Milestone Report: Text-to-image Implementation of Style 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.

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) such as in AttentionGAN (Xu et al., 2017), MirrorGAN (Qiao et al., 2019), ObjGAN (Li et al., 2019), DMGAN (Zhu et al., 2019). Despite all these impressive results, alignment of generated image with the input text as well as the generation of high-resolution images still remain challenging.

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. 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 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 controlling the strength of image features at different scales.

We will evaluate our approach by calculating the inception score(Salimans et al., 2016) of generated images and the R-precision score(Xu et al., 2017) to check if the images and texts are aligned.

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.

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. Style-Based Attentional Generative Adversarial Network

As shown in Figure 1, SBA-GAN integrates both StyleGAN(Karras et al., 2019) and AttnGAN(Xu et al., 2017) and embodies the attention modules into the progressive generation process. Disentangled by mapping network, la-

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

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

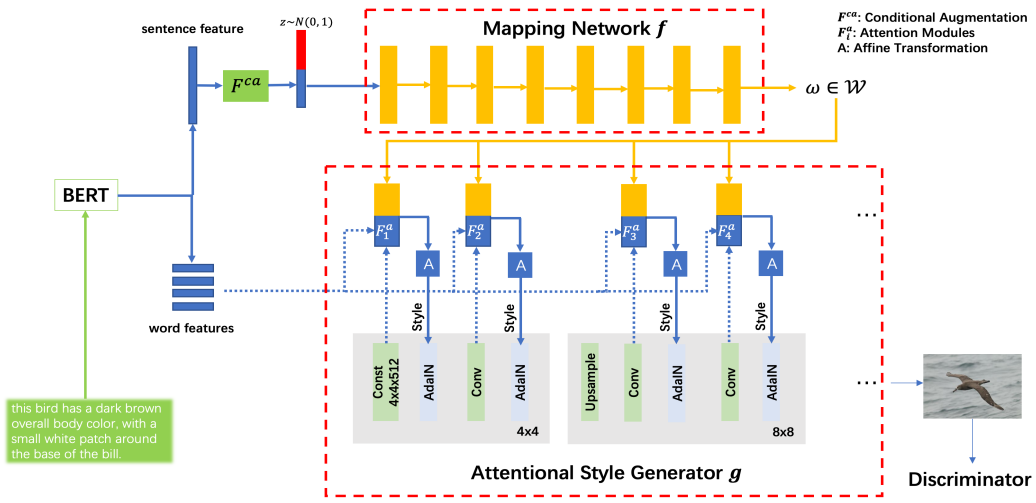

Figure 1. Structure of SBA-GAN. With the combination of ideas from(Karras et al., 2019),(Zhang et al., 2017), we first map the concatenated latent variable with augmented sentence feature to an intermediate latent space $\mathcal{W}$,which then is concatenated with attention from word feature to control the generator through adaptive instance normalization(AdaIN) at each convolution layer. Gaussian noise is added after each convolution(not showed in the figure). Here, A stands for a learned affine transform. The mapping network f consists of 8 layers and the synthesis network g consists of 10 layers – two for each resolution($4^2 - 256^2$). The output of the last layer is converted to RGB using a separate 1×1 convolution, similar to (Karras et al., 2018).

tent code w can control the style of generated images better. By attention modules, text can influence the objects generation better. Technically, SBA-GAN is mainly consist of two modules: **text encoder module**, **attentional style generator module**. We will introduce these modules below.

3.1. Text Encoder Module

First, we introduce the text encoder module that transforms the texts to features. **BERT**(Devlin et al., 2018) is a great breakthrough in the domain of pre-trained natural language processing. We borrow the pre-trained model and embed the given text description into both word-level features that work mainly in attention modules and sentence-level features that work mainly in generation process. In consideration of limited computation resources, we freeze all the layers in BERT.

Due to the diversity of the text domain 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.

Then we concatenate the augmented sentence embedding with random latent code $z \sim N(0,1)$, which is the input of the generator.

