Controllable Text-to-Image Generation with Enhanced Text Encoder and Edge-Preserving Embedding

Zeyu Wang\textsuperscript{a}, Xiaoyu Liang\textsuperscript{b}, Cheng Wang\textsuperscript{c}
Cornell Tech at Cornell University
\textsuperscript{a}zw597@cornell.edu, \textsuperscript{b}xl778@cornell.edu, \textsuperscript{c}cw859@cornell.edu

Abstract: Text-to-image generation through generative adversarial networks (GANs) has become a popular research topic in the fields of natural language processing and high-quality image synthesizing. By learning latent word features of queries, many state-of-the-art GANs generate images from coarse to fine in a multi-stage trend. While this mechanism is able to synthesize realistic images stage by stage, failures containing semantic mismatches and shape distortions still exist. In this project, we adopted a multi-stage text-to-image GAN as a baseline, and remodeled this structure in two ways: 1) modify the text encoder by pre-processing text queries with pre-trained language models to gain deeply mined text information, and 2) develop an edge-preserving structure to capture feature mismatches, denoted as “edge loss”, between text latent information and generated image edges. Experiments on benchmark dataset demonstrate that models of our method outperform the baseline model and increase Inception Scores by over 30%, and that our approaches are able to effectively generate synthetic images using natural language descriptions.

1. Introduction
Generative Adversarial Networks (GANs) are widely applied in image generation tasks, and development of natural language processing techniques further make it possible to generate images from text queries. State-of-the-art GAN models synthesize realistic images stage by stage, generating fake images from coarse to fine. While this mechanism provides many well-performed outcomes, failures of text-image mismatches and low-quality image generations still exist. In this project, we proposed two methods of improving a text-to-image generative adversarial network: enhancing the query text encoder and developing an edge-preserving text-image embedding space to capture edge loss.

The enhanced text encoder is presented by adopting pre-trained embedding models including Bi-directional Encoder Representations from Transformers (BERT) [1] and Global Vectors for Word Representation (GloVe) [8] to obtain deeply mined text information. Text encoders containing pre-trained models are able to extract more fine-grained word-level text features as discussed in the conditional AlignDRAW model [6] and its subsequent work [16, 14] whose performances highly outperformed those from previous works.

Our edge-preserving text-image embedding space is designed to mitigate contour (item edge) distortions of generated images by including an “edge loss” in the training process. This embedding space is trained solely by adopting the Siamese network structure as well as the concept of triplet loss [10, 12]. Specifically, in the latent embedding space, In particular, we selected only the hard triplets (the distance between the matching item and the anchor is much longer than the distance between the non-matching item and the anchor) during the training process for a relatively quick convergence [2].
The contribution of our work can be addressed in the following two aspects:

- Better exploit the text information by modifying the text encoder with the use of pre-trained language models
- Enforce model to pay attention to item edges by introducing edge loss in the loss function

The rest of the paper is organized as follows. The next section presents the related work with backgrounds. The proposed methods are elaborated in the third section, and the fourth section illustrates their implementations. The fifth section compares the experimental results, and the final section concludes the paper and states the future work.

2. Related Work

Similar to other text-to-image generating models, our approaches adopted a multi-stage GAN structure as our backbone. We attempted to improve the backbone by deeply mining text information to build an enhanced text encoder and by developing an edge-preserving text-to-edge mapping to reduce edge distortion.

Multi-stage structure. Multi-stage GAN has become a commonly adopted architecture for its ability to generate images from coarse to fine. Zhang et al. proposed a two-stage GAN architecture (StackGAN++) [16] to generate high-resolution images: Stage-I GAN generates low-resolution images with primitive shapes and colors given textual description, and Stage-II GAN processes the previous stage results and text description to yield fine realistic images. Such multi-stage networks have been extensively utilized as backbone models in the text-to-image GAN literature [15, 16, 5, 14].

