Towards Spatially Disentangled Manipulation of Face Images With Pre-Trained StyleGANs

Yunfan Liu, Qi Li, *Member, IEEE*, Qiyao Deng, and Zhenan Sun, *Senior Member, IEEE*

***Abstract*— Generative Adversarial Networks with style-based generators could successfully synthesize realistic images from input latent code. Moreover, recent studies have revealed that interpretable translations of generated images could be obtained by linearly traversing in the latent space. However, in most existing latent spaces, linear interpolation often leads to 'spatially entangled modification' in the manipulation result, which is undesirable in many real-world applications where local editing is required. To solve this problem, we propose to manipulate the latent code in the *'style space'* and analyze its advantage in achieving spatial disentanglement. Furthermore, we point out the weakness of simply interpolating in the style space and propose *'Style Intervention'*, a lightweight optimization-based algorithm, to further improve the visual fidelity of manipulation results. The performance of our method is verified with the task of attribute editing on high-resolution face images. Both qualitative and quantitative results demonstrate the advantage of image translation in the style space and the effectiveness of our method on both real and synthetic images.**

***Index Terms*— Generative adversarial networks, style-based generators, facial attribute manipulation.**

I. Introduction

GENERATIVE adversarial networks with style-based generators (e.g., StyleGAN [1] and StyleGANv2 [2]) have received significant research attention in various computer vision tasks [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. Compared to the structure of previously proposed generator networks [13], [14], [15] (referred to as 'traditional generators' in this paper), convolutional layers in style-based generators are equipped with adaptive normalizing modules (e.g., adaptive instance normalization (AdaIN) [16], [17]) to compute modulating parameters from the latent code. These parameters adjust the relative importance of different feature maps, controlling the semantic of generated images at different scales. Therefore, given a pre-trained StyleGAN generator, semantic manipulation could be performed by simply modifying such normalizing coefficients (or *'style codes'*), which saves researchers from training the entire model from scratch.

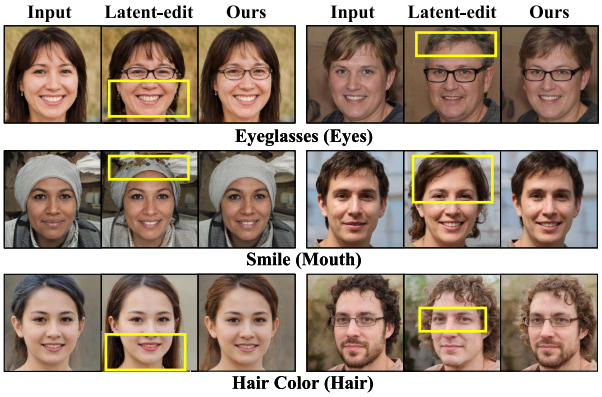

Fig. 1. Examples of spatial entanglement in image translation. The target attribute for each row is annotated underneath, and the related object for each attribute is labeled in parenthesis. 'Latent-edit' refers to results obtained by linear interpolation in the input latent space, and please note image changes irrelevant to the target object (within yellow boxes).

Due to the high controllability on generated images, numerous studies have been conducted to analyze the organization of the latent space of pre-trained StyleGAN models. Unsupervised approaches [18], [19], [20], [21], [22] typically use classical unsupervised machine learning techniques, e.g., Principal Component Analysis (PCA), to discover the collection of linear directions in the latent space to render interpretable semantic changes. Supervised methods [23], [24], [25], [26], on the other hand, edit the latent code under the guidance of attribute labels. However, although existing methods could successfully manipulate the semantic of input images, the translation remains 'spatially entangled', i.e., image content irrelevant to the target attribute is also modified (see Fig. 1). This makes existing methods impractical in many real-world applications where local editing is required.

One intuitive solution for this problem would be using a binary mask to select the target region and merge pixels elsewhere to the edited image. However, it is extremely difficult, if not impossible, to obtain an accurate mask for translation with large shape deformations on high-resolution images. Other studies attempt to solve this issue either by

Manuscript received 31 May 2022; revised 26 August 2022 and 19 September 2022; accepted 2 October 2022. Date of publication 10 October 2022; date of current version 5 April 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62276263, Grant 62076240, and Grant U1836217; and in part by the Beijing Natural Science Foundation under Grant 4222054. This article was recommended by Associate Editor Z. Yang. *(Corresponding author: Qi Li.)*

Yunfan Liu and Zhenan Sun are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the National Laboratory of Pattern Recognition, Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yunfan.liu@cripac.ia.ac.cn; znsun@nlpr.ia.ac.cn).

Qi Li and Qiyao Deng are with the National Laboratory of Pattern Recognition, Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: qli@nlpr.ia.ac.cn; dengqiyao@cripac.ia.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3213662.

