# Fine-grained Image-to-Image Editing from Text Captions

Saptarashmi Bandyopadhyay
saptab1@umd.edu

Matthew A. Gwilliam
mgwillia@umd.edu

David Li
dli7319@umd.edu

Soumik Mukhopadhyay
smukhopa@umd.edu

Marco Volpi
mvolpi@terpmail.umd.edu

## Abstract

*Image editing is one of the most sought after applications of image processing. With the advances in computer vision techniques and generative models as well as popularity of software like Adobe Photoshop image editing has come to the tips of our hands. But even today for editing images it is required to visually select regions that need manipulation. But as we progress in the field of visual grounding and generative modelling, one possible way to edit images would be using text captions. This makes the task of image manipulation easier for a layman who does not have the skills required to operate niche software.*

*In this work, we look into editing images using text captions in an unsupervised setting. This has hidden the underlying task of understanding the scene as well as the editing text and accordingly manipulating the image. For this we employ popular networks like StyleGAN2 and Bert respectively and try to get the attributes in the same latent style space.*

## 1. Introduction

Advances in deep learning and computer vision over the past several years have allowed researchers to produce high-quality realistic looking images using a technique called generative adversarial networks, or GANs. While these advances have led to the creation of breathtakingly realistic generated images, some even with fine-grained semantic control, little is still understood about how generative adversarial networks map their latent space to the manifold of realistic images.

Within GAN research, many papers have proposed various techniques to control the generated output of the GAN. For instance, Bau *et al*. [4, 3] dissect individual feature maps to determine what they represent in the final image by computing a correlation with segmented regions in the final image. That find that different channels of feature maps correspond to different objects in the image, suggesting that GANs learn a disentangled representation. Park *et al*. propose GauGAN [19] which uses a spatially adaptive normalization to generate images from segmentation masks. Both of these methods leverage the spatial correlations of regions in the feature maps with regions in the output images associated with CNNs.

In another line of computer work, researchers have focused on extracting correspondences between text captions, which only contain semantic information often without spatial information, and images, typically without semantic information. Recent work in this area have created different end-to-end text-to-image pipelines and image editing from text pipelines.

In this paper, we study whether pre-trained models can be combined and fine-tuned in a multi-modal way to achieve the same goals of multi-modal semantic image editing. By using pre-trained models, elements of our pipeline already model information from only a single modality, either text or images, and must learn a shared embedding or latent space between both modalities in the fine-tuning stage. Specifically, we seek to answer the following research questions:

1. Can BERT learn to ground StyleGAN2's AdaIN style vectors?
2. Can StyleGAN2 be used to semantically edit images by algebraically manipulating style vectors?
3. Can visual grounding with BERT improve the semantic editing of images?
4. How can we combine models pre-trained on different modalities for multi-modal tasks?

## 2. Related Works

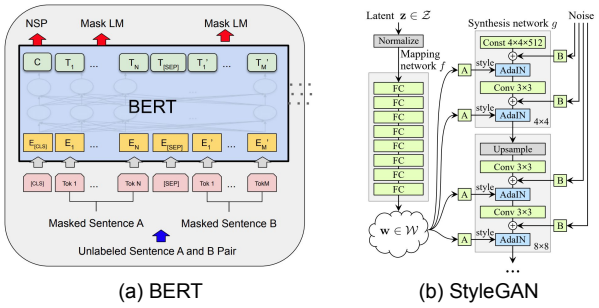
### 2.1. Image Generation with GAN

A popular line of work in computer vision and unsupervised learning focuses on image generation using generative adversarial networks [7]. Generative adversarial networks (GANs), originally proposed by Goodfellow *et al*.

[7], train a generator to produce fake samples from an underlying training distribution using a discriminator which distinguishes between real and generated samples. By back-propagation through the discriminator to the generator, the min-max optimization process eventually allows the generator to learn to produce images indistinguishable from the real training samples.

Among GAN architectures, one of the most popular models is StyleGAN [12]. StyleGAN, shown in Figure 1, allows control of the synthesized image by applying adaptive instance normalization [9] (AdaIN) at every layer of the image generation process and can generate high-resolution images using a progressive growing approach from ProGAN [10]. Inputs to AdaIN, or *style* vectors, are synthesized using a mapping network trained jointly with the image synthesis network. Since the original development of StyleGAN, further work has been done to remove artifacts in StyleGAN2 [13] and train on smaller datasets in StyleGAN2-Ada [11].

