Style Intervention: How to Achieve Spatial Disentanglement with Style-based Generators?

Yunfan Liu$^{1,3}$  Qi Li$^{1,2}$  Zhenan Sun$^{1,2,3}$  Tieniu Tan$^{1,2,3}$

$^1$ Center for Research on Intelligent Perception and Computing, CASIA  
$^2$ National Laboratory of Pattern Recognition, CASIA  
$^3$ School of Artificial Intelligence, University of Chinese Academy of Sciences  

yunfan.liu@cripac.ia.ac.cn, {qli, znsun, tnt}@nlpr.ia.ac.cn

Abstract

Generative Adversarial Networks (GANs) with style-based generators (e.g. StyleGAN) successfully enable semantic control over image synthesis, and recent studies have also revealed that interpretable image translations could be obtained by modifying the latent code. However, in terms of the low-level image content, traveling in the latent space would lead to ‘spatially entangled changes’ in corresponding images, which is undesirable in many real-world applications where local editing is required. To solve this problem, we analyze properties of the ‘style space’ and explore the possibility of controlling the local translation with pre-trained style-based generators. Concretely, we propose ‘Style Intervention’, a lightweight optimization-based algorithm which could adapt to arbitrary input images and render natural translation effects under flexible objectives. We verify the performance of the proposed framework in facial attribute editing on high-resolution images, where both photo-realism and consistency are required. Extensive qualitative results demonstrate the effectiveness of our method, and quantitative measurements also show that the proposed algorithm outperforms state-of-the-art benchmarks in various aspects.

1. Introduction

Generative Adversarial Networks (GANs) [10] with style-based generators have received significant research attention as they enable interpretable control of synthesized images at different scales [15]. The main difference between these networks and traditional generators is that each convolutional layer is weighted via adaptive instance normalization (AdaIN [13, 20]) with parameters computed based on the input latent code. Therefore, adjusting these ‘style coefficients’ modifies the relative strength of feature maps at each scale, which enables direct manipulation of the generation result and thus is widely adopted in many studies [14, 19, 27, 29].

Due to the high controllability of style-based generators, numerous studies have been conducted to analyze the organization of latent space and solve for latent directions for interpretable semantic changes. Unsupervised approaches [9, 11, 22, 26] use classical unsupervised machine learning techniques, e.g. Principal Component Analysis (PCA), to discover the collection of latent directions for semantic control. Supervised methods [3, 5, 21, 25], on the other hand, edit latent codes under the guidance of attribute labels in certain translation tasks. However, although semantic attributes of the input image could be correctly manipulated by existing methods, the translation of specific image content remains ‘spatially entangled’, as shown in Figure 1. This makes existing methods impractical in many real-world applications where local editing is required (e.g. portrait manipulation).

One intuitive solution for this problem would be introducing a binary mask to select the foreground and merge background to the edited image. However, it is extremely difficult, if not impossible, to obtain an accurate mask for

Figure 1. Examples of spatial entanglement in image translation. Names of the object to be manipulated are labeled on the left and the attribute for each row is annotated underneath.
translation with large shape deformations. From another perspective, manipulating feature maps along the spatial dimension [4, 24], e.g., introducing ‘dense style code’ which is spatially-variant, would help restrict image changes within the target area. However, this would heavily increase the number of parameters in the model as well as the cost of training, especially for images with high resolutions.

Notably, given any object-level visual concept (e.g., a facial component) in an image, both its semantic and spatial information are encoded in certain internal representations of the generator [6, 7]. Therefore, limiting manipulations to normalizing coefficients (i.e., style codes) of those feature maps would largely reduce the influence on synthesizing other image content. To this end, instead of the highly entangled latent space, we investigate the ‘style space’, i.e., the space spanned by all possible style vectors, to achieve fine-grained controls on local translations. Concretely, we propose a lightweight optimization-based framework, named ‘Style Intervention’, that could discover feature maps closely relevant to the target visual concept and perform precise image manipulation. Unlike previous methods, Style Intervention does not train any additional network (as in [5, 25]), or require the annotation of any extra attribute (as in [21]), since it makes the full use of intermediate variables. Although we stick to the generator of StyleGANv2 [16] in this paper, Style Intervention is a general framework which could adapt to any style-based generator, and the objective function could also be flexibly customized to suit various translation tasks. We verify the effectiveness of the proposed algorithm by the problem of facial attribute editing on high-resolution images, and extensive experimental results demonstrate that our method outperforms previous state-of-the-art in terms of both visual fidelity and spatial disentanglement.