Digital Object Identifier 10.1109/TCSVT.2022.3213662

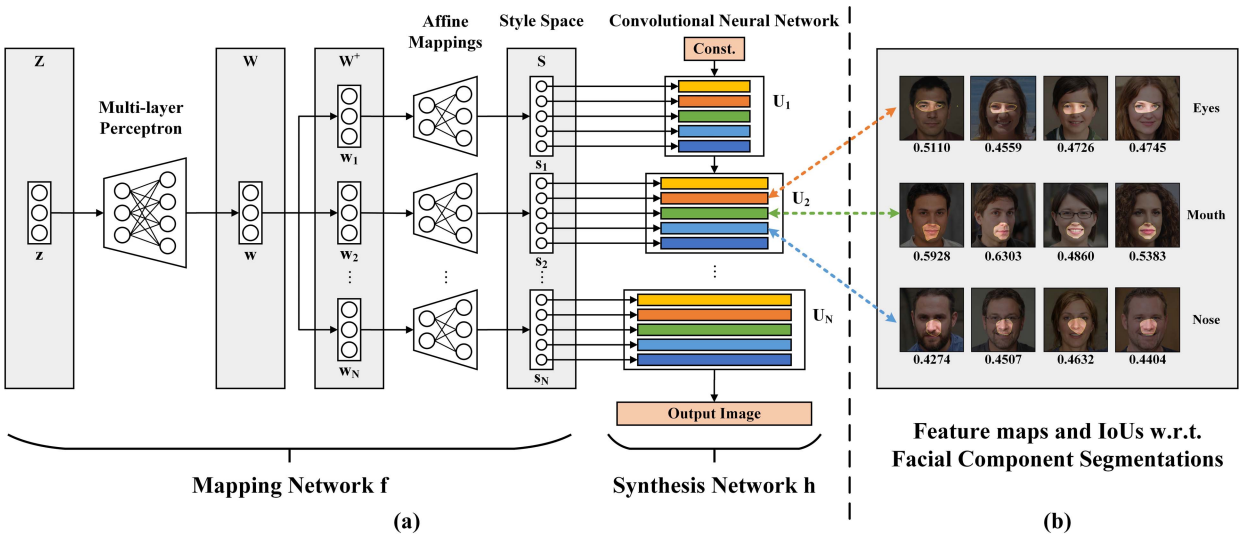

Fig. 2. A typical framework of style-based generators. (a) An illustration of the internal structure of a StyleGAN generator, which consists of a mapping network f and a synthesis network h. The mapping network f computes the style code s from the input latent code z via several steps, and the synthesis network h generates the output image with the guidance of s from a constant input. Each element of s is used to adjust the relative weight of a feature map within h. (b) An illustration of the overlap between feature maps and image content. We compute binarized feature maps by setting the top 5% activated spatial locations to 1 and others to 0, and overlay them with the corresponding image. We find out that feature maps at several scales consistently have large overlaps with the distribution of facial components in the output image. The intersection-over-union (IoU) values between binarized feature maps and facial component segmentations (shown on the right of each row) is labeled below each row.

introducing 'dense style code' which is spatially-variant [27], [28], [29], or by training auxiliary networks to maintain the pixel-level consistency [23], [24], [30]. However, these solutions heavily increase the number of parameters to be trained as well as the computational cost, especially for images with high resolutions.

To solve existing problems, we aim to propose a method to efficiently perform spatially disentangled manipulation of high-resolution images with a pre-trained StyleGAN generator. Notably, for any local object in a given image (e.g., a facial component), it is guaranteed that both its semantic and spatial information are encoded in certain feature maps within the convolutional network [31], [32]. Therefore, these feature maps are responsible for synthesizing the target object, and limiting the manipulation of corresponding normalization coefficients (i.e., style codes)[1] would largely reduce the influence on other image content, as the influence on other feature maps is suppressed. However, the intrinsically entangled structure of the mapping network in StyleGAN generators prevents manipulating single style code in existing latent spaces (i.e., $\mathcal{Z}$, $\mathcal{W}$, and $\mathcal{W}^+$), as modifying one single element of the latent code would potential change style code associated with superfluous feature maps (see Fig. 2 for details).

To this end, instead of existing highly entangled latent space, we investigate the **'style space'** (denoted by $\mathcal{S}$), i.e., the space spanned by all possible style codes, to achieve fine-grained controls on the translation of local objects. Concretely, extensive experiments are conducted to demonstrate the advantage of $\mathcal{S}$ over $\mathcal{Z}$, $\mathcal{W}$, and $\mathcal{W}^+$. Furthermore, we also point out the weakness of merely manipulating the style code, and propose a lightweight optimization-based framework, named **'Style Intervention'**, to deal with the problem and perform precise and realistic image manipulation. Unlike previous methods, Style Intervention does not train any additional deep network (as in [24] and [30]), or require the annotation of any extra attribute (as in [25]). We verify the effectiveness of the proposed algorithm with the task of facial attribute editing on high-resolution images, and extensive experimental results demonstrate that our method outperforms previous state-of-the-art approaches in terms of both visual fidelity and spatial disentanglement.

Our main contributions could be summarized as follows,

1) Based on the observation of the network structure of StyleGAN generators, we analyze both the advantage and weakness of the **style space** for spatially disentangled manipulation on high-resolution face images.
2) We propose an optimization-based algorithm, named **Style Intervention**, for precise image translation with high visual fidelity. It is lightweight, as no additional deep network training is involved, and flexible, as the objective function could be customized to fit various semantic translations.
3) We choose the problem of attribute editing on high-resolution face images as the test bench of the propose algorithm. Extensive experiments are conducted on both real and synthetic images, and results demonstrate the advantage of our method in controlling local translation while ensuring the visual fidelity of generated images.

II. Related Works

A. Facial Attribute Editing With Traditional GANs

Image-to-image translation (I2I) has long been a popular research topic in the computer vision community since the

[1]We use 'style codes' interchangeably with 'normalization coefficients' in this paper.

advent of GAN models [33], [34], [35], [36], [37]. Facial attribute editing (FAE) [38], [39], [40] is one of the most important research topics of I2I 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, many methods have been proposed and they could be generally divided into two categories, i.e., *traditional GAN based methods* and *pre-trained style-based generators based methods*.

The generator of traditional GAN based methods [41], [42], [43], [44], [45] typically receives latent variables, including semantic data (e.g., attribute label vectors) and the feature embedding of input images, at the input layer, and synthesize output images with a stack of convolutional layers. AttGAN [42] controls the semantic of output images by attribute vectors and regulates the behavior by image reconstruction. StarGAN [41] solves image translation among multiple domains with a single model, where the target domain is indicated with a label vector. Although satisfactory editing results could be obtained by these methods, they could hardly be adapted to images with high resolutions (e.g., 1024×1024) due to the heavy computational cost. Another solution would be editing down-sampled images and then restore the image via super-resolution [46], [47], [48]. However, this increases the chance of introducing noise, and ghosting artifacts caused by image editing may be enlarged by super-resolution. Therefore, researchers resort to re-using 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 to perform FAE.

B. GANs With Style-Based Generators

Inspired by studies on style transfer [16], [1] proposes the StyleGAN model for image synthesis, whose generator consists of a mapping network and a synthesis network. To generate an image, 'style codes' are first computed via the mapping network based on the input latent vector, and then incorporated into the synthesis network via adaptive instance normalization (AdaIN) modules. Mathematically, the modulation process could be written as

$$\mathrm{AdaIN}(\boldsymbol{x_i}, \boldsymbol{y}) = \boldsymbol{y_{s,i}} \frac{\boldsymbol{x_i} - \mu(\boldsymbol{x_i})}{\sigma(\boldsymbol{x_i})} + \boldsymbol{y_{b,i}} \tag{1}$$

where $\boldsymbol{x_i}$ denotes the feature map on the i-th channel and $\boldsymbol{y_{s,i}}$, $\boldsymbol{y_{b,i}}$ are the corresponding normalizing coefficients. StyleGANv2 [2] further analyzes and improves the network structure in [1], where feature map modulation is implemented by using transformed convolution weights.

Due to the remarkable success of such 'style-based generators' in image synthesis, a great number of studies on image translation have also resorted to using style codes to control the behavior of generator. Attribute-Decomposed GAN [4] computes the style code from reference images to encode attribute information, which help control the appearance of generated person images. In [5], identity and pose from different 3D body meshes are integrated via spatially adaptive normalization modules. PSGAN [3] adopts spatially-variant coefficients of instance normalization for makeup transfer.

