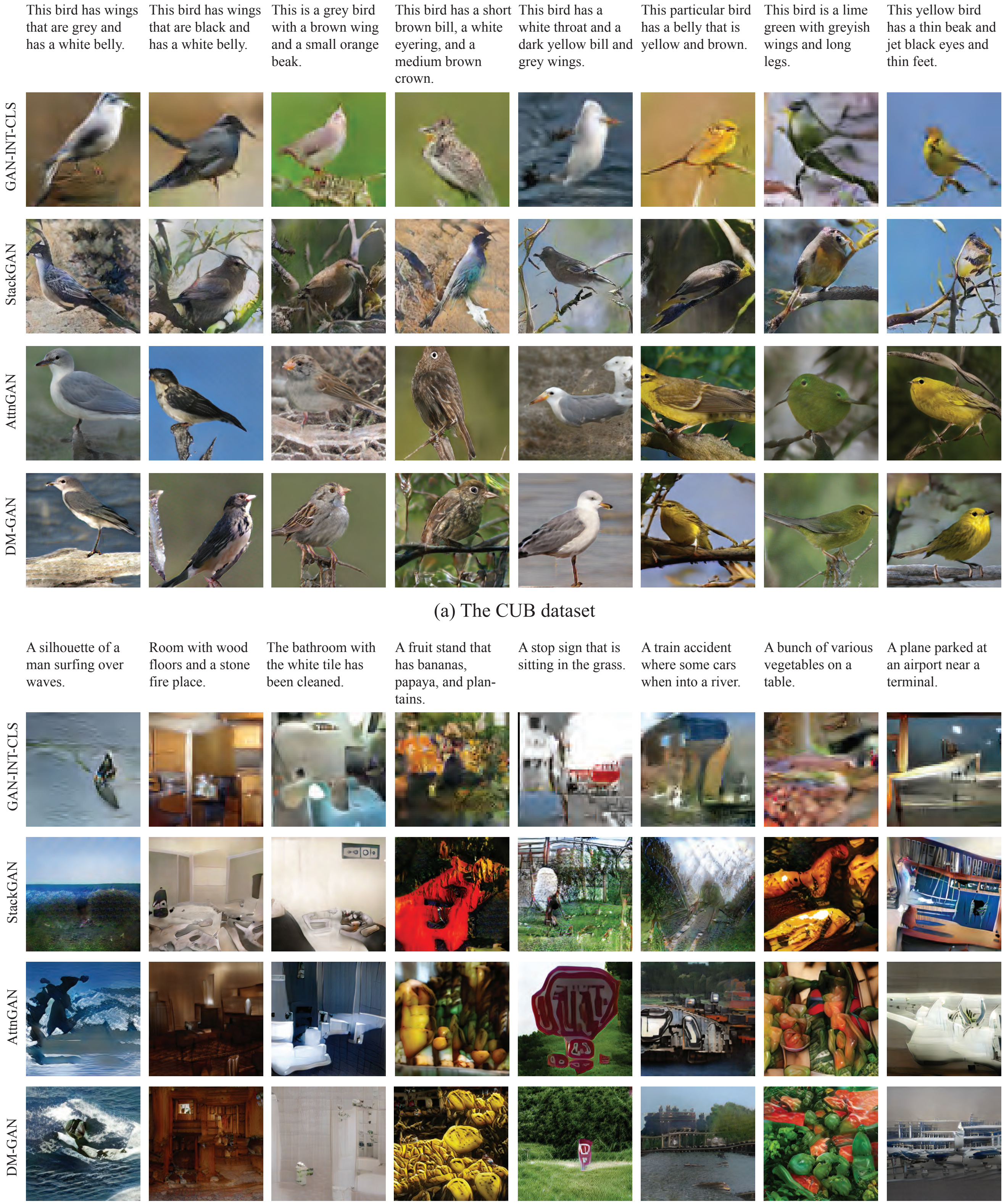
Our DM-GAN method has the capacity to better understand the logic of the text description and present a more clear structure of the images. Observing the samples generated on the CUB dataset in Figure 3(a), with single character, although DM-GAN and AttnGAN both perform well in accurately capture and present the character's feature, our DM-GAN model better highlights the main subject of the image, the bird, differentiating from its background. It demonstrates that, with the dynamic memory module, our DM-GAN model is able to bridge the gap between visual contents and natural languages. In terms of multi-subjects-image generation, for example, the COCO dataset in Figure 3(b), it is more challenging to generate photo-realistic images when the text description is more complicated and contains more than one subject. DM-GAN precisely captures the major scene based on the most important subject and arrange the rest descriptive contents logically, which improves the global structure of the image. For instance, DM-GAN is the only successful method clearly identifies the bathroom with required components in the column 3 in Figure 3(b). The visual results show that our DM-GAN is more effective to capture important subjects using a memory writing gate to dynamically select important words.

