# Text Embeddings in Generative Adversarial Text-to-Image Synthesis

**Ali MohammadMehr**
UBC CS Department
alimm@cs.ubc.ca

**Farnoosh Javadi**
UBC CS department
fjavadi@cs.ubc.ca

**Arya Rashtchian**
UBC CS department
aryara@cs.ubc.ca

## Abstract

The usage of a specific type of text embedding in state-of-the-art generative adversarial text-to-image synthesis models, derived us to question the importance of using text embeddings generated for zero-shot image retrieval task in these models. To assess the importance of the embedding in the task of text-to-image synthesis with GANs, we tested Skip Thoughts embedding and Visual Semantic Embedding to generate the text embeddings. Through quantitative and qualitative evaluation, we show that while Skip-Thought seems to impact the performance of a GAN in generating realistic images negatively, a GAN trained with Visual Semantic Embeddings achieves as high performances as the baseline model, strengthening our hypothesis that high performance can be achieved with a simple embedding such as Visual Semantic Embedding as an alternative to baseline model's embedding which performed well in zero-shot image retrieval.

## 1 Introduction

In this paper we are interested in translating visual concepts from characters to pixels or in other words we are interested in creating realistic image from text which is in the form of single-sentence human-written descriptions. Text to image synthesis gained interest in the research community in the past few years, but it is far from being solved. This task requires the generated images to be not only realistic but also semantically consistent, i.e., the generated images should preserve specific object sketches and semantic details described in text.

This challenging problem includes two subproblems: first, learn a text feature representation that captures the important visual details; and second, use these features to synthesize a compelling image that a human might mistake for real. Fortunately, deep learning has enabled enormous progress in both subproblems - natural language representation and image synthesis in the previous several years. Our contribution in this work is related to the second sub problem and it is about assessing the importance of quality of text embedding on the realisticness of generated images. To do so, by using the exact same GAN and encoder as [6] as a baseline, we also tried two different text encoders called Skip-thought vectors [4] and Visual-semantic embeddings [3]. The first one only takes text into account while the encoder is being trained. However, the second one, in addition to caption, it also takes corresponding image into account while the encoder is being trained. Finally, by comparing inception score, FID, and generated images of these three models with different text encoders, it can be seen that GAN with Visual Semantic Embedding achieves similar good results as the baseline model. This means that there is no specific need to use the baseline text embeddings good at zero-shot image retrieval, which are used by the baseline model and recent state-of-the-art models in text-to-image synthesis task, if we care about the quality and diversity of the images generated by GANs.

## 2 Related Work

Recently, it has been shown that Generative Adversarial Networks (GAN) have promising perfor-

mance for generating sharp images [1]. Built upon the generative models, conditional image generation, which is generating images based on some conditioning variables, has also been studied recently. With conditional GAN, Reed et al. [6] successfully generated plausible images for birds and flowers based on text descriptions embedded for the task of zero-shot image retrieval in [5]. Zhang et al.[9] introduce StackGAN, and while doing so, they introduce a Conditioning Augmentation (CA) module on top of the text embeddings introduced by Reed et al.[5] which makes it seem like they believe performance of StackGAN is boosted by the CA module, and therefore they recognize a high value for the embeddings in the task of text-to-image synthesis.

## 3 Data set

The publicly available dataset used in this paper is the Caltech CUB-200 birds dataset [8]. This dataset is one of datasets which are usually used for research on text to image synthesis. The CUB-200 dataset includes 11,788 pictures of 200 types of birds. This dataset include only photos, but no descriptions. Nevertheless, we used the publicly available captions collected by Reed et al. [reference] for these datasets using Amazon Mechanical Turk. Each of the images has five descriptions. They are at least ten words in length, they do not describe the background, and they do not mention the species of the flower or bird.

Figure 1: Caption: This bird is white with blue on its back and has a long pointy beak.

## 4 Model

Our general approach is to train a deep generative adversarial network conditioned on text features encoded by a text encoder. The model consists of two parts, one for encoding captions and one for generating images conditioned on the text embeddings. For the first part we implemented and compared three different text encoding models and for the second part we used a DC-GAN (Deep Convolutional Generative Adversarial Network).

