Text-guided Image Manipulation via Local Feature Editing

Tianhao Zhang* Lu Jiang Weilong Yang
Google Research
Mountain View, CA, USA
<bryanzhang,lujiang,weilongyang>@google.com

Abstract

Conditional image generation is a popular area in computer vision. In recent years, the conditional input has expanded from image-only to various different modalities including labels, text, etc. In this paper we propose a task that allows user to edit an input image using text instructions. Specifically, the modification can be targeting on part of the image while keeping the rest unchanged. We propose a GAN-based method to tackle this problem. The model decomposes this task into finding where and how to modify the image. We show that the proposed model reaches better results on the CSS dataset and also visualize interesting intermediate result during the generation process.

1. Introduction

Image synthesis from text has been a focal research area in the computer vision community. It is typically set up as a conditional image generation problem where a generative adversarial network (GAN) [3] is learned to generate realistic images from the text input either in the format of natural languages [24, 16, 22, 11, 25] or scene graphs [8, 23].

Recent works in this field have shown promising progress. However, given the same input sentence, an ideal GAN model would easily generate dozens if not hundreds of "correct" images that all match the descriptions. A user may only need one of them but currently, there is no way for the user to convey this idea to the model. On the other hand, in a more common scenario, a user may have already found a reference image and wants to apply changes to the image in order to add, remove, or modify the image content. The user can express the visual difference in a sentence with the aim of generating a target image that not only visually resembles the input image but also incorporates the changes specified in the text. The previous study showed [19] it as an important use case in session search and product retrieval.

Conditional image generation based on both text and image has been largely overlooked in recent works. Only a few works have been proposed to compose images conditioned on both text and image. Text Adaptive GAN [14] was proposed to solve the task of changing visual attributes of a single object with text on fine-grained datasets [21]. As the color is the only attribute that can be changed, the task is more similar to style transfer where the style is given by the text input. Another method called GeNeVA [2] was proposed to do iterative image generation given text descriptions. Objects are added to the image step by step based on text descriptions and previous images. The authors even created a dialog environment to boost interaction.

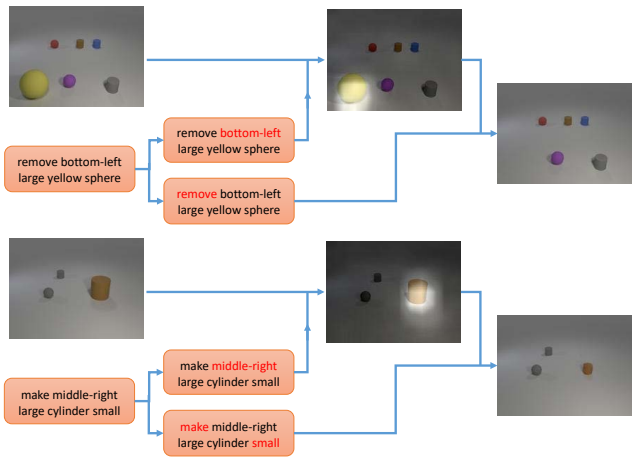


Figure 1: Examples of our image generation process. For each example, two different text attentions will focus on different part of the input text instruction and work for spatial localization and local editing respectively.

Recent GAN models have achieved impressive quality in the image-to-image generation. The main research question we study in this paper is how to generate an image by composing two different input modalities, namely the input image and the text. In other words, how to learn a meaningful cross-modal image generation.

In this work, we propose a new method called Local-

*Work done as a Google AI Resident.

GAN for image generation conditioned on both the text and image input. The key idea is to treat the input text as a "neural operator" to apply local editing to the image feature. The text operator is learned to disentangle two key aspects in the feature editing process, i.e. spatial region (where to edit) and specific computation (how to edit). For the former "where to edit", the proposed operator uses a dual-attention to map the attended spatial-indicative words in a sentence to the attended spatial region (or equivalently a binary mask) in 2D-image. The text operator applies the modification only on the attended spatial region. Regarding the second question "how to modify", we learn operational words (verb, nouns, adjective) to directly generate parameters in our feature modeling networks. The modified image features will be used to synthesize the target image in a generative adversarial model.

