Caption-to-Image Conditional Generative Modeling

Jennie Chen
jenniechen@stanford.edu

Wenli Looi
wlooi@stanford.edu

1 Introduction

Text-to-image generation is a challenging task with many potential applications. Many approaches have been explored in recent years, the majority of which focus on using generative adversarial networks (GANs). One of the biggest challenges in text-to-image generation is ensuring that the generated image is not only visually realistic, but also semantically aligned with the input text; after all, a photo-realistic result that is unrelated to the text does not properly address the task.

In this project, we will address the task of caption-to-image generation both by using variations of ACGANs and by modifying the MirrorGAN model proposed by Qiao et al. [1]. More specifically, we hope to modify the initial embeddings used by both approaches to see if more complex ways of encoding the caption can allow us to produce better images. By improving the parts of our models closely related to semantically aligning the generated image to the input caption, we hope to be able to generate high quality images that clearly correspond to the conditioning text.

2 Related Work

Many GAN variations have been explored for the task of caption-conditioned image generation.

Auxiliary Classifier GANs (ACGANs) [2] are a form of GAN for conditional image synthesis. Instead of providing the discriminator with the class label, as traditionally done in conditional GANs, the discriminator is tasked with predicting the class label. The authors claim that this increases the generation performance as the generator learns to both generate realistic images that the discriminator classifies as the correct class.

Xu et al.'s AttnGAN [3] uses an attention mechanism that allows the image generator to focus on different aspects of the text for drawing diffferent regions of the image. In addition, the AttnGAN uses a deep attentional multimodal similarity model (DAMSM) that computes the similarity between the generated image and the input text; this allows the model to ensure the image is not only well generated to seem photorealistic, but also relates well to the text description.

The MirrorGAN of Qiao et al. [1] is similarly composed of three stages. The caption is first turned into a semantic text embedding; the embedding is then fed into cascaded image generators using both word-level and sentence-level attention. Finally, an image captioning model is used to align the caption from the generated image with the input text description. This idea is similar to the DAMSM from the AttnGAN; however, where the DAMSM attempts to compute a text and image embedding that are in the same space, MirrorGAN attempts to directly translate the generated image into a textual equivalent.

In this project, we make use of several different word/sentence embeddings. InferSent [4] is a sentence embedding model developed by Facebook that is claimed to be useful at various downstream tasks. BERT [5] is a language representation model achieving state-of-the-art performance on many tasks like measuring the similarity of two sentences. A limitation of BERT, however, is that while it can be used to produce word embeddings, computing sentence similarity requires both sentences to be fed into the model. Sentence-BERT [6] is a modification of BERT that produces semantically meaningful sentence embeddings that can be compared with cosine similarity. The authors claim that Sentence-BERT greatly reduces the computation needed to compute the sentence similarity since a quadratic number of model invocations is no longer needed.

3 Dataset

We use the CUB-200-2011 birds dataset [7], processed in the same fashion as Zhang et al. for the StackGAN [8]. The dataset consists of a total of 11,788 images in 200 classes of bird species. For our purposes, we separate out 150 classes (altogether 8,855 images) for training and the remaining 50 classes (2,933 images) for testing. Images are cropped to ensure that the bird bounding boxes in each image have an object-image size ratio greater than 0.75, therefore roughly normalizing the size of the bird in each picture.

Captions are included for these images from Reed et al. [9], collected through the use of Amazon Mechanical Turk. Each image in the dataset is associated with 10 captions.

4 Evaluation

We performed both qualitative and quantitative evaluation on our models.

Qualitatively, we sampled images from our model and manually compare to the text input, looking at whether the sampled image looks like a bird and matches the caption.

Quantitatively, we used the Inception Score. This metric uses a Inception V3 model (pre-trained on the birds dataset) to classify many generated models; predictions are then combined to capture both image quality (how well the generated image looks like a specific object) and image diversity (whether there was a wide range of objects generated). The metric uses the following formula:

$$IS(x) = \exp\left(\mathbb{E}_x\left[KL\left(p(y|x)||p(y)\right)\right]\right)$$

where x represents the image and y represents the class.

