StyleMC: Multi-Channel Based Fast Text-Guided Image Generation and Manipulation

Umut Kocasari  Alara Dirik  Mert Tiftikci  Pinar Yanardag
Boğaziçi University
Istanbul, Turkey
{umut.kocasari, alara.dirik, mert.tiftikci}@boun.edu.tr, yanardag.pinar@gmail.com

Figure 1. Examples of text-driven manipulations using our method. For each example, the middle column shows the input images, left and right columns show manipulated results toward negative and positive directions, respectively.

Abstract

Discovering meaningful directions in the latent space of GANs to manipulate semantic attributes typically requires large amounts of labeled data. Recent work aims to overcome this limitation by leveraging the power of Contrastive Language-Image Pre-training (CLIP), a joint text-image model. While promising, these methods require several hours of preprocessing or training to achieve the desired manipulations. In this paper, we present StyleMC, a fast and efficient method for text-driven image generation and manipulation. StyleMC uses a CLIP-based loss and an identity loss to manipulate images via a single text prompt without significantly affecting other attributes. Unlike prior work, StyleMC requires only a few seconds of training per text prompt to find stable global directions, does not require prompt engineering and can be used with any pre-trained StyleGAN2 model. We demonstrate the effectiveness of our method and compare it to state-of-the-art methods. Our code can be found at http://catlab-team.github.io/stylemc.
1. Introduction

Generative Adversarial Networks (GANs) [9] have revolutionized generative modeling in computer vision since their emergence. Due to their powerful image synthesis capabilities, they are widely used for various visual tasks including image generation [37], image manipulation [31], super-resolution [27], and domain translation [39].

Despite their success, how to control the results generated by GANs remains an active research question. Previous work on controlled generation has shown that it is possible to generate images that belong to certain categories or have certain attributes by training conditional models [18]. However, conditional GANs require large amounts of labeled data for each target attribute. InfoGAN [4] is an approach that aims to develop models that generate a disentangled latent space in which each latent dimension controls a particular attribute. However, these approaches provide only limited control, depending on the granularity of available supervised information.

Recent research addressing these problems and aiming at controllable generation includes simple methods such as modifying the latent codes of images [23] and more complex methods such as searching for directions and interpolating latent vectors within pre-trained GAN models such as StyleGAN [13]. Another branch of work aims at finding disentangled directions in the latent space of GANs in a more principled way. Most of this work discovers domain-independent and interpretable directions such as zoom-in, rotation, and translation [21, 10, 30], while other frameworks propose to find a set of domain-specific directions such as hair color or gender on face images [25] or cognitive features [8]. The directions found are then used to modify a generated image by controlling the latent code by a certain amount to enhance or negate the target attribute in the generated image. Other work uses style space of StyleGAN2 to discover disentangled attributes and manipulate images for both coarse (e.g., gender, identity) and fine (e.g., hairstyle, eyes) visual features [33]. Recent work has also shown that these image manipulation methods can also be applied to real images by finding a latent code that accurately reconstructs the input image [24]. The latent code of the inverted real image can then be fed as input to GAN to perform processing operations directly on real images [1].

Recent works such as StyleCLIP and Paint by Word [2] uses CLIP [22] to manipulate real-world images via user-specified text prompts. However, both methods require hours of preprocessing or training to find stable directions. In this work, we propose a method to find image-independent manipulation directions in the latent space of pre-trained StyleGAN2 models using user-specified text prompts. Our method takes advantage of the joint representational power of CLIP and the generative power of StyleGAN2 while benefiting from the following key observations:

- We use style space of StyleGAN2 which is shown to be its most disentangled latent space [33]. Using the style space, we find multiple style channels to compute a global direction that can perform complex manipulations.
- Unlike previous work, our method finds directions using only layers up to 256 × 256 resolution within StyleGAN2, providing a significant speedup. We then use the found directions to apply manipulations and generate images at high resolutions such as 1024 × 1024.
- Our method uses only 128 randomly generated images to find stable and global manipulation directions regardless of the given text prompt.

Unlike previous work, such as StyleCLIP, our approach requires only a few seconds of training to find stable directions and does not require prompt engineering. In addition, our method is input agnostic, and can be applied to inverted real images as well as randomly generated images. We demonstrate the manipulation capabilities and efficiency of our method on a variety of datasets. Our results show that the discovered directions can successfully perform the desired processing while operating significantly faster.

The rest of this paper is organized as follows. Section 2 discusses related work on latent space manipulation. Section 3 presents our framework and Section 4 discusses quantitative and qualitative results. Section 5 discusses the limitations and implications of our work and Section 6 concludes the paper.

