Text-to-image Implementation with Style-based Attentional GAN

Fei Zheng fz2277 [*1] **Chirong Zhang cz2533** [*1] **Xiaoxi Zhao xz2740** [*1]

Abstract

In this project, we propose a Style-Based Attentional Generative Adversarial Network (SBA-GAN) that allows unsupervised disentanglement of high-level attributes and an attention-driven refinement for text-to-image generation. Borrowing from StyleGan literature and AttnGan structure, this new generator can synthesize details at different regions of image by paying attentions to relevant parts in the text and by interpolating styles into different resolutions in the image. On CUB dataset, our generated 256*256 images have a higher inception score compared to existing methods. Detailed style and attention analysis is also performed by visualizing the different layers of SBA-GAN.

1. Introduction

Text-to-image (T2I) generation is a an important machine learning task and an active research area in both computer vision and natural language processing. It is a fundamental problem in art generation, logo design, interior design and other computer-aided image synthesize. Recent years, significant progress has been made in text-to-image using generative adversarial networks (Goodfellow et al., 2014).

In contrast to general image generation problems, T2I generation is conditioned on texts instead of starting with random noise alone. Different T2I approaches have been made to generate more realistic and text-relevant images, such as in AttentionGAN (Xu et al., 2017), MirrorGAN (Qiao et al., 2019), ObjGAN (Li et al., 2019), DMGAN (Zhu et al., 2019). However, due to entanglement caused by the input latent space which must follow the probability density of the training data, it still remains challenging to generate high-resolution images align with the input text.

To address this issue, we propose a Style-Based Attentional Generative Adversarial Network (SBA-GAN). Motivated by the Style-Based Generator Architecture for Generative Adversarial Networks (GAN)(Karras et al., 2019), which implemented style transfer techniques(Gatys et al., 2016) in the generative network. The overall architecture of the SBA-GAN is illustrated in Figure 1. The two main parts of SBA-GAN is BERT (text encoder) and an attentional style generator. Unlike previous networks which pervasively utilize recurrent neural network (RNN)(Hochreiter et al., 1997) as the text encoder, we proposes to use the pre-trained BERT (Devlin et al., 2018). The other new component is the attentional style generator which adopts the general architecture of AttnGAN(Xu et al., 2017) and incorporate the style module described in (Karras et al., 2019).

To generate a single image, our generator starts from a learned constant input instead of a latent random variable and it takes the sentence-level and word-level vectors features to put "constraints" on the generated images. The conditional loss is calculated to guarantee that the text and image are aligned. Our generator also adjusts the "style" of the image at each convolution layer based on the latent code, which can directly control the strength of image features at different scales. Our code is available at https://github.com/zhengfei0908/SBA-GAN.

We evaluate our methods using the same loss functions in AttnGAN(Xu et al., 2017) and calculate the inception score(Salimans et al., 2016) of generated images. [1] Our inception score on CUB dataset (Wah et al., 2011) is slightly higher than the existing results.

2. Related work

Based on the DCGAN(Radford et al., 2016), Reed(2016) has proposed an architecture to do text-to-image translation. In his algorithm GAN-CLS(Reed et al., 2016), he introduces a third type of input consisting of real images with mismatched text in discriminator in addition to the real/fake inputs. This provides an additional signal to the generator. Also, he explores the disentangling of style and content by inverting the generator for style and it turns out captions alone are not informative for style prediction.

[1]Department of Statistics, Columbia University, New York, USA.

Final Project of COMS4995 Deep Learning, Columbia University, New York.

[1]The calculation of inception score in AttnGAN is the same with StackGAN which uses the deprecated tensorflow1 and the corresponding inception model, we found another way to calculate the inception score which might be different from the AttnGAN

AttnGAN (Xu et al., 2017) first came up with the idea to synthesize fine-grained details at different subregions of the image by paying attention to the relevant words in the natual language desciption. MirrorGAN(Qiao et al., 2019) borrow the idea of circleGAN(Zhu et al., 2017) and generate image from text and text back from image.