Since the ideal label distribution $p(y|x)$ and ideal marginal distribution $p(y)$ should be very different, with the first having one clear peak and the second being relatively uniform, we use KL divergence to measure the similarity of the two distributions. A high KL divergence means that the two distributions are very different; therefore, the higher the Inception Score, the better.

We note that although Inception Score is good for capturing the quality and diversity of generated images, it can't help us understand how well the generated images semantically match with the input captions. For that, we rely on our qualitative evaluation.

5 ACGAN

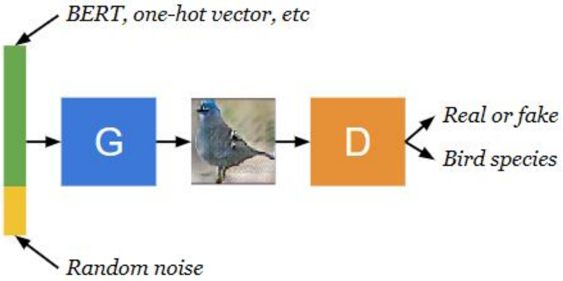


Figure 1: ACGAN Architecture

The ACGAN (Auxiliary Classifier GAN) is a variant of the traditional GAN architecture where the generator G generates images $X_{fake} = G(c, z)$ where c is the class label one-hot vector and z is random noise (we used 128-dimensional z). In our case here, c may also be a sentence embedding, such as from InferSent or BERT. The input is combined with some random noise and given to the generator, which generates an image. The discriminator the estimates realism of the image $P(real = 1|X)$ and class labels, $P(C|X)$. Note that even when the generator is given a sentence embedding and not the bird species, the discriminator still tries to predict the bird species only. See the future work section for possible alternatives.

Here, L_S is the log-likelihood of being real and L_C is the likelihood of the correct class:

$$L_S = \mathbb{E}[\log P(real = 1|X_{real})] + \mathbb{E}[\log P(real = 0|X_{fake})]$$
$$L_C = \mathbb{E}[\log P(C = c|X_{real})] + \mathbb{E}[\log P(C = c|X_{fake})]$$

The discriminator tries to maximize $L_S + L_C$ while the generator tries to maximize $L_C - L_S$.

6 MirrorGAN

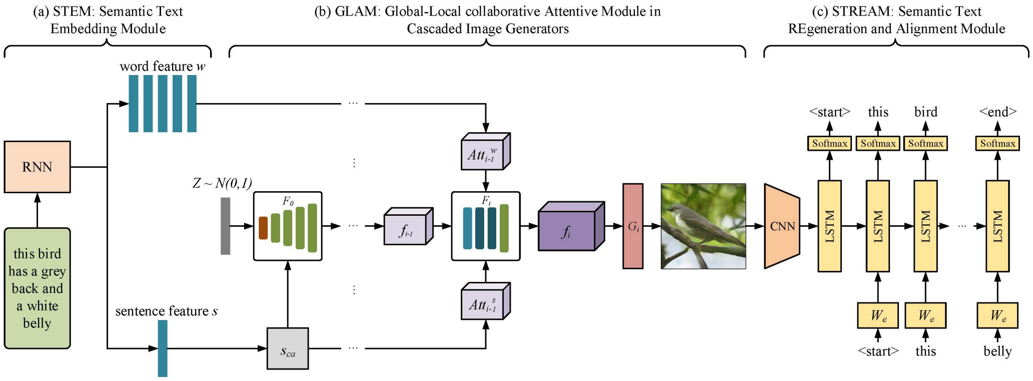

(a) STEM: Semantic Text Embedding Module
(b) GLAM: Global-Local collaborative Attentive Module in Cascaded Image Generators
(c) STREAM: Semantic Text REgeneration and Alignment Module
word feature w
RNN
this bird has a grey back and a white belly
sentence feature s
Z ~ N(0,1)
F_0
f_{i-1}
F_i
f_i
G_i
Att^w_{i-1}
Att^s_{i-1}
S_{ca}
CNN
LSTM
LSTM
LSTM
LSTM
Softmax
Softmax
Softmax
Softmax
<start>
this
bird
<end>
W_e
W_e
W_e
<start>
this
belly

Figure 2: MirrorGAN Architecture - figure taken from Qiao et al. [1]

As described above, the MirrorGAN is composed of three separate modules: the STEM module, which represents the caption as a text embedding; the GLAM module, which cascades multiple image generation networks using both word-level and sentence-level attention; and the STREAM module, which uses image captioning to regenerate a text description from the generated image.