Our main contributions could be summarized as follows,

1. Unlike most previous studies focusing on the input latent space, we investigate the style space of style-based generators to achieve spatial disentanglement. To the best of our knowledge, this work is the first to explore the possibility of controlling precise local translations with globally applied style coefficients.

2. We propose Style Intervention, a general optimization-based solution for precise image translation applicable to any style-based generator. It is lightweight, as no additional network structure or annotation data is required, and flexible, as the objective function could be customized to fit any translation task.

3. We choose the task of facial attribute editing as the test bench of the propose algorithm. Extensive experiments are conducted on high-resolution images and results demonstrate the advantage of our method in controlling local translation while ensuring the visual fidelity of generated images.

2. Related Work

2.1. Latent Space Analysis of GANs

Due to the remarkable success of StyleGAN [15] and StyleGANv2 [16] in image synthesis, many studies have been conducted to analyze the property of their latent space where style codes are computed.

Unsupervised approaches [9, 11, 22, 26] adopt classical unsupervised machine learning techniques, e.g., PCA, to solve for important latent directions and interpret their semantic meanings in generation results. Supervised methods, on the other hand, solve for the manipulated latent vector with the supervision of semantic labels. StyleRig [25] further parameterizes the latent space of StyleGAN using a 3DMM model and achieve accurate 3D control for portrait images. GANPaint [5] proposes to use another deep network to generate perturbations that correct intermediate feature maps of the generator. InterFaceGAN [21] trains linear support vector machines (SVMs) to classify latent codes based on attribute annotations, where the normal vector of each hyperplane represents the latent direction of the corresponding attribute.

2.2. Facial Attribute Editing with GANs

Facial attribute editing (FAE) is one of the most important research topics of image translation due to its wide range of practical applications. FAE aims at authentically manipulating the target attribute of an input face image while keep irrelevant image content intact. To solve this problem, numerous methods built on conditional GAN models [8, 12, 17, 18, 28] are proposed, and they typically focus on designing new loss functions or improving network structures. However, when it comes to images with high resolutions (e.g., 1024 × 1024), training these models from scratch is computationally expensive and the quality of generation results is hardly satisfying. Therefore, researchers have resorted to reusing large-scale style-based generators pre-trained on high-quality images as powerful generative priors, and control their behavior by manipulating the latent code along interpretable semantic directions.

3. Method

A style-based generator \( G : \mathcal{Z} \to \mathcal{I} \) consists of a mapping network \( f : \mathcal{Z} \to \mathcal{S} \), which computes the style code \( s \) based on the input latent vector \( z \), followed by a synthesis network \( h : \mathcal{S} \to \mathcal{I} \), which modulates each feature map by component of \( s \) and renders the final output \( (G = f \circ h) \). Due to such complex structure of \( G \), traveling in the input latent space \( \mathcal{Z} \), as in most previous stud-
In previous studies on model intervention [6, 7, 9, 11, 21, 22, 25, 26], we denote the collection of all feature maps are globally applied. Following the convention in previous age translations by modifying style codes although they inconsistently align with the spatial distribution of various classes.

This demonstrates the potential feasibility of localizing images of superfluous feature maps, which increases the chance of modifying non-relevant ones.

Similar to previous studies on model intervention [6, 7, 23], we observe that feature maps at certain channels consistently align with the spatial distribution of various classes of c, after being upsampled and thresholded (see Figure 2). This demonstrates the potential feasibility of localizing image translations by modifying style codes although they are globally applied. Following the convention in previous studies [7, 6], we denote the collection of all feature maps as U, and we use the subscript c and t to distinguish components (e.g. U and s) responsible for generating c from irrelevant ones.

