Text to Image Generation with Wasserstein Attentional Generative Adversarial Networks

Ziming Yin Lan Xiao
University of Toronto
40 St George St, Toronto, ON
lxiao, ziming@cs.toronto.edu

Abstract

Generating high quality images given a text description is a challenging problem and has great values in applications. Currently most of the approaches regarding this problem are based on conditional GAN (cGAN) [13], which conditions the generation on some additional information, a text description in the text-to-image task. cGAN is able to capture the overall theme from the sentence embedding of the text description, but lacks fine-grained details which are encoded in words of the text description. cGAN also suffers from mode collapse in which the generator produces limited varieties of samples. AttnGAN et al.[19] was proposed to address the first problem and achieve the state-of-the-art results on generating the CUB dataset and COCO dataset. In AttnGAN, a multi-stage GAN is trained to fill in realistic details at different spatial locations of the image by paying attention to the relevant words in the text description. In this project, we propose Wasserstein Attentional Generative Adversarial Network (WAGAN) to address the mode collapse problem by using a more stable GAN training process. Based on AttnGAN, we train our model by minimizing a different loss which effectively estimate the Wasserstein distance between the real data distribution $\mathcal{P}_r$ and the generator distribution $\mathcal{P}_g$ [3]. We conduct extensive experiments on heavily benchmarked datasets and present the generated samples.

1. Introduction

Recent advances in computer vision and natural language processing have lead to many exciting areas of research such as image captioning, visual question answering and text to image synthesis [2, 1, 16]. Among the others, text to image synthesis is particularly challenging because it involves generating high dimensional image from natural language. Text to image synthesis has many potential real world applications. For example, this allows people without strong art background to produce artistic content. This also allows automatic photo editing. In this project we address the problem of novel images generation according to a human annotated natural language description. For example, given the text "a black bird with black feet, a stripe head, a wide tail, a short beak that is pointy" (one text description in our training data), we expect the algorithm to be able to generate relevant image. This problem can be broken down into a few sub-problems. First, we need to train a text encoder that 'understands' the visual difference between words, e.g. the encoder need to be able to distinguish different colours, which are often clustered together in traditional word embeddings. Secondly, conditioning on the text description, the image generator should be able to generate photo-realistic images that are consistent with the text and diverse in styles. Recently, the development of Generative Adversarial Networks (GAN) [4] has shown promising results in generating realistic images. Several text to image synthesis methods based on GAN have been proposed. By conditioning on both generator and discriminator, GANs are able to generate images that are highly related to the given text. A prevailing approach is to use LSTM [7] to generate a global sentence embedding as condition for both the discriminator and the generator. Xu *et al.*[19] further improves the quality of the generated samples by extracting word-level embeddings and uses image to guide which word to attend to when generating fine-grained details at different sub-regions of the image.

However, it is very difficult to train GANs, and the original training loss is not an informative indication of model convergence. The stability of GAN training is only guaranteed for specific architectures and depends heavily on the chosen hyper-parameters. One common failure case for GAN training is mode collapse. For example, in MNIST dataset, ten modes are presented corresponding to digits '0' to '9'. In the case of mode collapse, only samples of one digits can be generated. To address this issue, we propose Wasserstein Attention Generative Network (WAGAN) [3]

that minimize the Wasserstein distance during training. It has been shown to have better theoretical convergence guarantee. Our code is available at https://github.com/ZimingY/AttnGAN. The main contributions of our work are:

- We train a Wasserstein Attentional GAN (WAGAN) for synthesizing photo-realistic images from text descriptions.
- We conduct experiments on several heavily benchmarked datasets to demonstrate the effectiveness and drawbacks of our model design.
- We summarize our different training procedures and provide guidelines on how to systematically fine-tune the hyper-parameter for WAGAN. This provides useful information for designing future WGAN models.

2. Related Work

Generative methods that produce novel samples from high-dimensional data distributions such as images have seen remarkable progress in recent years with the advances of deep learning techniques. Currently the are a few important classes of methods including autoregressive models [17], variational autoencoders [10], and generative adversarial networks [4]. GANs produces sharp and photo-realistic images while suffers from limited variation and unstable training. Radford [15] introduced a class of GAN called deep convolutional generative adversarial networks (DCGAN), and it becomes the most widely used networks in image generation. However, DCGAN suffers from the aforementioned problems. To address these problems, people have proposed many extensions based on DCGAN model.

