CS6910 Project
DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis

Yadukrishnan R Menon (ME18B084)
Ahal Martin V (ME18B001)

Abstract

In the task of data-driven, text-based image generation, improvements in generative models, especially the style-based GAN architecture has shown to surpass or equal the existing state-of-the-art.The current methods of image generation based on text heavily relies on the initial image being higher quality and each word we feed into the network has some importance which is neglected in the current methods as we feed in the existing word representations. In this report, we try and reproduce the paper " DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis" by Zhu et.al.[2019] to generate high quality images. The authors propose a GAN archiecture, that introduces a dynamic memory module to refine blurry image contents, when the initial images aren't well generated. This paper also utilises two gates, a memory gate which selects the important text information based on the initial image content, which enables our method to accurately generate images from the text description and a response gate to adaptively combine the information read from the memories and the image features.

1 Introduction

In landscape of generative modelling, the recent development of Generative Adversarial Networks (GANs) [1] for high quality image synthesis with a text descriptor dictating how the images are to be generated. We reach one step closer to AI for the successful combination of the natural language and the associated visual content. A way in which this is achieved is via multi stage methods [2] where images created are of low resolution and is refined into a higher resolution image. Despite the multistage methods giving tremendous results, this approach chiefly deals with two problems:

- **Initial Image Quality** : The generated result of a Generative Adversarial Network is heavily dependant on the initial quality of the images. If the initial fuzzy images are badly generated, then the multi-stage refinement process of images won't generate high-quality images.
- **Varying level of information from text** : The words of an input sentence convey different levels of meaning about the visual content. Current refinement process is a bit ineffective as the same word representations are being used for different image refinements. Visual information must also be taken into consideration while forming textual meaning.

This paper has the following key contributions:

- A novel Dynamic Memory Generative Adversarial Network (DM-GAN) which contains a memory mechanism to manage badly-generated initial images. A **key-value memory** structure (which were a part of **memory networks**) is added to the GAN framework.
- A **memory writing gate** to dynamically select the words that resonate more with the generated image
- A **response gate** that adaptively fuses information from the image as well as the memory.

2 DM-GAN

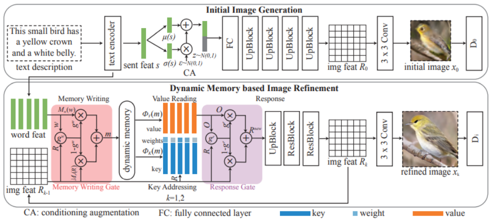
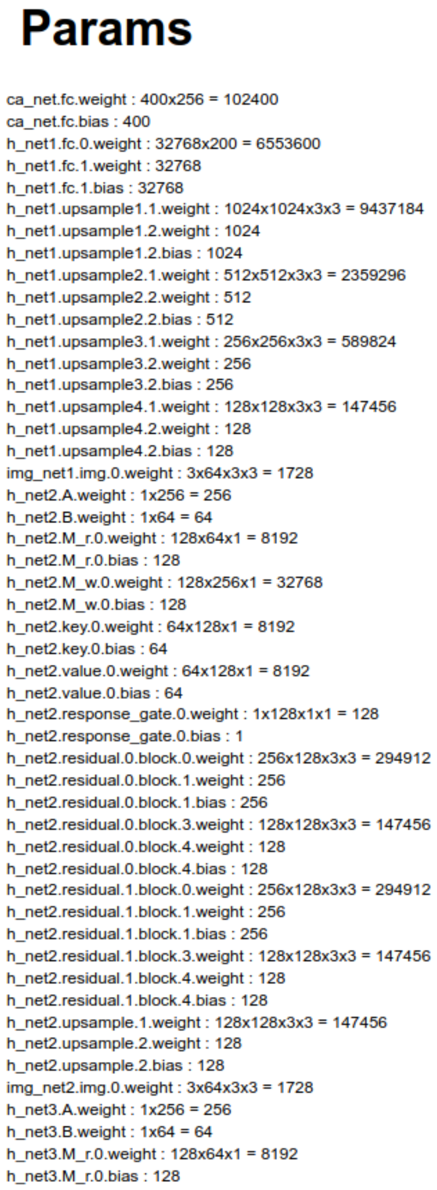