3.1. Translation in the Style Space S

Following the previous discussion, two immediate questions one might ask would be 1) how to solve the division \( \mathbb{U} = (\mathbb{U}_c, \mathbb{U}_t) \), and 2) afterwards, how to edit s to precisely manipulate the semantic of c. These questions are extremely difficult to answer due to the intrinsic complexity of style-based generators, as modifying one single style code would affect all subsequent feature maps and thus may cause a series of unpredictable consequences.

However, this issue could be considered from another perspective, that is, the property of the resultant style code, denoted as \( \hat{s} \), after the manipulation. Based on Proposition 1, if \( f_\alpha \) is implemented by a separating hyperplane with the normal vector denoted as \( n_\alpha \), the corresponding displacement vector \( \Delta s = \hat{s} - s \) should satisfy the following conditions:

\[
\Delta s \approx k \cdot n_\alpha, \text{ where } k \text{ is constant (1)}
\]

\[
\Delta s_\pi = 0 \text{ (2)}
\]

Eq. 1 indicates that \( \hat{s} \) should render the desired translation effect of \( \alpha \), while Eq. 2 suggests that \( s_\pi \) is supposed to remain intact. In this way, interpolating along \( \Delta s \) would lead to the desired edited image after forwarding through h.

How to approach \( \Delta s_\pi \)? Based on Eq. 1 and Eq. 2, it is natural to approximate it using the normal vector of a separating hyperplane in \( S \), denoted as \( \Delta s_\pi \), which classifies style codes according to labels of the translated attribute \( \alpha \). Moreover, sparsity regularization is imposed on \( \Delta s_\pi \) to meet the requirement of Eq. 2, which encourages the displacement vector to modify fewer style codes. In this way, we unify translating the input image and reducing irrelevant changes in a single framework.

**Proposition 1.** Given the input style code s, a style-based synthesis network \( h: S \rightarrow \mathcal{I} \), and an attribute classifier \( f_\alpha : S \rightarrow \mathbb{R} \) for the attribute \( \alpha \) of c, where

\[
f_\alpha(s) \begin{cases} > 0 & \text{for } h(s) \text{ with positive label of } \alpha \\ < 0 & \text{for } h(s) \text{ with negative label of } \alpha \\ \end{cases}
\]

the style code \( \hat{s} \) of an ideally translated image should satisfy the following conditions,

\[
f_\alpha(s) \cdot f_\alpha(\hat{s}) < 0 \text{ (4)}
\]

\[
\hat{s}_\pi = s_\pi \text{ (5)}
\]

3.2. Style Intervention with Self-supervision

Although \( \Delta s_\pi \) has properties similar to the optimal style displacement vector \( \Delta \hat{s} \), it does not provide adequate clues to render natural translation effects. This is because \( \Delta s_\pi \) is obtained via a discriminative task, and thus only contains the minimum information to translate the input image so as to change the attribute label of c, which does not guarantee the realism of generation results (as shown in Figure 3).
However, since the result of translating in $Z (z \rightarrow z')$ is visually plausible, the corresponding displacement vector in $S$, i.e. $\Delta s = f(z') - f(z)$, should contain sufficient information to synthesize rich textural details. To this end, we naturally propose to combine the translation in both $Z$, which introduces realistic semantic changes but is short on spatial disentanglement, and $S$, which is able to exert precise modifications but may be less satisfactory in generating photo-realistic results, for authentic and accurate manipulation of image components.

To achieve this, an intuitive method would be changing each component in $\Delta s$, according to the corresponding value in $\Delta s_n$, and check whether it enhances the translation result. However, in most cases, the generation of $c$ jointly depends on multiple feature maps, and thus solving for one single component in $\Delta s$ at a time could not lead to reasonable results. Therefore, we propose to tackle the problem by solving for the intervention coefficient $s = \{s_i\}_{i=1}^n$, where $n$ is the number of convolutional layers. Each $s_i \in [0, 1] \cup \{0\}$ indicates the degree of intervention for all channels in the $i$th convolutional layer, where $l_i$ is the the number of feature maps within. Hence, the merged displacement of style code, denoted as $\Delta s_m$, could be computed as

$$\Delta s_m (\Lambda) = (1 - \Lambda) \cdot \Delta s + \Lambda \cdot \Delta s_n$$