C. Latent Space Analysis for Style-Based Generators

The most prominent feature of style-based generators is that the semantic of output images could be interpretably manipulated by editing the corresponding latent vector. Therefore, numerous studies have been conducted to analyze the structure of latent spaces and explore the relationship between latent vectors and the semantic of generated images. These methods could be generally divided into two categories, i.e., supervised and unsupervised methods, based on whether semantic labels are used in the learning process.

Unsupervised methods [18], [19], [20], [21], [22] adopt classical unsupervised machine learning techniques to solve for representative latent directions, and also interpret their semantic meanings in generation results. Supervised methods [19], [23], [24], [25], [30], [49], on the other hand, solve for the manipulated latent vector with the supervision of semantic labels. However, these methods either produce spatially entangled changes in the manipulation result, or train additional deep networks to maintain the pixel-level consistency in non-target area. To solve this problem, a contemporary study, named StyleSpace Analysis [50], proposes to solve for the most activated image region given an element in the style code to achieve spatial disentanglement.

Another line of work [10], [51], [52], [53], [54], named 'GAN Inversion', aims to solve for a latent code that minimizes the error between the generated image and an given input, and explores the possibility of applying pre-trained generators to real images. In general, most existing works focus on manipulating the latent code in $\mathcal{Z}$, $\mathcal{W}$, or $\mathcal{W}^+$ space. However, in this work, we explain why traveling in these latent spaces would cause spatially entangled image changes, and propose an efficient method to tackle the problem.

III. Method

A. Problem Analysis

In this subsection, we analyze the source of spatial entanglement in image translation from the perspective of network structure. As shown in Fig. 2 (a), a StyleGAN generator $G : \mathcal{Z} \rightarrow \mathcal{I}$ consists of a mapping network $f : \mathcal{Z} \rightarrow \mathcal{S}$, which computes the style code $\boldsymbol{s} \in \mathcal{S}$ based on the input latent vector $\boldsymbol{z} \in \mathcal{Z}$ ($\boldsymbol{s} = f(\boldsymbol{z})$), followed by a synthesizing convolutional neural network $h : \mathcal{S} \rightarrow \mathcal{I}$, where each feature map is modulated by a component of $\boldsymbol{s}$ and renders the final output ($I = h(\boldsymbol{s})$). More concretely, the input latent code $\boldsymbol{z} \in \mathbb{R}^{512}$ sampled from a normal distribution $\mathcal{N}(0, I)$ is first mapped to an intermediate latent vector $\boldsymbol{w} \in \mathcal{W}$ with the same dimension. Such mapping function is implemented with a multi-layer perceptron (MLP) network, which is constructed by a stack of fully connnected layers followes by activation layers. Afterwards, $\boldsymbol{w}$ is replicated by the number of convolutional blocks in h (denoted as N), and each of them is fed into an affine mapping network (also implemented by MLP) to compute the actual style code $\boldsymbol{s}$ at its level. These style codes serve as the normalizing coefficients which modulate

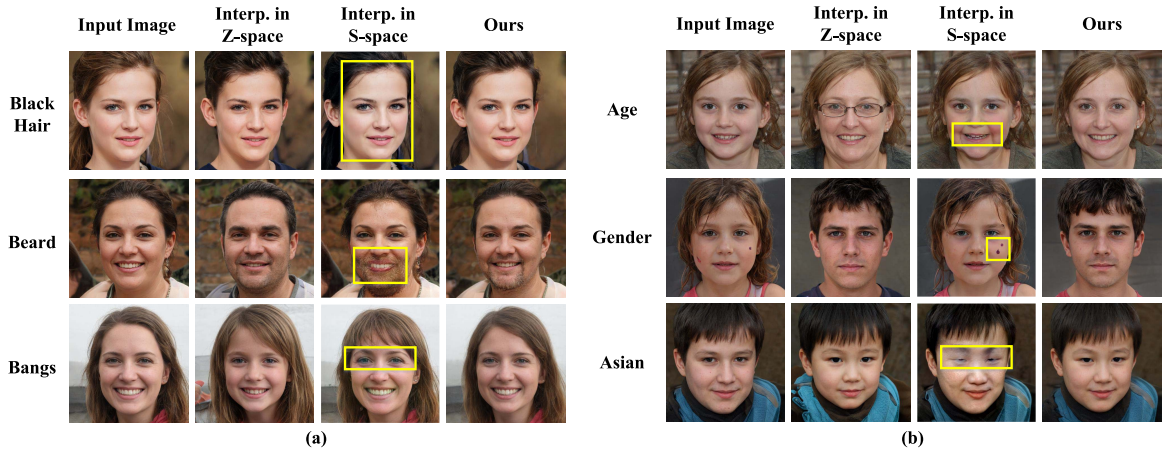

Fig. 3. Examples of editing results obtained by interpolating in $\mathcal{Z}$ ('Interp. in Z-space'), interpolating in $\mathcal{S}$ ('Interp. in S-space'), and the proposed method on attributes associated with (a) local objects and (b) more global targets. For 'Black Hair' and 'Bangs', unnaturalness caused by interpolating in $\mathcal{S}$ is mainly reflected in color distortion of the facial region (highlighted by yellowing bounding boxes). Zoom in for a better view of details.

the feature maps (denoted as $\mathbb{U} = \bigcup_{i=1}^{N} U_i$) in the synthesis network h.

For face images, a close inspection of the feature maps shows that some spatial channels consistently align with the distribution of a certain facial component class (denoted as c) after being re-scaled and thresholded (as shown in Fig. 2 (b)) [31], [32], [55]. Intuitively, these feature maps are responsible for synthesizing the corresponding facial component, and thus modifications should be restricted to the associated style codes (denoted as $\boldsymbol{s}_c$) .[2] However, as shown in Fig. 2 (a), all mapping functions, i.e., $\mathcal{Z} \rightarrow \mathcal{W}$ and $\mathcal{W}/\mathcal{W}^+ \rightarrow \mathcal{S}$, are implemented by fully connected layers (with activation layers). Therefore, modifying one single element of the latent code in $\mathcal{Z}$ / $\mathcal{W}$ / $\mathcal{W}^+$ will change the result in multiple output channels, which would eventually increase the chance of influencing superfluous feature maps and cause entangled changes in the output image.

B. Translation in the Style Space

As discussed in the previous subsection, the spatial entanglement of image translation is essentially caused by the intrinsic structural entanglement of StyleGAN generators. This inspires us to directly manipulate the style code in $\mathcal{S}$ instead of existing latent spaces $\mathcal{Z}$ / $\mathcal{W}$ / $\mathcal{W}^+$, which enables adjusting individual feature maps and achieve better disentanglement.