3.2. Attentional Style Generator Module

Next we introduce the style-base attentional generator module. We adopt the general structure described in StyleGAN(Karras et al., 2019). In the mapping network part, we use a 8-layer MLP to disentangle the latent code z into latent code w.

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

In the generation part, we follow the practice of StyleGAN and use the structure of ProgressiveGAN(Karras et al., 2018) to generate images from resolution 4^2 to 256^2 step by step. During each step, we mix the information from both previous image feature f and word-level features w by **attention module**, which is proposed by AttnGAN(Xu et al., 2017).

$$Att = FC(\sum_{l=0}^{L-1}(Uw^l)(softmax(f^T(Uw^l)))^T), \tag{3}$$

where $Att \in \mathcal{R}^T$, T is the dimension of attetion vector and $U \in \mathcal{R}^{M\times D}$ is to transform the words to the space with same dimension as image. Full connected layer FC is aimed to transform the re-weighted image feature into the space with same dimension as attention vector.

Then we concatenate the attention code with disentangled latent code w and use a learnable affine transformations, which is denote by A in Figure 1, to specialize it to *styles* $y = (y_s, y_b)$ to control the adaptive instance normalization(AdaIN) operation after each convolution layer of the

generation network. The AdaIN operation is defined as:

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}, \quad (4)$$

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.

For **Discriminator** part. We follow the practice of ProgressiveGAN(Karras et al., 2018) and reverse generator network except attetion module to attain image features.

3.3. Objective functions

Following common practice, we employ the GAN loss that embodies both conditional and unconditional. The loss is defined as

$$\mathcal{L}_{G'} = -\frac{1}{2}\mathrm{E}_{\hat{x}\sim P_G[log(D(\hat{x})]} - \frac{1}{2}\mathrm{E}_{\hat{x}\sim P_G[log(D(\hat{x},\bar{e})]} \quad (5)$$

$$\begin{aligned}\mathcal{L}_D = &-\frac{1}{2}\mathrm{E}_{\hat{x}\sim P_{data}[log(D(\hat{x})]} - \frac{1}{2}\mathrm{E}_{\hat{x}\sim P_G[log(1-D(\hat{x},\bar{e})]} \\ &-\frac{1}{2}\mathrm{E}_{\hat{x}\sim P_{data}[log(D(\hat{x},\bar{e})]} - \frac{1}{2}\mathrm{E}_{\hat{x}\sim P_G[log(1-D(\hat{x},\bar{e})]},\end{aligned} \quad (6)$$

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 include the cosine similarity score in generation part since this can make generated image and text match better. For **text encoder** part, we use BERT to attain sentence embedding $\bar{e}$, as described above. For **image encoder**, we adopt Inception v3(Salimans et al., 2016) to attain image embedding $\bar{c}$. The cosine similarity score can be calculated as $R(\bar{c}, \bar{e}) = (\bar{c}^T\bar{e})/(\|\bar{c}\|\|\bar{e}\|)$.

4. Experiment

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 (Wah et al., 2011) and COCO(Lin et al., 2014) datasets.

CUB dataset consists of 11788 images of 200 categories of birds with about 10 sentences to describe each image. and COCO(Lin et al., 2014) datasets. COCO dataset is a larger dataset with 123,287 images and 886,284 instances.

4.1.2. Evaluation Criteria

We will evaluate our approach by calculating the inception score(Salimans et al., 2016) of generated images and the R-precision score proposed in AttnGAN (Xu et al., 2017) to check if the images and texts are aligned. Inception score is a quantitative evaluation measure to the objectiveness and diversity of generated images but cannot reflect whether the generated image is well conditioned on the given text description.

R-precision is on the other hand, an evaluation metric to measure the similarity of images and the corresponding texts. We calculate the cosine similarities between the image feature and the text feature and count the average accuracy at the different settings: top-r. If the ground truth falls into the top-r candidates, then the image and text are considered aligned, otherwise not.

4.1.3. Implementation Details

4.2. Main Results

We've finished the coding of our main structure, which can be found on github https://github.com/zhengfei0908/COMS4995-StyleGAN

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.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv:1710.10196*, 2018.

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.

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.

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.

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.