With the appearance of StyleGAN(Karras et al., 2019), researchers have proposed new network structure using it and have observed a great increse in FID score(Heusel et al., 2018). LOGAN (Oeldorf & Spanakis, 2019) which proposes a conditional GAN structure with StyleGAN implemented, has successfully generate conditional logos with high FID. But the paper did not discuss the alignment between the image and the logo label and indeed, some generated logos do not seem to be properly generated conditioned on the given label. Our model, tries to address this problem and introduce the similarity measure between text and the generated images.

3. Methods: Style-Based Attentional Generative Adversarial Network

As shown in Figure 1, SBA-GAN integrates StyleGAN(Karras et al., 2019) and AttnGAN(Xu et al., 2017). Unlike common practice, we separate latent variable z from conditional sentence feature vector and use disentangled latent variable w to control the style of images through the style module(AdaIN) during generation process. Technically, SBA-GAN can be divided into two main parts: text encoder and attentional style generator.

3.1. Text Encoder

First, we introduce the text encoder module that transforms the texts to features. We try two types of the text encoders, plain RNN with LSTM units(Hochreiter et al., 1997) and BERT(Devlin et al., 2018). LSTM helps to handle long term dependency problems and is now a standard method in natural language processing while BERT is a great breakthrough in the domain of pre-trained NLP. In previous work in text-to-image field like AttnGAN(Xu et al., 2017) and MirrorGAN(Qiao et al., 2019), BERT is rarely applied to extract information from sentence. In our experiments, we compare the pre-trained BERT with our baseline, LSTM and BERT version outperform the baseline.

In terms of how the text encoders are used, we embed the given text descriptions into both word-level features that work mainly in attention modules and sentence-level features that work mainly in generation process.

Due to the diversity of the text domain or language and in order to enhance the model robustness, we introduce some noises in sentence-level features by using Conditional Augmentation(Zhang et al., 2017). In Figure 1, we use F^{ca} to represent this module.

$$\bar{e}_{ca} = F^{ca}(\bar{e}), \tag{1}$$

where $\bar{e} \in \mathcal{R}^D$, $\bar{e}_{ca} \in \mathcal{R}^{D'}$, D is the dimension of embedding features and D' is the dimension after augmentation.

Unlike common practice in conditional GAN, we only input the augmented sentence feature into next generator rather than concatenate it with random latent variable z.

3.2. Attentional Style Generator

Next we introduce the style-base attentional generator. We adopt the general architecture described in AttnGAN(Xu et al., 2017) because of its outstanding results.

In the mapping network g, we use a 8-layer MLP to disentangle the latent variable z into latent variable w. And as proposed in StyleGAN(Karras et al., 2019), the mapping network consists of 8 fully connected layers without bias.

$$w = MLP(z) \tag{2}$$

In the generation part, we follow the practice of AttnGAN(Xu et al., 2017) and use the structure of StackGAN(Zhang et al., 2017) to generate images with resolution 64^2, 128^2 and 256^2.

During each generation block, attention modules F_i^a mix word-level features $e \in \mathcal{R}^{D\times T}$ and image features $h \in \mathcal{R}^{\hat{D}\times T}$ from previous block and retrieve the most relevant word vectors to generate different sub-regions of the image. Word-level features are first converted into the common semantic space with same dimension as image features,

$$e^{'} = Ue, \tag{3}$$

where $U \in \mathcal{R}^{\hat{D}\times D}$. The result $F_i^a(e,h) \in \mathcal{R}^{\hat{D}\times N}$($N$ is the number of sub-regions) is feature vector of sub-regions, which can be represented as $(c_0, c_1, ..., c_{N-1})$. For the j^{th} sub-region,

$$c_j = \sum_{i=0}^{T-1} \beta_{j,i} e_i^{'}, \beta_{j,i} = \frac{e^{s_{j,i}^{'}}}{\sum_{k=0}^{T-1} e^{s_{j,k}^{'}}}, \tag{4}$$

where $s_{j,k}^{'} = h_j^T e_i^{'}$. $\beta_{j,i}$ indicates the attention weight of the i^{th} word for the j^{th} sub-region of the image.