2. Related Work

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are two-part networks consisting of a generator and a discriminator [9] trained simultaneously in an adversarial manner. StyleGAN [13] and StyleGAN2 are among the popular GAN approaches that generate high-quality images. They map the input latent code to an intermediate latent space using a mapping network. BigGAN [3] is another large-scale model that uses skip-z connections, as well as a class vector. In this work, we work with pre-trained StyleGAN2 models.
2.2. Latent Space Manipulation

Several methods have been proposed to exploit the latent space of GANs for image manipulation, which can be divided into two broad categories: supervised and unsupervised methods. Supervised approaches typically benefit from pre-trained attribute classifiers that train the optimization process to discover meaningful directions in the latent space, or use labeled data to train new classifiers that directly aim to learn directions of interest [8, 25]. Other work shows that it is possible to find meaningful directions in latent space in an unsupervised way [30, 11, 29]. GANSpace [10] proposes to apply Principal Component Analysis (PCA) [32] to randomly sampled latent vectors of the intermediate layers of BigGAN and StyleGAN models. A similar approach is used in SeFA [26], where they directly optimize the intermediate weight matrix of the GAN model in closed form. LatentCLR [36] proposes a contrastive learning approach to find unsupervised directions that are transferable to different classes.

2.3. Text-Based Image Manipulation

Text-based image manipulation methods aim to generate images that contain visual attributes corresponding to the given text input without changing irrelevant attributes [7, 38, 19, 16, 17]. One of the recent works that uses image-text matching methods is TedGAN [34], which inverts real images using the inversion module of StyleGAN and then learns the correspondences between visual and linguistic attributes.

Other recent work uses image-text matching methods such as CLIP, to harness the power of joint image-text representations. CLIP is a multimodal contrastive learning framework with two encoder modules that aim to map the image and text pairs to the same embedding space. To achieve this, it maximizes the similarity between the embeddings of the matched image and text instances while minimizing that of the unmatched instance, resulting in a powerful bidirectional mapper. Recent works such as StyleCLIP and Paint by Word use CLIP to provide feedback to the generated images. StyleCLIP provides three different methods, namely Latent Optimization, Latent Mapper and Global Directions, which we refer to as StyleCLIP-LO, StyleCLIP-LM and StyleCLIP-GD for the rest of this paper. StyleCLIP-LO directly optimizes the latent code given an image and a text prompt. StyleCLIP-LM uses a latent residual mapper trained on a particular text prompt. StyleCLIP-GD maps a text prompt in an input-independent global direction. Paint by Word uses user-specified masks to perform manipulations within a specified region. However, both approaches require hours of preprocessing and training time. Our method aims to overcome this limitation by proposing a more efficient way to find stable and accurate directions to introduce or emphasize desired attributes in images.

3. Methodology

3.1. Background on Style Space

The generation process of StyleGAN2 consists of several latent spaces, namely \(Z, W, W^+\) and \(S\). More formally, let \(G\) denote a generator acting as a mapping function \(G: Z \rightarrow X\) where \(X\) is the target image domain. The latent code \(z \in Z\) is drawn from a prior distribution \(p(z)\), typically chosen to be Gaussian. The \(z\) vectors are transformed into an intermediate latent space \(W\) using a mapper function consisting of 8 fully connected layers. The latent vectors \(w \in W\) are then transformed into channel-wise style parameters, forming the \(style\) space, denoted \(S\), which is the latent space that determines the style parameters of the image. The \(W^+\) space is an extended version of \(W\) that uses a different intermediate latent vector on each layer of the synthesis network. It has been shown that the \(W^+\) space better reflects the disentangled nature of the latent space than the \(W\) space and is therefore more commonly used for image inversion [28].

The synthesis network of the generator in StyleGAN2 consists of several blocks, each block having two convolutional layers for synthesizing feature maps. Each main block has an additional \(1 \times 1\) convolutional layer that maps the output feature tensor to RGB colors, referred to as \(tRGB\). The three different style code vectors are denoted as \(s_{11}, s_{12},\) and \(s_{iRGB}\), where \(i\) indicates the block number. Given a block \(i\), the style vectors \(s_{11}\) and \(s_{12}\) of each block consist of style channels that control disentangled visual attributes. The style vectors of each layer are obtained from the intermediate latent vectors \(w \in W\) of the same layer by three affine transformations, \(w_{i1} \rightarrow s_{i1}, w_{i2} \rightarrow s_{i2}, w_{i2} \rightarrow s_{iRGB}\).