The self-supervising objective function for solve the optimal intervention coefficient $\Lambda^*$ contains three parts:

- **Pixel-level Loss $\mathcal{L}_{pix}$**: We explicitly penalize $\Lambda$ for modifying image content other than $c$. Let us denote the binary mask for $c$ obtained from the semantic segmentation of $G(z)$ as $m_c$, where pixel values within $c$ is set to 1 and elsewhere to 0. The pixel-level loss could be formulated as

$$\mathcal{L}_{pix} = \|(1 - m_c \cdot (h(s + \Delta s_m (\Lambda)) - h(s))\|_2$$

- **Attribute Loss $\mathcal{L}_{attr}$**: To ensure that the edited image does present the target semantic change, we introduce the attribute loss $\mathcal{L}_{attr}$, which could be computed as

$$\mathcal{L}_{attr} = -\frac{\Delta s_n \cdot \Delta s_m (\Lambda)}{\|\Delta s_n\|_2 \cdot \|\Delta s_m (\Lambda)\|_2}$$

It is obvious that $\mathcal{L}_{attr}$ is exactly the cosine similarity between $\Delta s_n$ and $\Delta s_m (\Lambda)$, which encourage the resultant style code $\Delta s_m$ to still render the desired attribute change presented by $\Delta s_n$.

- **L2-norm Loss $\mathcal{L}_{norm}$**: Although introducing $\Delta s_n$ helps with disentangling spatial changes in image, it inevitably makes the resultant style code deviate from the manifold expanded by $s = f'(z)$, which could possibly make the corresponding image less natural. For this reason, we would like to limit the overall degree of intervention using the L2-norm Loss, which could be written as $\mathcal{L}_{norm}(\Lambda) = \|\Lambda\|_2$.

Therefore, the overall objective function could be written as

$$\mathcal{L} = \mathcal{L}_{pix} + \lambda_{attr} \mathcal{L}_{attr} + \lambda_{norm} \mathcal{L}_{norm}$$

where $\lambda_{attr}$ and $\lambda_{norm}$ control the relative importance of $\mathcal{L}_{attr}$ and $\mathcal{L}_{norm}$, respectively. The optimal intervention coefficient $\Lambda^*$ could be solved by $\Lambda^* = \arg \min_\Lambda \mathcal{L}$.

Since each convolutional layer is independently normalized [15], we propose a layer-wise optimization algorithm to reduce the interaction between different convolutional layers. Concretely, from the coarsest level ($i = 1$) to the finest level ($i = n$), we solve for one single $\lambda_i$ at a time while keep the intervention coefficient of other layers fixed. This is because feature maps with low spatial resolutions mainly control global attributes, and thus style codes of these channels should be determined before those adjusting fine-grained details. Please refer to the supplementary material for details of the proposed algorithm.

### 3.3. Differences Compared to Previous Methods

InterFaceGAN [21] is the study closest to ours, but we focus on disentanglement in the spatial dimension instead of the semantic level, which is more practical for real applications. Moreover, our method does not require labels of extra attributes to perform orthonormalization, not to mention that many image changes, such as identity, could hardly be described by attribute labels. Image2StyleGAN [1] and its improvement [2] proposes to investigate the extended latent space previous to affine layers, which aims at providing more freedom for GAN inversion. StyleRig [25] and GANPaint [5] involve training deep networks along with the StyleGAN model, which are much more computational expensive than our method. Unsupervised approaches [9, 11, 22, 26] are inappropriate for this task as they focus on discovering interpretable latent semantics, instead of solving for the latent direction for the target attribute.

### 4. Experiment

#### 4.1. Experimental Setup

**Data Preparation** We use the facial attribute editing (FAE) task, which has received considerable attention in recent studies on style-based generators, to verify the effectiveness of our method. In this paper, 12 attributes directions are considered for the qualitative experiment, which is a large extension comparing to previous studies [11, 21, 22, 26]. We collect 100,000 synthetic samples and adopt the online face analysis toolkit Face++ [4] to label...
5 facial attributes (‘Gender’, ‘Age’, ‘Eyeglasses’, ‘Mouth Open’, and ‘Smile’). For the rest attributes, we randomly select 10,000 samples and manually annotate them, which will be released soon. Please note our method could be easily extended to other tasks (e.g. scene editing), as input images are not required to be faces and other variables (e.g. labels and masks) could be easily obtained.

