HIGH SCHOOL EDITION
Journal of Student Research
Volume 10 Issue 4 (2021)

A Picture is Worth a Thousand Words: Using Cross-Modal Transformers and Variational AutoEncoders to Generate Images from Text

Satyajit Kumar[1] and Ehsan Adeli[#]

[1]Portola High School, Irvine, CA, USA
[#]Advisor

ABSTRACT

Text-to-image generation is one of the most complex problems in deep learning, where the application of Recurrent Neural Networks (RNNs) and Generative Adversarial Networks (GANs) has seen significant success. However, GANs prioritize the sharpness of the image rather than covering all the nuances of the text. Given that Transformers have recently outperformed RNNs and other neural network models in both the text and image spaces, we explored whether Transformer models can perform better in multi-modal tasks such as text-to-image synthesis. Our conclusion based on evaluating five Transformer based models on the MS-COCO dataset showed that Transformers perform better but would need a significant amount of memory and compute resources.

Introduction

Image generation from text captions is a challenging problem that has seen success with a combination of Generative Adversarial Networks (Goodfellow et al., 2014) and language models such as Long Short Term Memory neural networks (Schuster et al., 1997), as shown in Xu et al. (2018) and Qiao et al. (2019). While GANs can generate hyper-realistic images, their results heavily depend on the hyperparameters of the model and, therefore, are exceedingly challenging to train (Kurach et al., 2019) as opposed to Variational AutoEncoders or VAEs (Kingma and Welling, 2014) and Autoregressive models which are easier to train and whose results are easily reproducible. Similarly, in the text captions, the Transformer model proposed in Vaswani et al. (2017) outperforms the more inefficient RNN (Devlin et al., 2019).

Most of the prior methods involved using an RNN, typically an LSTM, to encode the input sentence and provide output hidden states for the GAN to decode and output an image. Using discriminator networks, the images would become sharper and more realistic over the training process. However, the RNN was not able to capture all the nuances of a sentence, and generally failed to represent the input accurately for absurd examples such as: “A combination of a giraffe and a turtle.”

However, recently the DALL-E model introduced in Ramesh et al. (2021), using Transformers, has greatly improved the content of the images produced with a method that combines Transformers and VAEs. They found that the model was able to generate logical samples for absurd text inputs that would force the model to not only produce high-quality images but images that captured the nuances of the sentence. This was a great improvement over GAN and RNN based neural networks, since the Transformer itself is a great improvement over the RNN, and the VAE allowed it to translate output tokens directly into the latent space.

The purpose of a VAE is to encode images in a structured latent space such that different latent vectors can output every possible image. Transformers are known to be able to extrapolate the pairwise relationships

between tokens to generate an output. A question that remains to be answered is that if the output of a Transformer were to be the input to the latent space of a VAE, would it be possible to generate samples of higher image quality without sacrificing memory or generalizability? This work aims to answer that question.

Methods

Like the current state-of-the-art for this task, DALL-E, we utilize the latent space of the VAE to produce images given the output tokens from the transformer. However, each method is distinct and explores a different approach to combining both models. With this said, given the limitations of the compute resources we had at our disposal, some architectures of models were not trainable.

BERT Encoder with VQ-VAE

For the first approach, we used a combination of a pre-trained BERT feature encoder with a VQ-VAE. Vanilla VAEs lack the crispness and quality of images produced by the GAN, so as proposed by Oord et al. (2017), we utilize a VQ-VAE for this task.

First, we calculated a text embedding with a pretrained BERT feature encoder. Since only taking the mean-pooled output would lose information, we mean-pool and concatenate the output of four distinct layers to make up for this. The combined output size of these four layers is a vector of size 4096, which we transform and give to the trained VQ-VAE decoder.

The transformation network, which transforms the text vector of size 4096 into the VQ-VAE latent vector of size 131072, consists of eight fully connected layers. The first seven layers were all 4096 to 4096, while the last layer reformed this into the image feature vector of size 131,072. We trained this network to generate vectors in the VQ-VAE's latent space, given the text encoding. During the evaluation stage, the text was encoded through the BERT feature encoder, transformed through the fully connected layers, and finally decoded with the VQ-VAE decoder to output the final image.