6.1 Modules

The STEM module, or Semantic Text Embedding Module, consists of a RNN network that takes in a text description and extracts both word embeddings and sentence embeddings. The base STEM module from the original MirrorGAN is a bidirectional GRU with 128 hidden units, producing 256-dimensional embeddings.

The GLAM module, or Global-Local Collaborative Attentive Module, has a structure very similar to AttnGAN. At each stage of the cascading image generators, we use both a word-level and sentence-level attention model. Each model takes in the relevant embedding and visual feature. The embedding e is converted into a common semantic space of visual features using a perceptron layer U and is then multiplied with the input visual feature vector f to get an attention score. An attentive context feature is then computed by taking the inner product of the attention score with the converted word embedding Ue. The resulting attentive context features from the two models are concatenated with each other as well as with the input visual feature vector to compose the new visual feature vector.

The STREAM module, or Semantic Text REgeneration and Alignment Module, computes a text description from the generated image; this generated description can then be compared to the original text description in order to semantically align them. The STREAM module in the original MirrorGAN uses a common encoder-decoder framework; the encoder is an Inception V3 network pretrained on ImageNet, while the decoder is an LSTM with 512 hidden units.

6.2 Loss Functions

Both the generator and the discriminator use a loss based on both visual realism - how realistic a generated image looks - and semantic consistency - how well the image matches with sentence semantics. From this, we have the following equation for the loss functions of generator G_i and discriminator D_i:

$$\begin{aligned}\mathcal{L}_{G_i} &= -\frac{1}{2}E_{I_i\sim p_{I_i}}\log(D_i(I_i)) - \frac{1}{2}E_{I_i\sim p_{I_i}}\log(D_i(I_i, s))\\ \mathcal{L}_{D_i} &= -\frac{1}{2}E_{I_i^{GT}\sim p_{I_i^{GT}}}\log(D_i(I_i^{GT})) - \frac{1}{2}E_{I_i\sim p_{I_i}}\log(1-D_i(I_i))\\ &\quad -\frac{1}{2}E_{I_i^{GT}\sim p_{I_i^{GT}}}\log(D_i(I_i^{GT}, s)) - \frac{1}{2}E_{I_i\sim p_{I_i}}\log(1-D_i(I_i, s))\end{aligned}$$

where I_i is a generated image sampled from distribution p_{I_i} in the i^{th} stage, I_i^{GT} is a real image sampled from distribution $p_{I_i^{GT}}$ in the i^{th} stage and s is the input sentence embedding.

For the generator, we also use a text-semantic reconstruction loss aligning the original text description with the resulting description from the STREAM module. This loss is described as

$$\mathcal{L}_{stream} = -\sum_{t=0}^{L-1}\log p_t(T_t)$$

where T is the text description and L represents the sentence length.

The final objective functions of the generator and discriminator across all m stages are defined below:

$$\mathcal{L}_G = \sum_{i=0}^{m-1}\mathcal{L}_{G_i} + \lambda\mathcal{L}_{stream} \qquad , \qquad \mathcal{L}_D = \sum_{i=0}^{m-1}\mathcal{L}_{D_i}$$

7 Methods

Due to constraints in both time and compute, we limited our work to generating 64x64 color images. The ground truth bird images were appropriately downsized to match.

7.1 Baselines

For our baselines, we trained ACGAN models conditioned on one-hot class (bird species) and InferSent [4] vector, using a pre-trained 4096-dimensional InferSent model from Facebook. The one-hot model addresses a slightly different task (class-to-image generation), but allows us to understand what kind of image quality we can expect to reach. Other than the one-hot models, all other models are only provided with the sentence embedding and not the bird species.

We also use Qiao et al.'s implementation of the MirrorGAN as a baseline in order to compare how the performance our modified MirrorGAN compares. Due to our limitations, the GLAM module of our MirrorGAN only has one attention model. We use a publicly-available implementation of MirrorGAN on GitHub [10].

Both ACGANs were trained for 2000 epochs each using an Adam optimizer with learning rate 0.0002. The MirrorGAN was only trained for 450 epochs due to time limitations, using an Adam optimizer with learning rate 0.0001.

