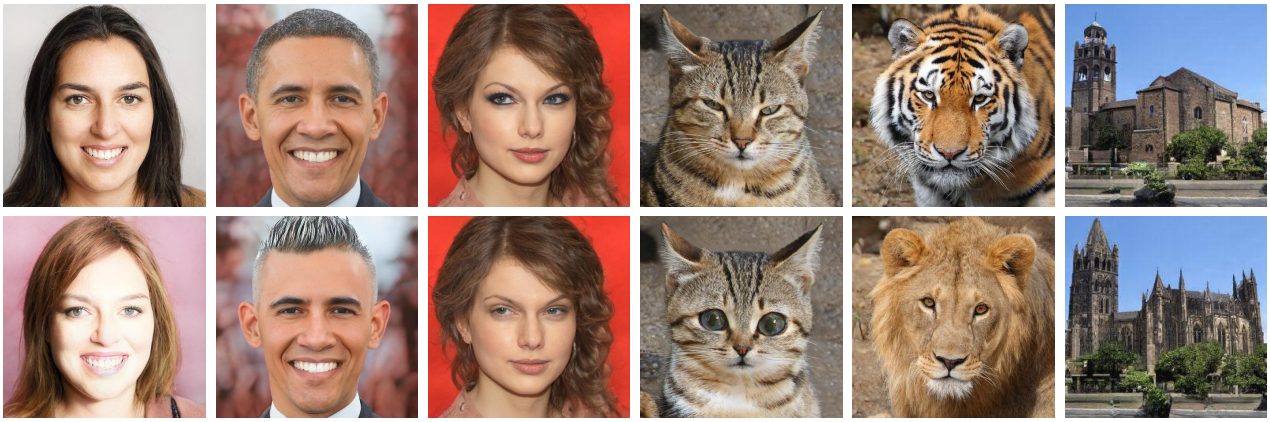
StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery

Or Patashnik[†*] Zongze Wu[‡*] Eli Shechtman[§] Daniel Cohen-Or[†] Dani Lischinski[‡]
[‡]Hebrew University of Jerusalem [†]Tel-Aviv University [§]Adobe Research



"Emma Stone" "Mohawk hairstyle" "Without makeup" "Cute cat" "Lion" "Gothic church"

Figure 1. Examples of text-driven manipulations using StyleCLIP. Top row: input images; Bottom row: our manipulated results. The text prompt used to drive each manipulation appears under each column.

Abstract

Inspired by the ability of StyleGAN to generate highly realistic images in a variety of domains, much recent work has focused on understanding how to use the latent spaces of StyleGAN to manipulate generated and real images. However, discovering semantically meaningful latent manipulations typically involves painstaking human examination of the many degrees of freedom, or an annotated collection of images for each desired manipulation. In this work, we explore leveraging the power of recently introduced Contrastive Language-Image Pre-training (CLIP) models in order to develop a text-based interface for StyleGAN image manipulation that does not require such manual effort. We first introduce an optimization scheme that utilizes a CLIP-based loss to modify an input latent vector in response to a user-provided text prompt. Next, we describe a latent mapper that infers a text-guided latent manipulation step for a given input image, allowing faster and more stable text-based manipulation. Finally, we present a method for mapping text prompts to input-agnostic directions in StyleGAN's style space, enabling interactive text-driven image manipulation. Extensive results and comparisons demonstrate the effectiveness of our approaches.

[*] Equal contribution, ordered alphabetically. Code and video are available on https://github.com/orpatashnik/StyleCLIP

1. Introduction

Generative Adversarial Networks (GANs) [18] have revolutionized image synthesis, with recent style-based generative models [24, 25, 22] boasting some of the most realistic synthetic imagery to date. Furthermore, the learnt intermediate latent spaces of StyleGAN have been shown to possess disentanglement properties [9, 48, 19, 53, 58], which enable utilizing pretrained models to perform a wide variety of image manipulations on synthetic, as well as real, images.

Harnessing StyleGAN's expressive power requires developing simple and intuitive interfaces for users to easily carry out their intent. Existing methods for semantic control discovery either involve manual examination (e.g., [19, 48, 58]), a large amount of annotated data, or pretrained classifiers [49, 1]. Furthermore, subsequent manipulations are typically carried out by moving along a direction in one of the latent spaces, using a parametric model, such as a 3DMM in StyleRig [53], or a trained normalized flow in StyleFlow [1]. Specific edits, such as virtual try-on [27] and aging [2] have also been explored.

Thus, existing controls enable image manipulations only along preset semantic directions, severely limiting the user's creativity and imagination. Whenever an additional, unmapped, direction is desired, further manual effort and/or large quantities of annotated data are necessary.

In this work, we explore leveraging the power of recently introduced Contrastive Language-Image Pre-training (CLIP) models in order to enable intuitive text-based semantic image manipulation that is neither limited to preset manipulation directions, nor requires additional manual effort to discover new controls. The CLIP model is pretrained on 400 million image-text pairs harvested from the Web, and since natural language is able to express a much wider set of visual concepts, combining CLIP with the generative power of StyleGAN opens fascinating avenues for image manipulation. Figure 1 shows several examples of unique manipulations produced using our approach. Specifically, in this paper we investigate three techniques that combine CLIP with StyleGAN:

1. Text-guided latent optimization, where a CLIP model is used as a loss network [20]. This is the most versatile approach, but it requires a few minutes of optimization to apply a manipulation to an image.

