Zero-Shot Image Generation by Distilling Concepts from Multiple Captions

K J Joseph [1] **Arghya Pal** [1] **Vineeth N Balasubramanian** [1]

Abstract

Existing methods for generating an image from its description, use one single caption to generate a plausible image. A single caption by itself, would not be able to capture the variety of concepts that might be present in the image. We propose a generative model that will iteratively improve the concepts, and thereby the quality of the generated image by making use of multiple captions about a single image. This is achieved by ensuring 'cross-caption cycle consistency' between the captions and the intermediate image representations. We report quantitative and qualitative results to bring out the efficacy of the proposed approach in zero-shot image generations, where images are generated from descriptions of novel classes that are not seen during training.

1. Introduction

'A picture is worth a thousand words.' The information that is conveyed by the visual perception of an image is difficult to be captured by a single textual description (caption) of the image. In order to alleviate this semantic gap, standard image captioning datasets like MS COCO (Lin et al., 2014) and Pascal Sentences, (Rashtchian et al., 2010) provide five captions per image. In most cases, these captions contain complementary information.

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) has proven its mettle in synthesizing photo-realistic images. Recent text-to-image synthesis methods like (Reed et al., 2016b; Zhang et al., 2017b;a; Tao Xu, 2018) condition the GAN with an encoded representation of a caption. For generating an image, information form only one caption is used in these methods. They miss to make use of the variety in other captions about the same image. We hypothesize that we can improve the quality of the image by distilling concepts from multiple captions about the same image. It is analogous to having a painter update a canvas each time, after reading different description of the end image that he/she is painting.

In this work, we propose a deep generative model, DistillGAN, which iteratively updates its generated image features by taking into account different captions at each step. We ensure that the captions and the generated image features holds a cycle consistency. Concretely, let $F_i : \mathbf{t}_i \rightarrow \mathbf{I}_i$ and $G_i : \mathbf{I}_i \rightarrow \mathbf{t}_i$; where $\mathbf{t}$ represents a caption, $\mathbf{I}$ represents an image, F_i transforms the i^{th} caption to the corresponding image representation and G_i does the opposite. A network that is consistent with two captions is trained such that $G_2 \circ F_2 \circ G_1 \circ F_1(\mathbf{t}) \approx \mathbf{t}$. This model takes inspiration from Cycle-GAN (Zhu et al., 2017) which has demonstrated superior performance in unpaired image to image translation. The way in which cycle consistency helps to distill more information from captions in explained in section 3.1.

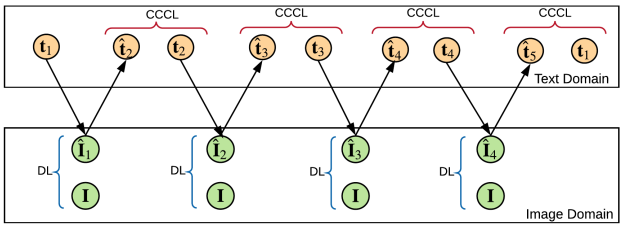
Our experiments on Caltech-UCSD Birds dataset (CUB) (Welinder et al., 2010) and Oxford-102 Flowers dataset (Nilsback & Zisserman, 2008) reveal that DistillGAN is able to generate plausible images even for classes with no training labels. Further, once the model has been trained, it can be used to generate new images which vary in pose and background, still consistent with the set of captions. Such zero-shot generation results and comparison with the other state-of-the-art methods is reported in Section 4.

2. Related Work

Text to Image Synthesis has received lot of interest in the recent years. Reed *et al.*(Reed et al., 2016b) used Conditional GANs (Mirza & Osindero, 2014) to generate 64×64 images from captions. This was the first end-to-end differentiable architecture from character level to pixel level generation. 'Deep Symmetric Structured Joint Embeddings' (Reed et al., 2016a) were used to generate the embeddings for the captions and in-turn was used for conditioning the GAN. StackGAN (Zhang et al., 2017b) and its follow up work, StackGAN++ (Zhang et al., 2017a) increased the spatial resolution of the generated image by adopting a two stage process. Similar to ours, their generations are also zero-shot. Hence we compare our results against them is Section 4. It is worth noting that all the methods so far

[1]Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, India. Correspondence to: Joseph K J <cs17mtech01001@iith.ac.in>.

Towards Learning with Limited Labels Workshop, 35^{th} *International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018.



$Legend$: $\{\mathbf{t}_i\}$: True Captions; $\{\hat{\mathbf{t}}_i\}$: Generated Captions;
$\{\hat{\mathbf{I}}_i\}$: Generated Images; $\mathbf{I}$: True Image;
CCCL: Cross-Caption Consistency Loss;DL: Discriminator Loss.