7.2 Experiments

We experimented with several variations of the ACGAN as well as a variation of the MirrorGAN by incorporating BERT embeddings using the pretrained BERT-Base (Uncased) model provided by Devlin et al. [5].

For our first experiment, we converted the input English caption into a series of 768-dimension word embeddings using the pre-trained BERT embeddings. These word embeddings are then averaged to form a sentence embedding, which is given to the ACGAN to train on.

Similarly, our second experiment calculated a sentence embedding based on BERT which is given as input to the ACGAN; however, unlike the first approach, the sentence embeddings are calculated directly from the caption using Reimers and Gurevych's Sentence-BERT [6], instead of aggregating individual word embeddings.

For our third experiment, we incorporateed BERT embeddings into the MirrorGAN by replacing the embedding layer of the STEM module with pre-trained BERT weights. This modification means that the MirrorGAN trains with 768-dimension embeddings as opposed to the previous 256-dimension embeddings.

As with our baselines, both ACGANs were trained for 2000 epochs using an Adam optimizer with learning rate 0.0002, while the MirrorGAN was trained for 450 epochs using an Adam optimizer with learning rate 0.0001.

8 Results

8.1 Generated Images

In order to qualitatively compare the output of our models, we chose two captions and used them as input to each of our models. See the captions below, followed by the actual class of bird:

this is a bird with a white belly, a blue wing and head and a small black beak - Cerulean Warbler
this bird is yellow with black and has a very short beak - American Goldfinch

The real images corresponding to these captions are as follows:

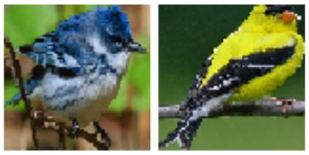
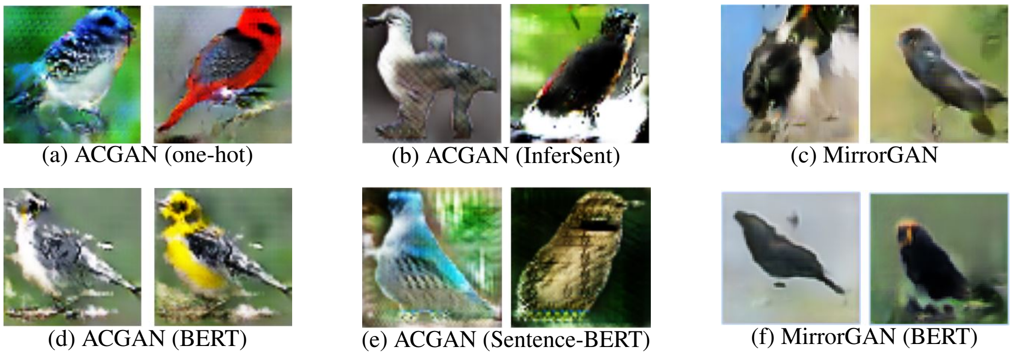


Figure 3: Real images - (Left) Cerulean Warbler, (Right) American Goldfinch

The generated images can be seen below:



(a) ACGAN (one-hot)

(b) ACGAN (InferSent)

(c) MirrorGAN

(d) ACGAN (BERT)

(e) ACGAN (Sentence-BERT)

(f) MirrorGAN (BERT)

Figure 4: Generated images - (Left) Cerulean Warbler, (Right) American Goldfinch

From these generated images, we can see that the ACGAN models are generally the best at producing bird-like images, with outlines and forms that clearly resemble birds in addition to bird-like details such as feathering and beaks. The images generated from the MirrorGAN models, in contrast, sometimes approach a vaguely bird-like shape but are very blurry or lack details such as beaks and feathering. On occasion, the output is even too blurry or general to tell that it is supposed to approximate a bird.

In terms of matching the caption, the ACGAN Sentence-BERT model seems to perform the best, incorporating both the blue and white coloring of the Cerulean Warbler and the yellow coloring and short beak of the American Goldfinch. The ACGAN one-hot model was able to caputre the blue of the Cerulean Warbler (and indeed very closely matched the real image) but produced a red and black bird instead of a yellow and black bird for the American Goldfinch, while the ACGAN BERT model did a good job of matching the caption for the American Goldfinch but completely missed the "blue" requirement of the Cerulean Warbler.