Figure 1. Three examples of images generated by this model architecture.

As seen in Figure 1, this model architecture does not achieve good visual results, and the images are delusional and blurry. The model achieves an inception score of 1.6 ± 0.01 on the COCO dataset, which doesn't compare to the other models, which all achieve an inception score of more than 7.88 ± 0.07. These results show

that this specific architecture is not able to convey the nuances in sentences and translate them over to the VQ-VAE's latent space.

Siamese-BERT with VQ-VAE

In the next method, we alleviate one of the above problems by improving the text encoder network. This method has the same encoder architecture as the previous one; however, it is trained in a siamese-triplet fashion. Since the previous method's text embedding did not have clusters in its embedding space due to the concatenation, the model couldn't correlate similar sentences together. We replace this with a siamese-triplet BERT encoder since it produces SOTA sentence embeddings, as shown in Reimers et al. (2019). This method of training an encoder not only ensures that similar sentences cluster together but also ensures that dissimilar sentences are sufficiently isolated. The output of this BERT encoder is a vector of size 768. We train another fully connected network to transform this into the VQ-VAE latent vector.

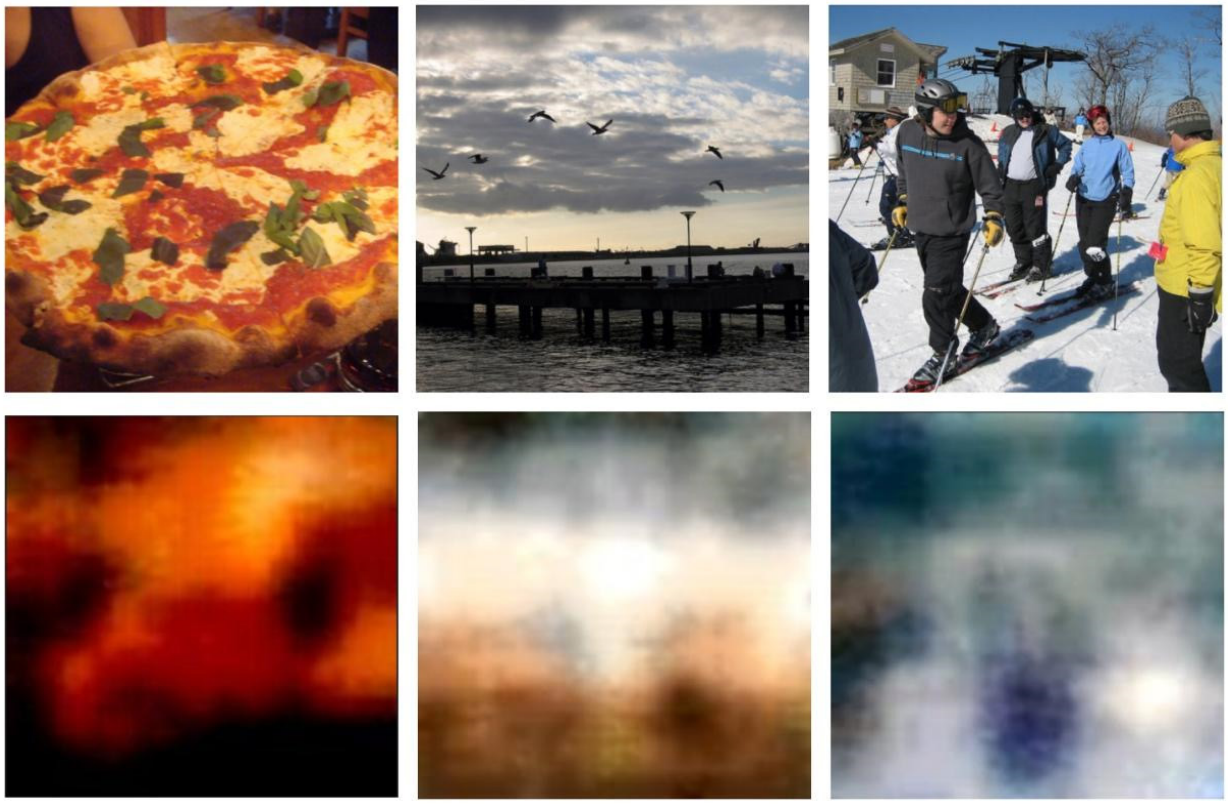

Figure 2. Top row is ground truth, while the bottom row is generated from the evaluation set.