Figure 4 indicates that our DM-GAN model is able to refine badly initialized images and generate more photo-realistic high-resolution images. So the image quality is obviously well-improved, with more clear background and convincing details. In most cases, the initial stage generates a blurry image with rough shape and color, so that the background is fine-tuned to be more realistic with fine-grained textures, while the refined image will be better conditioned on the input text and provide more photo-realistic high-resolution images. In the fourth column of Figure 4, no white streaks can be found on the bird body from the initial image with 64×64 resolution. The refinement process helps to encode ”white streaks” information from text description and add back missing features based on the text description and image content. In order word, our DM-GAN model is able to refine the image to match the input text description.

To evaluate the diversity of our DM-GAN model, we generate several images using the same text description, and multiple noise vectors. Figure 5 shows text descriptions and synthetic images with different shapes and backgrounds. Images are similar but not identical to each other, which means our DM-GAN generates images with high diversity.

4.3. Ablation Study

In order to verify the effectiveness of our proposed components, we evaluate the DM-GAN architecture and its variants on the CUB dataset. The control components between architectures include the key-value memory (M), the writing gate (WG) and the response gate (RG). We define a baseline model which removes M, WG and RG from DM-GAN. The memory is written according to partial text information (Eq.3). The response operation simply concatenates the image features and the memory output (Eq.6). The performance of the DM-GAN architecture and its variants is reported in Table 3. Our baseline model produces slightly better performance than AttnGAN. By integrating these components, our model can achieve further improvement which


This bird has wings that are grey and has a white belly.
This bird has wings that are black and has a white belly.
This is a grey bird with a brown wing and a small orange beak.
This bird has a short brown bill, a white eyering, and a medium brown crown.
This bird has a white throat and a dark yellow bill and grey wings.
This particular bird has a belly that is yellow and brown.
This bird is a lime green with greyish wings and long legs.
This yellow bird has a thin beak and jet black eyes and thin feet.
GAN-INT-CLS
StackGAN
AttnGAN
DM-GAN
A silhouette of a man surfing over waves.
Room with wood floors and a stone fire place.
The bathroom with the white tile has been cleaned.
A fruit stand that has bananas, papaya, and plantains.
A stop sign that is sitting in the grass.
A train accident where some cars when into a river.
A bunch of various vegetables on a table.
A plane parked at an airport near a terminal.
GAN-INT-CLS
StackGAN
AttnGAN
DM-GAN

(a) The CUB dataset

(b) The COCO dataset

Figure 3. Example results for text-to-image synthesis by DM-GAN and AttnGAN. (a) Generated bird images by conditioning on text from CUB test set. (b) Generated images by conditioning on text from COCO test set.