Preliminary experimental results on the CSS dataset [19] show the proposed method performs favorably against strong baseline methods. The results show that 1) the generated images are more realistic with improved quality, and 2) the generation is better guided by the text input compared with the baseline methods.

2. Method

Our goal is to learn a generative adversarial network from the multimodal input including a reference image and a sentence which states the desired modification to the image and conveys needed information to the image generation network. Example text inputs include adding or removing an object at a certain location, or changing the properties/attributes of an object. We call this task *text-guided image manipulation*.

More formally, Given $\mathbf{x}$ as the input image and $\mathbf{t}$ as the input text (a word sequence), the proposed conditional image generator G is supposed to synthesize a target image $G(\mathbf{x}, \mathbf{t})$ in the pixel space that has applied the modification specified in the text instruction. Here, the ground truth output image is denoted by $\mathbf{y}$.

2.1. Architecture

The proposed model is based on Generative Adversarial Network [3] which includes a generator network G and a discriminator network D.

The generator consists of five modules: an image encoder E_G, a text encoder E_T, an image decoder Dec, as well as two novel modules including a spatial localizer and a local feature editor. As shown in Figure 2, the image encoder takes in the input image $\mathbf{x}$ and produces an encoded image feature $f_0 = E_G(\mathbf{x}) \in \mathbb{R}^{H \times W \times C}$ where $H \times W$ is the resolution and C is the number of channels. The text encoder E_T takes as input a text sentence $\mathbf{t}$ of l words and outputs a sequence text embeddings $T_1, T_2 \cdots, T_l$ for each word. We use the BERT [1] model as our text encoder.

The text input is treated as a neural operator to perform local image editing in the feature space. The text operator is learned first to localize the spatial region for the modification (where to edit) and then to retrieve specific operation to be performed on the image feature. This is achieved by two modules: spatial localizer and local feature editor. The spatial localizer takes in the text embeddings as well as the input image feature and generates a spatial attention map m that highlights the region that requires editing. The local feature editor takes advantage of m and to make changes to the input image feature f_0 and output a modified feature f_{out}. The spatial localizer and the local feature editor both include a self attention module to achieve attended information from text embeddings that is useful for their own purposes. Both modules will be discussed more in detail in Section 2.2. Finally the image decoder decodes the modified feature to an RGB output image $G(\mathbf{x},\mathbf{t}) = Dec(f_{out})$.

We employ the same as the encoder and decoder as in the Pix2pix-HD [20] model. We split its generator (which uses an encoder-decoder structure) into an encoder and a decoder. Specifically, we use layers before the bottleneck (downsampling convolutional layers) as the image encoder E_G and use the remainder of the layers (bottleneck residual blocks and upsampling convolutional layers) as the decoder Dec. For simplicity, the encoder and decoder are pretrained and held fixed during training.

Unlike most image generation tasks, we apply the discriminator at the feature level instead of the image level. The discriminator is expected to tell $E_G(\mathbf{y})$, the feature of ground truth output from the generated feature $G(\mathbf{x},\mathbf{t})$. We use a patch-based discriminator [7] that takes the concatenation of the image and the attention mask as input. The network consists of only three convolutional layers. We also adopt the feature matching loss introduced in [20]. The spectral normalization and hinge loss introduced in [13] are also used in our model. The losses will be discussed more in detail in Section 2.3.

2.2. Feature Editing

Image editing guided by text can be extremely difficult stemming from the significant disparate feature space of the image and text. The intuition is that we just want to "modify" the image feature based on the text feature, rather than create an entirely different feature space. Therefore we propose to model the image generation process as an image modification process of the following 3 stages:

1. Locate the region where changes should be applied
2. Edit the image feature of the located region
3. Incorporate the edited feature and compute the global feature for the image.