Conditional GAN [13] is an extension of GAN where both the generator and the discriminator receive additional conditioning variables such as attributes or class labels. Based on this formualtion, cGAN can generate MNIST digits conditioned on a class label c [13]. Recently, several methods have been proposed to condition image generation on unstructured text. Reed *et al.* [16] developed GAN-CLS that generate images using cGAN conditioning on the text embedding φ_t to generate 64×64 samples. In their follow-up work, they are able to generate higher resolution samples (128×128) by conditioning also on the annotations of object locations.

By drawing analogy to how human painters draw, Zhang *et al.*[21] developed a multiple-layer GANs (StackGAN) in which they condition on text descriptions multiple times to synthesize the images progressively. The overall architecture is shown in Fig 1. The Stage-I GAN generates low-resolution images conditioned on the given text descriptions. Stage-II GAN takes the low-resolution images generated in Stage-I as input and conditions on the text descriptions again to generate high-resolution images (256×256) that look realistic. This sequential image generation process allows the model to first learn coarse contours and color in the first stage, and tune the image and complete details that are omitted in the first stage by reading the text again. This approach is able to generate high-resolution images without the need to annotate the images with object locations. Zhang *et al.* [21] also introduced a Conditioning Augmentation technique to encourage smoothness in the latent conditioning manifold. They apply a nonlinear transformation to the sentence embedding φ_t and generate additional conditioning variable $\hat{c}$ by sampling from $\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))$. This encourages robustness to small perturbations along the conditioning manifold. To further enforce the smoothness, they introduce a KL divergence between the standard Gaussian distribution and the conditioning Gaussian distribution.

$$D_{KL}(\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))||\mathcal{N}(0, I)) \quad (1)$$

This regularization term is included in the generator loss $\mathcal{L}_{Gi}, i \in [0, 1]$ and minimized at both stages.

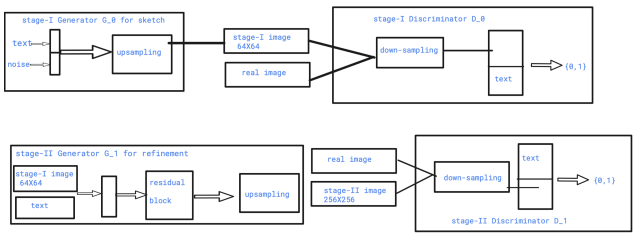

Figure 1. The architecture of StackGAN

Recently, the attention mechanism has gain popularity in training multi-modal neural network models. It learns an effective alignment between different modalities, such as image objects and agent actions in dynamic control problems [14], visual features and text embeddings in image captioning [1] and visual question answering [20, 11]. Mansimov *et al.*[12] introduced a conditional alighDRAW model that used soft attention to generate images from captions. Xu *et al.*[19] proposed Attentional Generative Adversarial Network to synthesize fine-grained details at different sub-regions of the images by paying attention to the important information at word level. The overall architecture is shown in figure 2. The model learns a bi-directional LSTM to map the unstructure text description to a vector space. The hidden states are concatenated to form a feature matrix $e \in R^{D \times T}$, where D is the word embedding dimen-