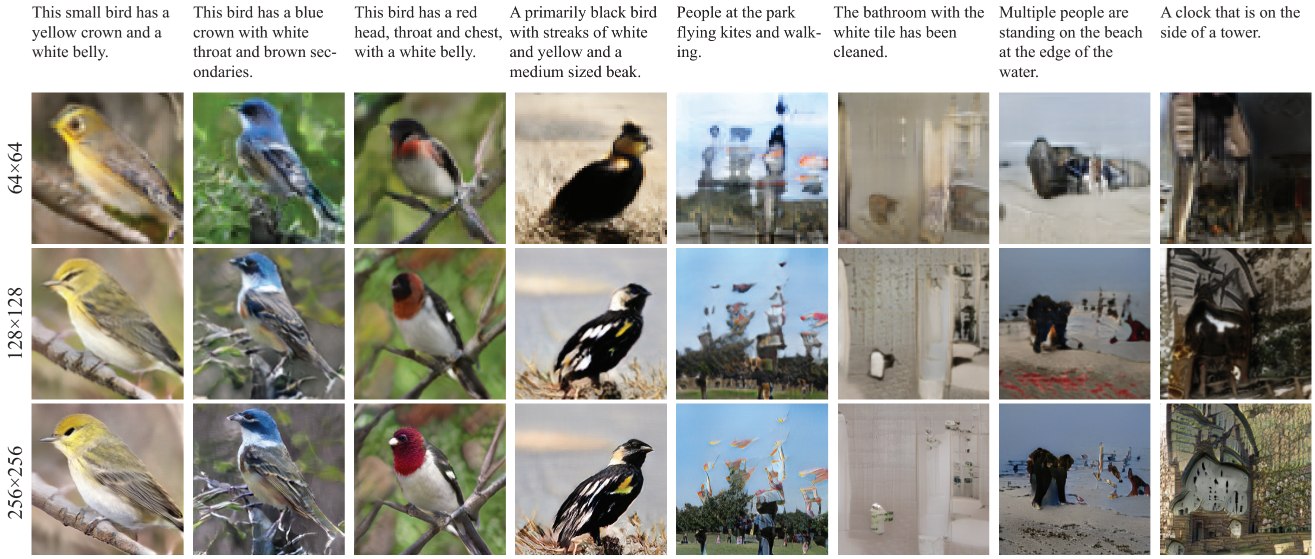

This small bird has a yellow crown and a white belly.
This bird has a blue crown with white throat and brown secondaries.
This bird has a red head, throat and chest, with a white belly.
A primarily black bird with streaks of white and yellow and a medium sized beak.
People at the park flying kites and walking.
The bathroom with the white tile has been cleaned.
Multiple people are standing on the beach at the edge of the water.
A clock that is on the side of a tower.
64×64
128×128
256×256

Figure 4. The results of different stages of our DM-GAN model, including the initial images, the images after one refinement process and the images after two refinement processes.

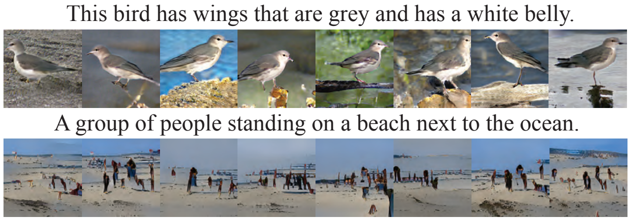
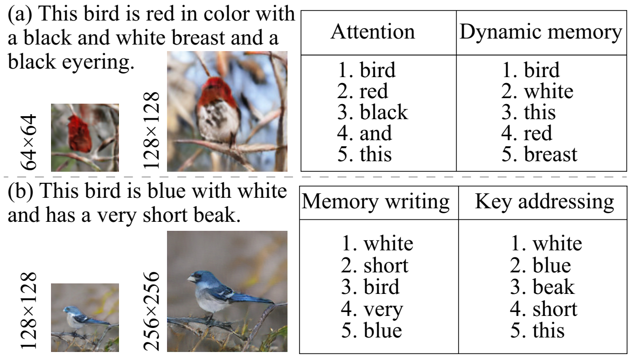

This bird has wings that are grey and has a white belly.
A group of people standing on a beach next to the ocean.

Figure 5. Generated images using the same text description.


(a) This bird is red in color with a black and white breast and a black eyering.
64×64
128×128
(b) This bird is blue with white and has a very short beak.
128×128
256×256

Attention	Dynamic memory
1. bird 2. red 3. black 4. and 5. this	1. bird 2. white 3. this 4. red 5. breast

Memory writing	Key addressing
1. white 2. short 3. bird 4. very 5. blue	1. white 2. blue 3. beak 4. short 5. this

Figure 6. (a) Comparison between the top 5 relevant words selected by attention module and dynamic memory module. (b) The top 5 relevant words selected by memory writing step and key addressing step.

demonstrates the effectiveness of every component.