2. A latent residual mapper, trained for a specific text prompt. Given a starting point in latent space (the input image to be manipulated), the mapper yields a local step in latent space.

3. A method for mapping a text prompt into an input-agnostic (global) direction in StyleGAN's style space, providing control over the manipulation strength as well as the degree of disentanglement.

The results in this paper and the supplementary material demonstrate a wide range of semantic manipulations on images of human faces, animals, cars, and churches. These manipulations range from abstract to specific, and from extensive to fine-grained. Many of them have not been demonstrated by any of the previous StyleGAN manipulation works, and all of them were easily obtained using a combination of pretrained StyleGAN and CLIP models.

2. Related Work

2.1. Vision and Language

Joint representations Multiple works learn cross-modal vision and language (VL) representations [12, 47, 52, 35, 30, 51, 29, 7, 32] for a variety of tasks, such as language-based image retrieval, image captioning, and visual question answering. Following the success of BERT [13] in various language tasks, recent VL methods typically use Transformers [55] to learn the joint representations. A recent model, based on Contrastive Language-Image Pre-training (CLIP) [42], learns a multi-modal embedding space, which may be used to estimate the semantic similarity between a given text and an image. CLIP was trained on 400 million text-image pairs, collected from a variety of publicly available sources on the Internet. The representations learned by CLIP have been shown to be extremely powerful, enabling state-of-the-art zero-shot image classification on a variety of datasets. We refer the reader to OpenAI's Distill article [17] for an extensive exposition and discussion of CLIP.

Text-guided image generation and manipulation The pioneering work of Reed *et al*. [45] approached text-guided image generation by training a conditional GAN [36], conditioned by text embeddings obtained from a pretrained encoder. Zhang *et al*. [62, 63] improved image quality by using multi-scale GANs. AttnGAN [60] incorporated an attention mechanism between the text and image features. Additional supervision was used in other works [45, 31, 26] to further improve the image quality.

A few studies focus on text-guided image manipulation. Some methods [14, 39, 33] use a GAN-based encoder-decoder architecture, to disentangle the semantics of both input images and text descriptions. ManiGAN [28] introduces a novel text-image combination module, which produces high-quality images. Differently from the aforementioned works, we propose a single framework that combines the high-quality images generated by StyleGAN, with the rich multi-domain semantics learned by CLIP.

Recently, DALL·E [43, 44], a 12-billion parameter version of GPT-3 [6], which at 16-bit precision requires over 24GB of GPU memory, has shown a diverse set of capabilities in generating and applying transformations to images guided by text. In contrast, our approach is deployable even on a single commodity GPU.

More recently, TediGAN [59] and Paint by Word [4], also pair a GAN with CLIP for text-guided image generation and manipulation. By training an encoder to map text into the StyleGAN latent space, TediGAN can generate an image corresponding to a given text. To perform text-guided image manipulation, TediGAN encodes both the image and the text into the latent space, and then performs style-mixing to generate a corresponding image. In Section 7 we demonstrate that the manipulations achieved using our approach reflect better the semantics of the driving text.

Rather than manipulating images, several concurrent projects use CLIP to guide text-to-image generation through optimization. Deep Daze [38] optimizes the weights of a neural implicit representation network, while [37, 41, 10, 16] optimize the latent space of BigGAN [5], StyleGAN [25] or VQGAN [15]. While text-to-image generation is an intriguing and challenging problem, we believe that the image manipulation abilities we provide constitute a more useful tool for the typical workflow of creative artists.

2.2. Latent Space Image Manipulation

Many works explore how to utilize the latent space of a pretrained generator for image manipulation [9, 53, 58]. Specifically, the intermediate latent spaces in StyleGAN

have been shown to enable many disentangled and meaningful image manipulations. Some methods learn to perform image manipulation in an end-to-end fashion, by training a network that encodes a given image into a latent representation of the manipulated image [40, 46, 2, 3]. Other methods aim to find latent paths, such that traversing along them result in the desired manipulation. Such methods can be categorized into: (i) methods that use image annotations to find meaningful latent paths [48, 1], and (ii) methods that find meaningful directions without supervision, and require manual annotation for each direction [19, 50, 56, 57].

While most works perform image manipulations in the $\mathcal{W}$ or $\mathcal{W}+$ spaces, Wu *et al*. [58] proposed to use the *StyleSpace* $\mathcal{S}$, and showed that it is better disentangled than $\mathcal{W}$ and $\mathcal{W}+$. Our latent optimizer and mapper work in the $\mathcal{W}+$ space, while the input-agnostic directions that we detect are in $\mathcal{S}$. In all three, the manipulations are derived directly from text input, and our only source of supervision is a pretrained CLIP model. As CLIP was trained on hundreds of millions of text-image pairs, our approach is generic and can be used in a multitude of domains without the need for domain- or manipulation-specific data annotation.

3. StyleCLIP Text-Driven Manipulation

In this work we explore three ways for text-driven image manipulation, all of which combine the generative power of StyleGAN with the rich joint vision-language representation learned by CLIP.

We begin in Section 4 with a simple latent optimization scheme, where a given latent code of an image in StyleGAN's $\mathcal{W}+$ space is optimized by minimizing a loss computed in CLIP space. The optimization is performed for each (source image, text prompt) pair. Thus, while this method is versatile, several minutes are required to perform a single manipulation, and it can be difficult to control. A more stable approach is described in Section 5, where a mapping network is trained to infer a manipulation step in latent space, in a single forward pass. The training takes a few hours, but it must only be done once per text prompt. The direction of the manipulation step may vary depending on the starting position in $\mathcal{W}+$, which corresponds to the input image, and thus we refer to this mapper as *local*.