The ACGAN InferSent model produced adequate bird images, but missed the details from the captions such as coloring. The MirrorGAN produced poor bird-images, but may have possibly taken in account some of the color-related words in the captions, though these may have been reflected in the background of the image as opposed to the bird; we can see that the left image has a blue background with a vaguely black and white shape that was likely intended to be the bird, while the right image has a yellow-green background. There appears to be some slight improvement when incorporating BERT embeddings into the MirrorGAN. On the left, we can see what is clearly a bird shape with what may be a "small black beak", although it lacks details of the caption such as "blue wing". The MirrorGAN BERT model did much better with the American Goldfinch caption, producing a yellow and black bird with a very short beak. Despite the improvement however, the generated images are still fuzzy and vague compared to the images produced by ACGAN models.

We were generally disappointed with the performance of our MirrorGAN models, which performed far below our expectations. However, this could be due to several factors. The MirrorGAN models generally trained at a much slower rate than our ACGAN models; accordingly, we were only able to train them for 450 epochs, compared to the much larger 2000 epochs of the ACGAN models. We did notice that our model continued to improve all the way to 450 epochs, so it is possible that if we continued to train our models for a few weeks, we could eventually produce much better bird images. Another important note is the fact that we had to reduce the GLAM module of our models to a single attention model in order to work with our time constraints. If we had the time to use multiple stages in order to get the full effect of cascading image generators, it is highly likely we could generate better images that more closely matched the caption; after all, the purpose of multiple stages was so subsequent stages could refine an initial image outline and fix any errors that might be present (such as incorrect coloring).

We also noticed that both of our MirrorGAN models suffered from mode collapse.

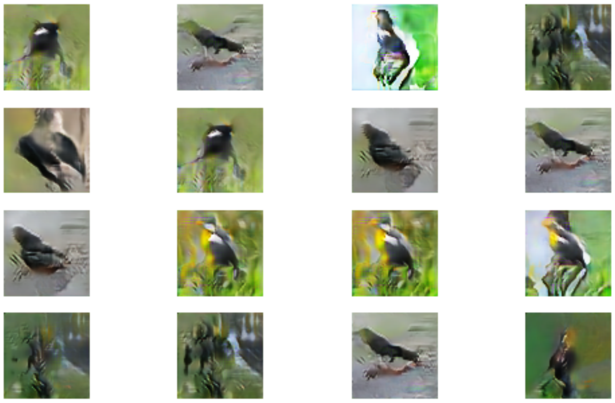


Figure 5: An example of MirrorGAN mode collapse

Figure 5 is an example of 16 images produced by the MirrorGAN model, given the same caption but with different random noise added each time. We can see that many of the images are very similar (for example, the one in the very top left and the second image in the second row). The same trend was seen in the MirrorGAN BERT model and also in the ACGAN models as shown in figure 6. There, you can see that all of he generated images have similar backgrounds, although there is some variation in the bird shapes.

8.2 Quantitative Evaluation

We evaluated the Inception Score of each model using a pre-trained bird classifier provided with the StackGAN [8] code on GitHub. As stated before, the ground truth bird images were resized to 64 x 64 so that the metric is comparable across models.

As expected, the Inception Score of all the generated models are lower than that for real images. We can see that the ACGAN one-hot model has a high Inception Score, matched only by the ACGAN Sentence-BERT model; however, we must note that the ACGAN one-hot was trained for an easier task, only having to generate from a class rather than from a specific caption.

We note that at least with the ACGANs, using more complex embeddings seem to improve the model; using Sentence-BERT embeddings led to a higher Inception Score than just averaged BERT embeddings, which was again better than InferSent embeddings. The ACGAN with Sentence-BERT embeddings was the best at the caption-to-image generation task, having the highest Inception Score. We also note that quantitatively, the