As shown in Figure 2, this model achieves slightly better results than the previous architecture. The inception score also improves considerably, to 3.1 ± 0.01. However, this is still nowhere near the other GAN-based models in previous works. To quantify the reason for these results, we calculate a cosine similarity and Euclidean distance on the embeddings produced by the transformer encoder as well as after the subsequent fully connected layers. We calculate the pairwise similarity based on if the sentences used to create the embeddings are similar or dissimilar.

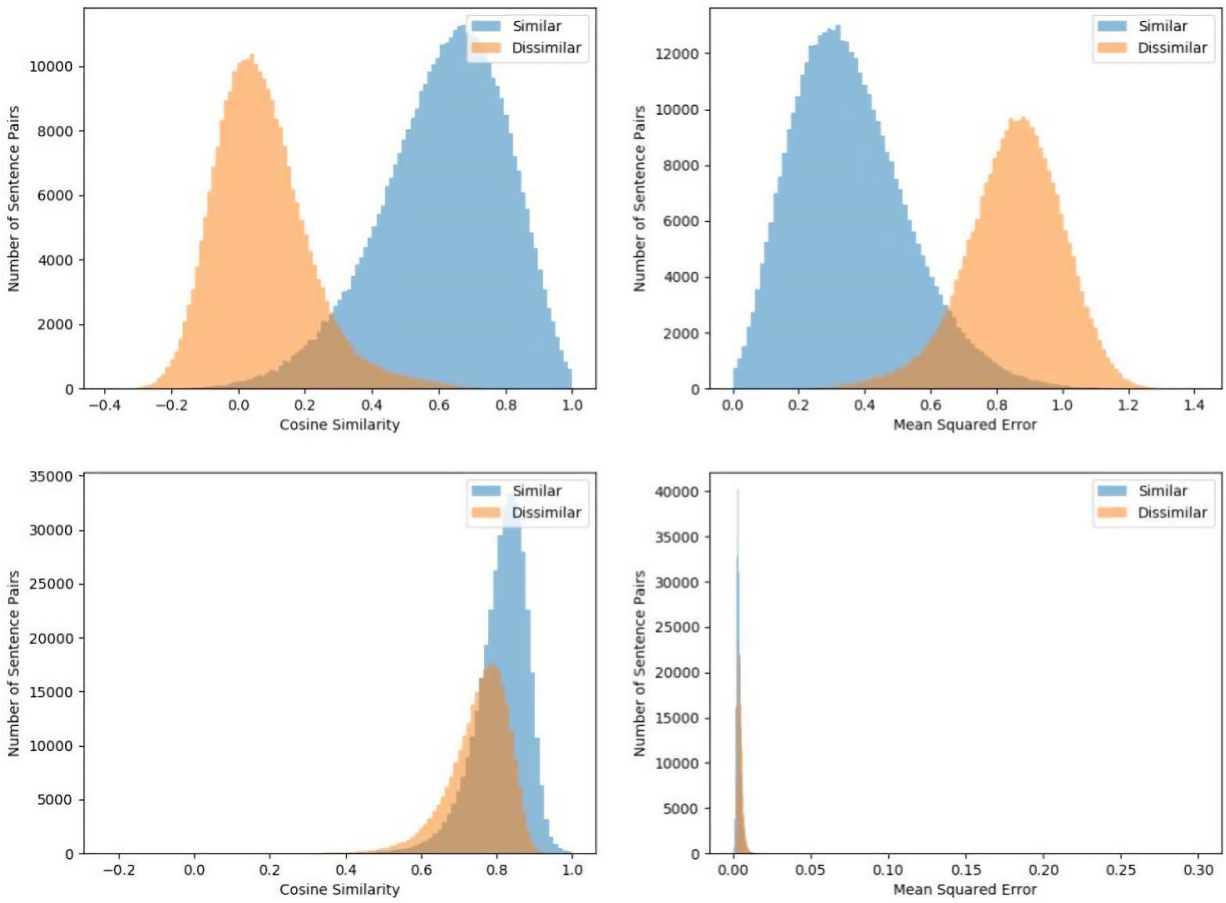

Figure 3. Histograms showing cosine and MSE similarity between pairs of similar and dissimilar sentences after the transformer and fully connected layers. Top row: after transformer, bottom row: after fully connected layers, first column: cosine similarity, second column: MSE.

The top row of Figure 3 showcases the sentence pairs after the siamese-triplet transformer. The similar sentence pairs consist of sentences that captioned the same image in the COCO dataset. The dissimilar sentence pairs consist of sentences that do not have any words in common. The broad distribution of values after the transformer shows that the embedding space is spaced out well. On the other hand, after the fully connected layers (bottom row of Figure 3), the MSE is abnormally low, which shows that the values in the embedding space are all close to one another.

Directly after the transformer encoding, the histograms in the top row of Figure 3 show that similar sentence pairs have higher cosine similarity than dissimilar sentence pairs while having smaller MSE than them. This indicates that the text encoding encodes the sentences well enough that the similarity is translated over to the fully connected layers. However, after the fully connected layers, that information is lost, as shown in the second row of Figure 3. Both the similar and dissimilar pairs have around the same mean similarity, so this indicates that the fully connected layers lose the information given by the text encoding.