Our experiments with the local mapper reveal that the manipulation directions are often similar to each other, despite different starting points. Also, since the manipulation step is performed in $\mathcal{W}+$, it is difficult to achieve fine-grained visual effects in a disentangled manner. Thus, in Section 6 we explore a third text-driven manipulation scheme, which transforms a given text prompt into an input agnostic (i.e., *global* in latent space) mapping direction. The direction is computed in StyleGAN's style space $\mathcal{S}$ [58], which is better suited for fine-grained and disentangled visual manipulation, compared to $\mathcal{W}+$.

	pre-proc.	train time	infer. time	input image dependent	latent space
optimizer	–	–	98 sec	yes	$\mathcal{W}+$
mapper	–	10 – 12h	75 ms	yes	$\mathcal{W}+$
global dir.	4h	–	72 ms	no	$\mathcal{S}$

Table 1. Our three methods for combining StyleGAN and CLIP. The latent step inferred by the optimizer and the mapper depends on the input image, but the training is only done once per text prompt. The global direction method requires a one-time preprocessing, after which it may be applied to different (image, text prompt) pairs. Times are for a single NVIDIA GTX 1080Ti GPU.

Table 1 summarizes the differences between the three methods outlined above, while visual results and comparisons are presented in the following sections.

We have also experimented with optimizing and mapping directly in the $\mathcal{S}$ latent space. Our results (in the supplementary material) reveal that optimizing in $\mathcal{S}$ yields more disentangled edits, however, it is harder to achieve global changes. For our latent mapper method, we found no advantage for operating in $\mathcal{S}$.

4. Latent Optimization

A simple approach for leveraging CLIP to guide image manipulation is through direct latent code optimization. Specifically, given a source latent code $w_s \in \mathcal{W}+$, and a directive in natural language, or a *text prompt* t, we solve the following optimization problem:

$$\arg\min_{w \in \mathcal{W}+} D_{\text{CLIP}}(G(w), t) + \lambda_{\text{L2}} \left\| w - w_s \right\|_2 + \lambda_{\text{ID}} \mathcal{L}_{\text{ID}}(w), \tag{1}$$

where G is a pretrained StyleGAN[1] generator and D_{CLIP} is the cosine distance between the CLIP embeddings of its two arguments. Similarity to the input image is controlled by the L_2 distance in latent space, and by the identity loss [46]:

$$\mathcal{L}_{\text{ID}}(w) = 1 - \langle R(G(w_s)), R(G(w)) \rangle, \tag{2}$$

where R is a pretrained ArcFace [11] network for face recognition, and $\langle \cdot, \cdot \rangle$ computes the cosine similarity between it's arguments. We solve this optimization problem through gradient descent, by back-propagating the gradient of the objective in (1) through the pretrained and fixed StyleGAN generator G and the CLIP image encoder.

In Figure 2 we provide several edits that were obtained using this optimization approach after 200-300 iterations. The input images were inverted by e4e [54]. Note that visual characteristics may be controlled explicitly (beard, blonde) or implicitly, by indicating a real or a fictional person (Beyonce, Trump, Elsa). The values of λ_{L2} and λ_{ID} depend on the nature of the desired edit. For changes that shift towards another identity, λ_{ID} is set to a lower value.

[1] We use StyleGAN2 [25] in all our experiments.

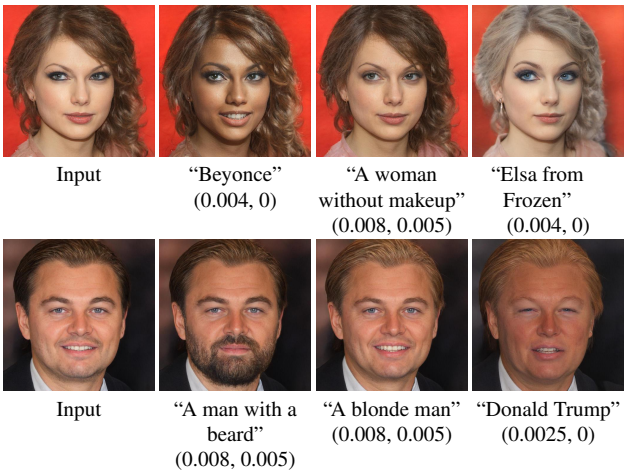


Figure 2. Edits of real celebrity portraits obtained by latent optimization. The driving text prompt and the $(\lambda_{\text{L2}}, \lambda_{\text{ID}})$ parameters for each edit are indicated under the corresponding result.

5. Latent Mapper

The latent optimization described above is versatile, as it performs a dedicated optimization for each (source image, text prompt) pair. On the downside, several minutes of optimization are required to edit a single image, and the method is somewhat sensitive to the values of its parameters. Below, we describe a more efficient process, where a mapping network is trained, for a specific text prompt t, to infer a manipulation step $M_t(w)$ in the $\mathcal{W}+$ space, for any given latent image embedding $w \in \mathcal{W}+$.