Style modules inject style from disentangled latent space $\mathcal{W}$ to generation process. Following the practice in StyleGAN(Karras et al., 2019), we first use an affine transformation to specialize w to *style* $y \in \mathcal{R}^{2\times\hat{D}}$, which control the adaptive instance normalization(AdaIN) operation in each generation block. The AdaIN operation is defined as:

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}, \tag{5}$$

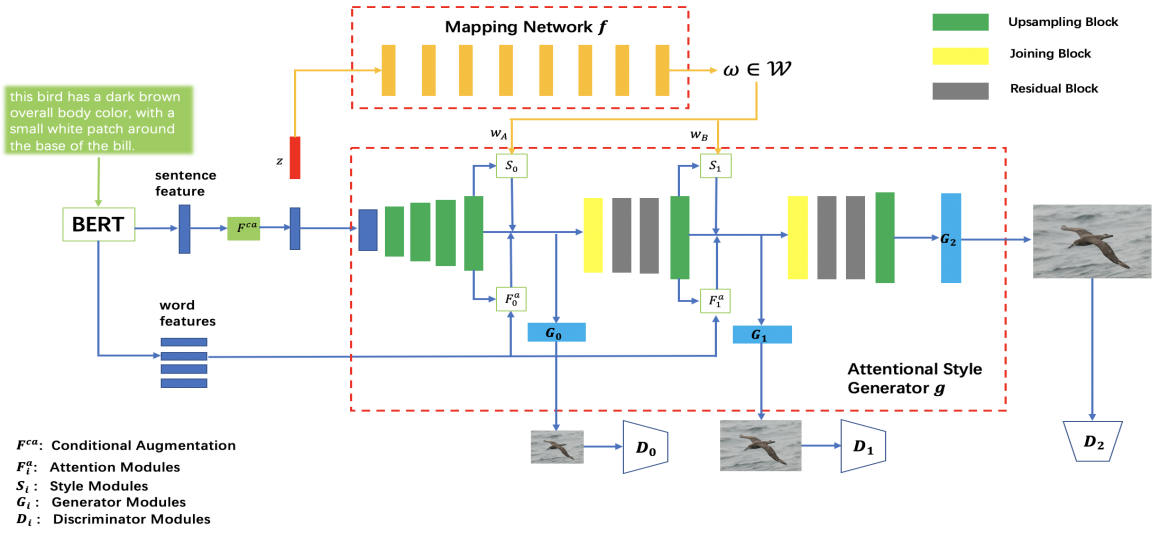

Figure 1. The architecture of proposed SBA-GAN. The figure is adapted mainly from AttnGAN(Xu et al., 2017). The mapping network f maps latent variable z to an intermediate latent space $\mathcal{W}$, as proposed in StyleGAN(Karras et al., 2019). Attentional style generator g generates images with different resolutions from a conditionally augmented sentence feature. During each step, style modules(S_0 and S_1) and attention modules(F_0^a and F_1^a) inject style and word features information respectively.

where each feature map x_i is normalized separately, and then scaled and biased using the corresponding scalar components from *style* y. So the dimension of y is twice the number of features on that layer.

Finally, we combine original image features, attentional image features and style injected image features to generate images at the next stage.

3.3. Objective functions

Following common practice in text-to-image field, we first employ the GAN loss that embodies both conditional and unconditional. During each stage of training process, we train the generator and discriminator alternatively. Specifically, the generator G_i is trained to minimize the generator loss:

$$\mathcal{L}_{G_i} = -\frac{1}{2}\mathrm{E}_{\hat{x}\sim P_{G_i}}[log(D_i(\hat{x})] - \frac{1}{2}\mathrm{E}_{\hat{x}\sim P_{G_i}}[log(D_i(\hat{x},\bar{e})] \quad (6)$$