Deep mining of text information. While text-to-image generators are widely applied in synthesizing images that match text description, big improvements were always achieved when the text/image encoding techniques were boosted. The conditional AlignDRAW model [6] and its relevant work [16, 14] extracted deep-mined fine-grained word-level text features using enhanced text encoders and achieved better performances than previous models. Recently, the use of scene graph [4] and semantic layout [3] enhanced the generating process even more. They not only took the fine-grained text characteristics into consideration, but also conducted in-depth mining of semantic information by modeling the entities and their relations in the text, so as to deal with more complex scenarios.

Text-to-image retrieval. Metric learning in an embedding space is commonly applied in text-to-image [12] or image-to-image [10, 2, 11] retrieval tasks. In 2016, Wang et al. proposed a structure-preserving embedding method [12] by learning joint embedding of images and texts using a two-branch neural network with multiple layers of linear projections. Mao et al. [7] proposed a multi-constraint framework to enhance the image encoding/decoding process whose perceptual loss combines several statistical characteristics of the compressed images, including mean square error, feature loss, and adversarial loss.

Triplet Sampling. Loss Measurements usually focus on differences between two comparable values, i.e., pairwise distances. The concept of triplets has been gradually introduced in many recognition/classification/clustering related works [10, 11, 2, 12]. A triplet contains an anchor, a positive (same class as the anchor) and a negative (different class from the anchor), forcing same-class items to gather. Developed triplet mining techniques [2, 10] were also introduced recently which significantly improves training efficiency.

Based on the work above, we noticed that a better text-to-image GAN model can probably be achieved by integrating different embedding models and modifying the objective function by including edge losses that preserve item shapes.

3. Proposed Model

In this section, we introduced the backbone architecture and elaborated our proposed methods to improve the backbone model.
3.1. Backbone architecture

We adopted the controllable text-to-image generative adversarial network (controlGAN) [5] as the backbone architecture, and explored two approaches for improving model performance: in the first approach, we modified the text encoder which encodes the input query to make better use of the text information; in the second approach, we captured the edge losses of images and added them to the objective function as an attempt to preserve item edges.

In the controlGAN architecture, the main task of the network is to generate realistic images that semantically match the input text queries. For each given sentence query, a pre-trained bidirectional recurrent neural network encodes this query into a sentence feature vector $s \in \mathbb{R}^D$ as well as a word feature matrix $w \in \mathbb{R}^{D \times L}$. Conditioning augmentation, $F_{ca}$, takes the sentence feature $s$ as input and converts it to the conditioning vector $s'$. The resulting $s'$ is then concatenated with a noise vector $z$ (sampled from standard normal distribution) to serve as the inputs for $F_1$. As shown in Figure 1, the attention modules $F_{attn}$ take the word feature $w$ and latent features $v_i$ as inputs and output attentive word-context features. These attentive features are further concatenated with the latent feature $v_i$ and then serve as inputs for the following stage:

$$v_i = \begin{cases} F_1(z, F_{ca}(s)), & i = 1 \\ F_1(v_{i-1}, F_{attn_{i-1}}(w, v_{i-1})), & i = 2, 3 \end{cases}$$  \quad (1)

In each stage $i$, the neural network $F_i$ produces a hidden latent feature $v_i$, which works as an input to the corresponding generator $G_i$ to produce a synthetic image $I_i$:

$$I_i = G_i(v_i), \quad i = 1, 2, 3$$  \quad (2)

In general, the controlGAN takes the sentence feature vector $s$ and word feature matrix $w$ into a three-stage network, learns the unique visual attributes of image sub-regions and synthesizes images from coarse to fine. To address our ideas, we conducted the following approaches of modifying both the text encoder and the objective function.
3.2. Approach 1: Enhance the text encoder to make better use of text information

One of the popular approaches for improving a text-to-image generation is to make better use of text information. In the work of controlGAN, a bidirectional recurrent neural network is used as the text encoder which outputs two components for each given sentence query: 1) a sentence feature \( s \) to represent the whole sentence, and 2) a word feature matrix \( w \) that contains word-level information of the sentence query. Although recurrent neural networks are commonly used as encoding tools, we found that the sentence queries can be further pre-processed before being parsed into the network. As a result, we utilized pre-trained word embedding models to process sentence contents as well as adopted Part of Speech tagging to remove words that make semantically trivial contributions.