Architecture The architecture of our text-guided mapper is depicted in Figure 3. It has been shown that different StyleGAN layers are responsible for different levels of detail in the generated image [24]. Consequently, it is common to split the layers into three groups (coarse, medium, and fine), and feed each group with a different part of the (extended) latent vector. We design our mapper accordingly, with three fully-connected networks, one for each group/part. The architecture of each of these networks is the same as that of the StyleGAN mapping network, but with fewer layers (4 rather than 8, in our implementation). Denoting the latent code of the input image as $w = (w_c, w_m, w_f)$, the mapper is defined by

$$M_t(w) = (M_t^c(w_c), M_t^m(w_m), M_t^f(w_f)). \quad (3)$$

Note that one can choose to train only a subset of the three mappers. There are cases where it is useful to preserve some attribute level and keep the style codes in the corresponding entries fixed.

Losses Our mapper is trained to manipulate the desired attributes of the image as indicated by the text prompt t, while preserving the other visual attributes of the input image. The CLIP loss, $\mathcal{L}_{\text{CLIP}}(w)$ guides the mapper to minimize the cosine distance in the CLIP latent space:

$$\mathcal{L}_{\text{CLIP}}(w) = D_{\text{CLIP}}(G(w + M_t(w)), t), \quad (4)$$

where G denotes again the pretrained StyleGAN generator. To preserve the visual attributes of the original input image, we minimize the L_2 norm of the manipulation step in the latent space. Finally, for edits that require identity preservation, we use the identity loss defined in eq. (2). Our total loss function is a weighted combination of these losses:

$$\mathcal{L}(w) = \mathcal{L}_{\text{CLIP}}(w) + \lambda_{L2} \left\| M_t(w) \right\|_2 + \lambda_{\text{ID}} \mathcal{L}_{\text{ID}}(w). \quad (5)$$

As before, when the edit is expected to change the identity, we do not use the identity loss. The parameter values we use for the examples in this paper are $\lambda_{\text{L2}} = 0.8, \lambda_{\text{ID}} = 0.1$, except for the "Trump" manipulation in Figure 9, where the parameter values we use are $\lambda_{\text{L2}} = 2, \lambda_{\text{ID}} = 0$.

In Figure 4 we provide several examples for hair style edits, where a different mapper used in each column. In all of these examples, the mapper succeeds in preserving the identity and most of the other visual attributes that are not related to hair. Note, that the resulting hair appearance is adapted to the individual; this is particularly apparent in the "Curly hair" and "Bob-cut hairstyle" edits.

It should be noted that the text prompts are not limited to a single attribute at a time. Figure 5 shows four different combinations of hair attributes, straight/curly and short/long, each yielding the expected outcome. This degree of control has not been demonstrated by any previous method we're aware of.

Since the latent mapper infers a custom-tailored manipulation step for each input image, it is interesting to examine the extent to which the direction of the step in latent space varies over different inputs. To test this, we first invert the test set of CelebA-HQ [34, 21] using e4e [54]. Next, we feed the inverted latent codes into several trained mappers and compute the cosine similarity between all pairs of the resulting manipulation directions. The mean and the standard deviation of the cosine similarity for each mapper is reported in Table 2. The table shows that even though the mapper infers manipulation steps that are adapted to the input image, in practice, the cosine similarity of these steps for a given text prompt is high, implying that their directions are not as different as one might expect.

6. Global Directions

While the latent mapper allows fast inference time, we find that it sometimes falls short when a fine-grained disentangled manipulation is desired. Furthermore, as we have seen, the directions of different manipulation steps for a given text prompt tend to be similar. Motivated by these observations, in this section we propose a method for mapping

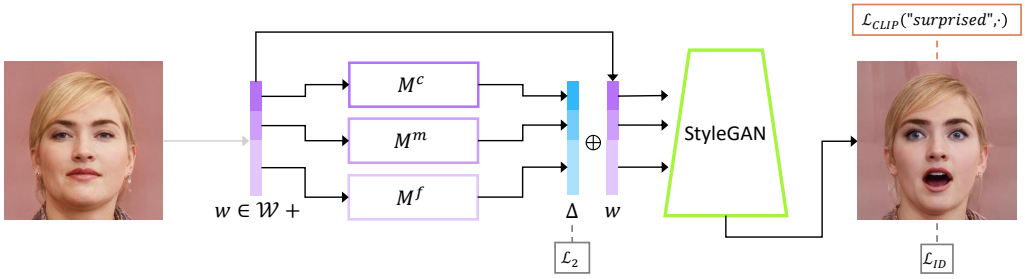


Figure 3. The architecture of our text-guided mapper (using the text prompt "grey hair", in this example). The source image (left) is inverted into a latent code w. Three separate mapping functions are trained to generate residuals (in blue) that are added to w to yield the target code, from which a pretrained StyleGAN (in green) generates an image (right), assessed by the CLIP and identity losses.

	Mohawk	**Afro**	**Bob-cut**	**Curly**	**Beyonce**	**Taylor Swift**	**Surprised**	**Purple hair**
Mean	0.82	0.84	0.82	0.84	0.83	0.77	0.79	0.73
Std	0.096	0.085	0.095	0.088	0.081	0.107	0.893	0.145

Table 2. Average cosine similarity between manipulation directions obtained from mappers trained using differnt text prompts.

Input "Mohawk hairstyle" "Curly hair" "Bob-cut hairstyle" "Afro hairstyle"

Figure 4. Hair style edits using our mapper when training M^c and M^m only. The driving text prompts are indicated below each column. All input images are inversions of real images.

"Straight short hair" "Straight long hair" "Curly short hair" "Curly long hair"

Figure 5. Controlling more than one attribute with a single mapper. The driving text for each mapper is indicated below each column.

a text prompt into a single, global direction in StyleGAN's style space $\mathcal{S}$, which has been shown to be more disentangled than other latent spaces [58].