### 4.1 Text Encoder

Since the purpose of our project was to assess the quality of text embeddings on the quality of generated images, we used three different text encoders, that are explained in the following.

#### 4.1.1 Deep Symmetric Structured Joint Embeddings

The baseline model had followed approach of [5] to obtain a visually-discriminative vector representation of text descriptions by using deep convolutional and recurrent text encoders that learn a correspondence function with images. The text classifier induced by the learned correspondence function $f_t$ is trained by optimizing the following loss:

$$\frac{1}{N}\sum_{n=1}^{N}\Delta(y_n, f(v_n)) + \Delta(y_n, f(t_n))$$

where $\Delta$ is the 0-1 loss,$v_n$ are the images,$t_n$ are text descriptions, and $y_n$ are the class labels. Classifiers $f_v$ and$f_t$ are parametrized as follows:

$$f_v(v) = \mathbb{E}_{t\sim\tau(y)}[\phi(v)^T\rho(t)]$$

$$f_t(t) = \mathbb{E}_{v\sim\nu(y)}[\phi(v)^T\rho(t)]$$

where $\phi$ is the image encoder , $\rho$ is the text encoder ,$\tau(y)$is the set of text descriptions of class y and likewise $\nu(y)$ for images.

#### 4.1.2 Skip-Thought Vectors

For the first model that just takes text descriptions into account for generating text embeddings, we used Skip-Though vectors. Because it is shown that those vectors can efficiently capture the semantic of sentences. Skip-Thought model [4] is an encoder decoder that tries to reconstruct the surrounding sentences of an encoded passage. Sentences that share semantic and syntactic properties are thus mapped to similar vector representations. In this model an encoder maps words to a sentence vector and two decoders are used to generate the previous and next sentences of the input. One decoder is used for the next sentences and another

decoder is used for the previous sentences. Their parameters are learned separately.

#### 4.1.3 Visual-Semantic Embeddings

The second model that we used for encoding text descriptions is the model for producing Visual-Semantic embeddings proposed in [3], which encodes a sentence considering its corresponding image besides its text. The model learns a multimodal joint embedding space with images and text as it is shown in figure 2. In this model

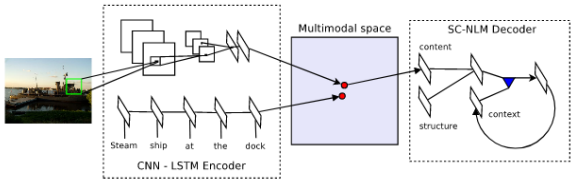


Figure 2: An overview of the Visual Semantic model that learns joint image-sentence embeddings

images and corresponding text descriptions are projected into a common latent space by a CNN and a LSTM repectively, then the model tries to maximize the cosine similarity of each image-text pair, and minimize the cosine similarity of each image-text pair. If $\Theta$ denotes all the learnable parameters in the model, the following pairwise ranking loss is optimized during training.

$$\min_{\Theta}(\sum_{x}\sum_{k} max\{0, \alpha - s(x,v) + s(x,v_k)\}+$$
$$\sum_{v}\sum_{k} max\{0, \alpha - s(v,x) + s(v,x_k)\})$$

Where $s$ is the score function and $s(x,v) = x.v$, $v_k$ is a non-descriptive sentence for image embedding $x$, and vice-versa for $x_k$.

### 4.2 Image Generator

For generating images we used deep generative adversarial networks.

#### 4.2.1 GAN

GANs consist of a generator G and a discriminator D that compete in a two-player minimax game: The discriminator tries to distinguish real training data from synthetic images, and the generator tries to fool the discriminator. Concretely, D and G play the following game on V(D,G):

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_data}(\log D(x))+$$
$$\mathbb{E}_{x\sim p_z(z)}(log(1 - D(G(z))))$$

Where z comes from a Gaussian noise.