Given an input image I with the style code $\boldsymbol{s}$ ($I = h(\boldsymbol{s})$), as well as the target attribute α to be manipulated, the editing result could be obtained by simple linear interpolation, i.e., $I' = h(\boldsymbol{s}') = h(\boldsymbol{s} + \Delta\boldsymbol{s}^{(\alpha)})$, where $\Delta\boldsymbol{s}^{(\alpha)}$ denotes the displacement vector in $\mathcal{S}$ associated with α. Thus, $-\Delta\boldsymbol{s}^{(\alpha)}$ corresponds to the attribute change in the opposite semantic direction. In this paper, we do not deliberately distinguish $\Delta\boldsymbol{s}^{(\alpha)}$ from $-\Delta\boldsymbol{s}^{(\alpha)}$, and generally refer to the associated facial semantic in both directions with the name of target attribute.

[2]We re-use c in the subscript to denote feature maps or style codes responsible for synthesizing the target object ($\bar{c}$ for irrelavant ones).

Suppose we have an attribute classifier F_α defined in $\mathcal{S}$ which satisfies,

$$F_\alpha(\boldsymbol{s}) \begin{cases} > 0 & \text{for } h(\boldsymbol{s}) \text{ has the attribute } \alpha \\ < 0 & \text{for } h(\boldsymbol{s}) \text{ does not have the attribute } \alpha \end{cases} \tag{2}$$

then the edited style code $\boldsymbol{s}'$ should satisfy

$$F_\alpha(\boldsymbol{s}) \cdot F_\alpha(\boldsymbol{s}') < 0 \tag{3}$$

Based on the relationship between local image object and feature maps in h, ideally, the displacement vector should satisfy

$$\Delta\boldsymbol{s}_{\bar{c}}^{(\alpha)} = 0 \tag{4}$$

which indicates that the style code of feature maps irrelevant to the target object (denoted by the subscript $\bar{c}$) remains unchanged. If F_α is implemented using a linear classifier, given a sample style code s with $F_\alpha(s) < 0$, interpolating s along the **normal vector of the separating hyperplane** of F_α (denoted as $\Delta\boldsymbol{s}_n^{(\alpha)}$) will increase the value of $F_\alpha(s)$ and finally reaches $F_\alpha(s) > 0$ (i.e., flip of the attribute label). This is exactly what we expected by traveling along $\Delta\boldsymbol{s}^{(\alpha)}$ in $\mathcal{S}$, and thus it is natural to approximate $\Delta\boldsymbol{s}^{(\alpha)}$ with $\Delta\boldsymbol{s}_n^{(\alpha)}$ to satisfy Eq. 3. Moreover, the **sparsity regularization** on $\Delta\boldsymbol{s}_n^{(\alpha)}$ is also involved to satisfy Eq. 4. Therefore, interpolating along $\Delta\boldsymbol{s}_n^{(\alpha)}$ would achieve spatially disentangled translation of the attribute α.

C. Style Intervention With Self-Supervision

Although $\Delta\boldsymbol{s}_n$ has desired properties for generating spatially disentangled semantic changes, it does not provide sufficient information to render natural translation effects in certain cases. This is because $\Delta\boldsymbol{s}_n$ is obtained via a discriminative task (i.e., attribute classification) with sparsity constraints, and thus only contains the minimum information to translate the input image so as to change the attribute label of α (We omit the superscript of α in this subsection for simplicity). Therefore, it does not guarantee the realism of generation

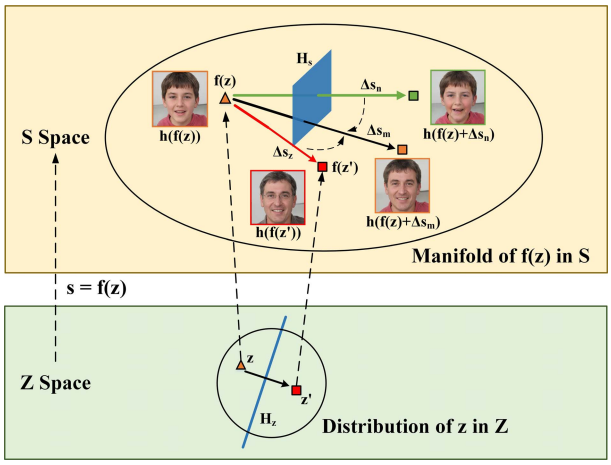

Fig. 4. The framework of the proposed method Style Intervention. H_z and H_s denote the separating hyper-plane of the target attribute in $\mathcal{Z}$ and $\mathcal{S}$, respectively. We use H_z to show that corresponding images of z and z' have different attribute labels, and z' could be obtained by interpolation z along the normal vector of H_z. $\Delta \boldsymbol{s}_z$ denotes the displacement vector between $f(z)$ and $f(z')$, and it is combined with the normal vector of H_s, i.e., $\Delta \boldsymbol{s}_n$, to obtain the final displacement vector $\Delta \boldsymbol{s}_m$.

Algorithm 1 Spatially Disentangled Image Translation by Style Intervention

Input: mapping network f; input latent code z and edited latent code z'; normal vector for the attribute α, $\Delta \boldsymbol{s}_n^{(\alpha)}$; learning rate γ; maximum iteration step t_m

Output: intervention coefficients $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_i\}_{i=1}^N$

$\boldsymbol{\Lambda} \leftarrow \mathbf{0}$, $\Delta \boldsymbol{s}_z \leftarrow f(z') - f(z)$;
$\Delta \boldsymbol{s}_n^{(\alpha)} \leftarrow \frac{\|\Delta \boldsymbol{s}_z\|_2}{\|\Delta \boldsymbol{s}_n^{(\alpha)}\|_2} \Delta \boldsymbol{s}_n^{(\alpha)}$
for $i \leftarrow 1$ *to* N **do**
  for $t \leftarrow 1$ *to* t_m **do**
    $\boldsymbol{\Lambda} \leftarrow clamp\ (\boldsymbol{\Lambda}, 0, 1)$;
    $\Delta \boldsymbol{s}_m \leftarrow (\mathbf{1} - \boldsymbol{\Lambda}) \cdot \Delta \boldsymbol{s}_z + \boldsymbol{\Lambda} \cdot \Delta \boldsymbol{s}_n^{(\alpha)}$;
    Compute the overall objective function $\mathcal{L}$;
    $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_i + \gamma \cdot \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}_i}$;
  end
end

results. As shown in Fig. 3, although interpolation in the style space $\mathcal{S}$ generates more spatially disentangled image changes than in $\mathcal{Z}$, the results still suffer from unnaturalness.