Let $s \in \mathcal{S}$ denote a style code, and $G(s)$ the corresponding generated image. Given a text prompt indicating a desired attribute, we seek a manipulation direction Δs, such that $G(s + \alpha\Delta s)$ yields an image where that attribute is introduced or amplified, without significantly affecting other attributes. The manipulation strength is controlled by α. Our high-level idea is to first use the CLIP text encoder to obtain a vector Δt in CLIP's joint language-image embedding and then map this vector into a manipulation direction Δs in $\mathcal{S}$. A stable Δt is obtained from natural language, using prompt engineering, as described below. The corresponding direction Δs is then determined by assessing the relevance of each style channel to the target attribute.

More formally, denote by $\mathcal{I}$ the manifold of image embeddings in CLIP's joint embedding space, and by $\mathcal{T}$ the manifold of its text embeddings. We distinguish between these two manifolds, because there is no one-to-one mapping between them: an image may contain a large number of visual attributes, which can hardly be comprehensively described by a single text sentence; conversely, a given sentence may describe many different images. During CLIP training, all embeddings are normalized to a unit norm, and therefore only the direction of embedding contains semantic information, while the norm may be ignored. Thus, in well trained areas of the CLIP space, we expect directions on the $\mathcal{T}$ and $\mathcal{I}$ manifolds that correspond to the same semantic changes to be roughly collinear (i.e., have large cosine similarity), and nearly identical after normalization.

Given a pair of images, $G(s)$ and $G(s+\alpha\Delta s)$, we denote their $\mathcal{I}$ embeddings by i and $i + \Delta i$, respectively. Thus, the difference between the two images in CLIP space is given by Δi. Given a natural language instruction encoded as Δt, and assuming collinearity between Δt and Δi, we can determine a manipulation direction Δs by assessing the relevance of each channel in $\mathcal{S}$ to the direction Δi.

From natural language to Δt In order to reduce text embedding noise, Radford *et al*. [42] utilize a technique called prompt engineering that feeds several sentences with the same meaning to the text encoder, and averages their embeddings. For example, for ImageNet zero-shot classification, a bank of 80 different sentence templates is used, such as "a bad photo of a {}", "a cropped photo of the {}", "a black and white photo of a {}", and "a painting of a {}". At inference time, the target class is automatically substituted into these templates to build a bank of sentences with similar semantics, whose embeddings are then averaged. This process improves zero-shot classification accuracy by an additional 3.5% over using a single text prompt.

Similarly, we also employ prompt engineering (using the same ImageNet prompt bank) in order to compute stable directions in $\mathcal{T}$. Specifically, our method should be provided with text description of a target attribute and a corresponding neutral class. For example, when manipulating images of cars, the target attribute might be specified as "a sports car", in which case the corresponding neutral class might be "a car". Prompt engineering is then applied to produce the average embeddings for the target and the neutral class, and the normalized difference between the two embeddings is used as the target direction Δt.

Channelwise relevance Next, our goal is to construct a style space manipulation direction Δs that would yield a change Δi, collinear with the target direction Δt. For this purpose, we need to assess the relevance of each channel c of $\mathcal{S}$ to a given direction Δi in CLIP's joint embedding space. We generate a collection of style codes $s \in \mathcal{S}$, and perturb only the c channel of each style code by adding a negative and a positive value. Denoting by Δi_c the CLIP space direction between the resulting pair of images, the relevance of channel c to the target manipulation is estimated as the mean projection of Δi_c onto Δi:

$$R_c(\Delta i) = \mathbb{E}_{s \in \mathcal{S}}\{\Delta i_c \cdot \Delta i\} \tag{6}$$

In practice, we use 100 image pairs to estimate the mean. The pairs of images that we generate are given by $G(s \pm \alpha \Delta s_c)$, where Δs_c is a zero vector, except its c coordinate, which is set to the standard deviation of the channel. The magnitude of the perturbation is set to $\alpha = 5$.

Having estimated the relevance R_c of each channel, we ignore channels whose R_c falls below a threshold β. This parameter may be used to control the degree of disentanglement in the manipulation: using higher threshold values results in more disentangled manipulations, but at the same time the visual effect of the manipulation is reduced. Since various high-level attributes, such as age, involve a combination of several lower level attributes (for example, grey hair, wrinkles, and skin color), multiple channels are relevant, and in such cases lowering the threshold value may be preferable, as demonstrated in Figure 6. To our knowledge, the ability to control the degree of disentanglement in this manner is unique to our approach.

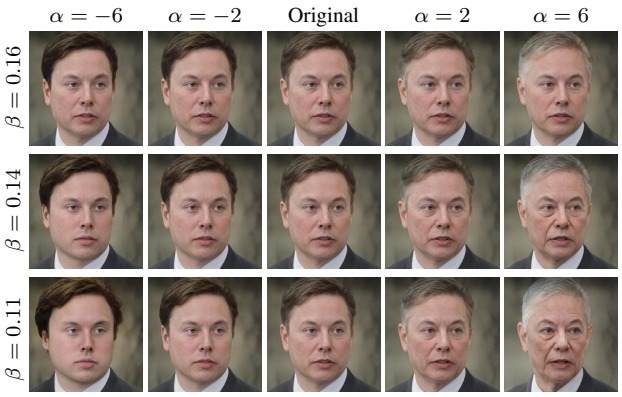


Figure 6. Image manipulation driven by the prompt "grey hair" for different manipulation strengths and disentanglement thresholds. Moving along the Δs direction, causes the hair color to become more grey, while steps in the $-\Delta s$ direction yields darker hair. The effect becomes stronger as the strength α increases. When the disentanglement threshold β is high, only the hair color is affected, and as β is lowered, additional correlated attributes, such as wrinkles and the shape of the face are affected as well.