In our work, we use the style space \(S\) to perform manipulations, as it is shown to be the most disentangled, complete and informative space [33] compared to \(W\) and \(W^+\).

3.2. StyleMC: Style Space based Multi-Channel Directions

Given a pre-trained StyleGAN2 generator \(G\) and a style code \(s \in S\), let \(G(s)\) represent the generated image. Our method takes a text prompt \(t\) such as ‘A woman wearing makeup’ as input and finds a manipulation direction \(\Delta s\) such that \(G(s + \Delta s)\) generates a manipulated image in which the target attribute specified by \(t\) is present or enhanced, while other attributes
randomly generated images from StyleGAN2 or to real images inverted by a StyleGAN2 encoder such as e4e [28]. Note that we do not optimize the latent code as in the StyleCLIP-LO method [20]. Instead, we find a global direction for a given text prompt \( t \) that can be applied to any image. Our method works significantly faster than the other methods by taking advantage of the following insights:

- **Low-resolution layers.** The training process of the StyleGAN2 model starts by focusing on low-resolution features and progressively focuses on higher resolutions, shifting its attention to finer details. As shown in [14], the low-resolution images are not significantly affected by the higher-resolution layers. Inspired by this observation, we only use layers up to \( 256 \times 256 \) to find a manipulation direction within the \( S \) space. This strategy eliminates the additional time required for computation within the style blocks corresponding to \( 512 \times 512 \) and \( 1024 \times 1024 \) resolutions.

- **Small batch of images.** We compute our loss function using only a small batch of randomly generated images to find the direction \( \Delta s \). Regardless of the text prompt, we find that using a batch of 128 images is sufficient to find stable and generalizable directions. The reason for this strategy is that the manipulation effect caused by the style channels remains consistent across all images [33].

- **Operating on \( S \).** We operate directly on \( S \)-space, which is shown to be more disentangled than \( W \) and \( W^+ \) spaces [33]. Our method is significantly faster than other methods at finding the desired direction \( \Delta s \) due to the disentangled nature of the \( S \) space. In addition, our framework identifies and uses multiple style channels to compute directions and hence, can perform more complex edits. As shown in previous work [14, 33], while single channel manipulations can achieve edits such as changing the hair color or gender of a face image, they fail at more complex manipulations such as changing age, which typically requires a combination of multiple channels, such as wrinkle, grey hair and eyeglasses. Therefore, our method incorporates multiple style channels and captures complex manipulations such as age or change of personal identity while finding the desired direction.

Refer to the ablation study in Section 4.3 to see how these insights affect manipulation and computation time.
4. Experiments

We evaluate the proposed method on StyleGAN2 on a variety of datasets, including FFHQ [13], LSUN Car, Church, Horse [35], AFHQ Cat, Dog, Wild [5] and MetFaces [12] datasets. We also compare our method to the state-of-the-art text-based manipulation methods StyleCLIP and TediGAN, and the unsupervised methods GANspace, SeFa, and LatentCLR. Next, we discuss our experimental setup and then present results for several StyleGAN2 models.

4.1. Experimental Setup

For manipulation experiments on real images, we use the e4e method and map their latent codes from the $W^+$ space to the $S$ space. The dimension of the $s$ vectors we use to generate images is 9088. Following [33], we exclude $s_{RGB}$ layers since they cause entangled manipulations and transform the entire image. Additionally, we exclude style channel parameters of the last 4 blocks when finding $\Delta s$ as they represent very fine-grained features and are difficult to be used for editing tasks. For all experiments, we set the coefficient of the CLIP loss term to 1, while the coefficient of the identity loss term takes values between 0.1 and 6. Since it is desirable to retain features other than the target attribute, e.g., facial identity for FFHQ, the identity loss is increased so that the found direction is not affected by common model biases such as making target face younger as applying makeup. However, for manipulations that involve a complete change in identity, such as Donald Trump, the identity loss coefficient should be set to a low value. In most of our experiments, we used a coefficient value of 0.1, 0.5, or 2, depending on the complexity of the manipulation.