Much work has been devoted to understanding StyleGAN and controlling its output by careful manipulation of AdaIN parameters. Abdal *et al*. [1, 2] develop an algorithm to map real images to an extended style space of StyleGAN and use it to generate high-quality edited images.



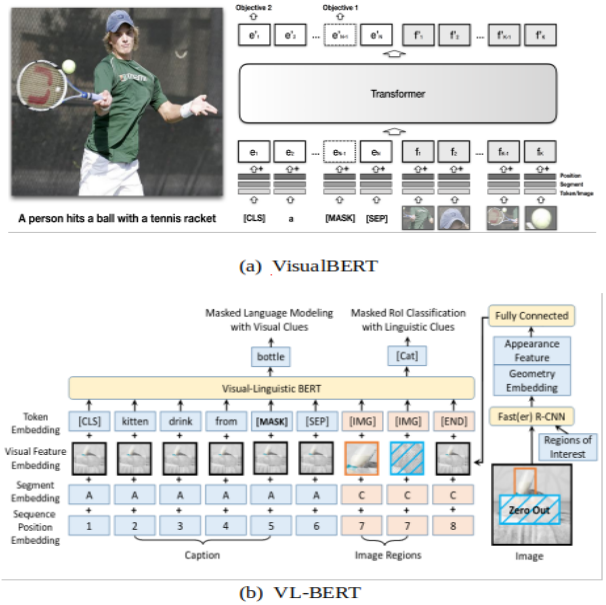

(a) BERT (b) StyleGAN

Figure 1: Overview of (a) BERT which operates on natural language text and (b) StyleGAN which generates images using AdaIN parameters.

## 2.2. Image Generation from Text

Recently, the community has devoted significant attention to the relationship between text and images. Within the task of image generation, some of this work has focused on generating images from text descriptions [20]. Research in that area has focused on generating layouts for more complex scenes using coarse-to-fine multi-stage approaches [8, 18]. Other work has made improvements to the fidelity and appearance of images generated from text, with the goal of generating photo-realistic images of single subjects and even complex scenes [25, 27, 29].

While many papers have proposed image manipulation and editing algorithms [15, 28], our paper is most closely related to algorithms that are text-driven. Specifically, ManiGAN makes edits to simple images based on captions [16]. Our method proposes to accomplish something similar, but using BERT to learn the text representation and StyleGAN2 to generate the actual images. TIM-GAN has a related approach, but is distinct in that image editing is done using a natural language instruction (that describes an edit) rather than a caption that describes an image [26].




(a) VisualBERT

(b) VL-BERT

Figure 2: VisualBERT [17] (a) performs visual grounding by inputting image captions and image regions-of-interest (RoIs) into a BERT transformer model. VLBERT [22] (b) preforms a similar task, while describing the geometry of each token in the output.

## 2.3. Visual Grounding

Another line of work in multi-modal text and vision research that is beneficial for text-guided image editing focuses on visual grounding, or identifying the correspondence between words in a sentence and regions in an image.

Li *et al*. [17] developed VisualBERT, a BERT model pre-trained at masked language modeling and sentence-image prediction by taking both image captions and image regions-of-interest (RoIs) as inputs shown in Figure 2. Each embedding is a summation of a visual feature representation, a segment embedding and a position embedding. After pre-training, VisualBERT can ground visual concepts as seen by examining the attention matrices (i.e. the products of weight and key matrices) and can be fine-tuned for visual question answering and visual commonsense reasoning.

Su [22] develop VL-BERT by pre-training on masked language modeling and masked RoI classification as seen in Figure 2. Unlike VisualBERT, caption segments of VL-

BERT consist of both a token embedding as well as a visual feature embedding which describes the appearance and geometry of each token.

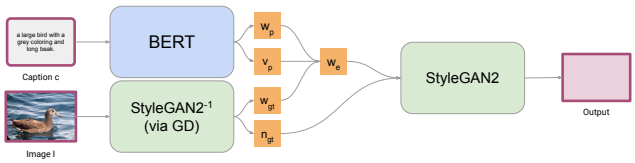



Figure 3: Our initial pipeline combines a BERT model that interprets captions with a StyleGAN2 model that generates realistic looking images. At inference time, an edited style vector is generated and passed into StyleGAN2 to generate an edited image.

# 3. Method

## 3.1. Initial Pipeline

Our inference pipeline, shown in Figure 3, consists of 2 components: a style edit generator based on Bert [6] and a pre-trained StyleGAN2 [14] image generator. The role of the style generator is to convert captions to style vectors, identifying which components of the style vector are well-defined by the caption.