In summary, given a target direction Δi in CLIP space, we set

$$\Delta s = \begin{cases} \Delta i_c \cdot \Delta i & \text{if } |\Delta i_c \cdot \Delta i| \geq \beta \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Figures 7 and 8 show a variety of edits along text-driven manipulation directions determined as described above on images of faces, cars, and dogs. The manipulations in Figure 7 are performed using StyleGAN2 pretrained on FFHQ [24]. The inputs are real images, embedded in $\mathcal{W}+$ space using the e4e encoder [54]. The figure demonstrates text-driven manipulations of 18 attributes, including complex concepts, such as facial expressions and hair styles. The manipulations in Figure 8 use StyleGAN2 pretrained on LSUN cars [61] (on real images) and on generated images from StyleGAN2-ada [23] pretrained on AFHQ dogs [8].

7. Comparisons and Evaluation

We now turn to compare the three methods presented and analyzed in the previous sections among themselves and to other methods. All the real images that we manipulate are inverted using the e4e encoder [54].

Text-driven image manipulation methods: We begin by comparing several text-driven facial image manipulation methods in Figure 9. We compare between our latent mapper method (Section 5), our global direction method (Section 6), and TediGAN [59]. For TediGAN, we use the authors' official implementation, which has been recently updated to utilize CLIP for image manipulation, and thus is

Figure 7. A variety of edits along global text-driven manipulation directions, demonstrated on portraits of celebrities. Edits are performed using StyleGAN2 pretrained on FFHQ [24]. The inputs are real images, embedded in $\mathcal{W}+$ space using the e4e encoder [54]. The target attribute used in the text prompt is indicated above each column.

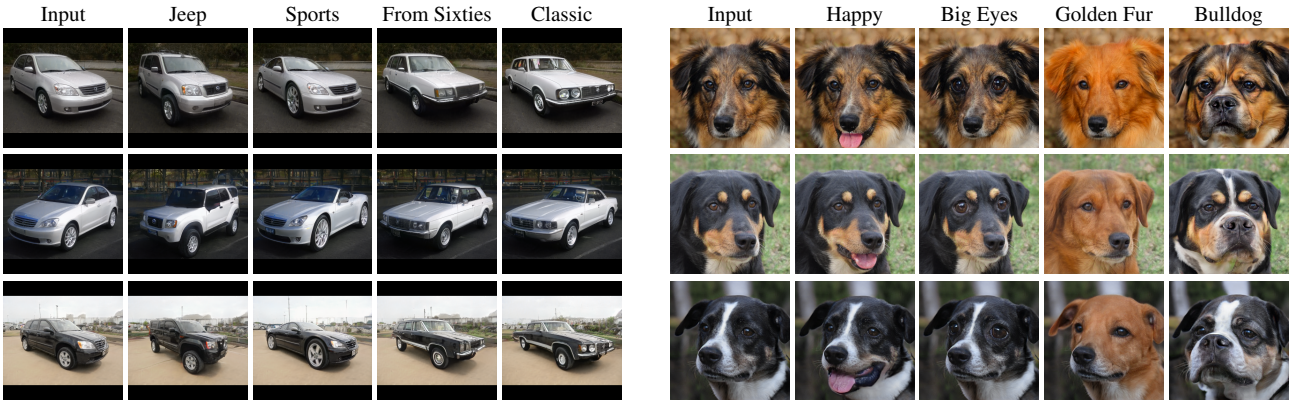


Figure 8. A variety of edits along global text-driven manipulation directions. Left: using StyleGAN2 pretrained on LSUN cars [61]. Right: using StyleGAN2-ada [23] pretrained on AFHQ dogs [8]. The target attribute used in the text prompt is indicated above each column.

somewhat different from the method presented in their paper. We do not include results of the optimization method presented in Section 4, since its sensitivity to hyperparameters makes it time-consuming, and therefore not scalable.

We perform the comparison using three kinds of attributes ranging from complex, yet specific (e.g., "Trump"), less complex and less specific (e.g., "Mohawk"), to simpler and more common (e.g., "without wrinkles"). The complex "Trump" manipulation, involves several attributes such as blonde hair, squinting eyes, open mouth, somewhat swollen face and Trump's identity. While a global latent direction is able to capture the main visual attributes, which are not specific to Trump, it fails to capture the specific identity. In contrast, the latent mapper is more successful. The "Mohawk hairstyle" is a less complex attribute, as it involves only hair, and it isn't as specific. Thus, both our methods are able to generate satisfactory manipulations. The manipulation generated by the global direction is slightly less pronounced, since the direction in CLIP space is an average one. Finally, for the "without wrinkles" prompt, the global direction succeeds in removing the wrinkles, while keeping other attributes mostly unaffected, while the mapper fails. We attribute this to $\mathcal{W}+$ being less disentangled. We observed similar behavior on another set of attributes ("Obama","Angry","beard"). We conclude that for complex and specific attributes (especially those that involve identity), the mapper is able to produce better manipulations. For simpler and/or more common attributes, a global