Figure 3. A variety of manipulations on StyleGAN2 FFHQ model. Rows 1-4 shows inverted real images and Rows 5-8 shows randomly generated images. The text prompt used for the manipulation is above each column.
4.2. Qualitative Results

Our method is capable of performing edits in a variety of domains (see Figure 1). Moreover, Figure 3 shows that our method can perform complex edits on the FFHQ dataset, ranging from stylistic edits such as *curly hair* or *beard* to emotional edits such as *relieved* or excited. As Figure 4 (a) shows, our method can successfully perform a variety of simple and complicated edits such as *white horse* and *vintage car* with models trained on the LSUN Car and LSUN Horse datasets. Furthermore, Figure 4 (b) shows that our method can handle a variety of simple (e.g., *fur color*, *eyes*) and complex (e.g., *species*, *expression*) manipulations with models trained on the AFHQ Cat and AFHQ Dog datasets. Finally, Figure 4 (c) shows manipulations with a model trained on the MetFaces dataset.

4.3. Ablation Study

Our method benefits from several important insights, such as using a small batch size and low-resolution layers to find manipulation directions for speedup. We also find multiple channels and use identity loss for manipulation quality and effectiveness. In this section, we perform ablation studies to understand the contribution of each component.

**Low-resolution layers** Our method is able to find directions at low-resolution layers to speedup the process. To test the effectiveness of this approach, we find directions at resolutions $128 \times 128$, $256 \times 256$, and $1024 \times 1024$ (see Figure 5 (b)). Our results show that $128 \times 128$ does not encapsulate enough signals to find suitable directions, while $256 \times 256$ and $1024 \times 1024$ achieve comparable manipulations. In terms of computation time, working with resolution $128 \times 128$ requires 3s, while $256 \times 256$ and $1024 \times 1024$ require 5s and 18s, respectively. Therefore, we use only the layers up to $256 \times 256$ resolution to find the desired directions while achieving a significant speedup, and then use the found directions to apply manipulations at high resolutions such as $512 \times 512$ or $1024 \times 1024$.

**Small batch size** Our method uses only 128 images to find directions. To understand how the results and computation time change with different batch sizes, we apply our method to a batch of 32, 128, and 1024 images (see Figure 5 (b)). Our experiment shows that using a batch of 32 images leads to slightly unnatural manipulations, while a batch of 128 images can compete with the results obtained with 1024 images. In terms of computation time, determining...
direction with 32 images requires 1.2s, while 128 and 1024 require 5s and 35s, respectively. Thus, using a batch of 128 images leads to the desired manipulation performance while providing significant speedup. We also note that the directions are not sensitive to the attributes of the images in the batch, and that using the same set of images for any given text prompt and 1 requires 5s and 35s, respectively. Thus, using a batch of 128 images leads to the desired manipulation performance while providing significant speedup. We also note that the directions are not sensitive to the attributes of the images in the batch, and that using the same set of images for any given text prompt works in practice.

**Single-channel vs. Multi-channel** Next, we perform an ablation study by manipulating images using a single channel. Similar to our multi-channel method, we use a CLIP-based loss:

$$L_s = 1 - \langle CLIP(G(s + \Delta s)), CLIP(t_1) - CLIP(t_2) \rangle$$

(4)

where \( s \) corresponds to the latent code of the image and \( t_1 \) and \( t_2 \) are the user-specified text inputs. Our experiments show that single-channel yields less stable directions for a single text prompt \( t \). To mitigate this effect, we use two text prompts \( t_1 \) and \( t_2 \), where \( t_1 \) contains the target/positive attributes (e.g. ‘A man with Mohawk hairstyle’) and \( t_2 \) the neutral/negative version of the target attribute (e.g. ‘A man with hair’). We also exclude identity loss, since the single-channel based changes are disentangled and affect only a single attribute without changing the identity of the image.

Figure 5 (a) shows a comparison between single-channel and multi-channel manipulations. We note that the single-channel approach can successfully handle simple manipulations such as blonde without affecting other attributes, while it fails at more complex manipulations such as mohawk. This is because complex style changes such as mohawk cannot be performed by manipulating a single channel, and therefore result in unrelated changes such as adding eyeglasses. On the other hand, our multi-channel method can successfully handle the complex mohawk manipulation and is able to find a more prominent blonde style.

<table>
<thead>
<tr>
<th>Method</th>
<th>Preproc</th>
<th>Training Time</th>
<th>Inference Time</th>
<th>Input Agnostic</th>
<th>Latent Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>StyleCLIP-LO</td>
<td>-</td>
<td>-</td>
<td>98s</td>
<td>no</td>
<td>W+</td>
</tr>
<tr>
<td>StyleCLIP-LM</td>
<td>-</td>
<td>10-12h</td>
<td>75ms</td>
<td>no</td>
<td>W+</td>
</tr>
<tr>
<td>StyleCLIP-GD</td>
<td>4h</td>
<td>-</td>
<td>72ms</td>
<td>yes</td>
<td>S</td>
</tr>
<tr>
<td>TediGAN</td>
<td>-</td>
<td>12h+</td>
<td>21s</td>
<td>yes</td>
<td>W+</td>
</tr>
<tr>
<td>Ours</td>
<td>-</td>
<td>5s</td>
<td>65s</td>
<td>yes</td>
<td>S</td>
</tr>
</tbody>
</table>

Table 1. Time comparison of our method with three StyleCLIP methods and TediGAN.