The style edit generator needs to be able to parse the caption $c$ using its understanding as a general purpose language model as well as understand the style-vectors used by StyleGAN. To accomplish this, we use Bert to first parse the caption. Next, we add two fully-connected heads $(f_{style}, f_{weight})$ which converts embeddings generated by Bert into style vectors $w_p$ and weight vectors $v_p$. As Bert generates embeddings of size 768 and StyleGAN uses style vectors of size 512, our head generates two 512-dim vectors:

$$u = \text{Bert}(c) \tag{1}$$
$$w_p = f_{style}(u) \tag{2}$$
$$v_p = \text{sigmoid}(f_{weight}(u)) \tag{3}$$

Given the new style $w_p$ and style weights $v_p$ as well as the original style $w_{gt}$, we generate an edited style by mixing the two styles according to the style weights:

$$w_e = v_p * w_p + (1 - v_p) * w_{gt} \tag{4}$$

### 3.1.1 Training and Losses

Our training proceeds in two stages. In the first stage we pretrain a StyleGAN2 model to generate realistic images. In this stage, it is essential that the domain of the training set accurately resembles the domain of the images we wish to edit so that StyleGAN2 is able to recreate such images. However, as training StyleGAN2 does not require captioned training data, it is possible to augment any captioned training dataset with uncaptioned data during this training stage.

In the second stage, we fix the pretrained StyleGAN and optimize Bert to generate style vectors and weights. As the exact properties of style vectors may vary between trained StyleGAN models, during the training and inference process, we fix a single StyleGAN model.

### 3.1.2 Inference

At inference time, we wish to generate an edited image $I_{edit}$ from an original image $I$ and edited caption $c_{edit}$. We first project the original image into the latent space of the StyleGAN generator $G$. This yields the original style $w_{gt}$ and noise $\eta_{gt}$ corresponding to the original image:

$$w_{gt}, \eta_{gt} = \underset{w,\eta}{\text{argmin}} \|G(w, \eta) - I\|_1 \tag{5}$$

This optimization is done using the Adam optimizer.

Next, the edited caption is fed through BERT following Equation 1, Equation 2, and Equation 3 to get a style edit $w_p$ and weight $v_p$. Following Equation 4, we compute the edited style vector $w_e$ from the style edit, weight, and projected style vector. When we pass this edited style vector $w_e$ along with $\eta_{gt}$ through the StyleGAN2 generator $G$ we expect to get the image edited $I_{edit}$ with the attributes present in the edited caption $c_{edit}$.

## 3.2. Contrastive Learning Pipeline

Building on top of the initial pipeline we try to achieve visual grounding in a contrastive learning setting as shown in Figure 4. The problem with the initial pipeline was that nowhere did we generate the edited image during the training process. In fact we were just training the Bert to generate styles from the captions of the same image. In this process the Bert tends to learn to make the style weights $v_p$ to be small enough to get the majority of the style from the $w_{orig}$. Due to this, the inference image is almost the same as that of the original image as shown in Figure 6.

To make sure that the model starts to learn to identify distinctive attributes as well as conditionally update the style vectors corresponding to these attributes, we use a conditional contrastive learning setting inspired from [23]. In the input we have two images-caption pairs - $(I, c)$ and $(I_{rand}, c_{rand})$. What we want to do here is to edit the first image $I$ using the caption of the second pair $c_{rand}$. To do so we get the Bert generated style $w_{rand}$ and weights $v_{rand}$ for the second caption $c_{rand}$ using Equations (1) - (3). The styles are then mixed and passed through the StyleGAN2 generator to get the edited image as follows -