In this project, we adopted two pre-trained models to initialize the word embedding part. The sentence queries are first encoded by GloVe [8] and BERT [1] to extract latent text information, and the resulting numerical representations are fed into the recurrent neural network to output more detailed sentence features and word features.

3.3. Approach 2: Develop an edge-preserving structure to capture the edge matching loss

To mitigate the randomness of the generated images, the controlGAN conducted perceptual loss calculation which is an important approach to quantize the fidelity of the generated images. While pixel-wise difference is sensitive to the colors, the controlGAN extracts image features for loss measurements instead of computing the pixel-wise differences. The \( \text{relu2}_2 \) layer of VGG16 is used for feature extraction of both fake images and ground truth images because VGG16 net contains relatively more parameters than other convolutional neural networks and is more likely to capture detailed pixel information.

While this mechanism is able to synthesize realistic fake images, we noticed that many generated results, though semantically align with the text query, are visually out of shape. As shown in Figure.2, this generated image contains certain bird-related features like feathers and beaks but lacks an acceptable shape of bird.

In our attempt, we developed an edge-preserving structure which maps text information to corresponding image edges. The method is able to learn joint embedding of query texts and image edges using a two-branch neural network. This embedding space is trained separately by a metric learning process, and the well-trained structure is added to the backbone model to compute the edge matching loss \( L_{\text{edge}} \) between text query and the item edge of the generated image. We will elaborate the training process of this embedding space in the following subsections.

![Figure 2: A failure example of generated bird.](image)

3.3.1. Edge information captured by Canny Edge Detector

As shown in Figure.3, the image branch extracted edge information by applying a canny edge detection algorithm to images. In order to reduce noise of background items in an image, we utilized the bounding box data to crop out the main items in boxes that tightly surround them. The canny edge detector then removes other noise in the image with a 5x5 Gaussian filter which filters the cropped image...
images horizontally and vertically with a Sobel operator to get the pixel gradients in both directions. After obtaining the gradient magnitudes, item edges are detected at which the pixel gradients change dramatically. For each image, this algorithm generates a corresponding edge map containing only the item edges which serves as an image input to the modified Siamese network.

3.3.2. Modified Siamese network of a text-image embedding
While the Siamese network [10] commonly takes only images as inputs to learn image similarities for image clustering tasks, our modified version takes both the text query and the edge image as inputs, and learns their joint embedding. As shown in Figure 3, our model has two branches that share the same architecture: both contain fully connected layers with corresponding weight matrices, and each is followed by a ReLU activation layer. After the last fully connected layer, the L2 batch normalization is applied to generate 300-dimensional vector representations from each branch. Based on this modified Siamese network, the

![Figure 3: The Siamese network structure. There are two branches sharing the same structure in this network, one for images and the other for texts.](image)

...text and the image edge will share a latent space where vectors from the two modalities can be compared directly.

3.3.3. Triplet sampling
**Triplet Loss.** In the edge-preserving embedding space, our goal is to pull same-class images and texts together while separate different-class images or texts by an enforced margin $m$ (Figure 4), so that only same-class images and texts are closed to each other, forming class clusters. The concept of triplet loss is used in this network training process. A triplet includes an anchor with a certain label, a same-label item of the anchor (positive), and a different label item of the anchor (negative). For a...
given set of triplet \((a, p, n)\), we trained the modified Siamese network with bi-directional ranking constraints which take into account both the text branch and the image branch.

In detail, for an anchor text \(a_{text} (a_t)\), we decreased the distance \(d(a_t, p_i)\) between the anchor and its matching image \(p_{image} (p_i)\) while increased the distance \(d(a_t, n_i)\) between the anchor and its non-matching image \(n_{image} (n_i)\) to keep the non-matching away by an enforced distance margin \(m\):

\[
d(a_t, p_i) + m < d(a_t, n_i), \forall p_i \in I^+, \forall n_i \in I^- \quad (3)
\]

Figure 4: For a \((a, p, n)\) set with hard positive and hard negative, the learning process pulls \(p\) over by a distance at least as long as \(m\), and pushes \(n\) away by a distance at least as long as \(m\).

where \(I^+\) and \(I^-\) denotes the set of matching and non-matching images of the anchor text. Similarly, for an anchor image \(a_{image} (a_i)\), we have:

\[
d(a_i, p_t) + m < d(a_i, n_t), \forall p_t \in T^+, \forall n_t \in T^- \quad (4)
\]

where \(T^+\) and \(T^-\) denotes the set of matching and non-matching texts of the anchor image.