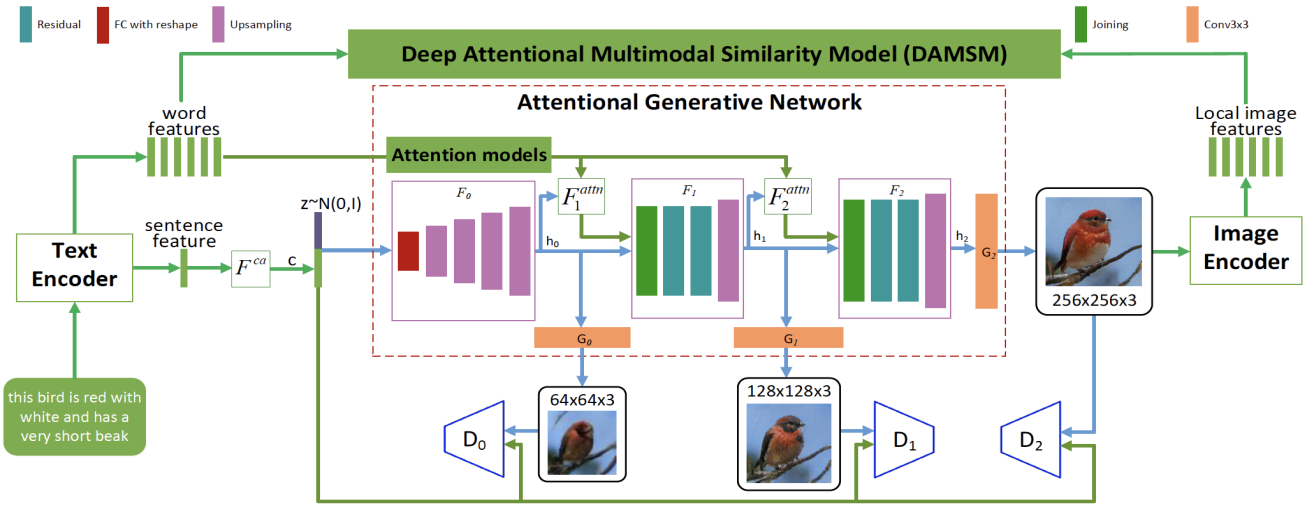

Figure 2. The architecture of AttnGAN

sion and T is the sequence length. The i^{th} column e_i is the word embedding for the i^{th} word. The last hidden state represents the sentence embedding $\bar{e} \in R^D$. The sentence embedding $\bar{e}$ is converted to the conditioning vector $\hat{c}$ using Conditional Augmentation described above. The model employs a multi-stage GAN structure similar to the StackGAN, but instead of conditioning on the sentence embedding multiple times, AttnGAN only uses the sentence embedding to generate low-resolution image $h_0 \in R^{D \times N}$ in the first stage, where N is the number of spatial locations. In the subsequent stages, the image features from the previous layer h_{i-1} are used to query the word feature matrix e by calculating an attention layer to form a word-context vector c. The word-context vector is a dynamic representation of word vectors relevant to specific spatial locations in the sample. Conditioning on both the word-context vectors and the image features, AttnGAN is able to focus on words that are most relevant to the sub-regions being drawn. This progressively synthesizes a higher resolution image with more details at each stage. To enforce the matching between the generated samples and the corresponding text descriptions, a Deep Attentional Multimodal Similarity Model (DAMSM) is applied to the last stage generator output. DAMSM calculates the fine-grained text-image matching loss L_{DAMSM} based on attention mechanism. Since we intend to use this sub-module in our architecture, we refer interested readers to the original paper for further details.

3. Wassersterin Attentional GANs

3.1. Preliminaries

Generative Adversarial Networks (GAN) are composed of two competing models that are alternatively trained to play a two-player minimax game. The generator G learns a mapping from input noise variable $p_z(z)$ to data space. The discriminator $D(x)$ takes input from either the real data distribution $\mathcal{P}_r$ or the generator distribution $\mathcal{P}_g$, and outputs the probability that x come from $\mathcal{P}_r$. Formally, the game between the generator G and the discriminator D is the minimax objective:

$$\min_G \max_D \mathbf{E}_{x \sim \mathcal{P}_r}[\log(D(x))] + \mathbf{E}_{\tilde{x} \sim \mathcal{P}_g}[\log(1 - D(\tilde{x}))]$$

Conditional GAN extend the original idea of GAN by conditioning both D and G on additional variable c, yielding $G(z, c)$ and $D(x, c)$, where z is a noise vector sampled from a standard normal distribution. The objective function of cGAN is:

$$\begin{aligned} &\min_G \max_D V\{G, D\} \qquad (2) \\ =&\mathbf{E}_{x \sim p(x)}[\log D(x|c)] + \mathbf{E}_{z \sim p(z)}[\log(1 - D(G(z)|c))] \end{aligned}$$

Though GANs are powerful generative models, they suffer from training instability and mode collapse. Arjovsky *et al.* [3] provides an analysis of the convergence properties of the value function being optimized by GANs. They proposes the Wassersterin GAN, which leverages the Wassersterin distance to produce a value function that has better theoretical properties than the original formulation. The Wassersterin distance $W(q, p)$ is defined as the minimum cost of transporting mass in order to transform distribution q into distribution p. The dual form of a WGAN value function can be written as :

$$\min_G \max_D \mathbf{E}_{x \sim p(r)}[D(x)] + \mathbf{E}_{\tilde{x} \sim p(g)}[D(\tilde{x})]$$

where D is the set of 1-Lipschitz functions and can be optimized in the parametric function space using a neural

network. To enforce the Lipschitz constrain, we require the gradient norm of the critic's output with respect to its input is at most one everywhere. Gulrajani *et al.*[5] introduce a penalty on the gradient norm when it deviates from one.