However, since the result of translating in $\mathcal{Z}$ ($z \rightarrow z'$) is visually plausible, the corresponding displacement vector in $\mathcal{S}$, i.e., $\Delta \boldsymbol{s}_z = f(z') - f(z)$ where f is the mapping function of StyleGAN (see in Fig. 4), should contain sufficient information to render rich textural details of the translation process. To this end, we naturally propose to combine the advantages of translating in both $\mathcal{Z}$ and $\mathcal{S}$, i.e., high visual fidelity and semantical disentanglement, respectively, for authentic and accurate manipulation of image components. Our method could be conviniently adapted to translations in other latent spaces (e.g., $\mathcal{W}$ or $\mathcal{W}^+$) by simply substituting $\Delta \boldsymbol{s}_z$ with displacement vectors in the corresponding space.

To achieve this, an intuitive method would be changing each component in $\boldsymbol{s}$ according to the corresponding value in $\Delta \boldsymbol{s}_n$, and check whether it enhances the translation result. However, in most cases, the synthesis of the target facial component to be edited jointly depends on multiple feature maps [31], [32], [55], and thus solving for one single component at a time could lead to unreasonable results. Therefore, we propose to tackle the problem by solving for the **intervention coefficient** $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_i\}_{i=1}^N$. Each $\boldsymbol{\lambda}_i \in [0, 1]^{l_i}$ indicates the degree of intervention for all channels in the i th convolutional layer, where l_i is the number of feature maps within. Hence, the merged displacement of style code, denoted as $\Delta \boldsymbol{s}_m$, could be computed as

$$\Delta \boldsymbol{s}_m(\Lambda) = (\mathbf{1} - \boldsymbol{\Lambda}) \cdot \Delta \boldsymbol{s}_z + \boldsymbol{\Lambda} \cdot \Delta \boldsymbol{s}_n \tag{5}$$

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

- **Pixel-level Loss** $\mathcal{L}_{pix}$**:** We explicitly penalize $\boldsymbol{\Lambda}$ for modifying image content irrelated to the target attribute, which is indicated using a binary mask m_c of the related facial component c. Specifically, m_c is obtained based on the semantic segmentation of input images, where pixel values within c is set to 1 and elsewhere to 0. Therefore, the pixel-level loss could be formulated as

$$\mathcal{L}_{pix} = \|(1 - m_c) \cdot (h(\boldsymbol{s} + \Delta \boldsymbol{s}_m(\boldsymbol{\Lambda})) - h(\boldsymbol{s}))\|_2 \tag{6}$$

  The detailed approach for computing m_c is discussed in Section IV-A.
- **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

$$\mathcal{L}_{attr} = -\frac{\Delta \boldsymbol{s}_n \cdot \Delta \boldsymbol{s}_m(\boldsymbol{\Lambda})}{\|\Delta \boldsymbol{s}_n\|_2 \cdot \|\Delta \boldsymbol{s}_m(\boldsymbol{\Lambda})\|_2} \tag{7}$$

  It is obvious that $\mathcal{L}_{attr}$ is exactly the cosine similarity between $\Delta \boldsymbol{s}_n$ and $\Delta \boldsymbol{s}_m(\boldsymbol{\Lambda})$, which encourages the resultant style code $\Delta \boldsymbol{s}_m$ to still render the desired attribute change presented by $\Delta \boldsymbol{s}_n$.
- **L2-norm Loss** $\mathcal{L}_{norm}$**:** Although introducing $\Delta \boldsymbol{s}_n$ helps with disentangling spatial changes in image, it inevitably makes the resultant style code deviate from the manifold expanded by $\boldsymbol{s} = f(z)$, 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 $\mathcal{L}_{norm}(\boldsymbol{\Lambda}) = \|\boldsymbol{\Lambda}\|_2$.

Therefore, the overall objective function could be written as

$$\mathcal{L} = \mathcal{L}_{pix} + \lambda_{attr}\mathcal{L}_{attr} + \lambda_{norm}\mathcal{L}_{norm} \tag{8}$$

where λ_{attr} and λ_{norm} control the relative importance of $\mathcal{L}_{attr}$ and $\mathcal{L}_{norm}$, respectively. The optimal intervention coefficient $\boldsymbol{\Lambda}^*$ could be solved by $\boldsymbol{\Lambda}^* = \arg\min_{\boldsymbol{\Lambda}} \mathcal{L}$. Details of the proposed method, named **Style Intervention**, are shown in Algorithm 1. Note that the edited latent code z' in this work is computed by simply linearly interpolating z along the semantic direction Δz, which is the normal vector of trained linear SVMs in $\mathcal{Z}$ (similar to [25]). However, there

is no restriction on how z' should be computed, and thus the proposed algorithm could be easily adapted to various tasks where z' is already available.

D. Differences With Previous Methods

StyleSpace Analysis [50] is the contemporary study most similar to ours, but there are still many differences in between. Concretely, our method mainly focuses on studying the disentanglement of the style space w.r.t. spatial regions in the output image (spatial dimension), while StyleSpace cares more about the correlation between style code and image attributes (semantic level). As for image manipulation, StyleSpace proposes to detect locally active image regions for a given channel of style code by tracing the gradient, but we consider the normal vector of linear SVMs trained with sparsity constraints as the direction for interpolation. Moreover, we also point out the limitation of editing in S and solve it by proposing an optimization-based algorithm. GH-Feat [56] aims at manipulating AdaIN coefficients, which are computed by a learned encoder network instead of a fixed mapping network. InterFaceGAN [25] also uses linear interpolation for disentangled attribute manipulation. However, we not only perform attribute editing in another latent space (i.e., $\mathcal{S}$ rather than $\mathcal{Z}$ or $\mathcal{W}$), but also 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 [51] and its advanced version Image2StyleGAN++ [52] propose to investigate the extended latent space previous to affine layers (i.e., $\mathcal{W}^+$). Although image manipulations are more spatially disentangled, the focus of their work is on solving the GAN inversion problem, and we will also demonstrate the advantage of $\mathcal{S}$ over $\mathcal{W}^+$ in Section IV-B. StyleRig [30], StyleFlow [23], GANPaint [24], and Latent-Transformer [49] involve training auxiliary networks along with the pre-trained StyleGAN model, which are much more computational expensive than our method. Unsupervised approaches [18], [19], [20] are inappropriate for this task as they focus on discovering interpretable latent semantics, instead of solving for the latent direction for the target attribute.

IV. Experiment

A. Experimental Setup

1) Implementation Details: In this paper, we implement the attribute classifier $\mathcal{F}$ with linear support vector machine (SVM), and thus the normal vector of the corresponding separating hyper-plane serves as the interpolation direction for semantic translation. For translation in $\mathcal{S}$, the L1-norm is involved as the regularization term to satisfy sparsity constraint. Similarly, to compute the edited latent code z', we also perform linear interpolation in $\mathcal{Z}$ for simplicity without sparsity constraint.