$$I_{edit} = G(v_{rand} * w_{rand} + (1 - v_{rand}) * w_{gt}, \eta) \tag{6}$$

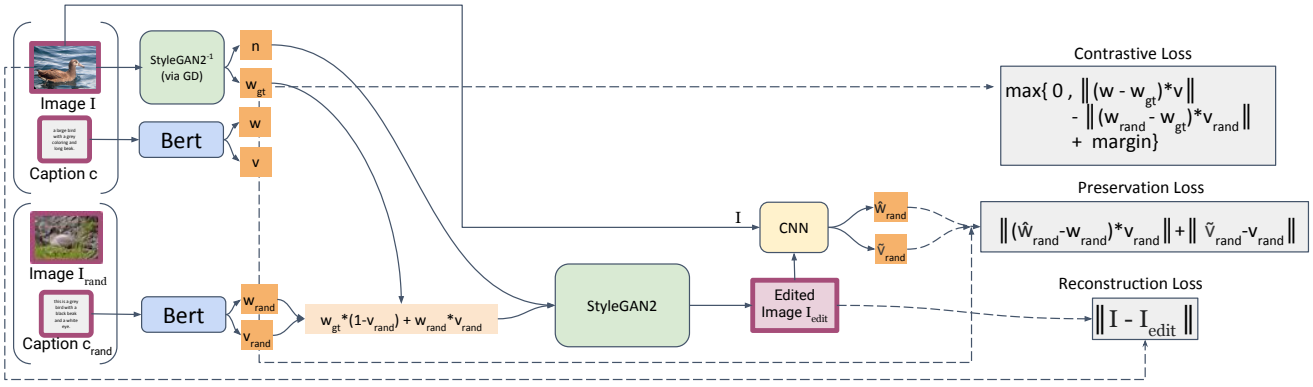



Figure 4: Overview of our contrastive attribute training pipeline which uses three losses during training: a contrastive loss, a preservation loss, and a reconstruction loss.

Another convolutional neural network(CNN) is present that takes as input $I$ and $I_{edit}$ and tries to predict back $w_{rand}$ an $v_{rand}$. This is to make sure that the style information that was present in the second caption $c_{rand}$ is preserved in the edited image $I_{edit}$ and doesn't get lost while mixing. This is similar to the ideas presented in [21] and [5] where the information preservation is being implemented in an unsupervised manner without actually having the attribute labels.

$$\hat{w}_{rand}, \tilde{v}_{rand} = \text{CNN}(I, I_{edit}) \tag{7}$$

The inference in this pipeline is the same as that presented in Section 3.1.2.

#### 3.2.1 Training and Losses

Here we have three losses. The first loss is a triplet style loss which tries to reward the conditional similarity of attributes in the style of the first image $I$ and the first caption $c$ while penalising the conditional similarity of the attributes in the style of first image $I$ and the second caption $c_{rand}$.

$$\begin{aligned}\mathcal{L}_{contrastive} = &\max(0, \|(w - w_{gt}) * v\|_1 \\ &- \|(w_{rand} - w_{gt}) * v_{rand}\|_1 + margin)\end{aligned} \tag{8}$$

It is highly probable that when the second image is chosen randomly with respect to the first image, they will have have very distinct attributes and hence this loss is intuitively reasonable.

The second loss is the preservation loss that makes sure that the editing caption style information is preserved in the edited image given that we have no control over the pretrained StyleGAN2 generator.

$$\begin{aligned}\mathcal{L}_{preservation} = &\|(\hat{w}_{rand} - w_{rand}) * v_{rand}\|_1 + \\ &\|\tilde{v}_{rand} - v_{rand}\|_1\end{aligned} \tag{9}$$

Here the first term corresponds to matching the styles while the second term is for matching weights for the correct attributes.

The third loss is a simple reconstruction loss to make sure that the edited image structure corresponds to the original image structure.

$$\mathcal{L}_{reconstruction} = \|I - I_{edit}\|_1 \tag{10}$$

### 3.3. Visual Grounding Pipeline

Finally, we attempted to replace BERT with VLBERT which is pre-trained for visual grounding. Our idea to include Visual grounding is based on the 3 ideas listed below.

- Efficient editing by aligning text to be replaced with image sub-region, identifying the associated attention matrices from the dictionary and replacing the older attention weights in the image sub-region with the new ones creating the edited image.
- Quantitative evaluation: Other than the image sub-region aligned to the text being edited, all other non-aligned image sub-regions are expected to be the same in original and edited image. So their similarity measured can be calculated.
- Use VisualBERT with visual feature representations instead of BERT, just having textual embeddings to predict the style vectors as shown in Figure 5.

The challenges that we faced while fine-tuning and testing was that the down-stream tasks for Visual grounding models with BERT are limited to VCR, VQA, RefCOCO, the pre-trained model is based on COCO which has to be fine-tuned for any downstream task, and training requires almost 70G of data and features for which we did not have adequate storage as seen in our training and testing log files.