The new objective is

$$L = \mathbf{E}_{\tilde{x}\sim\mathcal{P}_g}[D(\tilde{x})] - \mathbf{E}_{x\sim\mathcal{P}_r}[D(x)] + \lambda\mathbf{E}[(||\nabla_{\hat{x}}D(\hat{x})||_2 - 1)^2] \quad (3)$$

where $\hat{x}$ is sampled uniformly along straight lines between pairs of points sampled from the data distribution $\mathcal{P}_r$ and the generator distribution $\mathcal{P}_g$, $\hat{x} = \alpha G(z) + (1-\alpha)x$.

The gradient of the critic function with respect to its input behaves better than the original GAN, which makes optimizing the generator easier. Empirically, it is also observed that WGAN value function appear to be a good indication of sample quality.

Besides, since WGAN penalize the norm of the critic's gradient with respect to each input independently, while batch normalization create correlations between samples in the same batch, batch normalization in the critic is avoided.

3.2. Model Architecture

We build our model based on AttnGAN [19]. We keep the multi-layer structure which generate high-resolution samples progressively by paying attention to relevant word embeddings at each layer. The specific architecture is shown in table 1. A few important modifications are introduced in order to incorporate the Wasserstein training objective. In the rest of this subsection, we present details of our model WAGAN.

Stage I GAN

Let $\bar{e}$ represent the sentence embedding, We generate the conditioning vector $\hat{c}$ by applying Conditioning Augmentation to the sentence vector $F^{ca}(\bar{e})$. v is a noise vector sampled from a standard normal distribution. We concatenate v and $\hat{c}$ to generate a 64×64 image by applying a series of transposed convolution layers.

Instead of learning a discriminator $D(x)$ which output the probability that x came from the real data distribution rather than the generator distribution, Stage-I GAN learns a critic to approximate the Wasserstein distance, which is a regression problem. The sentence embedding $\bar{e}$ is spatially replicated to form a $M_d \times M_d \times N_d$ matrix. The input image, either from P_{data} or P_g, is passed through a series of convolution layers until the spatial dimension is $M_d \times M_d$. Then, the image feature is concatenated with the sentence embedding along the depth dimension. The resulting tensor passes through a 1×1 convolution layer to learn a multi-modal representation of the image and the text. Finally, a fully connected layer with 1 scalar output is used to produce the Wasserstein distance. Note that since the resulting scalar is no longer a probability, we do not apply *sigmoid* activation to the output of the critic.

In the case of Wasserstein GANs, as the critic D gets closer to the optimal critic D^*, it better approximate the Wasserstein distance $W(P_{data}, P_r)$, which allows the generator to be better updated. Therefore, we train the critic for n_{critic} times for every generator update. In this experiment, we fix $n_{critic} = 5$. Since the use of *batchnorm* should be avoided in the critic for WGAN formulation, we use *InstanceNorm2d* as a drop in replacement where *batchnorm* is originally used. The objective function of critic D is

$$L_D = \frac{1}{2}(\mathbf{E}_{\tilde{x}\sim\mathcal{P}_g}[D(\tilde{x})] + \mathbf{E}_{\tilde{x}\sim\mathcal{P}_g}[D(\tilde{x},\bar{e})]) - \frac{1}{2}(\mathbf{E}_{x\sim\mathcal{P}_r}[D(x)] + \mathbf{E}_{x\sim\mathcal{P}_r}[D(x,\bar{e})]) + \lambda\mathbf{E}[(||\nabla_{\hat{x}}D(\hat{x})||_2 - 1)^2] \quad (4)$$

As for the gradient penalty, since we have text description as condition, we calculate $\nabla_{\hat{x}}D(\hat{x})$ and $\nabla_{\hat{x}}D(\hat{x}|c)$ and use the average as our penalty in the critic objective function.