And the discriminator D_i is trained to minimize the discriminator loss:

$$\begin{aligned}\mathcal{L}_{D_i} = &-\frac{1}{2}\mathrm{E}_{x\sim P_{data}}[log(D_i(x)] - \frac{1}{2}\mathrm{E}_{\hat{x}\sim P_{G_i}}[log(1-D_i(\hat{x})] \\ &-\frac{1}{2}\mathrm{E}_{x\sim P_{data}}[log(D_i(x,\bar{e})] - \frac{1}{2}\mathrm{E}_{\hat{x}\sim P_{G_i}}[log(1-D_i(\hat{x},\bar{e})],\end{aligned} \quad (7)$$

where x is from the true image distribution P_{data} and $\hat{x}$ is from the model distribution P_G. The unconditional loss determines whether the image is real or fake and the conditional loss determines whether the image matches the sentence or not.

Apart from the common GAN loss, we also employ DAMSM loss(Xu et al., 2017). DAMSM was proposed in AttnGAN and achieved good results. We consider this loss for two reasons. One is that we regard this loss as the critical reason that AttnGAN can achieve such good results. The other is that we also need to pre-train our BERT. DAMSM can be used to train BERT individually, which is fit for our limited computation resource.

So the total generator loss and the discriminator loss can be defined as below:

$$\mathcal{L}_G = \sum_{i=0}^{L-1} G_i + \lambda L_{DAMSM} \quad (8)$$

$$\mathcal{L}_D = \sum_{i=0}^{L-1} D_i \quad (9)$$

4. Results

Experimentation is carried out to evaluate the proposed SBA-GAN. We compare our SBA-GAN result with previous state-of-the-art GAN models for text to image synthesis.

Figure 2. Examples of images generated by our best model(BERT with style) conditioned on customized text descriptions. The vertical variate is from the different latent variable z.

4.1. Experiment Setup

4.1.1. Dataset

Same as previous text-to-image methods(Xu et al., 2017; Qiao et al., 2019), our method is evaluated on CUB dataset.(Wah et al., 2011)

The CUB bird dataset contains 8,855 training images and 2,933 test images belonging to 200 categories, each bird image has 10 text description.

4.1.2. Evaluation Criteria

We will evaluate our approach by calculating the inception score(Salimans et al., 2016) of generated images. Inception score is a quantitative evaluation measure to the objectiveness and diversity of generated images.

4.1.3. Implementation Details

For the text encoder part, we freeze most layers but the last layer of pre-trained BERT model *bert-base-uncased*. The embedding dimensions for words features and sentence feature are both 256. The max sentence length is 20. For the image encoder part, we follow the practice of AttnGAN(Xu et al., 2017) and use pre-trained InceptionV3(Szegedy et al., 2015) to extract image features. Considering that we also use a pre-trained BERT in text encoder part, not a RNN model trained from scratch, we make the last three feature layers trainable. This proves to be a good practice. The image feature dimension is 256 and each image is divided into 17×17 sub-regions to calculate image-text matching score(*i.e.*, DAMSM loss). Because of the limited computation source, we employ the same hyperparameters as AttnGAN(Xu et al., 2017) for DAMSM module. The generator and discriminators are trained alternatively with learning rate 0.0002. For the baseline model(only adding styles), we train 600 epoches and for the latter models, we train only 200 epoches.

4.2. Main Results

We conduct four different models based on AttnGAN(Baseline model) and compare them by inception score. "+ style" is the baseline model combined with style(s_0 and s_1 in Figure2). "+ BERT" replaces LSTM text encoder in "+ style" with BERT text encoder. In "+ Style mixing", s_0 and s_1 are generated from different w and is trained on "+ BERT".

The result shows that both BERT and style module help to improve the results.

Baseline	4.36
+ style	5.05
+ BERT	5.12
+ Style mixing	4.75

Table 1. Inception score from different models

4.2.1. Generation Results

Subjective visualization is presented in Figure 3,4. All the figures are generated from our "+BERT" model. It can be seen that the image details can be generated precisely by our model. i.e. text information can be presented in the figure. For example, in the first column, all three plots are able to present blue wings and red belly as described in the text as well as the short information in the third column.