As for the StyleGAN generator, we typically focus on using the StyleGANv2 model [2] pre-trained on the FFHQ dataset [1] with resolution 1024×1024. Specifically, the publicly available Pytorch [57] re-implementation of StyleGANv2 [58] is adopted in all experiments. To obtain the complete style code, we concatenate the modulating coefficients of all residual blocks from the coarsest to the finest level. At each resolution, style codes of two convolutional layers and one to_rgb layer are concatenated in order. For the optimization process, we adopt the Adam optimizer and roughly set balancing coefficients λ_{attr} and λ_{norm} to $1e^{-2}$ and $1e^{-6}$, respectively. Moreover, to improve the time efficiency at test phase, we further adopt a 'layer-wise early stopping' technique to terminate the optimization process at a certain layer, if the difference of loss value between two iterations is less than a pre-defined threshold ϵ (empirically set to $1e^{-3}$).

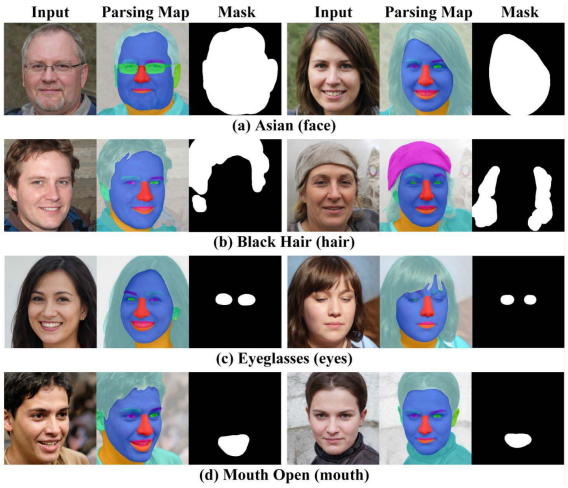

Fig. 5. Examples of parsing maps and binary masks for various attribute and facial components. Activated regions of masks (white area) are dilated with a ellipse kernel of size 50 to allow slight modification to pixels outside the target area near the boundary.

As for the semantic mask m_c used for computing the pixel-level loss $\mathcal{L}_{pix}$, it is constructed following the protocol shown in TABLE I, which defines the relationship between parsing classes in c and the manipulated attributes α). Parsing maps and binary masks for various attribute and facial components are visualized in Fig 5. Notably, our method is not limited to a certain way of obtaining m_c, and users are free to adjust m_c between different trials of optimization.

2) Data Preparation: The effectiveness of the proposed method is verified on the task of facial attribute editing (FAE), which has received considerable attention in recent studies on style-based generators. To train the attribute classifier $\mathcal{F}$ in both $\mathcal{Z}$ and $\mathcal{S}$, we create a synthetic data using the same pre-trained model for image manipulation. To be specific, we randomly sampled 100,000 latent code $\boldsymbol{z} \sim N(0, I)$ in $\mathcal{Z}$, and compute the corresponding style code $\boldsymbol{s} = f(z)$ as well as the output image $I = G(\boldsymbol{z})$. For each image, binary labels for 12 facial attributes are manually annotated. To be objective, 5 attributes of them ('Gender', 'Age', 'Eyeglasses', 'Mouth Open', and 'Smile') are labeled using the third-party online face analysis toolkit Face++ [59]. For the rest attributes, we randomly select 10,000 sample images and manually annotate them with binary labels.

TABLE I

THE RELATIONSHIP BETWEEN FACIAL COMPONENTS (DENOTED AS c), FACIAL ATTRIBUTES (DENOTED AS α), AND NAME OF PARSING CLASSES

Facial Components (c)	Facial Attributes (α)	Parsing Classes
Hair	Gender, Age	Hair
Mouth	Smile, Mouth Open	Mouth, Upper & Lower Lip
Eyes	Eyeglasses	Left & Right Eye
Face	Gender, Age, Asian, Bangs, Beard	Left & Right Brow\Eye\Ear, Upper & Lower Lip, Nose, Mouth, Skin, Eyeglasses

TABLE II

CLASSIFICATION ACCURACY (%) OF LINEAR SVMS TRAINED IN $\mathcal{S}$ ON BOTH TRAINING AND VALIADATION SET. SPARSITY VALUES DEMONSTRATE THE PROPORTION (%) OF ZERO ELEMENTS IN $\Delta \boldsymbol{s}_n^{(\alpha)}$

Attribute (α)	Gender	Age	Smile	Eye-glasses	Mouth Open	Asian	Bangs	Black Hair	Blond Hair	Grey Hair	Curvy 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
Sparsity	93.60	98.14	96.08	96.92	94.98	96.25	93.61	93.64	93.80	98.55	93.25	97.76

To evaluate the performance of the proposed method and compare with other state-of-the-art approaches, we adopt **CelebA-HQ** [43], which contains 30,000 face images in-the-wild of resolution 512×512, as the benchmark dataset for both qualitative and quantitative experiments. In each experiment, images in CelebA-HQ are resized to align with the input resolution of the target model. For real images used in all experiments, the corresponding latent code in $\mathcal{Z}$ / $\mathcal{W}$ / $\mathcal{W}^+$ or $\mathcal{S}$ of each input image is obtained via an optimization-based method similar to [51].

3) Benchmark Methods: In this study, we choose InterFaceGAN [25], StyleSpace [50], and Latent-Transformer [49] as representatives for FAE methods using pre-trained StyleGAN generators. InterFaceGAN manipulates face images by simple linear interpolation in $\mathcal{Z}$, and achieves semantic disentanglement by subtracting the project on non-target attribute directions. We also re-implement InterFaceGAN in the $\mathcal{W}^+$ space for real image manipulation. StyleSpace directly manipulates the style code most relative to the target attribute. Latent-Transformer learns a mapping function in the $\mathcal{W}^+$ space for latent code manipulation. StarGAN [41] and AttGAN [42] are considered as benchmark methods with traditional GAN structure (i.e., without style guidance). Publicly available implementation of these two approaches are used for testing [60], [61], and we resize input images to corresponding resolutions for fairness (256×256 for AttGAN and 128×128 for StarGAN).

4) Evaluation Metrics: Since the proposed method aims to achieve spatially disentangled attribute manipulation of face images, a natural and intuitive measurement would be directly evaluating the **Mean Squared Error (MSE)** of pixel values in the non-target area. However, since such MSE metric is directly optimized by the pixel-level loss in the overall objective function, i.e., $\mathcal{L}_{pix}$, it might be biased towards our method. Therefore, to provide a more comprehensive analysis on the performance of our method as well as a fairer comparison against benchmark methods, **Structural Similarity Index Measure (SSIM)** and **Face Verification Score (ID)** are also adopted for measuring the image consistency in different levels of abstraction. Specifically, SSIM evaluates the preservation of structural information computed over the entire image, and ID reflects to what extent the identity information is preserved in manipulated results. The face verification score is also obtained via Face++ API for objectivity.