**Training Configurations** In this paper, we typically focus on the StyleGANv2 model [16] pre-trained on the FFHQ dataset [15] with resolution $1024 \times 1024$. We concatenate the output of all affine layers to form the style code $s \in \mathbb{R}^{9088}$. For the optimization process, we adopt the Adam optimizer and set balancing coefficients $\lambda_{\text{attr}}$ and $\lambda_{\text{norm}}$ to $1e^{-2}$ and $1e^{-6}$, respectively. Please refer to the supplementary material for more detailed information of how the mask $m_c$ is computed and what facial components are considered as the target for each attribute.

### 4.2. Separability of the Style Space $S$

In Section 3.1, we approximate the ideal displacement vector in $S$ by $\Delta s_n$, i.e. the normal vector of a hyperplane classifying style codes by attribute labels. Therefore, it is necessary to inspect the separability of style codes in the first place since this is the cornerstone of our method.

According to the result shown in Table 1, all facial attributes are almost linearly separable in the style space. This indicates that style codes are not just simple concatenations of normalization coefficients, but also contain discriminative semantic information of image attributes. Moreover, the high sparsity of normal vectors (over 93%) induced by the L1-norm suggests that only a small part of $U$ is responsible for ‘flipping’ the corresponding attribute, which also proves the possibility for disentangling spatial translation by restricting modifications of the style code to $s_c$.

### 4.3. Spatially Disentangled Image Translation

From figure 4, it is clear that the proposed method achieves spatial disentanglement on translations of different facial components in various semantic directions. For example, the hair region could be vividly modified to show different colors or styles, without affecting any other image content. This is because our method combines interventions in the style space $S$, which precisely locate feature maps controlling characteristics of the target object, with translations in the latent space $Z$, which provide rich textural information to render output images. Notably, our

![Sample results of manipulating different facial component along various semantic directions](image_url)

**Figure 4.** Sample results of manipulating different facial component along various semantic directions (labeled on the left).

| Attribute | Gender | Age | Smile | Eye-glasses | Mouth Open | Asian | Bangs | Black Hair | Blond Hair | Grey Hair | Straight Hair | Beard |
|-----------|--------|-----|-------|-------------|------------|-------|-------|------------|------------|-----------|---------------|-------|
| Train     | 97.81  | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 93.83 | 96.32 | 95.48 | 100.00 | 86.02 | 100.00 |
| Validate  | 93.00  | 98.14 | 97.70 | 90.58 | 97.59 | 80.40 | 85.03 | 82.98 | 78.54 | 94.34 | 81.17 | 90.14 |
| Sparcity  | 93.60  | 98.14 | 96.08 | 96.92 | 94.98 | 96.25 | 93.61 | 93.64 | 93.80 | 98.55 | 93.25 | 97.76 |

Table 1. Classification accuracy (%) of linear SVMs on both training and validation set. Sparcity values demonstrate the proportion (%) of zero elements in $\Delta s_n$. 

---

5 facial attributes (‘Gender’, ‘Age’, ‘Eyeglasses’, ‘Mouth Open’, and ‘Smile’). For the rest attributes, we randomly select 10,000 samples and manually annotate them, which will be released soon. Please note our method could be easily extended to other tasks (e.g. scene editing), as input images are not required to be faces and other variables (e.g.

---

5 facial attributes (‘Gender’, ‘Age’, ‘Eyeglasses’, ‘Mouth Open’, and ‘Smile’). For the rest attributes, we randomly select 10,000 samples and manually annotate them, which will be released soon. Please note our method could be easily extended to other tasks (e.g. scene editing), as input images are not required to be faces and other variables (e.g. labels and masks) could be easily obtained.