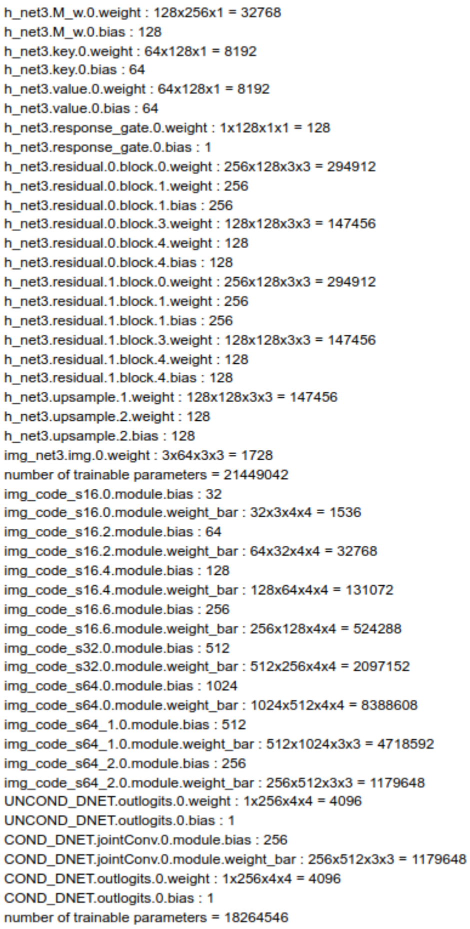
Figure 1: Architecture of DM-GAN for text to image sysnthesis.



Params

```
ca_net.fc.weight : 400x256 = 102400
ca_net.fc.bias : 400
h_net1.fc.0.weight : 32768x200 = 6553600
h_net1.fc.1.weight : 32768
h_net1.fc.1.bias : 32768
h_net1.upsample1.1.weight : 1024x1024x3x3 = 9437184
h_net1.upsample1.2.weight : 1024
h_net1.upsample1.2.bias : 1024
h_net1.upsample2.1.weight : 512x512x3x3 = 2359296
h_net1.upsample2.2.weight : 512
h_net1.upsample2.2.bias : 512
h_net1.upsample3.1.weight : 256x256x3x3 = 589824
h_net1.upsample3.2.weight : 256
h_net1.upsample3.2.bias : 256
h_net1.upsample4.1.weight : 128x128x3x3 = 147456
h_net1.upsample4.2.weight : 128
h_net1.upsample4.2.bias : 128
img_net1.img.0.weight : 3x64x3x3 = 1728
h_net2.A.weight : 1x256 = 256
h_net2.B.weight : 1x64 = 64
h_net2.M_r.0.weight : 128x64x1 = 8192
h_net2.M_r.0.bias : 128
h_net2.M_w.0.weight : 128x256x1 = 32768
h_net2.M_w.0.bias : 128
h_net2.key.0.weight : 64x128x1 = 8192
h_net2.key.0.bias : 64
h_net2.value.0.weight : 64x128x1 = 8192
h_net2.value.0.bias : 64
h_net2.response_gate.0.weight : 1x128x1x1 = 128
h_net2.response_gate.0.bias : 1
h_net2.residual.0.block.0.weight : 256x128x3x3 = 294912
h_net2.residual.0.block.1.weight : 256
h_net2.residual.0.block.1.bias : 256
h_net2.residual.0.block.3.weight : 128x128x3x3 = 147456
h_net2.residual.0.block.4.weight : 128
h_net2.residual.0.block.4.bias : 128
h_net2.residual.1.block.0.weight : 256x128x3x3 = 294912
h_net2.residual.1.block.1.weight : 256
h_net2.residual.1.block.1.bias : 256
h_net2.residual.1.block.3.weight : 128x128x3x3 = 147456
h_net2.residual.1.block.4.weight : 128
h_net2.residual.1.block.4.bias : 128
h_net2.upsample.1.weight : 128x128x3x3 = 147456
h_net2.upsample.2.weight : 128
h_net2.upsample.2.bias : 128
img_net2.img.0.weight : 3x64x3x3 = 1728
h_net3.A.weight : 1x256 = 256
h_net3.B.weight : 1x64 = 64
h_net3.M_r.0.weight : 128x64x1 = 8192
h_net3.M_r.0.bias : 128
```