Figure 1. Figure shows how DistillGAN maintain cycle consistency across four captions $(\mathbf{t}_1, \cdots, \mathbf{t}_4)$. A generator G converts $\mathbf{t}_i$ to an image $\hat{\mathbf{I}}_i$. Discriminator at each step forces $\hat{\mathbf{I}}_i$ to be realistic. A cross-caption consistency network converts $\hat{\mathbf{I}}_i$ back to a caption $(\hat{\mathbf{t}}_{i+1})$ and forces it to be close to $\mathbf{t}_{i+1}$. In the last step, $\hat{\mathbf{t}}_5$ is ensured to be consistent with the initial caption $\mathbf{t}_1$, hence completing a cycle. Meanwhile, the concepts in $\hat{\mathbf{I}}_i$ is incrementally improved.

in the literature uses only one caption to generate images, while our method iteratively improves the image quality by distilling concepts from multiple captions.

Interestingly, there is a recent work by Sharma *et al.*(Sharma et al., 2018) which improves the image quality by taking into account the dialogues (questions and answers) about an image along with the captions. They generate a dialogue embedding for the whole dialogue about an image and condition the generations on this embedding along with the caption embedding. Hence, all the dialogue information is used just once, in their setting, while we update the image features iteratively based on each caption. This clearly separates our work from theirs.

Cycle Consistent adversarial networks (CycleGAN (Zhu et al., 2017)) has shown very impressive results in unpaired image to image translation. CycleGAN learns two mappings, $G : A \rightarrow B$ and $F : B \rightarrow A$ using two generators G and F. A and B can be unpaired images from any two domains. For learning the mapping, they introduce a cycle-consistency loss, $F(G(A)) \approx A$ and $G(F(B)) \approx B$. Standard discriminator loss ensures that the images generated by G and F are plausible. Several methods like (Yi et al., 2017; Zhang et al., 2018; Liu et al., 2017; Kim et al., 2017) with the similar concept has been proposed in literature. All of them consider only pairwise cycle consistency. Our proposed approach imposes a transitive consistency across multiple captions. We call this cross-caption cycle consistency and is explained in Section 3.1.

3. Distilling Concepts from Multiple Captions

3.1. Ensuring Cross-Caption Consistency

DistillGAN generates an image, starting from noise and a set of captions, $C = \{\mathbf{t}_1, \cdots, \mathbf{t}_N\}$. Figure 1 gives a simplified overview of the distilling process and Figure 2 explains the architecture of DistillGAN. Let us take an example of synthesizing an image by distilling information from four captions. In the first iteration, a generator network (G_1) takes noise and the first caption, $\mathbf{t}_1$, as its input, to generate an image, $\hat{\mathbf{I}}_1$, which is passed to the discriminator network (D_1), which verifies whether it is real or not. As in a usual GAN setup, generator tries to create better looking images so that it can fool the discriminator. DistillGAN passes the generated image features to a 'Cross-Caption Consistency Network' (CCCN) which will learn to generate a caption for the image. While training, the Cross-Caption Consistency Loss ensures that the generated caption should be similar to the second caption, $\mathbf{t}_2$.

Next, features used to generate $\hat{\mathbf{I}}_1$ and $\mathbf{t}_2$ is fed to generator (G_2) to generate $\hat{\mathbf{I}}_2$. While D_2 urges G_2 to make $\hat{\mathbf{I}}_2$ similar to the real image $\mathbf{I}$, the CCCN ensures that the learned image representation is consistent for generating the next caption in sequence. This repeats until when $\hat{\mathbf{I}}_4$ gets generated, where the CCCN will ensure that the generated caption is similar to the first caption, $\mathbf{t}_1$. Hence we complete a cycle: $\mathbf{t}_1 \rightarrow \mathbf{t}_2 \rightarrow \mathbf{t}_3 \rightarrow \mathbf{t}_4 \rightarrow \mathbf{t}_1$, while generating $\hat{\mathbf{I}}_1 \cdots \hat{\mathbf{I}}_4$ in-between. $\hat{\mathbf{I}}_4$ contains the concepts from all the captions and hence is much richer in quality.

3.2. End to End Architecture

We seek to model the recurrent conditional distribution:

$$P_i(\mathbf{I}_i|\mathbf{I}_{i-1}, \mathbf{t}_i); \text{where } i = 0, \cdots, N, 0 \qquad (1)$$

In the above equation, the image, $\mathbf{I}_i$ is a refinement on the top of $\mathbf{I}_{i-1}$ using text, $\mathbf{t}_i$. However, the last N^{th} image is conditioned on the first caption (hence $i = 0, \cdots, N, 0$) to maintain our proposed cycle-consistency. We restricted our model to 3 captions per image and observed a exemplary improvement with respect to other state-of-the-art text to image synthesis. All qualitative and quantitative demonstrations is reported in Section 4.