We converted the constraints to our training objective in the standard way using hinge loss. The resulting loss function is given by:

\[
L_{triplet} = \sum_{a_t, p_i, n_i} \max[0, m + d(a_t, p_i) - d(a_t, n_i)] \quad \text{(text branch training)}
\]

\[
+ \sum_{a_i, p_t, n_t} \max[0, m + d(a_i, p_t) - d(a_i, n_t)] \quad \text{(image branch training)}
\]

(5)
As we minimize the triplet loss, $L_{\text{triplet}}$, we are pushing $d(a, p)$ to be 0 and $d(a, n)$ to be greater than $d(a, p) + m$ (text branch), and are pushing $d(a, p)$ to be 0 and $d(a, n)$ to be greater than $d(a, p) + m$ (image branch). As these triplets become easy negatives (Figure 5), $L_{\text{triplet}}$ becomes 0.

**Triplet Selection.** For a given pair of an anchor and a positive, there are three types of possible negatives: 1) easy negatives, 2) semi-hard negatives, and 3) hard negatives. As shown in Figure 5, for a $(a, p, n)$ set, the distance between an easy negative and the anchor $d(a, n)$ is already much longer than the distance between the positive and the anchor $d(a, p)$, which results in zero-valued loss (no need to train). Therefore, easy negatives are considered as invalid since they cause slow convergence to the whole training process. In this project, we sampled only the hard triplets for a quicker convergence [2]. That is to say, for each anchor, we calculated pairwise distances in a batch, selected the hardest positives and hardest negatives among the batch, and optimized our loss function using stochastic gradient decent. As Figure 4 shows, the training process of triplet loss enforced the distance between an anchor and a positive to be close while separate the anchor and a negative by a distance at least as long as the margin $m$.

4. Experiment

In this section, we experimented our enhanced text encoder and edge-preserving structure on CUB bird dataset [13].

4.1. Dataset

Our work is evaluated on CUB bird dataset consisting 200 bird species with around 60 images per specie. The total of 11788 images are split in 8,855 training images and 2,933 test images with 10 corresponding captions per image.

4.2. Implementation

We maintained the training setup of the backbone network which used Adam optimizer with the learning rate $2e-4$. To evaluate our enhanced text encoder with pre-trained models, we co-trained the recurrent neural network text encoder with inputs processed by BERT and GloVe, respectively.

To train the edge-preserving embedding structure, we utilized bounding box data to crop out the main item in an image, and extracted its edge information by applying a canny edge detection algorithm, generating a new edge map that contains only the item edges, as shown in Figure 3. We then removed color messages from the captions, encoded text features and edge features, and parsed...
the encode results into the network. In particular, we trained the image embedding branch to take 256*256 sized images and map each of them to a 300-dimensional feature vector. We also trained the text embedding branch to parse sentence features to 300-dimensional feature vectors. With this mechanism, the edge-preserving structure is able to compute the edge matching score $L_{\text{edge}}$ of a sentence query and the item edge of its generated image.

The total loss of our network is defined by the summation of the perceptual loss, the Deep Attentional Multimodal Similarity Model loss (DAMSM loss) and our proposed edge loss:

$$\lambda * L_{\text{edge}} + L_{\text{perceptual}}(I, I') + L_{\text{DAMSM}}(I, I')$$  \tag{6}

where $\lambda$ is a hyper-parameter which is set to be 1e-3 for computational efficiency. The edge loss is captured by the pre-trained Siamese network and measures the difference between the encoded text query and encoded fake image edge. The perceptual loss is calculated by measuring the mismatch of text latent features and image latent features extracted by VGG16 network [5]. The DAMSM [14] loss measures the difference between the word feature matrices and encoded images.