The generator loss consists three part, the adversarial loss that minimizes the Wasserstein distance, and a regularization term to enforce smoothness over the conditioning manifold. Since the critic output an approximation of the Wasserstein distance instead of probability, the log trick is removed.

$$L_G = -\frac{1}{2}(\mathbf{E}_{\tilde{x}\sim\mathcal{P}_g}(D(\tilde{x})) + \mathbf{E}_{\tilde{x}\sim\mathcal{P}_g}(D(\tilde{x},\bar{e}))) + \lambda D_{KL}(\mathcal{N}(\mu(\varphi_t),\Sigma(\varphi_t))||\mathcal{N}(0,I)) \quad (5)$$

Stage II/III GAN

In the next two layers, normal GAN setup is used to generate high-resolution images with vivid details. Since the setups for stage-II and stage-III GAN are identical, we will describe them here together. Conditioning on the image features generate from the previous layer and a word-context features generate from $F^{attn}(e,h)$, stage-II/III GAN can correct defects in the previous layer and completes previously ignored word-level details to generate more photo-realistic samples.

The attention model $F^{attn}(e,h)$ takes the word feature $e \in R^{D\times T}$ and the image features from the previous stage GAN $h \in R^{\hat{D}\times N}$ as input, where D represent the hidden

Table 1. The architecture of WAGan

Discriminator Architecture			Generator Architecture		
D_0	D_1	D_2	G_0	G_1	G_2
input: 3 $\times$64 $\times$ 64	input: 3 $\times$128 $\times$ 128	input: 3 $\times$256 $\times$ 256	input: sentence embedding	input: h_{code1}	input: h_{code2}
conv2d(3,64) LeakyReLU	conv2d(3, 64) LeakyReLU	conv2d(3, 64) LeakyReLU	GLU() linear(256, 400)	Attention module $\rightarrow$ c_{code}	Attention module $\rightarrow$ c_{code}
conv2d(64,128) InstanceNorm2d LeakyReLU	conv2d(64, 128) BatchNorm LeakyReLU	conv2d(64, 128) BatchNorm LeakyReLU	cat with noise linear() BatchNorm GLU()	cat h_{code1} and c_{code}	cat h_{code1} and c_{code}
conv2d(128,256) InstanceNorm2d LeakyReLU	conv2d(128, 256) BatchNorm LeakyReLU	conv2d(128, 256) BatchNorm LeakyReLU	upsample(2) con2d(512,512) BatchNorm GLU()	residual block	residual block
conv2d(256,512) InstanceNorm2d LeakyReLU	conv2d(256, 512) BatchNorm LeakyReLU	conv2d(256, 512) BatchNorm LeakyReLU	upsample(2) con2d(256,256) BatchNorm GLU()	upsample(2) conv2d() BatchNorm GLU() $\rightarrow h_{code2}$	upsample(2) conv2d() BatchNorm GLU() $\rightarrow h_{code3}$
If condition cat with text 256	conv2d(512, 1024)	conv2d(512, 1024) conv2d(1024, 2048)	upsample(2) con2d(128,128) BatchNorm GLU()	conv2d(32,3) Tanh()	conv2d(32,3) Tanh()
conv2d(768,512) InstanceNorm2d LeakyReLU	conv2d(1024,512) BatchNorm LeakyReLU	conv2d(2048,1024) BatchNorm LeakyReLU	upsample(2) con2d(64,64) BatchNorm GLU() h_{code1}		
conv2d(512,1)	If condition: cat with text 256 conv2d(768,512) BatchNorm LeakyReLU conv2d(512,1) sigmoid()	conv2d(1024,512) BatchNorm LeakyReLU	conv2d(32,3) Tanh()		
If no condition conv2d(512,1)	If not condition: conv2d(512,1) sigmoid()	If condition: same as D_1			
		If not condition: same as D_1			

dimension for word embeddings, T represent the sequence length, $\hat{D}$ represent the hidden dimension for image features, and N represent the number of image locations. The image feature and the word feature are first converted into a common semantic space using fully connected layer with ReLU activation, i.e. $e' = Ue$, where $U \in R^{\hat{D} \times D}$. Then, we query the e' with image feature h, resulting in a weight matrix β which it's $(i, j)^{th}$ element represent how much attention should j^{th} image location pay to the i^{th} word. Formally,

$$\beta_{i,j} = \frac{exp(h_j^T e'_i)}{\sum_{k=0}^{T-1} exp(h_j^T e'_k)} \tag{6}$$

Then, we calculate the word-context vector c_j for the j^{th}

image location by applying the weight:

$$c_j = \sum_{i=0}^{T-1} \beta_{j,i} e'_i \tag{7}$$

The image features and the corresponding word-context features are combined to form a multmodal context vector, based on which the GAN generates new images features in the surrounding proximity.