Besides, the precision of our model can be shown in figure 3 as well. Two sentences which only differ in color are used to generate two plots. In the first row, the bird has white belly, gray check patch and yellow crown while the bird in the second row has yellow belly, white check patch and gray crown instead.

Figure 3. Examples of images generated by our best model(BERT with style) conditioned on customized text descriptions. The vertical variate is from the different latent variable z.

4.2.2. Attention Analysis

Attention visualization is presented in Figure3,4 which further proves that our model is able to learn how to generate the plot precisely. Figure3 shows that attention can be allocated to correct positions. For example, attention of none-semantic words like "a", "bird","with" are allocated globally while attention of semantic words like "white" and "yellow" are allocated on the right place.

Figure 4 present how the model learn to generate plot. For example, the attention of word "green" in the stage one and stage two are different. In the first plot, the highlight part is approximated in a large area while in stage 2, it is located at the wings accurately.

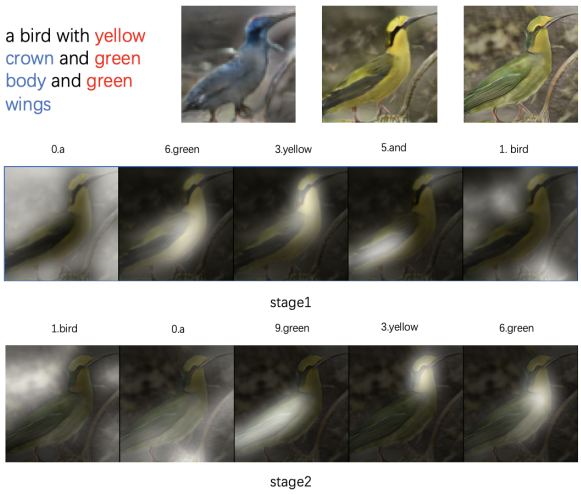

Figure 4. Examples of attention images generated by our best mode conditioned on customized text descriptions.

4.2.3. Style Mixing Analysis

As proposed in StyleGAN(Karras et al., 2019), mixing regularization can encourage the styles to localize and we employ this strategy to make our generator more robust. To be specific, we generate two random latent variables z_A and z_B, pass them through mapping network respectively and get w_A and w_B. When generating an image, we should switch randomly from one to another with a certain probability. This technique prevents the network from assuming that styles are correlated.

For simplicity, we implement style mixing technique by injecting two different random latent variable in order without randomness. Figure 5 shows some styles mixing results. The pattern is not that significant but we can still detect a little. For example, the yellow bird on the third row and the third column has a long tail and a short beak, which is a combination of the original two birds. The worse inception score might be due to the training methods we use.

5. Discussion

5.1. Conclusion

In this project, we proposed a Style-based Attentinal GAN, named SBA-GAN, based on AttnGAN(Xu et al., 2017) and StyleGAN(Karras et al., 2019). We use BERT as our text encoder and employ adaptive instance normalization(*i.e.*,style injection) and style mixing technique. As a result, BERT encoder and adaptive instance normalization achieve inception score 5.05 and 5.12 respectively. Style mixing does not perform well as we thought first because of the computation

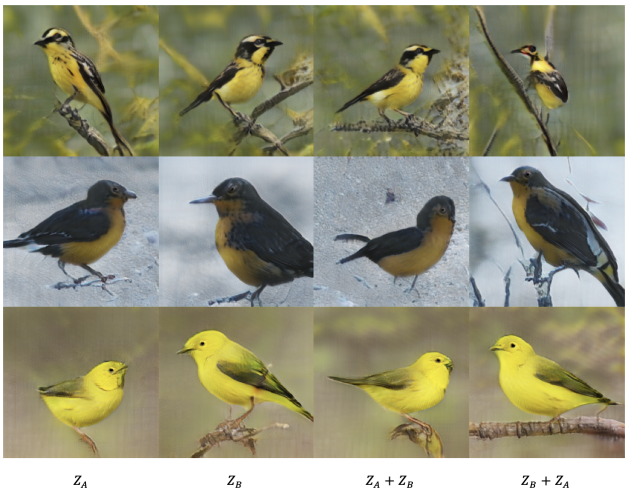

Figure 5. Examples of mixing style results. The first and the second column are generated by single random latent variable Z_A and Z_B respectively. The third column are generated by injecting Z_A first and Z_B second and the fourth column reverse.