**Training Configurations** In this paper, we typically focus on the StyleGANv2 model [16] pre-trained on the FFHQ dataset [15] with resolution $1024 \times 1024$. We concatenate the output of all affine layers to form the style code $s \in \mathbb{R}^{9088}$. For the optimization process, we adopt the Adam optimizer and set balancing coefficients $\lambda_{\text{attr}}$ and $\lambda_{\text{norm}}$ to $1e^{-2}$ and $1e^{-6}$, respectively. Please refer to the supplementary material for more detailed information of how the mask $m_c$ is computed and what facial components are considered as the target for each attribute.

### 4.2. Separability of the Style Space $S$

In Section 3.1, we approximate the ideal displacement vector in $S$ by $\Delta s_n$, i.e. the normal vector of a hyperplane classifying style codes by attribute labels. Therefore, it is necessary to inspect the separability of style codes in the first place since this is the cornerstone of our method.

According to the result shown in Table 1, all facial attributes are almost linearly separable in the style space. This indicates that style codes are not just simple concatenations of normalization coefficients, but also contain discriminative semantic information of image attributes. Moreover, the high sparsity of normal vectors (over 93%) induced by the L1-norm suggests that only a small part of $U$ is responsible for ‘flipping’ the corresponding attribute, which also proves the possibility for disentangling spatial translation by restricting modifications of the style code to $s_c$.

### 4.3. Spatially Disentangled Image Translation

From figure 4, it is clear that the proposed method achieves spatial disentanglement on translations of different facial components in various semantic directions. For example, the hair region could be vividly modified to show different colors or styles, without affecting any other image content. This is because our method combines interventions in the style space $S$, which precisely locate feature maps controlling characteristics of the target object, with translations in the latent space $Z$, which provide rich textural information to render output images. Notably, our
method could also deal with the translation of abstract facial attributes, i.e., ‘Age’, ‘Gender’, and ‘Asian’, where we perform the foreground-background disentanglement.

However, one may challenge the effectiveness of our method by attributing the success of precise attribute translations to the explicitly enforced image-level consistency (i.e. Eq. 7), instead of the disentanglement of style codes. To this end, we fix $\Lambda^*$ and change the coefficient of $\Delta s_n$ to check if the translation could be interpolated only within the target area. If results in Figure 4 are simply outcomes of the optimization process where unwanted modifications happen to be minimized, this operation would again lead to changes in irrelevant regions. According to results shown in Figure 5, interpolating along $\Delta s_n$ in the style space does not affect other image content and smooth gradual changes are achieved. This proves that the result of our method is not simply the solution to a superficial optimization problem, but the outcome of a reasonable combination of translations in two spaces ($Z$ and $S$).

4.4. Visualization of Intermediate Results

Apart from verifying the effectiveness of our method in terms of final generation results, it would also be interesting to investigate the internal mechanism of the proposed algorithm. To this end, we employ the translation of attribute ‘Mouth Open’ as a case study, and visualize intermediate optimization results to gain a better understanding of how Style Intervention approaches the final output.

According to Figure 6 (a), it is clear that changes between successive convolutional layers are discrete and monotonous. The discreteness implies that certain facial component is treated as a whole and embedded in neighboring feature maps, instead of being treated as a collection of local texture patches distributed among all convolutional layers. For example, eyeglasses incorrectly synthesized by editing in $Z$ are removed as a whole by the 4th layer (red box), rather that gradually disappear through several steps. Similarly, the mustache is also accurately removed by the 10th layer (yellow box). This demonstrates the possibility of manipulating images on the basis of visual components (i.e. with spatial disentanglement) by modifying individual
style code, which is the theoretical basis of our method.

As for the monotonicity of image changes, it proves the rationality of performing the optimization in a layer-wise manner, that is, solving for the optimal intervention coefficient of one convolutional layer would not damage the outcome of optimizing previous layers. This is because the convolutional layers are interdependent since the input of each layer is normalized before being modulated according to the style code. Therefore, the discrepancy between each intermediate result and the input in non-target image regions keeps decreasing throughout the whole optimization process, which is also verified in Figure 6 (b) as the error drops monotonously. We have also tried to enlarge the range of interpolation in $Z$, and it turns out that although the error between input images and results of editing in $Z$ increases, the proposed method converges well and reaches low error at the final output layer.