This problem was caused by the large upsample ratio in the fully connected layers. If the VQ-VAE's latent space size decreased and the fully connected layers were removed, the information encoded in the text embedding could more efficiently be translated to the VQ-VAE.

Encoder-Decoder Transformer with VQ-VAE 2

To decrease the VQ-VAE's latent space size further without losing image quality, we replace the vanilla VQ-VAE with VQ-VAE 2, proposed in Razavi et al. (2019). This contains a hierarchical latent space that also produces sharper images. We first evaluate this model on the top latent encoding since that contains the geometry information. This latent encoding is of size 16384, which is a great improvement over the previously used VQ-VAE. We split this latent vector into 16 vectors of size 1024, and use these as the tokens on the decoder side of a transformer.

The transformer itself follows a BERT-style architecture for every encoder and decoder layer and has six layers for both. It uses the words in the sentence as the tokens to the encoder and outputs the decoder tokens, which are the 16 splits of the VQ-VAE 2 latent vector.

For the loss function, we applied triplet loss to the output since this would make the outputs have a distance between each other. For the anchor, we used the 16 ground truth vectors, while for the negative example we used a dissimilar sentence.

Transformer Encoder with VQ-VAE 2

We modified the approach above to contain a transformer encoder instead of the full transformer. We take the output of the transformer, then mean pool over time to get a fixed-size vector and calculate triplet loss with the anchor being the ground truth VQ-VAE 2 vector. Since we did not have a transformer decoder in this approach, it was unnecessary to split the latent vector into 16 different parts.

For both above methods, the results were roughly the same. The model failed to converge, and even after using the triplet loss, all the outputs collapsed into one point. We concluded that this was because the VAE and the transformer were incompatible with one another.