To generate realistic images with multiple levels (i.e., sentence level and word level) of conditions, the final objective function of the attentional generator is defined as the sum of unconditional loss and conditional loss:

$$\begin{aligned} L_{G_i} = & \sum_i -\frac{1}{2}\mathbf{E}_{\tilde{x}_i \sim \mathcal{P}_g}[\log(D_i(\tilde{x}_i))] \\ & - \sum_i \frac{1}{2}\mathbf{E}_{\tilde{x}_i \sim \mathcal{P}_g}[\log(D_i(\tilde{x}_i, \bar{e}))] \end{aligned} \tag{8}$$

The unconditional loss determines whether the image is real or fake while the conditional loss determines whether the image and the sentence match or not.

Each discriminator D_i is trained to classify the input into the class of real or fake by minimizing the cross-entropy loss defined by

$$\begin{aligned} L_{D_i} = & -\frac{1}{2}\mathbf{E}_{x \sim \mathcal{P}_r}[\log D_i(x)] - \frac{1}{2}\mathbf{E}_{\tilde{x}_i \sim \mathcal{P}_g}[\log(1 - D_i(\tilde{x}_i))] + \\ & - \frac{1}{2}\mathbf{E}_{x \sim \mathcal{P}_r}[\log D_i(x, \bar{e})] - \frac{1}{2}\mathbf{E}_{\tilde{x}_i \sim \mathcal{P}_g}[\log(1 - D_i(\tilde{x}_i))] \end{aligned} \tag{9}$$

where x is from the true data distribution p_{data} and $\tilde{x}_i$ is from the i^{th} layer generator distribution p_{gi}. We train the discriminator 1 time for every generator update. The discriminator D classifies the input images into two classes, real or fake, so that the last layer of the discriminator is a *sigmoid* to convert logit into probability.

In the last layer, we calculate a fine-grained text-image matching loss L_{DAMSM} [19]. With an attention mechanism, this loss is able to compute the similarity between the generated sample and the sentence using both the sentence level information and the detailed word level information. Thus, L_{DAMSM} provides additional error signal for training the generators.

3.3. Implementation Details

In the first iteration, we try to enforce the Wasserstein distance in all three layers of the attentional GAN, but the model does not show promising result. Inspired by Hung [8], we only enforce Wasserstein distance for the unconditional loss but keep the original value function for the conditional loss. Hung is able to train a conditional WGAN on four classes. For the conditional images, the discriminator loss is computed by standard cross entropy of the output probability, while for the unconditional samples generating by only looking at a noise vector, he used Wasserstein distance. The final loss was the sum of these two. Unfortunately, the inception score after training with this setting was still sub-optimal. To improve the performance, we introduce two training techniques, namely learning rate decay and Two-Timescale Update Rule (TTUR) [6]. TTUR refers to the usage of two different learning rates for the generator and the discriminator, which have nice theoretical guarantees that the networks will converge to a local Nash equilibrium. The convergence under TTUR is shown to be faster and the quality of the images are higher than in the case of classic method of training. Before jumping to our current model, since the previous two architectures do not work, we tried to fix it by introducing a more stable WGAN training process.

We use Adam optimizer [9] to train the text encoder, image encoder, discriminator, and generator. At each round, the critic is trained five iterations per one generator iteration. For the training process of encoder, the learning rate is 0.002 and batch size is 48. The learning rate of the critic and generator in the first stage is $5e-6$. For the following stages, the learning rate is $1e-4$. The batch size is 20 for all discriminator training and generator training. γ, γ_2, γ_3 and λ are set to be $4, 5, 10, 10$. We run 1000 epochs in total, and each epoch has 402 batches. The training of the discriminator D and generator G takes 44 hours.

Our final architecture is shown in Table 1. We choose to use WGAN only on the first stage. For the following two stages, we keep the architecture in *et al.*[19]. Then the loss for the first state is computed via Eq (4) and Eq (5). Loss for the second and third stage is computed by Eq (9) and Eq (8).

4. Experiments

We use CUB dataset [18] to help develop the algorithm. This is a heavily bench-marked dataset for text to image synthesis modeling. The CUB dataset contains 11788 images and 200 types of birds. The CUB dataset only contains images, but there are publicly available captions collected by Reed *et al.* [16] for the task of text to image synthesis. Each image has five descriptions, with at least 10 words in length. The descriptions only contains information about the features of the bird, not the background and they do not explicitly mention the bird types. We split the datasets into training and testing set such that they contain disjoint

classes. Since the goal is to generate realistic images, the validation set is merged together with training set. CUB dataset is split into training (8855) and testing (2933).