4.5. Comparison against Benchmark Methods

4.5.1 GANs with Style-based Generators

In order to demonstrate the superiority of the proposed method, we compare manipulation results against several other benchmarks, including editing in $Z$-space and InterFaceGAN [21]. InterFaceGAN eliminates entangled semantic translations by subtracting the projection on the corresponding direction in the latent space. To avoid the bias of trained attribute classifiers, we only conduct experiments on four facial attributes that could be recognized by Face++ online APIs for objectivity (we do not include ‘Mouth Open’ as it largely overlaps with ‘Smile’).

As the result shown in Figure 7, it is clear that our method could disentangle manipulations of the target facial component from other image content, and simultaneously render realistic translations. On the contrary, editing in $Z$-space clearly produces spatially entangled modifications to the input image. Although InterFaceGAN largely solves the problem of semantic entanglement, as shown in Figure 7 (b) and (d), the results still suffer from undesirable changes in terms of specific image content (e.g., background texture and personal identity). These unwanted image modifications could hardly be described by abstract attribute labels and thus could not be eliminated by subtracting the latent projection, unless supervision on the image level is imposed.

Notably, as shown in Figure 7 (c), removing the translation along ‘Age’ also eliminate the target attribute change ‘Eyeglasses’. This is because eyeglasses are more likely to appear in images with aged faces, and thus these two visual concepts are heavily entangled in the latent space. To solve this problem, most of previous studies on style-based generators resort to the intermediate latent space, i.e., $W$-space, for more disentangled representations. Therefore, we also compare the performance of our method against editing in the $W$-space. Based on the results shown in Figure 8, it is clear that compared to editing in $W$, our method could still achieve better spatial disentanglement in manipulating individual visual concept.

To provide a more comprehensive evaluation of our method, quantitative experiments are also conducted. We
Table 2. Comparison of quantitative results measured by MSE, SSIM and ID. ‘EG’ is the abbreviation for ‘Eyeglasses’. The metric ‘ID’ is the confidence score of face verification obtained by the Face++ online analysis toolkit.

| Method          | Gender | Age | EG  | Smile |
|-----------------|--------|-----|-----|-------|
| Z-space         | 0.034  | 0.015 | 0.022 | 0.016 |
| W-space         | 0.018  | 0.011 | 0.010 | 0.016 |
| InterFaceGAN    | 0.040  | 0.013 | 0.013 | 0.015 |
| Ours            | 0.015  | 0.008 | 0.004 | 0.003 |

Figure 9. Comparison between our method and approaches based on conditional GANs: StarGAN (128 × 128) and AttGAN (256 × 256).

Table 3. Results of user study. For each question, the proportion (%) of votes favoring our method is reported.

| Metric              | vs. AttGAN | vs. StarGAN |
|---------------------|------------|-------------|
| Attribute Translation| 82.85      | 78.08       |
| Attribute Preservation| 98.10      | 94.76       |
| Image Quality       | 100.00     | 99.53       |

Adopt Mean Squared Error (MSE), Structural Similarity Index Measure (SSIM), and face verification score (ID) as metrics to measure how image content of the input is preserved in generation results at different levels. According to results reported in Table 2, our method outperforms benchmark approaches almost in all cases, demonstrating the effectiveness of the proposed method in rendering precise translations of local regions. This is desirable in various practical applications, such as portrait image editing, where the accurate manipulation of individual object is required.

4.5.2 GANs with Traditional Structures

Since traditional GANs, i.e., GAN models with successive convolutional layers instead of style injection, also integrate supervision on the consistency of image content, we also compare the performance of our method to these models for a comprehensive analysis. Concretely, we consider StarGAN [8] and AttGAN [12] as representatives for state-of-the-art FAE methods based on conditional GANs. The officially released models are used for testing, and we resize input images to corresponding resolutions 2 3.