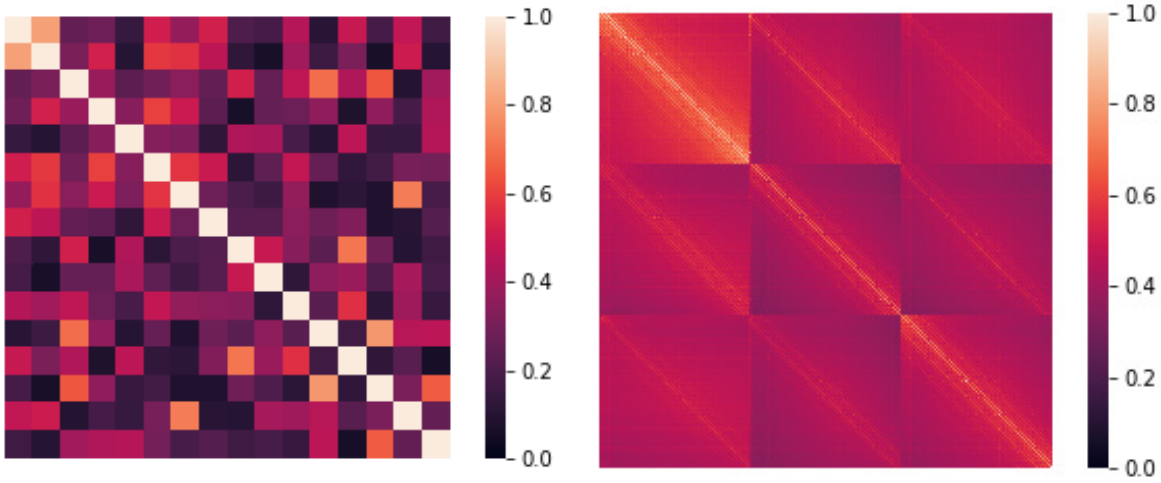

Figure 4: Correlation coefficient matrices of the 16 tokens of the VQ-VAE 2 (left) and the pixel distributions of 32x32 images generated by an Image Transformer (right). These plots clearly show the higher correlations the outputs of a transformer have in comparison to the seemingly random correlations of the VQ-VAE 2. Note that the Image Transformer Correlation Coefficient matrix has nine distinct sections due to there being three channels in the image, meaning that each axis will have three segments.

Image Transformer

The fifth and final method contains a pure transformer, going from the words in a sentence directly to the pixels in an image. However, due to memory constraints, we were forced to use fewer layers and parameters, while switching global self-attention to local self-attention for the decoder. We also downsampled the images to 32x32 since images of size 128x128 would be too large for these memory constraints. This type of model was shown in Parmar et al. (2018) to work on super-resolution. However, in this case, the input would be words, rather than pixels.

This method has several advantages over the previous four methods. Instead of creating one sentence embedding, this model uses the relationships between all words in a sentence to generate the image. This model also didn't have the pairwise connection problem since transformers have been shown to work in text applications as well as image applications.

Despite these advantages, this approach did not work since the ratio of super-resolution for this specific model was too high. For the original Image Transformer model, the ratio of super-resolution was 16, while on the other hand, this text-to-image transformer had a ratio that was on average around 200. Combined with the lack of GPU memory, conservative hyperparameters, and the usage of local self-attention, generating images from this model did not work. However, we believe that given enough compute resources and memory, it would be possible to create a transformer that would go directly from the words in a sentence to pixels in an image.

Results and Discussion

To evaluate the models in this paper and compare them to previous works, we use the Inception Score (Salimans et al. 2016) to evaluate its proficiency on the MS-COCO Dataset (Lin et al., 2014). It is important to note that the Inception score prioritizes the sharpness of the image over the content. This means that approaches using GANs, where sometimes the output image is illogical but sharp, would still score higher than methods that use VAEs like DALL-E, or the methods introduced in this paper.

Table 1. Inception Score of two models in this paper as well as previous models evaluated on the MS-COCO dataset.

Model	**Inception Score on MS-COCO**
BERT Encoder with VQ-VAE (ours)	*1.6 ± 0.01*
Siamese BERT with VQ-VAE (ours)	*3.1 ± 0.01*
GAN-INT-CLS	7.88 ± 0.07
StackGAN	8.45 ± 0.03
DALL-E (Sample Size 128, No Blur)	18.13 ± 0.44
AttnGAN	25.89 ± 0.47
MirrorGAN	26.47 ± 0.41
DM-GAN	30.49 ± 0.57

Another interesting point is that DALL-E, the current state-of-the-art qualitatively, does not achieve the highest inception score even though it generalized in ways the authors did not expect. It was able to adapt to absurd inputs and still produce a logical output, which the GAN-based models struggled with. This could be because the tokens that the transformer used to translate to the latent space of the VAE were fixed discrete

points, so the sharpness of the outputs degraded further, which had a large negative impact on the inception score.