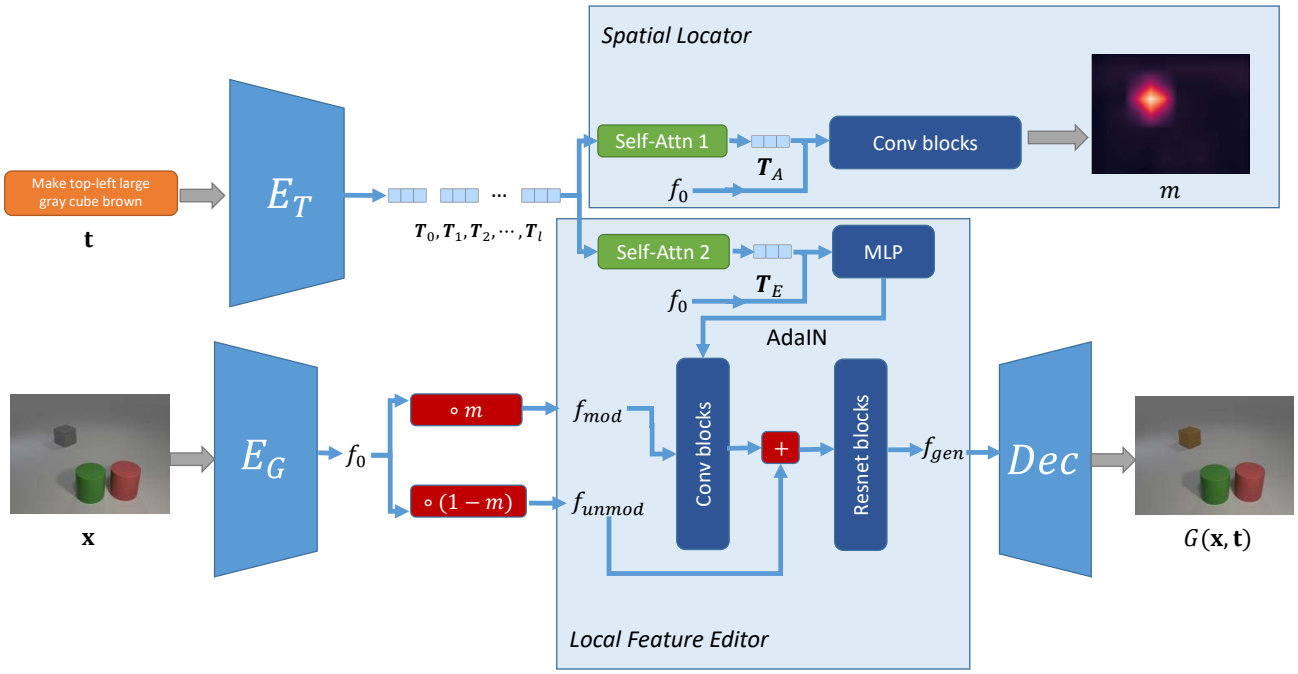


Figure 2: The generator architecture of our proposed model. The image encoder E_G encodes the input image $\mathbf{x}$. The text encoder E_T encodes a sentence of length l and outputs l text embeddings $T_1, T_2, \cdots, T_l$. The spatial localizer uses the text embeddings and input image feature to generate a spatial attention map m. Based on m and the text embeddings, the local feature editor edits the input image feature f_0 and outputs a modified feature f_{out}. The image decoder Dec then decodes the modified feature to an output image $G(\mathbf{x},\mathbf{t})$.

The text is treated as an operator to find 1) where to edit and 2) how to edit based on different "cue" (spatial or instructional) in the sentence. Given the text embeddings $T_1, T_2 \cdots, T_l$, we use a self-attention mechanism to summarize the salient words in the sentences Similar to [18], the scaled dot-product attention maps a query and a set of keys and values. The queries, keys, and values can be obtained by multiplying the input text embeddings with a learnable weight matrix.

Let each text embedding be a d_0-dimension column vector and $T = \begin{bmatrix} T1 & T2 & \cdots & T_l \end{bmatrix}^T$ then

$$Q = T \cdot W_Q$$
$$K = T \cdot W_K$$
$$V = T \cdot W_V$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_0 \times d}$ are weight matrices to learn and d is the dimension of the output embedding.