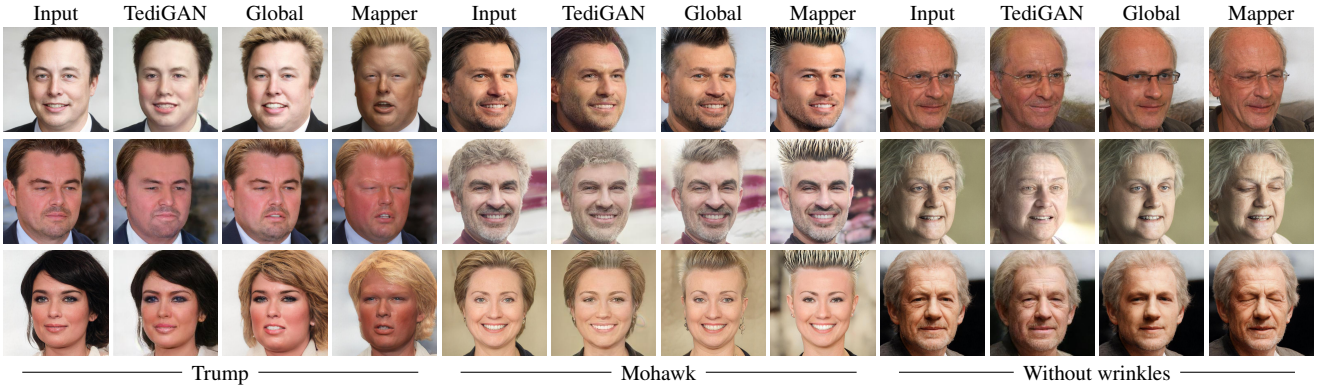


Figure 9. We compare three methods that utilize StyleGAN and CLIP using three different kinds of attributes.

direction suffices, while offering more disentangled manipulations. We note that the results produced by TediGAN fail in all three manipulations shown in Figure 9.

Other StyleGAN manipulation methods: In Figure 10, we show a comparison between our global direction method and several state-of-the-art StyleGAN image manipulation methods: GANSpace [19], InterFaceGAN [49], and StyleSpace [58]. The comparison only examines the attributes which all of the compared methods are able to manipulate (Gender, Grey hair, and Lipstick), and thus it does not include the many novel manipulations enabled by our approach. Since all of these are common attributes, we do not include our mapper in this comparison. Following Wu *et al*. [58], the manipulation step strength is chosen such that it induces the same amount of change in the logit value of the corresponding classifiers (pretrained on CelebA).

It may be seen that in GANSpace [19] manipulation is entangled with skin color and lighting, while in InterFaceGAN [49] the identity may change significantly (when manipulating Lipstick). Our manipulation is very similar to StyleSpace [58], which only changes the target attribute, while all other attributes remain the same.

In the supplementary material, we also show a comparison with StyleFLow [1], a state-of-the-art non-linear method. Our method produces results of similar quality, despite the fact that StyleFlow simultaneously uses several attribute classifiers and regressors (from the Microsoft face API), and is thus can manipulate a limited set of attributes. In contrast, our method requires no extra supervision.

Limitations. Our method relies on a pretrained StyleGAN generator and CLIP model for a joint language-vision embedding. Thus, it cannot be expected to manipulate images to a point where they lie outside the domain of the pretrained generator (or remain inside the domain, but in regions that are less well covered by the generator). Similarly, text prompts which map into areas of CLIP space that are not well populated by images, cannot be expected to yield a visual manipulation that faithfully reflects the semantics of the prompt. We have also observed that drastic manipulations in visually diverse datasets are difficult to achieve. For example, while tigers are easily transformed into lions (see Figure 1), we were less successful when transforming tigers to wolves, as demonstrated in the supplementary material.

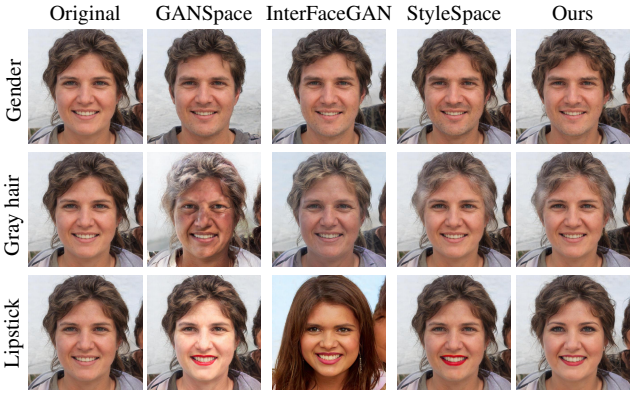


Figure 10. Comparison with state-of-the-art methods using the same amount of manipulation according to a pretrained attribute classifier.

8. Conclusions

We introduced three novel image manipulation methods, which combine the strong generative powers of StyleGAN with the extraordinary visual concept encoding abilities of CLIP. We have shown that these techniques enable a wide variety of unique image manipulations, some of which are impossible to achieve with existing methods that rely on annotated data. We have also demonstrated that CLIP provides fine-grained edit controls, such as specifying a desired hair style, while our method is able to control the manipulation strength and the degree of disentanglement. In summary, we believe that text-driven manipulation is a powerful image editing tool, whose abilities and importance will only continue to grow.

Acknowledgments We thank the anonymous reviewers for their comments. This work was supported in part by a gift from Adobe and by the Israel Science Foundation (grant no. 2492/20).

References

[1] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. StyleFlow: attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *CoRR*, abs/2008.02401, 2020.

[2] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a matter of style: Age transformation using a style-based regression model. *CoRR*, abs/2102.02754, 2021.