However, the large discrepancy between the transformer-based methods (our methods and DALL-E) suggests that the benefits of using fixed tokens to translate from the Transformer to the VAE outweigh the disadvantages. It would reduce the memory impact and make it easier for the transformer to generalize on sentences while only losing some of the sharpness of the image. Even with that, DALL-E had 12 billion parameters and took around 24 GB of memory, which greatly exceeds the limit of a 16 GB NVIDIA V100 GPU. All in all, although RNNs have their disadvantages in parallelizing and performance, they have the advantage of simply not taking as many resources to train as a transformer does.

Conclusion and Limitations

We have shown five different model architectures to solve the problem of text-to-image synthesis. Unlike previous methods that used models such as GANs and RNNs, we use five novel approaches using Transformers and VAEs to try to solve this problem. We concluded that unless the outputs of the transformer are discretized into tokens and the image is split into segments to conserve memory, combining the output of the transformer directly with the latent space of the VAE either does not converge, is unfeasible due to memory constraints, or is incompatible. However, if the vocabulary of the transformer can be sufficiently enlarged to capture more of the VAE's latent space without increasing memory significantly, we believe that future work would be able to greatly improve the quality of the outputs produced by models of this class while still retaining the generalizing capability of Transformers.

Acknowledgments

I would like to thank Dr. Ehsan Adeli for his guidance and support throughout this project and Prof. Fei-Fei Li for giving me the opportunity to do this work. Without their support, this work would not have been possible. I also would like to thank the Stanford Vision and Learning (SVL) Lab as a whole for giving me the resources necessary to do this project.

References

Cornia, M., Stefanini, M., Baraldi, L., & Cucchiara, R. (2020). Meshed-Memory Transformer for Image Captioning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10575-10584. https://doi.org/10.1109/CVPR42600.2020.01059

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL. https://doi.org/10.18653/v1%2FN19-1423

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., & Bengio, Y. (2014). Generative Adversarial Networks. ArXiv, abs/1406.2661. https://arxiv.org/abs/1406.2661

Kingma, D.P., & Welling, M. (2014). Auto-Encoding Variational Bayes. CoRR, abs/1312.6114. https://arxiv.org/abs/1312.6114

Kurach, K., Lucic, M., Zhai, X., Michalski, M., & Gelly, S. (2019). A Large-Scale Study on Regularization and Normalization in GANs. ICML. http://proceedings.mlr.press/v97/kurach19a.html

Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014). Microsoft COCO: Common Objects in Context. ECCV. https://arxiv.org/abs/1405.0312

Oord, A.V., Vinyals, O., & Kavukcuoglu, K. (2017). Neural Discrete Representation Learning. NIPS. https://arxiv.org/abs/1711.00937

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N.M., Ku, A., & Tran, D. (2018). Image Transformer. ArXiv, abs/1802.05751. https://arxiv.org/abs/1802.05751

Qiao, T., Zhang, J., Xu, D., & Tao, D. (2019). MirrorGAN: Learning Text-To-Image Generation by Redescription. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 1505-1514. https://doi.org/10.1109/CVPR.2019.00160

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-Shot Text-to-Image Generation. ArXiv, abs/2102.12092. https://arxiv.org/abs/2102.12092

Razavi, A., Oord, A.V., & Vinyals, O. (2019). Generating Diverse High-Fidelity Images with VQ-VAE-2. ArXiv, abs/1906.00446. https://arxiv.org/abs/1906.00446

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. ArXiv, abs/1908.10084. https://doi.org/10.18653/v1%2FD19-1410

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved Techniques for Training GANs. NIPS. https://arxiv.org/abs/1606.03498

Schuster, M., & Paliwal, K. (1997). Bidirectional recurrent neural networks. IEEE Trans. Signal Process., 45, 2673-2681. https://ieeexplore.ieee.org/document/650093

Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. ArXiv, abs/1706.03762. https://arxiv.org/abs/1706.03762

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., & He, X. (2018). AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1316-1324. https://doi.org/10.1109/CVPR.2018.00143