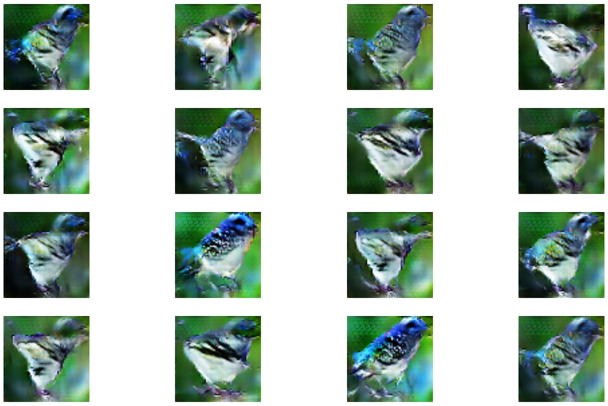


Figure 6: An example of ACGAN (one-hot) mode collapse

	Model	Inception score
	Real images	3.39 ± 0.22
Baselines	ACGAN (one-hot)	2.37 ± 0.15
	ACGAN (InferSent)	2.06 ± 0.10
	MirrorGAN	1.67 ± 0.05
Experiments	ACGAN (BERT)	2.32 ± 0.20
	ACGAN (Sentence-BERT)	2.37 ± 0.12
	MirrorGAN (BERT)	1.49 ± 0.04

Figure 7: Inception scores for models shown with estimated standard deviation

ACGANs all outperform the MirrorGANS. This is not surprising given what we saw qualitatively with the general blurriness of the MirrorGANs.

Interestingly, the MirrorGAN BERT model had a lower Inception Score than the base MirrorGAN model despite the generated images looking slightly better in many cases. This may be due to lack of diversity in generated images rather than worse image quality; we noticed that the MirrorGAN BERT model seemed to suffer from mode collapse more than the MirrorGAN model. We also note that although we trained both MirrorGAN models for the same number of epochs, the MirrorGAN BERT model used higher-dimension embeddings; it is possible that it needs more training epochs to equal the quantitative performance of the base MirrorGAN.

9 Conclusion

In this project, we approached the task of caption-to-image generation using ACGANs and MirrorGANs, with variations largely through the use of different embeddings to try to better align captions to the generated images. We found some success with using complex embeddings like BERT and Sentence-BERT with the ACGAN, able to produce bird-like images that could generally match aspects of the caption such as color. Simpler embeddings would often result in bird-like images, but ones that missed important details of the caption. Attempts with the MirrorGAN were less successful; the generated birds were often blurry and lacking in detail, and at times didn't even resemble a bird. However, this is likely due to the time constraints that led us reduce the number of stages in the MirrorGAN's GLAM module.

For the ACGAN model, future work may include changing the discriminator to predict the sentence embedding instead of the bird species. This will likely improve performance since it will allow the model to better capture differences between birds of the same species and give the generator more training signal about how the sentence maps to the image.

Given more time, we would like to increase the number of stages in the MirrorGAN so that every generated image has chances to be refined and corrected. Additionally, our project focused only on modifying the STEM module; however, there is a lot of work that can still be done with the STREAM module, which helps realign the generated image to the input caption. The current STREAM module is a relatively standard encoder-decoder framework, with an Inception V3 encoder and a vanilla LSTM decoder. The decoder especially could be improved in many ways, such as adding attention to the vanilla LSTM or replacing it altogether with a hierarchical LSTM as described by Song et al. [11] to take advantage of the successes of deep neural networks. By using a better STREAM module, the MirrorGAN may be able to ensure that its generated images are more semantically aligned to the input caption, making sure that images produced are not only realistic but relevant.

Code

Our code can be viewed at `https://github.com/looi/CS236`.

References

[1] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription, 2019. `arXiv:1903.05854`.

[2] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[3] Attngan: Fine-grained text to image generation with attentional generative adversarial networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 1316–1324. IEEE Computer Society, 12 2018. `doi:10.1109/CVPR.2018.00143`.

[4] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

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

[8] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, 2016. `arXiv:1612.03242`.

[9] Scott Reed, Zeynep Akata, Bernt Schiele, and Honglak Lee. Learning deep representations of fine-grained visual descriptions, 2016. `arXiv:1605.05395`.

[10] Komiya-M. Mirrorgan. `https://https://github.com/komiya-m/MirrorGAN`, 2019.

[11] Jingkuan Song, Xiangpeng Li, Lianli Gao, and Heng Tao Shen. Hierarchical lstms with adaptive attention for visual captioning, 2018. `arXiv:1812.11004`.