However, conditioning equation 1 on text description doesn't go well as human written text descriptions are very abstract and generally loose fine-grained details. So, we first encode the text description $\mathbf{t}_i$ through a pre-trained non-linear transformation (Reed et al., 2016a) to get an encoded vector representation $\phi(\mathbf{t}_i)$. This modifies equation 1 to the following:

$$P_i(\mathbf{I}_i|\mathbf{I}_{i-1}, \phi(\mathbf{t}_i)); \text{where } i = 0, \cdots, N, 0 \qquad (2)$$

To distill information among i captions (i.e. $i = 0, \cdots, N, 0$) in equation 2, we use a Generative Adversarial Framework (GAN) where we have i-number of GANs in a serial manner. Each generator G_i is conditioned on a set of convolutional layers, called Block B_i, which is a nonlinear transformation of backbone features, b_{i-1} (that are shared

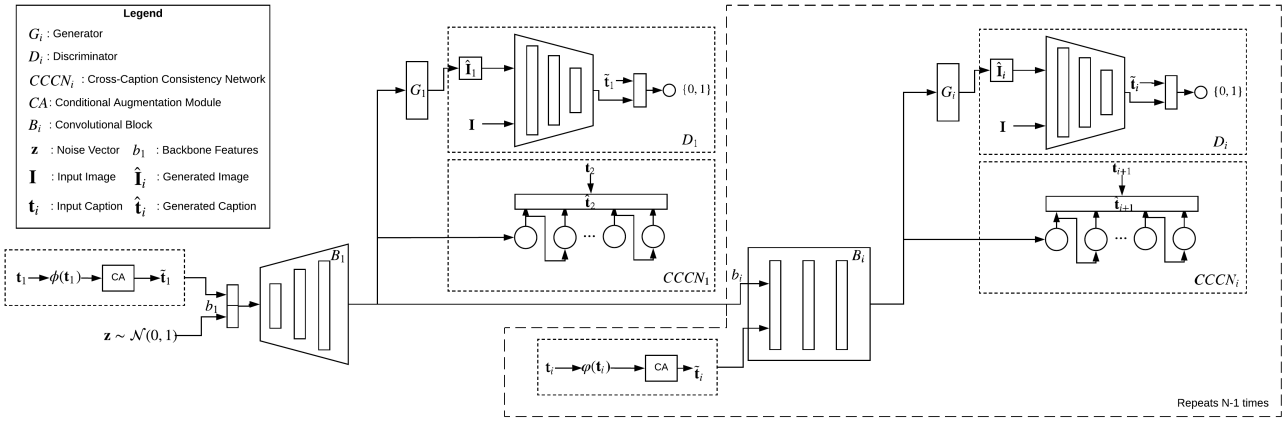


Figure 2. Architecture of DistillGAN.

across multiple GANs) of previous GAN G_{i-1} and text embedding of i^{th} caption $\phi(\mathbf{t}_i)$, i.e. $G_i(z_i|B_i(b_{i-1}, \phi(\mathbf{t}_i)))$. To start off, $b_0 \sim \mathcal{N}(0, 1)$.

Figure 2 shows the end to end architecture of DistillGAN. Training a network with multiple generators (G_i) and discriminators (D_i) can be very tricky. Inspired by the architecture in (Zhang et al., 2017a), we maintain a common shared backbone, from which multiple generators branch off. This backbone is implemented as a set of residual convolutional layers.

We will walk through the architecture. An embedding for the first caption, $\phi(\mathbf{t}_1)$ is generated using SJE (Reed et al., 2016a). $\phi(\mathbf{t}_1)$ is a high dimensional vector. We use Conditional Augmentation (Zhang et al., 2017b) to transform it to a lower dimensional conditioning latent variable, $\tilde{\mathbf{t}}_1$. A noise vector is sampled from a standard normal distribution. Both these vectors are concatenated together and is passed through a set of up-sampling convolutional layers to transform it into a tensor of size $64 \times 64 \times 128$. This is done in Convolutional Block, B_1 in Figure 2.

The first generator, G_1, immediately branches off from the backbone. It uses three 3×3 kernels to generate an image $(\hat{\mathbf{I}}_1)$ from the backbone features, shared with it. $\hat{\mathbf{I}}_1$ is passed to the discriminator, D_1. All discriminators are implemented as a set of sub-sampling convolutional layers which ends in a sigmoid function which generates probability of the image being fake / real. The same set of backbone features are passed to the Cross-Caption Consistency Network (CCCN), which uses an LSTM (explained in Section 3.2.1) to generate caption for the image.