B. Ablation Study

1) Separability of the Style Space $\mathcal{S}$: In Section III-B, we approximate the ideal displacement vector in $\mathcal{S}$ by $\Delta \boldsymbol{s}_n^{(\alpha)}$, i.e., the normal vector of a hyperplane classifying style codes by the label of attribute α. Therefore, it is necessary to inspect the separability of style codes since this is the cornerstone of our method.

According to the result shown in Table II, all 12 facial attributes considered in this study 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 $\mathbb{U}$ is responsible for changing the status of the corresponding attribute, which also proves the possibility for disentangling spatial translation by restricting modifications of the style code to $\boldsymbol{s}_c$.

2) Disentanglement of Different Latent Spaces: In Section III-A, we analyzed the cause of spatially entangled changed from the perspective of network structure, and lead to the conclusion that the style space $\mathcal{S}$ is more appropriate to achieve spatial disentanglement. To reveal the advantage of $\mathcal{S}$, in this subsection, we disregard the optimization-based algorithm and compare the intrinsic disentanglement of different latent spaces by simple interpolation. For an arbitrary input latent code, we interpolate it along the same semantic attribute in different spaces with normalized distances. Concretely, given a pair of input and edited latent code in $\mathcal{Z}$, denoted as $\boldsymbol{z}$ and $\boldsymbol{z}'$ respectively, the norm of the displacement vector could be computed as $\|\Delta \boldsymbol{z}\|_2 = \|\boldsymbol{z} - \boldsymbol{z}'\|_2$.

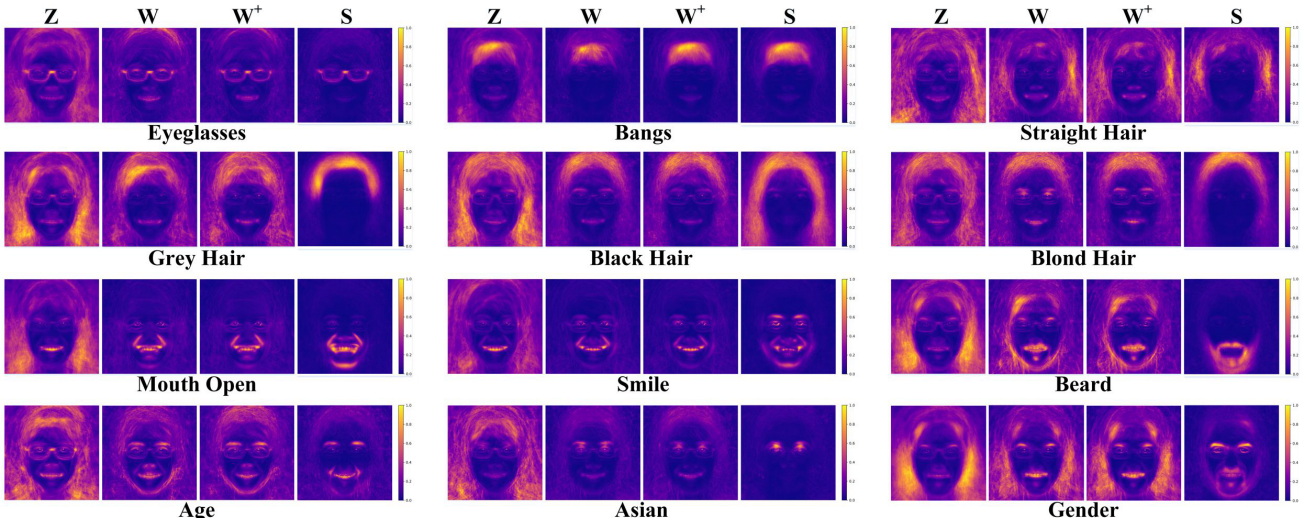

Fig. 6. Averaged residual images between random samples and their interpolation results in different latent spaces. For each spatial location, high intensity indicates larger image modifications. Higher level of concentration indicates better spatial disentanglement.

Afterwards, by forwarding through the style-based generator, we could obtain the corresponding latent code of $\boldsymbol{z}$ and $\boldsymbol{z}'$ in every latent space (i.e., $\boldsymbol{w}$ and $\boldsymbol{w}'$, $\boldsymbol{w}^+$ and $\boldsymbol{w}^{+\prime}$, $\boldsymbol{s}$ and $\boldsymbol{s}'$), and also the norm of displacement vectors (i.e., $\|\Delta \boldsymbol{w}\|_2$, $\|\Delta \boldsymbol{w}^+\|_2$, and $\|\Delta \boldsymbol{s}\|_2$). These norm values represent the discrepancy between the two samples in different latent spaces, and are used to normalize the corresponding distance of interpolation for fairness of comparison.

To better illustrate the locality of manipulating in $\mathcal{S}$, we compute the averaged residual images between 1,000 random samples and their interpolation results in different latent spaces. As shown in Fig. 6, image changes are more localized in $\mathcal{S}$ compared to other latent spaces, where they mainly locate within the target region and are disentangled with pixel elsewhere. This holds true for all considered attribute with different target facial component, indicating the intrinsic advantage of $\mathcal{S}$ over other latent spaces. In Fig. 7, we show the mean squared error of pixel values within non-target image area as the interpolated distance increases. Clearly, manipulation results obtained by interpolating in $\mathcal{Z}$ are heavily entangled, which is reflected by the rapidly increasing MSE of non-target pixels. Among all four latent spaces, $\mathcal{S}$ consistently outperforms $\mathcal{Z}$, $\mathcal{W}$, and $\mathcal{W}^+$ in achieving spatial disentanglement. Please note that all results in Fig. 6 and Fig. 7 are obtained by simple interpolation and no optimization is involved, indicating the intrinsic disentangling property of different latent spaces.

3) Effectiveness of Different Latent Spaces: In previous sections, we have shown that the proposed optimization-based method could produce more spatially disentangled image translations. However, it is still unclear whether the spatial disentanglement comes from the style space itself or the optimization process. Therefore, in this subsection, we investigate whether performing the proposed method in latent spaces other than $\mathcal{S}$ could still achieve comparable results.

As shown in Fig. 8, although optimization in $\mathcal{W}$ or $\mathcal{W}^+$ could improve the spatial disentanglement of image translation, the style space $\mathcal{S}$ achieves the best performance in all cases. Qualitative results in TABLE III are obtained by averaging over all 12 attributes on images in the synthetic dataset, and they also demonstrate that performing the proposed method in the style space achieves the best result under all metrics. Therefore, it is the intrinsic disentanglement of $\mathcal{S}$ that help generate more spatially disentangled image translation results.