Further, we visualize the most relevant words selected by the AttnGAN [28] and our DM-GAN. We notice that the attention mechanism cannot accurately select relevant words when the initial images are not well generated. We propose the dynamic memory module to select the most relevant words based on the global image feature. As Fig. 6 (a) shows, although a bird with incorrect red breast is generated, dynamic memory module selects the word, *i.e.*, ”white” to correct the image. The DM-GAN selects and combines word information with image features in two steps (see Fig. 6 (b)). The gated memory writing step first roughly selects words relevant to the image and writes them into the memory. Then the key addressing step further reads more relevant words from the memory.

5. Conclusions

In this paper, we have proposed a new architecture called DM-GAN for text-to-image synthesis task. We employ a dynamic memory component to refine the badly generated image, a memory writing gate to highlight important text information and a repose gate to fuse image and memory representation. Sometime the images synthesized by the DM-GAN are not satisfying. Because the complicated scenes of the COCO dataset make it challenging for most existing generative models, especially when more than one object needs to be generated. Our DA-GAN refines initial images with wrong color and rough shapes. However, the final results still rely heavily on the layout of multi-subjects in initial images. In the future, we will try to design a more powerful model to generate initial images with better layout.

Acknowledgment

This research has been supported by National Key Research and Development Program (2018YFB0904503) and National Natural Science Foundation of China (U1866602,61772456).

References

[1] Jiezhang Cao, Yong Guo, Qingyao Wu, Chunhua Shen, and Mingkui Tan. Adversarial learning with local coordinate coding. *ICML*, 2018.

[2] Ayushman Dash, John Cristian Borges Gamboa, Sheraz Ahmed, Marcus Liwicki, and Muhammad Zeshan Afzal. Tac-gan-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*, 2017.

[3] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE ICCV*, pages 5706–5714, 2017.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NIPS*, pages 2672–2680, 2014.

[5] Caglar Gulcehre, Sarath Chandar, Kyunghyun Cho, and Yoshua Bengio. Dynamic neural turing machine with continuous and discrete addressing schemes. *Neural computation*, 30(4):857–884, 2018.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, pages 6626–6637, 2017.

[8] D Kinga and J Ba Adam. A method for stochastic optimization. In *ICLR*, volume 5, 2015.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[12] Shuang Ma, Jianlong Fu, Chang Wen Chen, and Tao Mei. Da-gan: Instance-level image translation by deep attention generative adversarial networks. In *CVPR*, pages 5657–5666, 2018.

[13] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *ACL*, pages 1400–1409, 2016.

[14] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *CoRR*, 2014.

[15] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[16] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *CVPR*, pages 4467–4477, 2017.

[17] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, pages 2642–2651, 2017.

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[19] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[20] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, pages 1060–1069, 2016.

[21] Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. In *NIPS*, pages 217–225, 2016.

[22] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, pages 2234–2242, 2016.

[23] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NIPS*, pages 2440–2448, 2015.

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[25] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 7167–7176, 2017.

[26] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[27] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory Networks. In *ICLR*, 2015.

[28] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

[29] Mingkuan Yuan and Yuxin Peng. Text-to-image synthesis via symmetrical distillation networks. *arXiv preprint arXiv:1808.06801*, 2018.

[30] Han Zhang, Tao Xu, and Hongsheng Li. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE ICCV*, pages 5908–5916. IEEE, 2017.

[31] 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. *TPAMI*, 2018.

[32] Fengda Zhu, Linchao Zhu, and Yi Yang. Sim-real joint reinforcement transfer for 3d indoor navigation. In *CVPR*, 2019.

[33] Linchao Zhu, Zhongwen Xu, and Yi Yang. Bidirectional multirate reconstruction for temporal modeling in videos. In *CVPR*, July 2017.