In the mainline, the $64 \times 64 \times 128$ feature maps are transformed into $128 \times 128 \times 64$ feature maps using the Block B_i. The first layer in B_i spatially concatenates the latent representation of the next caption with the incoming feature maps. Further layers in B_i increases the spatial resolution. These backbone features are then send to the next set of Generator, Discriminator and CCCN. This is how the backbone network grows. The spatial size of the feature map progressively increases and the CCCN ensures that concepts from the new captions provided in each step will be imbibed into the features for generating the images. This is novel to our proposed approach.

The key components of the architecture is explained below:

3.2.1. Cross-Caption Consistency Network (CCCN)

CCCN is modeled as an LSTM which generates one word at each time-step conditioned on a context vector (derived by attending to specific regions of the image), the hidden state and the previously generated word. CCCN takes as input the same set of backbone features that the generator consumes. It is then pooled to reduce the spatial dimension. Regions of these feature maps are aggregated into a single context vector by learning to attend to these feature maps similar to the method proposed by (Xu et al., 2015). Each word is encoded as its one-hot representation.

There is one CCCN block per generator. CCCN is trained by minimizing the cross-entropy loss between each of the generated words and words in the true caption. The true caption for Stage i is $(i + 1)^{th}$ caption, and finally the first caption, as is explained in Section 3.1. The loss of each of the CCCN block is aggregated and back-propagated together.

3.2.2. Discriminator

The discriminator is implemented as a set of down-sampling convolutional layers. The text features $(\tilde{\mathbf{t}}_i)$ that were used to condition the generator is spatially replicated and further fused with a 1×1 convolution. Finally the features are reduced to a single neuron to compute the final score of D. Its loss function is calculated as:

$$\mathcal{L}_{D_i} = \mathbb{E}_{\mathbf{I}_i \sim p_{data}}[\log D_i(\mathbf{I}_i)] + \mathbb{E}_{\mathbf{s}_i \sim p_{G_i}}[\log(1 - D_i(\mathbf{s}_i))]$$

p_{data} is the original data distribution and p_{G_i} is the distribution of the corresponding generator network. The multiple discriminators are trained in parallel.

Figure 3. The top row shows three zero-shot generations and the corresponding captions consumed in the process. The first two images belong to Indigo Bunting, Tree Sparrow class of CUB dataset (Welinder et al., 2010) and the last image belongs to Peruvian Lily class of Flowers dataset (Nilsback & Zisserman, 2008). The bottom row shows some random samples of generated images. All the images are generated for classes that are unseen while training the model. (Kindly zoom in to see the detailing in the images.)

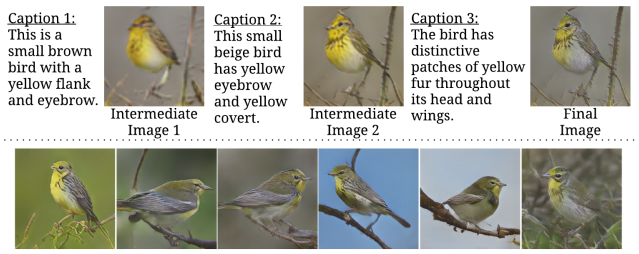


Figure 4. The top row shows how DistillGAN improves the quality of the image at each stage. Intermediate results and the corresponding captions consumed is shown. The bottom row shows generated birds of the same class, but with varying pose and background. These are generated by keeping the captions the same and varying the noise vector used to condition the GAN.

3.2.3. Generator

Multiple generators are trained together by minimizing the following loss function:

$$\mathcal{L}_G = \sum_{i=1}^{N} \mathcal{L}_{G_i}, \text{ where } \mathcal{L}_{G_i} = \mathbb{E}_{\mathbf{s}_i \sim p_{G_i}}[\log(1 - D_i(\mathbf{s}_i))] + \lambda D_{KL}(\mathcal{N}(\mu(\phi(\mathbf{t}_i), \Sigma(\mathbf{t}_i))||\mathcal{N}(0, 1))$$

The first term in $\mathcal{L}_{G_i}$ is the standard minimization term in the GAN framework which pushes the generator to generate better quality images. The D_{KL} term is used to learn the parameters of $\mu(\phi(\mathbf{t}_i))$ and $\Sigma(\mathbf{t}_i)$ of the Conditional Augmentation framework (Zhang et al., 2017b). It is learned very similar to the re-parameterization trick in VAEs (Kingma & Welling, 2013). λ is a regularization parameter, whose value we set to 1 for the experiments.