### 4.3 Final model

An overview of the final model is shown in 3. At first, the text embedding $\rho(t)$ , encoded by an encoder $\rho$ that can be either the baseline text encoder, or the skip-thought model or the visual semantic model, is concatenated with a sample noise z from the noise prior $z \sim \mathcal{N}(0,1)$ to give style to the generated image, and makes the results more diverse. Then the concatenation result is fed forward to generator G to generate an image $\hat{x} \leftarrow G(z, \rho(t))$. In the discriminator D, encoded image by several layers of 2D convolution is concatenated with the replicated text encoding $\rho(t)$ and the result is fed into another convolution followed by rectification to give the image a score $D(\hat{x}, \rho(t))$. This score, which is in range [0,1], determines how real the input image is.

## 5 Experiments

To compare the two text-to-image models that we devised with each other and with the baseline model, we had to train three different models. While training the models, we observed the mean output value of discriminator for real images; the mean output value of discriminator for fake images; the mean output value of discriminator for all of the real, fake, or unmatched images; and the mean output value of discriminator for fake images before updating generator's parameters to monitor the training procedure.

### 5.1 GAN Training Procedure

For training, we need matched and unmatched images and captions from CUB data set. In order to make changing the caption embedding easier, for each matching image and caption, we added an unmatched image to the sample, so each sample of the data includes a caption and an image that match and a unmatched image which is not matched with the image.

To train the three models, for each batch of samples which includes 64 real images, their matching captions, and a non-matching image for each caption, we first propagate the real image and it's caption through the discriminator and configure the labels so that the discriminator has a lower loss if it predicts "real." Then for each sample caption in the batch we generate a fake image by feeding the

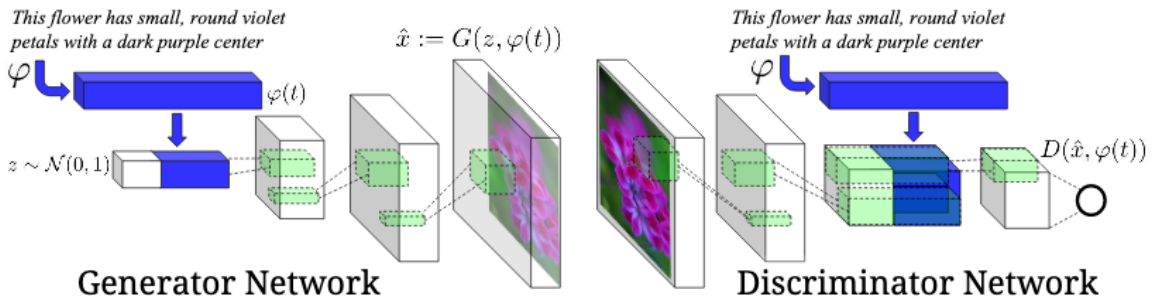


Figure 3: The architecture of the model. Text encoding $\rho(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and concatenated with image feature maps for further stages of convolutional processing.

generator with the caption embeddings and train the discriminator to output "fake" to them. After that, we feed the non-matching image and caption into the discriminator and train it to predict "fake" for this pair. In the end, we feed the caption embeddings of the samples in the batch to the generator and feed the generated image to discriminator. In this step since we will be updating the generator parameters, we configure the labels to "real," so that the generator is updated to fool the discriminator. After this final step, we take a new batch and start the procedure from beginning.

### 5.2 Training the Models

First, we trained the baseline model [6]. After 150 epochs of training the baseline model, the results of [6] are reproduced. To train the GAN with Skip Thoughts embedding, we used a pre-trained Skip Thoughts model to train the GAN for 150 epochs, updating only the GAN parameters. After 150 epochs, we refined both GAN parameters and Skip Thoughts parameters for 20 epochs. After 20 epochs, the model seemed to have converged based on the mean output values of discriminator. To train the GAN with Visual Semantic Embedding, we used a pre-trained VSE model to train the GAN for 150 epochs, updating only parameters of the GAN, similar to what we did for training the GAN with Skip Thoughts embedding. After 150 epochs, we refined both the GAN parameters and the VSE model parameters for 30 epochs. The GAN did not seem to have converged at the 20th epoch, and thus the refining of VSE was done for 10 additional epochs resulting in a total of 30 epochs of refining for GAN with VSE.