Figure 2: Params of DM-GAN - 1



```
h_net3.M_w.0.weight : 128x256x1 = 32768
h_net3.M_w.0.bias : 128
h_net3.key.0.weight : 64x128x1 = 8192
h_net3.key.0.bias : 64
h_net3.value.0.weight : 64x128x1 = 8192
h_net3.value.0.bias : 64
h_net3.response_gate.0.weight : 1x128x1x1 = 128
h_net3.response_gate.0.bias : 1
h_net3.residual.0.block.0.weight : 256x128x3x3 = 294912
h_net3.residual.0.block.1.weight : 256
h_net3.residual.0.block.1.bias : 256
h_net3.residual.0.block.3.weight : 128x128x3x3 = 147456
h_net3.residual.0.block.4.weight : 128
h_net3.residual.0.block.4.bias : 128
h_net3.residual.1.block.0.weight : 256x128x3x3 = 294912
h_net3.residual.1.block.1.weight : 256
h_net3.residual.1.block.1.bias : 256
h_net3.residual.1.block.3.weight : 128x128x3x3 = 147456
h_net3.residual.1.block.4.weight : 128
h_net3.residual.1.block.4.bias : 128
h_net3.upsample.1.weight : 128x128x3x3 = 147456
h_net3.upsample.2.weight : 128
h_net3.upsample.2.bias : 128
img_net3.img.0.weight : 3x64x3x3 = 1728
number of trainable parameters = 21449042
img_code_s16.0.module.bias : 32
img_code_s16.0.module.weight_bar : 32x3x4x4 = 1536
img_code_s16.2.module.bias : 64
img_code_s16.2.module.weight_bar : 64x32x4x4 = 32768
img_code_s16.4.module.bias : 128
img_code_s16.4.module.weight_bar : 128x64x4x4 = 131072
img_code_s16.6.module.bias : 256
img_code_s16.6.module.weight_bar : 256x128x4x4 = 524288
img_code_s32.0.module.bias : 512
img_code_s32.0.module.weight_bar : 512x256x4x4 = 2097152
img_code_s64.0.module.bias : 1024
img_code_s64.0.module.weight_bar : 1024x512x4x4 = 8388608
img_code_s64_1.0.module.bias : 512
img_code_s64_1.0.module.weight_bar : 512x1024x3x3 = 4718592
img_code_s64_2.0.module.bias : 256
img_code_s64_2.0.module.weight_bar : 256x512x3x3 = 1179648
UNCOND_DNET.outlogits.0.weight : 1x256x4x4 = 4096
UNCOND_DNET.outlogits.0.bias : 1
COND_DNET.jointConv.0.module.bias : 256
COND_DNET.jointConv.0.module.weight_bar : 256x512x3x3 = 1179648
COND_DNET.outlogits.0.weight : 1x256x4x4 = 4096
COND_DNET.outlogits.0.bias : 1
number of trainable parameters = 18264546
```

Figure 3: Params of DM-GAN - 2

The methodology is implemented via two stages as shown in the diagram above:

- Initial Image Generation
- Image Refinement using Dynamic Memory

At the initial image generation stage, firstly, the input text description is transformed into some internal representation by using a text encoder. The text description given to the model is converted to an internal representation of sorts (a sentence feature s and several word features W). Then, the generator predicts an initial image x_0 of rough quality and few details using the sentence feature s 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 normally sampled (i.e from a normal distribution).

At the the next stage of dynamic memory based image refinement, more fine-detail visual contents are added to the blurry 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 obtained from the previous stage.
This stage can be repeated multiple times to retrieve more relevant information and generate a high-resolution image with more fine-grained details.

The dynamic memory based image refinement stage has four components as succinctly described in the following sub-sections.

2.1 Memory Writing

In this step we store the text information into a key value structured memory by using a 1×1 convolution operation which embeds word features into the memory feature space for later use and is controlled by means of a memory gate.

2.2 Key Addressing and Value Reading

In key addressing we retrieve relevant memories using key memory by calculating similarity probability(softmax) of a certain image feature along with its key memory. During Value reading, the output memory representation is defined as the weighted sum of value memories according to the probability which we calculated as a part of key addressing

2.3 Response

After receiving the output memory, during response, we combine the current image and the output representation to provide a new image feature, thereby adaptively fusing of features of the image and the memory contents and this is controlled by means of a response gate.