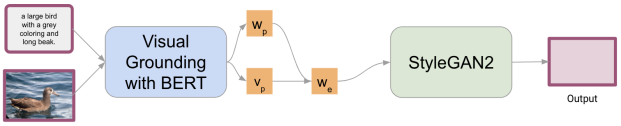



Figure 5: Extension of our previous pipeline to include Visual Grounding as input to StyleGAN2

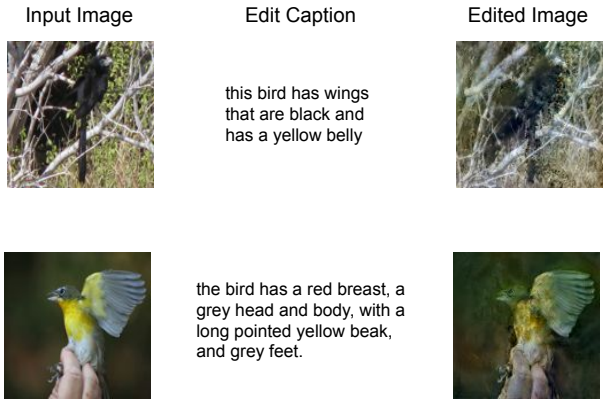



Figure 6: Qualitative results with our initial pipeline which uses a reconstruction loss on the cub dataset. Despite our best efforts, the edited image fails to reflect the new caption input into the network.


Figure 7: Qualitative results with the Contrastive Learning Pipeline. (a) Edited image with reconstruction loss with the first image; (b) Edited image when trained with reconstruction loss to both the first and the second image

# 4. Evaluation

## 4.1. Datasets and Models

For our evaluation, we focus on the Caltech-UCSD Birds-200-2011 (CUB) dataset which has captions from AttnGAN [24].

We started by pre-training StyleGAN2 on each dataset using the `stylegan2_pytorch`[1] implementation by Phil Wang. Afterwards, this pre-trained model is frozen and BERT is fine-tuned to ground the style latent space of StyleGAN2 according to the methods described in section 3. In our experiments, we start with the pre-trained `bert-base-uncased` model from Huggingface[2]. As the pre-trained BERT model inputs and outputs 768 dimensional word embeddings, we add a single fully-connected layer to the first output of BERT which outputs a 1024 dimensional vector. The first 512 entries of this output are treated as the style edit while the remaining 512 entries are passed through sigmoid to generate the style weights as shown in Equation 3.

## 4.2. Qualitative Results

We present qualitative results of our initial pipeline with reconstruction loss in Figure 6. From our results, we see that while BERT does output a modified style vector which changes the image, the edited image does not necessarily correspond well to the edited caption. Compared to the original input image, the edited image is far grainier and does not match the edited caption. Thus our preliminary experiments suggest that BERT may be unable to map the image captions to the AdaIN latent space of StyleGAN2 without extensive tuning.

Next, the qualitative results for the Contrastive Learning Pipeline has been presented in Figure 7. The results are poor and do not reflect the attributes present in the edit caption. But these results are inconclusive as the experiments could not be carried out with extensive tuning.

# 5. Conclusion

In this paper, we study whether the latent space of a pre-trained StyleGAN2 model can be semantically grounded. To understand this, we experiment with applying BERT to predict latent style vectors from image captions for the task of semantic image editing. We propose and test several different variants of training losses including losses in both the style space as well as image space. However, we found that our architecture combining BERT and StyleGAN2 was unable to generate edited images which correspond to the edited image captions input to BERT. Future work is to include visual grounding in our pipeline for efficient editing and quantitative evaluation.

# References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[1] stylegan2_pytorch: https://github.com/lucidrains/stylegan2-pytorch

[2] Huggingface BERT: https://huggingface.co/transformers/model_doc/bert.html

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[3] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 38(4), 2019. 1

[4] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 1

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29:2172–2180, 2016. 4

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 3

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014. 1, 2

[8] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[9] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2

[10] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 2

[11] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020. 2

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[13] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[15] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. In *Advances in Neural Information Processing Systems*, pages 2065–2075, 2019. 2

[16] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H.S. Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[17] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557, 2019. 2

[18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[19] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[20] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 2

[21] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6490–6499, 2019. 4

[22] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: pre-training of generic visual-linguistic representations. *CoRR*, abs/1908.08530, 2019. 2

[23] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 830–838, 2017. 3

[24] 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. 2018. 5

[25] Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. Semantics disentangling for text-to-image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2327–2336, 2019. 2

[26] Tianhao Zhang, Hung-Yu Tseng, Lu Jiang, Weilong Yang, Honglak Lee, and Irfan Essa. Text as neural operator:

Image manipulation by text instruction. *arXiv preprint arXiv:2008.04556*, 2020. 2

[27] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[28] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pages 597–613. Springer, 2016. 2

[29] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2