resource and training strategy.

5.2. Future Work

We plan our future work in the following directions: experiments on larger datasets, more hyperparameter tuning, higher resolution images, more style mixing and change of the loss functions.

First, we only experiment our results on CUB birds dataset, but for most T2I tasks, COCO dataset (Lin et al., 2014) is also used for experiments. Due to our limited computational power and the huge image amount in COCO dataset, we choose not to do this experiment this time. However, to better compare our methods to existing networks, the experiment on COCO dataset is strongly helpful. Also, the text (condition) we are using now are mainly short sentences(less than 18 words), we can thus try some longer sentences with more detailed descriptions in the future. Since we are adopting the structure of StyleGAN, we can also try to implement a T2I on human faces dataset.

Second, we may want to perform more hyperparameter tuning in the future. We now freeze all the layers except the last pooling layer in BERT and we adopt all the hyperparameters used in AttnGAN instead of tuning our own parameters. If time and computation power permitted, we could search across the combination of hyperparameters, unfreeze more layers in BERT to get a better result.

Third, we start our image generation from 64*64 images and finish the generative process when we reach 256*256. We could start from 16*16 or even smaller images and incorporate more styles in our generative process. Also, we want to finish at images with higher resolution 512*512 or even 1024*1024 as StyleGAN does.

Last, we can try change our loss function to WGAN loss(M.Arjovsky et al., 2017) or add spectral normalization to our generator (Zhang et al., 2018). We now can capture the color information in the text accurately but the shape, position of the birds can be wrong. Further some of the birds in our generated images do not look real (two heads, strange feet etc.). So we can try to change our loss function, alter the number of generator training between each discriminator and train for much longer periods.

References

Devlin, J., Chang, M., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018.

Gatys, L. A., Ecker, A. S., and M.Bethge. Image style transfer using convolutional neural networks. *IEEE*, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio., Y. Generative adversarial nets. *arXiv:1406.2661*, 2014.

Heusel, M., Ramsauer, H., T. Unterthiner, B. N., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv:1706.08500*, 2018.

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

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. *arXiv:1812.04948*, 2019.

Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., and Gao, J. Object-driven text-to-image synthesis via adversarial training. *arXiv:1902.10740*, 2019.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft coco: Common objects in context. *arXiv:1405.0312*, 2014.

M.Arjovsky, Chintala, S., and Bottou, L. Wasserstein gan. *arXiv:1701.07875*, 2017.

Oeldorf, C. and Spanakis, G. Loganv2: Conditional style-based logo generation with generative adversarial networks. *arXiv:1909.09974*, 2019.

Qiao, T., Zhang, J., Xu, D., and Tao, D. Mirrorgan: Learning text-to-image generation by redescription. *arXiv:1903.05854*, 2019.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generatice adversarial networks. *arXiv:1511.06434*, 2016.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text to image synthesis. *arXiv:1605.05396*, 2016.

Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *NIPS*, 2016.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. URL http://arxiv.org/abs/1512.00567.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *arXiv:1711.10485*, 2017.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv:1612.03242*, 2017.

Zhang, H., Goodfellow, I., D.Metaxas, and Odena, A. Self-attention generative adversarial networks. *arXiv:1805.08318*, 2018.

Zhu, J., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv:1703.10593*, 2017.

Zhu, M., Pan, P., Chen, W., and Yang, Y. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. *arXiv:1904.01310*, 2019.