5. Ablation Studies

To compare results from our approaches, we extracted the output of each model and used the Inception Score of generated testing images. According to quantitative results shown in Table.1, our approaches achieved higher Inception Scores than the backbone model, indicating that the proposed changes can effectively promote the generating process. In addition, we presented a visual comparison (Figure.6) of the generated images from different model settings given the same caption.

| Model                        | IS Mean | IS Std |
|------------------------------|---------|--------|
| ControlGAN                   | 2.9145  | 0.0600 |
| Ours (ControlGAN + EdgeLoss) | 3.8083  | 0.3646 |
| Ours (ControlGAN + Bert)     | 3.9473  | 0.4371 |
| Ours (ControlGAN + GloVe)    | 4.5177  | 0.2305 |

Effectiveness of enhanced text encoder. Compared to the controlGAN model which generates blue-toned image, the models with BERT (ControlGAN + Bert) and GloVe (ControlGAN + GloVe) are both able to correctly generate yellow-toned birds as shown in Figure.6. In detail, the model with BERT successfully synthesizes a bird image that align semantically with the text query, while the model with GloVe fails to generate a reasonable bird shape. We also noticed that the images generated from these two models both lack backgrounds compared to the real image, which can be attribute to the lack of background information in the text query. This observation indicates that successful image generations require the learning process of both the text query and the ground truth image, especially when ground truth images contain richer information than text queries.
Figure 6: Image comparison. With the given sentence query and ground truth image, the image output from model “ControlGAN + EdgeLoss” visually outperforms others.

**Effectiveness of edge matching loss.** The model with edge loss (ControlGAN+EdgeLoss) uses the same normal text encoder as the controlGAN model, so they both generated blue-toned birds. As shown in Figure 6, the edge matching mechanism not only captures the item edge information of the bird (the bird body), but also learns the adjacent edge information of the bird (the background). Even with a lack of background description in text query, the model with edge loss is still able to generate highly realistic backgrounds compared with the real image.

6. Conclusion

Based on controlGAN, we proposed better text encoders to utilize the text information and enforced the model to pay attention to the latent contour information by introducing edge matching loss. Our approaches have accelerated the training process of the backbone architecture and increased the Inception Score at the same epoch by 30.67% (model with edge loss), 35.44% (model with Bert), and 55.01% (model with GloVe). The image results from our enhanced text encoder approach accurately reflect detailed text information, and the image results from our edge-preserving approach show realistic bird shapes as well as rich backgrounds. These results demonstrated the advantages of our methods with respect to both high quality image-generation and efficiency of learning. Future work will explore optimization method of edge-preserving structure to capture latent information in both text queries and ground truth images.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *ArXiv*, abs/1703.07737, 2017.

[3] S. Hong, D. Yang, J. Choi, and H. Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7986–7994, 2018.

[4] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018.

[5] B. Li, X. Qi, T. Lukasiewicz, and P. Torr. Controllable text-to-image generation. In *Advances in Neural Information Processing Systems*, pages 2063–2073, 2019.
[6] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov. Generating images from captions with attention. arXiv preprint arXiv:1511.02793, 2015.

[7] Q. Mao, S. Wang, S. Wang, X. Zhang, and S. Ma. Enhanced image decoding via edge-preserving generative adversarial networks. In 2018 IEEE International Conference on Multimedia and Expo (ICME), pages 1–6. IEEE, 2018.

[8] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, 2014.

[9] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In Advances in neural information processing systems, pages 2234–2242, 2016.

[10] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015.

[11] C. Song, Y. Huang, W. Ouyang, and L. Wang. Mask-guided contrastive attention model for person re-identification. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1179–1188, 2018.

[12] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5005–5013, 2016.

[13] 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.

[14] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1316–1324, 2018.

[15] 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.

[16] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. IEEE transactions on pattern analysis and machine intelligence, 41(8):1947–1962, 2018.