We reduce matrix Q to a d_0-dimension vector $\hat{q}$ by average pooling. The output is

$$\text{Attention}(\hat{q}, K, V) = V^T \text{softmax}(\frac{K\hat{q}}{\sqrt{d}})$$

The two self-attention modules learn to focus on different words that are useful to 1) finding spatial region and to 2) suggesting the feature change. As a result, they produce two different text embeddings: T_A and T_E where T_A is used for spatial attention generation and T_E is used for local feature editing.

2.2.1 Spatial Localization

Given the attended text feature T_A, we up-sample it to the same resolution as the image feature f_0 using a transposed convolutional layer. The up-sampled feature is then concatenated with the image feature over the channel and passed through two more convolutional layers. We take the mean of the output layer over the channels and pass it through it a sigmoid activation layer to obtain an activation mask m. The spatial attention can be derived from m by normalization.

2.2.2 Local Feature Editing

This module applies the editing on the localized region. we use m to split the image feature f_0 into two sub-features:

$$f_{mod} = m \odot f_0, \quad f_{unmod} = (1 - m) \odot f_0$$

where $\odot$ is element wise dot product and f_{mod} and f_{unmod} represent the highlighted feature to be edited and the features to stay untouched, respectively. f_{mod} will be edited through two conv layers with 3×3 filters and then added back to f_{unmod}. The combined feature will be passed through two additional Residual blocks [4] to refine the consistency across the entire feature.

To guide the editing of f_{mod} using the text feature T_E, we adopt the Adaptive Instance Normalization (AdaIN) [6] that normalizes the output feature after each layer with parameters generated using the text and image feature. To do so, we first squeeze the image feature f_0 to a single vector by taking the mean over the height and width. The text feature T_E and the squeezed image feature are concatenated and passed through a three layer MLP to produce two sets of parameters, each of which is used for the AdaIN module following a conv layer.

2.3. Losses

Multiple losses are used in the training of our model to reach the best performance.

GAN Losses We adopt the hinge loss version of the conditional GAN loss [13, 12, 17]. We then replace the image with the feature and replace the conditional input with the attention map. The idea is that since we are not training the decoder, a discriminator on the feature level would result in a more direct and larger gradient during back propagation. The attention map can additionally let the D focus on the area that has been changed. Let $f_y = E_G(\mathbf{y})$. The loss is given by

$$\begin{aligned} L_D =& \mathbb{E}_{(m,f_y)} \min[0, -1 + D(m, f_y)] + \\ & \mathbb{E}_{(m,f_{gen})} \min[0, -1 - D(m, f_{gen})] \\ L_G =& -\mathbb{E}_{(m,f_{gen})} D(m, f_{gen}) \end{aligned}$$

Reconstruction losses We use the two reconstruction losses used in Pix2pix-HD [20] model: the feature matching loss and the perceptual loss using pretrained VGG network feature. The feature matching loss is useful for making the training more stable and the perceptual loss (which operates on image level) gives a more perceptive distance measure between images. Let $D^{(i)}$ denote the i-th layer feature. The feature matching loss is given by

$$L_{FM} = \sum_{i=1}^{L} \frac{1}{N_i} ||D^{(i)}(m, f_{gen}) - D^{(i)}(m, f_y)||_1$$

where L is the number of layers and N_i is the number of elements in the i-th layer.

Let $F^{(i)}$ denote the i-th layer VGG network feature. The perceptual loss is given by

$$L_{VGG} = \sum_{i=1}^{K} \frac{1}{M_i} ||F^{(i)}(\mathbf{y}) - F^{(i)}(G(\mathbf{x}, \mathbf{t})||_1$$

where K is the number of layers and M_i is the number of elements in the i-th layer.