Fig 3 shows the generated images at different epoch along with their attentional maps. The text description is on the top of each row, and marked with the colors learned by the model. For each word, the attention module highlights the sub-regions that the word focuses heavily. It is noticeable that the sub-regions that attention highlights are getting concrete and precise. Besides, the images of the birds become clearer with the increment of training iterations. Below are text descriptions for images after 500 epoch in Fig 3. The the descriptions are quite long and with various of details, the model captures these features pretty well.

5. This a bird with spot on crown, black short bill, white throat, white spots breast, and yellow wings.

6. Bird is blue and white color with a head and line around its neck.

7. Bird has yellow breast with a black and yellow crown and yellow belly

8. This is a small brown bird that has a beige breast and belly and a small beak.

9. This bird has black wings and tail with yellow upper covert and a bright red body and head.

10. White chest bird with gray wings and crown tail feather are very dark gray, the under being white.

We further test how sensitive the outputs are if using the sentences in the validation set, which are not seen during training. Some examples are shown in Fig 4. The generated images are pretty close to the text descriptions and also have convincing background. From the generated images, we do not observe mode collapse. The descriptions for the images are as following:

1. This bird is red with white and has a very short beak.

2. The bird has a yellow crown and a black eyering that is round.

3. A medium sized bird with a white underbelly and a yellow tipped head.

4. A colorful little bird with a bright yellow crow, black and yellow wings, and brown stripes on the chest.

The above generated figures show promising results of our architecture.

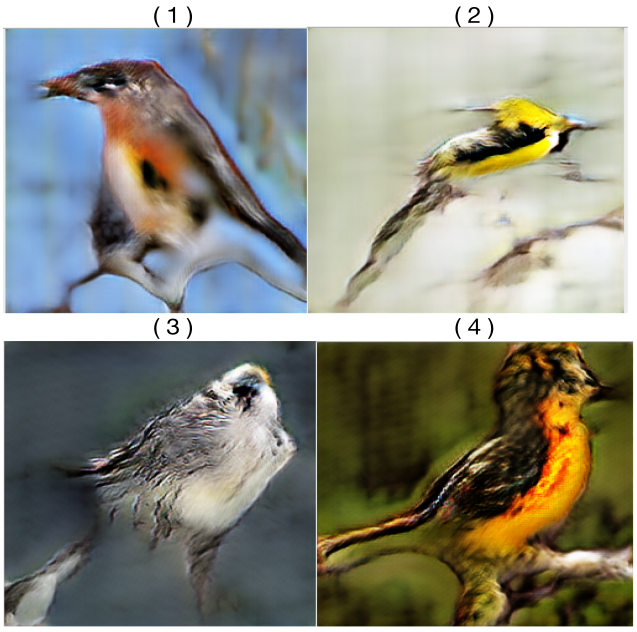

Figure 4. Example 256×256 results of our WAGAN model on validation set

5. Conclusion and Limitation

In this work, we proposed Wasserstein Attentional Generative Adversarial Network, a stable GAN architecture to generate high-resolution (256×256), photo-realistic images from text descriptions. We build our model based on the state-of-art model AttnGAN [19] but improved training objective, which approximates the Wasserstein distance. It has better theoretical convergence guarantee and empirically perform superior. The improved value function prevent mode collapse. Since the architecture of AttnGAN is complex, finding the right way to combine it with WGAN is non-trivial, we need to perform subtle changes to make the model converge. We conduct several experiments under different settings and find the architecture that works the best. Extensive experiment results demonstrate the effectiveness of our proposed WAGAN architecture. Currently WGAN is mostly explored in traditional GAN and DCGAN, our experiments present a novel and extended use case of Wasserstein distance value function. Due to the limited time and insufficient computing power, our model has not fully converged. Better performance can be achieved by fine-tuning the hyper-parameters such as n_{critic}, learning rate and gradient penalty coefficient.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Con-*

Figure 3. The generated images and their corresponding attention maps along iterations

ference on Computer Vision and Pattern Recognition, pages 6077–6086, 2018.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 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 *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[6] 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 *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] Nguyen Hung. improved-wgan-pytorch, 2018.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[11] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question

answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

[12] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*, 2015.

[13] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[14] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

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

[16] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.

[17] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

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

[19] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018.

[20] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[21] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017.