## 6 Evaluation and Results

To evaluate our models, and therefore to be able to analyze the affect of changing the text embedding to Skip Thought and Visual Semantic Embedding on the performance of the GAN, we used two different score metrics to do quantitative comparison. We also did Qualitative comparison of the three models by manually comparing the pictures generated by the models.

### 6.1 Inception Score

Inception score is a measure of diversity, measuring, on average, how different is the score distribution for a generated image from the overall class balance. In inception score, we desire that each image be close to only one class, but at the same time, we desire that the distribution of all the images be uniform across all the classes[7]. The result of computing inception score on the images generated by the three models can be seen in Table 1. Based on the scores, we can see that GAN with VSE has higher inception score than the baseline model while GAN with Skip Thoughts embedding has not achieved a high inception score.

### 6.2 Fréchet Inception Distance

FID is a measure of similarity between two datasets of images. It was shown to correlate well with human judgment of visual quality and is most often used to evaluate the quality of samples of Generative Adversarial Networks. FID is calculated by computing the Fréchet distance between two Gaussian distributions fitted to feature representations of the Inception network[2]. The result of computing FID on the three models can be seen in Table 1.

| Model | Inception Score | Fréchet Inception Distance |
|---|---|---|
| baseline GAN [6] | $2.65 \pm 0.03$ | 53.4 |
| GAN with Skip Thoughts Embeddings | $2.51 \pm 0.03$ | 85.7 |
| GAN with Visual Semantic Embeddings | $2.87 \pm 0.03$ | 52.3 |

Table 1: The results of quantitative evaluation of the three trained models. The results show that GAN with VSE can achieve almost as good results as the baseline, showing that text embedding trained for zero-shot image retrieval task is not a necessary ingredient to get good results in text-to-image synthesis task using GANs.

### 6.3 Qualitative Comparison

To compare the three models qualitatively, we have drawn some images generated by the three models which can be seen in Figure 4.

### 6.4 Results

Overall, by comparing inception score, FID, and generated images from the three models, it can be seen that GAN with Visual Semantic Embedding achieves similar good results as the baseline model. This means that there is no specific need to use the baseline text embeddings good at zero-shot image retrieval, which are used by the baseline model and recent state-of-the-art models in text-to-image synthesis task, if we care about the quality and diversity of the images generated by GANs.

## 7 Discussion

In the process of this project, for the first few weeks we all read a few related papers in the task of text-to-image synthesis. After that, while all the group members were mostly following all parts of the project, Arya started to look at the code related to CUB data set and its pre-processing and codes for computing inception score and FID, Ali started to look at existing codes for GANs and how they are trained, and Farnoosh looked at the code for Skip Thought and Visual Semantic Embedding and how they work. After gathering all the necessary code, we all collaborated in combining the codes, making them work on the server, monitoring the training and evaluating the models, during which every member mostly participated in the area they first looked at.

As for the lessons learned, the most important code-related lesson was to use Visdom as an alternative to MatPlotLib. Visdom is a visualization tool that can produce live data visualizations in a web-based environment. It is produced by Facebook and it is a very fun and awesome method to visualize graphs while training and makes monitoring the training very easy.

During the project we learned a lot about different types of text embeddings and the process of their training. We also learned the procedure of training GANs which is quite different from models we had seen in the assignments. In addition to these, we realized how sensitive GANs are to hyper-parameters making them work only with the right hyper-parameters which made us realize that flexibility of a network with respect to hyper-parameters is an important factor in comparing networks which is not usually evaluated by the papers introducing new networks for new tasks.

## References


[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[2] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6626–6637. Curran Associates, Inc., 2017.

[3] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.

[4] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *CoRR*, abs/1506.06726, 2015.

[5] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. pages 49–58, 2016.

[6] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text


Figure 4: Sample images generated by the three models.

to image synthesis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1060–1069. JMLR.org, 2016.

[7] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016.

[8] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[9] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. 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.