According to results shown in Figure 9, similar to traditional models which are designed to preserve image content irrelevant to the target translation, our method could also maintain the consistency between input and output images. We also conduct user study to comprehensively compare the performance between our method and traditional cGANs. 45 subjects participate in the survey and each is presented by 110 samples. More specifically, each subject is asked to rank the performance of these methods on three aspects, i.e., target editing, input preservation, and image quality. According to the result shown in Table 3, our method is preferred by the vast majority of users. Notably, more than 94.76% subjects think the proposed method outperforms previous approaches in preserving non-target image content. This demonstrates the effectiveness of manipulating individual style code in achieving spatial disentanglement, since all these methods impose supervision on image-level consistency.

5. Conclusion

Due to the intrinsic complexity of style-based generators and the spatial invariance of AdaIN normalization, image translations obtained by recent studies are spatially entangled and thus it is undesirable in practical applications. In this paper, we lift the dimension of feature space and propose to intervene the style code directly for the precise manipulation of individual feature map. A lightweight and flexible optimization-based algorithm is proposed based on the in-depth observation of internal mechanism of style-based generators. Both qualitative and quantitative results have demonstrated the ability of our method in achieving spatially disentangled translation the target object could be

---

2StarGAN: https://github.com/yunjey/stargan
3AttGAN: github.com/elvisyjlin/AttGAN-PyTorch
manipulated alone. Our method could be easily generalized to tasks beyond facial attribute editing, and may possibly inspire the future design of generators to incorporate information of spatial layout of images.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 4432–4441, 2019. 1

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8296–8305, 2020. 2

[3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. arXiv preprint arXiv:2008.02401, 2020. 1

[4] Yazeed Alharbi and Peter Wonka. Disentangled image generation through structured noise injection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5134–5142, 2020. 2

[5] David Bau, Hendrik Strobelt, William Peebles, Bolei Zhou, Jun-Yan Zhu, Antonio Torralba, et al. Semantic photo manipulation with a generative image prior. arXiv preprint arXiv:2005.07727, 2020. 1, 2, 3, 4

[6] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6541–6549, 2017. 2, 3

[7] 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. 2, 3

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8789–8797, 2018. 2, 8

[9] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 5744–5753, 2019. 1, 2, 3, 4

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in Neural Information Processing Systems (NIPS), pages 2672–2680, 2014. 1

[11] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. arXiv preprint arXiv:2004.02546, 2020. 1, 2, 3, 4

[12] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. IEEE Transactions on Image Processing (TIP), 28(11):5464–5478, 2019. 2, 8

[13] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the International Conference on Computer Vision (ICCV), pages 1501–1510, 2017. 1

[14] Wentao Jiang, Si Liu, Chen Gao, Jie Cao, Ran He, Jiasheng Feng, and Shuicheng Yan. Psgan: Pose and expression robust spatial-aware gan for customizable makeup transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5194–5202, 2020. 1

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4401–4410, 2019. 1, 2, 3, 4, 5

[16] 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 Conference on Computer Vision and Pattern Recognition (CVPR), pages 8110–8119, 2020. 2, 3, 5

[17] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5549–5558, 2020. 2

[18] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3673–3682, 2019. 2

[19] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5084–5093, 2020. 1

[20] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2337–2346, 2019. 1

[21] Yujun Shen, Jinjin Gu, Xiaou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 9243–9252, 2020. 2, 3, 4

[22] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. arXiv preprint arXiv:2007.06600, 2020. 1, 2, 3, 4

[23] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. International Conference on Learning Representations (ICLR), 2013. 3

[24] Ryohei Suzuki, Masanori Koyama, Takeru Miyato, Taizan Yonetsuji, and Huachun Zhu. Spatially controllable im-
age synthesis with internal representation collaging. *arXiv preprint arXiv:1811.10153*, 2018.

[25] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6142–6151, 2020.

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

[27] Jiashun Wang, Chao Wen, Yanwei Fu, Haitao Lin, Tianyun Zou, Xiangyang Xue, and Yinda Zhang. Neural pose transfer by spatially adaptive instance normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5831–5839, 2020.

[28] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y. Chang, and Shih-Wei Liao. Relgan: Multi-domain image-to-image translation via relative attributes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5914–5922, 2019.

[29] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5104–5113, 2020.