Triplet Loss We additionally add a triplet loss for faster learning. We used the classification loss introduced in [19], that is

$$L_{triplet} = \sum_{i=1}^{R} \log\{1 + \exp\{\kappa(f_y, f_{gen}) - \kappa(f_{neg_i}, f_{gen})\}\}$$

where f_{neg_i} is the i-th randomly selected negative image feature and κ is a similarity kernel defined as the negative l_2 distance. This loss intends to pull the generated image feature closer to the ground truth feature and more distant from the negative examples.

Regularizer for Spatial Localization In addition, we use the difference between the input image and ground truth image as a signal to help spatial attention map learning. Specifically, we take the absolute difference between the input and output image feature and normalize it to a single channel map scale from 0 to 1. The normalization consists of taking the average over the channels and divide it by the max value. That is

$$L_R = ||\text{norm}(E_G(\mathbf{y}), E_G(\mathbf{x})) - m||_1$$

We finally combine these loss using a weighted sum and the final generator loss will be:

$$\hat{L_G} = L_G + \alpha L_{FM} + \beta L_{VGG} + \gamma L_{triplet} + \delta L_R$$

3. Experiment

3.1. Baselines

We incorporate two closely-related image synthesis models as our baselines. Both models utilize text and image to generate the target image. Text-Adaptive GAN (TAGAN) [14] and Generative Visual Artist GAN (GeNeVAGAN) [2]

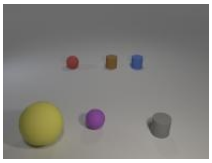
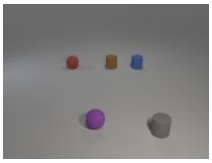
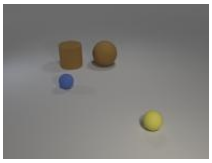
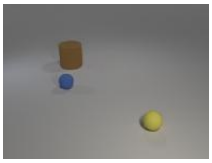
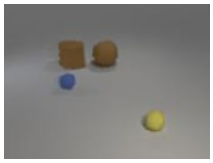
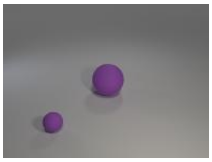
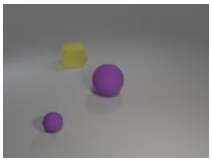
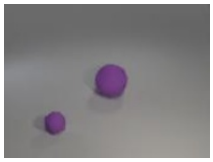
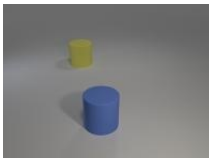
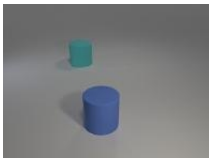
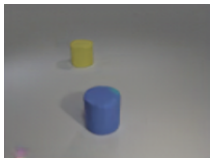
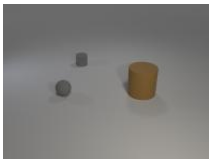
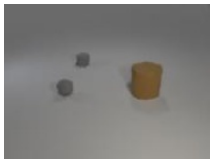
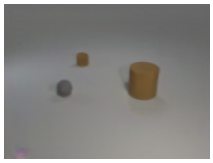
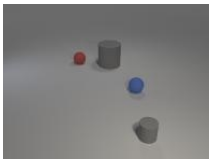
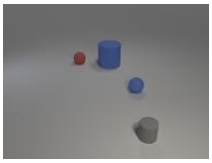
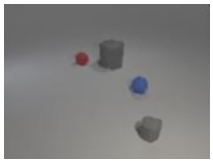
Input Image	Input Text	Ours	TA-GAN	GeNeVA
	remove bottom-left large yellow sphere			
	remove top-center large brown sphere			
	add large cube to top-left			
	make top-left large yellow object cyan			
	make middle-right large cylinder small			
	make top-center large cylinder blue			

Table 1: Qualitative Comparison between our method and baselines.

TA-GAN TA-GAN proposed a text-guided image manipulation task. Their setting is based on fine-grained datasets where each image contains only one single object from the same category (e.g. bird for the CUB dataset[21]) and there are multiple captions describing attributes of the object in the image. The image manipulation is done by transferring one image to fit another caption. The TA-GAN model explicitly trains the discriminator to learn whether an image matches a caption. Since we do not have captions describing the image, we fuse the input image and text modification to construct a caption for the output image, and use this as a baseline.