4. Experiments and Results

4.1. Datasets

The evaluate DistillGAN on CUB (Welinder et al., 2010) and Oxford-102 flowers dataset (Nilsback & Zisserman, 2008). CUB contains 200 bird species with 11,788 images and Oxford-102 dataset contains 8189 flower images of 102 classes. We use five captions per image collected by Akata *et al.*(Akata et al., 2015) to train the network. We split CUB and Oxford dataset into class disjoint training and test set.

Dataset	GAN-INT-CLS	GAWWN	StackGAN	StackGAN++	DistillGAN
CUB	2.88 ± .04	3.62 ± .07	3.70 ± .04	3.82 ± .06	**3.92 ± .11**
Oxford-102	2.66 ± .03	-	3.20 ± .01	-	**3.41 ± .17**

Table 1. Comparison with other text-to-image synthesis methods.

4.2. Results

We validate the efficacy of DistillGAN by comparing it with GAN-INT-CLS (Reed et al., 2016b), GAWWN (Reed et al., 2016a), StackGAN (Zhang et al., 2017b) and StackGAN++ (Zhang et al., 2017a). Inception Score (Salimans et al., 2016) is used as the evaluation metric. StackGAN, StackGAN++ and DistillGAN is trained for same number of epochs (1200) for fair comparison. We significantly improve the inception score when compared to previous methods by distilling information from multiple captions on CUB and Oxford Flowers dataset. Quantitative results are shown in Table 1.

Figure 3 shows some of the images generated by DistillGAN along with the captions used to generate them in the first row. Some random images generated from the captions from the CUB dataset is also shown. Note that none of the classes of the generated image has been used to train the model. Hence these are Zero-Shot generations. The progressive improvement of image quality in the different stages of DistillGAN is captured in Figure 4 top row. By changing the noise vector used to condition the first GAN we are able to generate different images that are consistent with all the captions. This is captured in the bottom row of Figure 4.

5. Conclusion

DistillGAN provides a framework for image synthesis by distilling concepts from multiple captions about a single image. One immediate enhancement to the proposed approach would be to include an end to end attention mechanism to the architecture. This might help to improve the quality of generation. We will explore this in a future work. The code and models for reproducing the results will be made available.

References

Akata, Zeynep, Reed, Scott, Walter, Daniel, Lee, Honglak, and Schiele, Bernt. Evaluation of output embeddings for fine-grained image classification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 2927–2936. IEEE, 2015.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Kim, Taeksoo, Cha, Moonsu, Kim, Hyunsoo, Lee, Jungkwon, and Kim, Jiwon. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

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

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Liu, Ming-Yu, Breuel, Thomas, and Kautz, Jan. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pp. 700–708, 2017.

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

Nilsback, M-E. and Zisserman, A. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Rashtchian, Cyrus, Young, Peter, Hodosh, Micah, and Hockenmaier, Julia. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 139–147. Association for Computational Linguistics, 2010.

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

Reed, Scott, Akata, Zeynep, Yan, Xinchen, Logeswaran, Lajanugen, Schiele, Bernt, and Lee, Honglak. Generative adversarial text-to-image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016b.

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Sharma, Shikhar, Suhubdy, Dendi, Michalski, Vincent, Kahou, Samira Ebrahimi, and Bengio, Yoshua. Chatpainter: Improving text to image generation using dialogue. *ICLR Workshops*, 2018.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang Han Zhang Zhe Gan Xiaolei Huang Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. 2018.

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

Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron, Salakhudinov, Ruslan, Zemel, Rich, and Bengio, Yoshua. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pp. 2048–2057, 2015.

Yi, Zili, Zhang, Hao, Tan, Ping, and Gong, Minglun. Dualgan: Unsupervised dual learning for image-to-image translation. *arXiv preprint*, 2017.

Zhang, Han, Xu, Tao, Li, Hongsheng, Zhang, Shaoting, Wang, Xiaogang, Huang, Xiaolei, and Metaxas, Dimitris. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv: 1710.10916*, 2017a.

Zhang, Han, Xu, Tao, Li, Hongsheng, Zhang, Shaoting, Wang, Xiaogang, Huang, Xiaolei, and Metaxas, Dimitris. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017b.

Zhang, Zizhao, Yang, Lin, and Zheng, Yefeng. Translating and segmenting multimodal medical volumes with cycle- and shape-consistency generative adversarial network. *arXiv preprint arXiv:1802.09655*, 2018.

Zhu, Jun-Yan, Park, Taesung, Isola, Phillip, and Efros, Alexei A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.