As for the main GAN architecture, the authors have implemented an AttnGAN (Attention Generative Adversarial Networks)[3], which is capable of refining the images to high-resolution ones by utilising the infamous multi-head attention mechanism.Even though each word present in the input sentence has a different level of information depicting the image content, AttnGAN takes all the words with no bias by employing an attention module to use the same word representation and this flaw is overcome by the dynamic memory module (inspired from Memory Networks [4]) discussed in the above subsections.

3 Experiments and Results

The experiments where chiefly conducted on the Caltech-UCSD Birds 200 (CUB-200) dataset.The CUB dataset contains 200 bird categories with 11,788 images, with 10 captions for each image in the dataset. The code is written in Pytorch and the bench marked in a NVIDIA RTX 2060 GPU. Considering the GPU limitations we have, the model was trained with a batch size of **2** as compared to **10** in the paper for the same amount of epochs and other hyper-parameters which is reflected in the tabular results below.

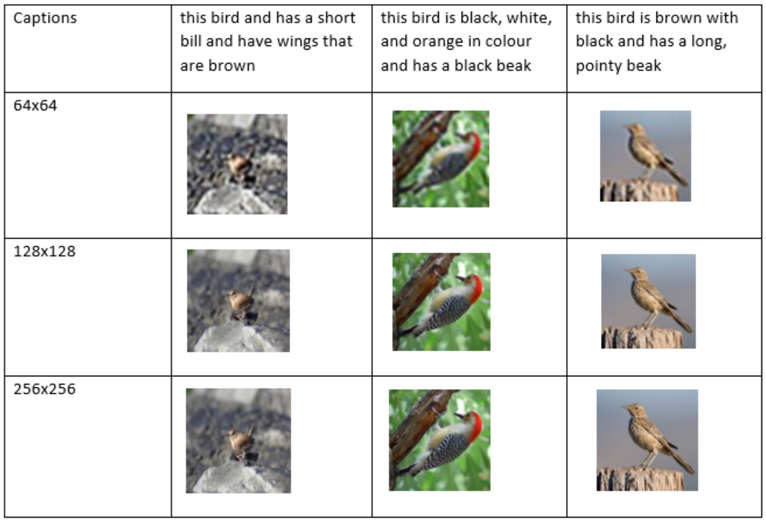
3.1 Metrics and Generated Results

For all these experiments, Frechet Inception Distance (FID)[5] and Inception Score (IS)[6] were used to evaluate the quality of the generated samples. A pretrained inception network is used to generate feature vectors from both the real and fake images. The distance between them is calculated as the FID score. A lower FID score indicates the closeness of the generated image with the real ones.The IS [22] uses a pretrained Inception v3 network [24] to compute the KL-divergence between the marginal and conditional class distribution. A small IS would imply that the generated model outputs a low diversity of images for all classes where each image clearly belongs to a specific class.

Model	Frechet Inception Distance	Inception Score
Paper	4.75 ± 0.07	16.09
Reproduced	**4.53**	**17.76**

Table 1: Comparison of the results in the paper and our reproduced version of the model.

Fig 4 indicates that the DM-GAN model is able to refine badly initialized images (64X64) and generate more photo-realistic high-resolution images.



Captions	this bird and has a short bill and have wings that are brown	this bird is black, white, and orange in colour and has a black beak	this bird is brown with black and has a long, pointy beak
64x64			
128x128			
256x256			

Figure 4: Different Stage results i.e the initial images, the images after one refinement process and the images after two refinement processes with their corresponding captions

4 Conclusion

In this paper, a new architecture called DM-GAN for text-to-image synthesis task has been introduced. The authors have constructed a dynamic memory component to refine the initial fuzzily generated image, a memory writing gate to emphasize important text information and a repose gate to adaptively fuse image and memory representation. Despite the model being too large and a bit complex as compared to the other models serving the same purpose and the final results relying a lot on the multi-subject layout in initial images, this paper still captures information more accurately and relate it to image features, which is a remarkable step towards true AI.

5 References

1. Generative Adversarial Nets - Ian Goodfellow et.al. 2014, NIPS

2. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks - Han Zhang et.al.2017, IEEE

3. Attngan: Fine-grained text to image generation with attentional generativeadversarial networks - Tao Zhu et.al 2018, CVPR

4. Key-value memory networks for directly reading documents - Alexander Miller et.al. 2016, ACL

5. Gans trained by a two time-scale update rule converge to a local nash equilibrium - Martin Heusel et.al. 2017, NIPS

6. Improved techniques for training gans - Tim Salimans et.al. 2016, NIPS