GeNeVA GeNeVA focuses on the task of sequential image generation based on text. Their proposed task decomposes image generation into multiple steps, with a given text instruction for each step. They proposed an iterative GAN model that generates images based on the previous image and current text input. We adapt their method to our task by treating our training pair as a single step image generation.

3.2. Dataset

We use the CSS dataset that is originally created by [19] for the image retrieval task. The dataset is generated using the CLEVR toolkit[9] and contains 3-D synthesized images with the presence of objects with different color,shape and size. There are three types of text modifications: Add, Remove and Change Attribute. Each text modification will specify the position of the target object and the expected operations. Examples can be seen in qualitative evaluations in Figure 1. Note that we generate the images to reduce misalignment for unchanged objects. The dataset includes 17K training data pairs and 17K tests.

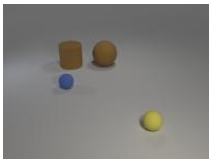
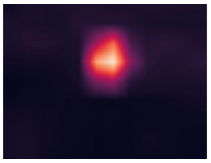
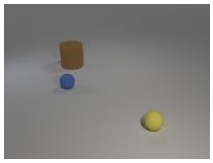
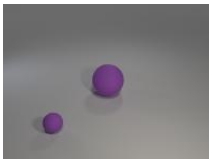
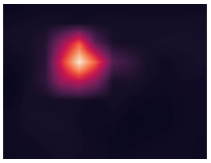
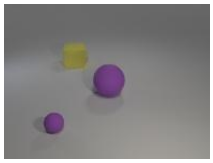
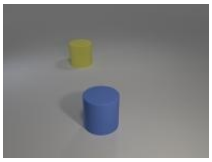
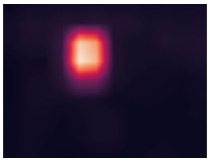
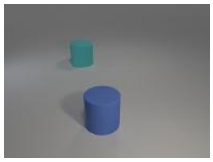
Input Image	Self-attn1	Self-attn2	Spatial attn	Result
	remove top-center large brown sphere	remove top-center large brown sphere		
	add large cube to top-left	add large cube to top-left		
	make top-left large yellow object cyan	make top-left large yellow object cyan		

Table 2: Visualization of intermediate results. We visualize the spatial attention map used in our method. We also highlighted texts with high attention score (≥ 0.2) in both self-attention modules.

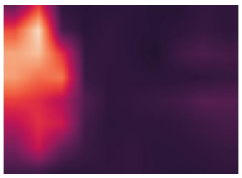
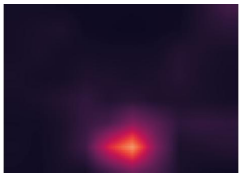
Input Image	Self-attn1	Self-attn2	Spatial attn	Result
	Small pine tree placed left side with tip cut off and a little bit of the left side cut off	Small pine tree placed left side with tip cut off and a little bit of the left side cut off		
	Top left is a cloud big size no cuts right corner is a sun almost all cut from top and right	Top left is a cloud big size no cuts right corner is a sun almost all cut from top and right		
	Fire to right of boy	Fire to right of boy		

Table 3: Additional results on the Codraw dataset.

3.3. Pretraining

As described in Section 2.1, the encoder and decoder are pretrained and fixed in future training. To do this, we directly train a reconstruction network for all images in the dataset using the Pix2pix-HD model. Then we split the pretrained model into the encoder and the decoder following Section 2.1

3.4. Implementation Details

We implemented our model in Pytorch and trained for 300 epochs using a single Tesla-V100 GPU. We use Adam optimizer with $\beta 1 = 0.5$ and $\beta 2 = 0.999$. The learning rate is 2e-5 for the first 150 epochs and linearly decays to 0 afterwards. We use the bert-base-cased model[1] pretrained on Wikipedia and Bookcorpus. Other hyper-parameter choices are: $alpha = 100$, $beta = 100$, $gamma = 10$, $delta = 100$

Model	IS ↑	FID ↓
Ground Truth	1.751	-
TA-GAN	1.696	13.950
GeNeVA	1.640	19.883
Ours	**1.727**	**1.474**

Table 4: IS and FID score comparisons.