**4.4. Comparison with Text-Guided Methods**

We compare our method to state-of-the-art text-driven manipulation methods, StyleCLIP-LM, StyleCLIP-GD and TediGAN for manipulating real face images with text prompts. Similar to [20], we excluded StyleCLIP-LO as it suffers from self-reported unstability issues. For comparison, we used the StyleGAN2 model trained on FFHQ and used a set of complex text prompts such as ‘Trump’, ‘Mohawk’ and ‘Makeup’ (see Figure 6). Most of these prompts require multiple attributes to be changed simultaneously, such as hair color, eyes, mouth, facial expression, and facial structure. For the ‘Donald Trump’ prompt, StyleCLIP-GD leads to some visual style changes but fails to capture the identity of Trump, while the TediGAN method leads to insignificant style changes. In contrast, our method and the StyleCLIP-LM method capture important features specific to the target identity, such as puffy face, blonde hair, pink skin, and squinty eyes. For the ‘Mohawk’ prompt, StyleCLIP-GD and TediGAN produce minimal changes, while our method produces competitive results with StyleCLIP-LM. For the ‘Makeup’ prompt, all methods achieve the target manipulation to some degree.

We compare the computation time for finding the manipulation directions and editing the images in Table 1. Compared to the other methods, our method does not require prompt engineering and takes significantly less time to find and perform the manipulations. More specifically, the StyleCLIP-LO method solves an
optimization problem in $W^+$ space to optimize the latent code directly, and requires several minutes of optimization to edit a single image. Although there is no preprocessing or training time, it is input-dependent and leads to unstable results. The StyleCLIP-LM method operates in $W^+$ space and trains a mapper network for a given text prompt. However, this method is also input dependent and requires 10-12 hours of training. StyleCLIP-GD finds an input-independent global direction instantaneously for any text prompt, but only after 4 hours of preprocessing. The disadvantage of this technique is that it requires hours of computation before any manipulation can be done, and it does not work well for complex and specific attributes, as can be seen in Figure 6. The TediGAN method, on the other hand, requires 12+ hours of preprocessing as it encodes both the image and the text into the latent space and trains an image inversion and encoding module. Nevertheless, TediGAN yields unfavorable results compared to our method and StyleCLIP methods. In contrast, our method is independent of the input and requires 5 seconds\(^2\) of training to find stable and global manipulation directions.

### 4.5. Comparison with Unsupervised Methods

Next, we compare how the directions found on FFHQ differ across state-of-the-art unsupervised methods such as GANspace, SeFa and LatentCLR. Figure 7 shows the visual comparison between directions that are commonly found by all methods, including Smiley, Lipstick and Young directions. As can be seen from the visuals, all methods perform similarly and are able to manipulate the images towards the desired attributes.

### 5. Limitations and Broader Impact

Our method is based on pre-trained StyleGAN2 and CLIP models, so the manipulation capabilities strongly depend on the datasets they were trained on. We note that while the joint representation capabilities of CLIP are powerful, they are still limited and may be biased towards certain attributes. Our framework has similar concerns as any other image synthesis tool that can be used for malicious purposes, as discussed in [15].

### 6. Conclusion

We introduced a fast and efficient method for text-guided image generation and manipulation. Unlike previous work that requires hours of preprocessing or training, our method requires only a few seconds per text prompt to find a global manipulation direction and generates results that are on par with state-of-the-art models such as StyleCLIP in terms of quality. In our experiments, we have shown that our method can be used to apply a variety of manipulations ranging from simple style changes such as hair color to complex changes such as gender, personal identity, species and provide control over the strength and direction of the manipulations.

### Acknowledgments

This publication has been produced benefiting from the 2232 International Fellowship for Outstanding Researchers Program of TUBITAK (Project No: 118c321). We also acknowledge the support of NVIDIA Corporation through the donation of the TITAN RTX GPU and GCP research credits from Google.

---

\(^2\)The time is reported as an average of 100 trainings.
References