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement, 2021.

[4] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *CoRR*, abs/2103.10951, 2021.

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[6] T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, T. Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv*, abs/2005.14165, 2020.

[7] Yen-Chun Chen, Linjie Li, Licheng Yu, A. E. Kholy, Faisal Ahmed, Zhe Gan, Y. Cheng, and Jing jing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[8] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Proc. CVPR*, pages 8188–8197, 2020.

[9] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of GANs. *arXiv preprint arXiv:2004.14367*, 2020.

[10] Katherine Crowson. VQGAN-CLIP. https://github.com/nerdyrodent/VQGAN-CLIP, 2021.

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. CVPR*, pages 4690–4699, 2019.

[12] Karan Desai and J. Johnson. VirTex: Learning visual representations from textual annotations. *ArXiv*, abs/2006.06666, 2020.

[13] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[14] H. Dong, Simiao Yu, Chao Wu, and Y. Guo. Semantic image synthesis via adversarial learning. *Proc. ICCV*, pages 5707–5715, 2017.

[15] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proc. CVPR*, pages 12873–12883, 2021.

[16] Federico A. Galatolo, Mario G.C.A. Cimino, and Gigliola Vaglini. Generating images from caption and vice versa via CLIP-guided generative latent space search. *arXiv preprint arXiv:2102.01645*, 2021.

[17] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, https://distill.pub/2021/multimodal-neurons/, 2021.

[18] 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.

[19] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. *arXiv preprint arXiv:2004.02546*, 2020.

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, 2016.

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv:1710.10196*, 2017.

[22] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

[23] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020.

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4401–4410, 2019.

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, pages 8110–8119, 2020.

[26] J. Y. Koh, Jason Baldridge, H. Lee, and Yinfei Yang. Text-to-image generation grounded by fine-grained user attention. *arXiv*, abs/2011.03775, 2020.

[27] Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. VOGUE: Try-on by StyleGAN interpolation optimization. *arXiv:2101.02285*, 2021.

[28] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. ManiGAN: Text-guided image manipulation. In *Proc. CVPR*, pages 7880–7889, 2020.

[29] Gen Li, N. Duan, Yuejian Fang, Daxin Jiang, and M. Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *Proc. AAAI*, 2020.

[30] Liunian Harold Li, Mark Yatskar, Da Yin, C. Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557, 2019.

[31] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, X. He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. *Proc. CVPR*, pages 12166–12174, 2019.

[32] Xiujun Li, Xi Yin, C. Li, X. Hu, Pengchuan Zhang, Lei Zhang, Longguang Wang, H. Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020.

[33] Yahui Liu, Marco De Nadai, Deng Cai, Huayang Li, Xavier Alameda-Pineda, N. Sebe, and Bruno Lepri. Describe

what to change: A text-guided unsupervised image-to-image translation approach. *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[34] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild, 2015.

[35] Jiasen Lu, Dhruv Batra, D. Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

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

[37] Ryan Murdock and Phil Wang. Big Sleep. https://github.com/lucidrains/big-sleep, 2021.

[38] Ryan Murdock and Phil Wang. Deep Daze. https://github.com/lucidrains/deep-daze, 2021.

[39] Seonghyeon Nam, Yunji Kim, and S. Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. In *NeurIPS*, 2018.

[40] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. Face identity disentanglement via latent space mapping. *ACM Trans. Graph.*, 39(6), Nov. 2020.

[41] Victor Perez. Generating images from prompts using CLIP and StyleGAN. https://towardsdatascience.com/generating-images-from-prompts-using-clip-and-stylegan-1f9ed495ddda, 2021.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *Image*, 2:T2, 2021.

[43] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, and Scott Gray. DALL·E: Creating Images from Text. https://openai.com/blog/dall-e/, 2021.

[44] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv:2102.12092*, 2021.

[45] S. Reed, Zeynep Akata, Xinchen Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[46] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a StyleGAN encoder for image-to-image translation. *arXiv:2008.00951*, 2020.

[47] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. *arXiv preprint arXiv:2008.01392*, 2020.

[48] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *Proc. CVPR*, pages 9243–9252, 2020.

[49] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: interpreting the disentangled face representation learned by GANs. *CoRR*, abs/2005.09635, 2020.

[50] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in GANs. *CoRR*, abs/2007.06600, 2020.

[51] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *Proc. ICLR*, 2020.

[52] Hao Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP/IJCNLP*, 2019.

[53] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. StyleRig: Rigging StyleGAN for 3d control over portrait images. *arXiv preprint arXiv:2004.00121*, 2020.

[54] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *CoRR*, abs/2102.02766, 2021.

[55] 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*, volume 30, 2017.

[56] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the GAN latent space. *arXiv preprint arXiv:2002.03754*, 2020.

[57] Binxu Wang and Carlos R Ponce. A geometric analysis of deep generative image models and its applications. In *Proc. ICLR*, 2021.

[58] Zongze Wu, Dani Lischinski, and Eli Shechtman. StyleSpace analysis: Disentangled controls for StyleGAN image generation. *arXiv:2011.12799*, 2020.

[59] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. TediGAN: Text-guided diverse face image generation and manipulation. *arXiv preprint arXiv: 2012.03308*, 2020.

[60] T. Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and X. He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proc. CVPR*, pages 1316–1324, 2018.

[61] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[62] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proc. ICCV*, pages 5907–5915, 2017.

[63] Han Zhang, T. Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:1947–1962, 2019.