3.5. Quantitative Evaluation

We conduct two commonly used quantitative evaluation methods for GAN: the Inception Score (IS)[15] and the Fréchet Inception Distance (FID)[5]. IS evaluates the image quality of a single dataset and FID evaluates the distance between two datasets.

The results are shown in Table 4. Our method beats both baselines on both metrics. Our method surpasses the baselines on FID by a huge margin, suggesting that the generated results are more similar to the ground truth images.

3.6. Qualitative Evaluation

We further demonstrate qualitative evaluation. 6 sets of tests are shown in Table 1. Both baselines can keep a relatively good image quality, however, neither of them can modify the input image properly. TA-GAN basically copies the input image. Even though GeNeVA can edit the image but it is obviously irrelevant to the input text. On the other hand, our model can properly do the correct modifications.

In addition, we visualize the intermediate results for our model in Table 2. The spatial attention maps are shown in the 4-th column which highlights the modified area. The attention map fits the text-specified object accurately which indicates the spatial localization to be working. We also highlight the words with a high attention score ($>$0.2) in both self-attention modules. As shown in the table, in the first self attention module, location words are highlighted. In the second self attention module, the attention focuses more on words that specify the target operation. These intermediate results explicitly show the undergoing generation process which helps to understand the full picture of the model

3.7. Other Dataset

To further verify the effectiveness of our method, we conduct additional experiment on the CoDraw [10] dataset. CoDraw is a synthetic dataset formed by sequences of images of children playing in the park. For each sequence, there is a conversation between a Teller and a Drawer. The teller gives text instructions on how to change the current image and the Drawer can ask questions to confirm details. For simplicity we only use the text given by the Teller. We also extract adjacent pairs of data from the sequences to form a dataset for our task. Due to time limitation, the baseline results are not available yet. Examples including intermediate results generated by our model are shown in Table 3

4. Conclusion

In this preliminary work, we study a multimodal image generation task where the generated image is conditioned on both a reference image and texts specifying the change to be made over the reference image. This problem setting corresponds to a critical scenario in real-world applications such as image and product retrieval, where users can manipulate the content of the image at hand to retrieve the target image. To address this problem, we propose a novel approach to treat the input text as a neural operator to "modify" the image feature from which the target image will be generated. We evaluate our method on the standard benchmark derived from CLEVR and shows promising results compared to two recent generative models. In the future, we plan to verify our work on more challenging datasets.

References

[1] 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. 2, 6

[2] Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W. Taylor. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 1, 4

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 7

[6] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 4

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 2

[8] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 1

[9] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 5

[10] Jin-Hwa Kim, Nikita Kitaev, Xinlei Chen, Marcus Rohrbach, Byoung-Tak Zhang, Yuandong Tian, Dhruv Batra, and Devi Parikh. Codraw: Collaborative drawing as a testbed for grounded goal-driven communication. *arXiv preprint arXiv:1712.05558*, 2017. 7

[11] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *CVPR*, 2019. 1

[12] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 4

[13] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 2, 4

[14] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: manipulating images with natural language. In *Advances in neural information processing systems*, pages 42–51, 2018. 1, 4

[15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 7

[16] Fuwen Tan, Song Feng, and Vicente Ordonez. Text2scene: Generating compositional scenes from textual descriptions. In *CVPR*, 2019. 1

[17] Dustin Tran, Rajesh Ranganath, and David Blei. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, pages 5523–5533, 2017. 4

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[19] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6439–6448, 2019. 1, 2, 4, 5

[20] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017. 2, 4

[21] 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. 1, 5

[22] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018. 1

[23] LI Yikang, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. Pastegan: A semi-parametric method to generate image from scene graph. In *NeurIPS*, 2019. 1

[24] 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. In *ICCV*, 2017. 1

[25] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *ICCV*, 2019. 1