TABLE III
QUANTITATIVE COMPARISON OF RESULTS OBTAINED BY PERFORMING THE PROPOSED METHOD IN DIFFERENT LATENT SPACES

	$\mathcal{Z}$-$\mathcal{W}$-edit	$\mathcal{Z}$-$\mathcal{W}^+$-edit	$\mathcal{Z}$-$\mathcal{S}$-edit
MSE (↓)	0.050	0.039	**0.017**
SSIM (↑)	0.609	0.653	**0.692**
ID (↑)	89.90	91.83	**94.97**

4) Effectiveness of Different Loss Terms: In the proposed method Style Intervention, three loss terms (i.e., pixel-level loss $\mathcal{L}_{pix}$, attribute loss $\mathcal{L}_{attr}$, and L2-norm loss $\mathcal{L}_{norm}$) are integrated to form the final objective function. In this subsection, we explore the contribution of each loss term, and qualitative results are shown in Fig. 9. It is clear that removing $\mathcal{L}_{pix}$ produces entangled manipulation results, since the algorithm fails to identify style codes responsible for controlling the attribute of the target attribute. The attribute change is less obvious when $\mathcal{L}_{attr}$ is absent, but it could be clearly observed in all other cases. $\mathcal{L}_{norm}$ is responsible for suppressing deviations from the natural image manifold, and removing it produces less natural (yet still disentangled) output image. In summary, all three loss terms jointly contribute to the successful generation of output images, and the quality will degrade if anyone of them is ablated.

C. Qualitative Results

1) On Synthetic Images: Image manipulation results on synthetic images for all 12 annotated facial attributes in both

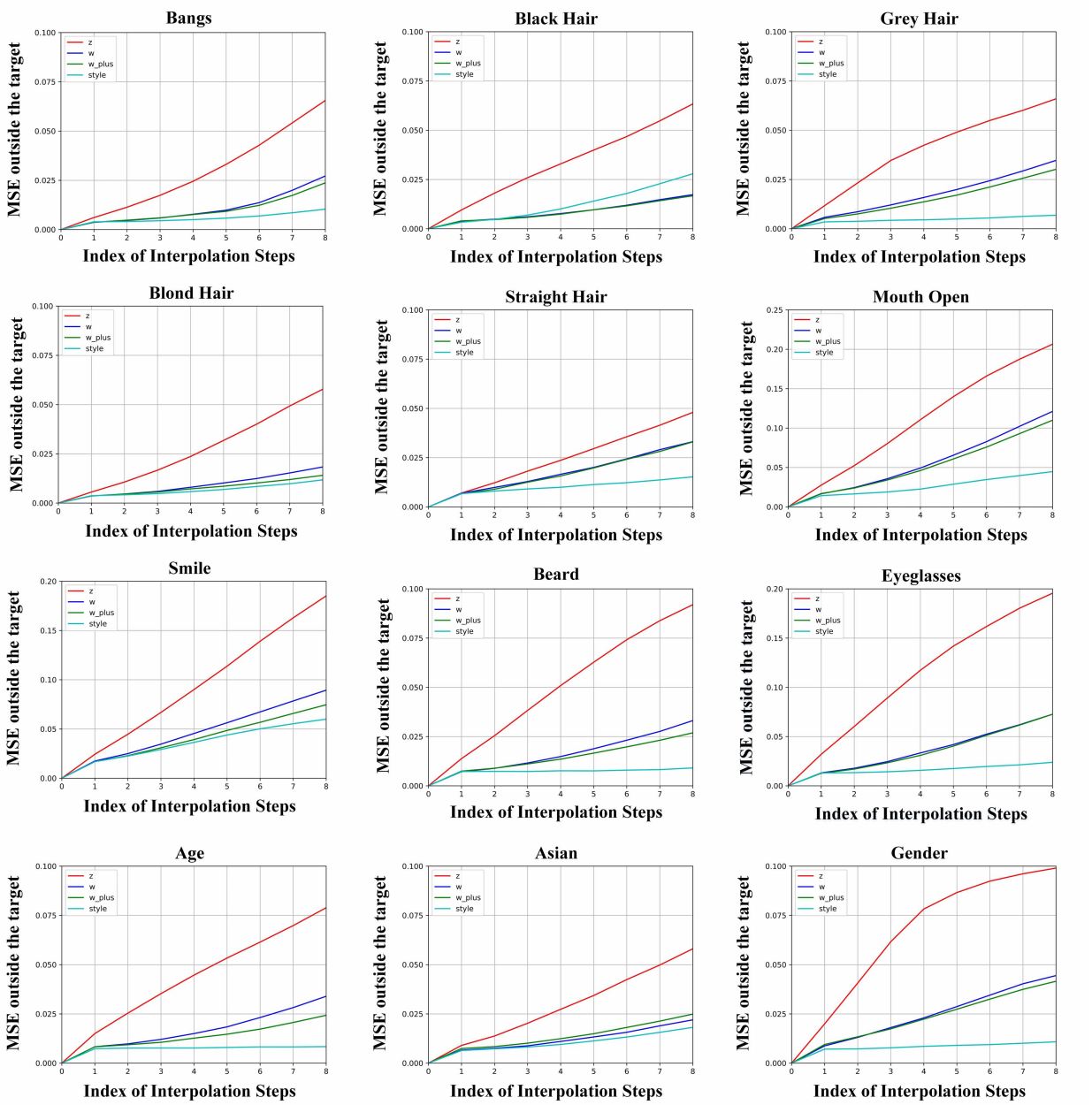

Fig. 7. The relationship between mean squared error (MSE) of pixel values in non-target regions and the interpolation step. Larger interpolation index indicates greater latent distance against the unedited image.

semantic directions are shown in Fig. 10. Although input images cover a wide range of population in terms of gender, race, and age with various pose and lighting conditions, spatially disentangled image translation is achieved in the output image for all cases, i.e., only pixels in the area related to the target attribute have been modified. Note that for the attribute 'Beard', photorealistic bread could be seamlessly attached to female without turning the face more masculine, indicating the ability of the proposed method in performing local editing regardless of the overall semantic.

2) On Realistic Images: To edit real images with pre-trained StyleGAN generators, they have to be firstly embedded into the latent space and then the obtained representations could be manipulated to produce attribute changes. Such embedding process is called 'GAN inversion', and it is well known that there is a tradeoff between the inversion quality and manipulability in the latent space [52], [54], [62]. Therefore, it is necessary to inspect whether the proposed method could be applied to real images. In this paper, similar to [51], embedded representations of all real images are computed by minimizing the reconstruction error, which is measured by both pixel-wise MSE loss and perceptual loss implemented with LPIPS [63].

Fig. 11 shows examples of celebrity image manipulation in different reconstruction spaces. For each latent space, we compare the performance of our method to simple linear interpolation. It is clear that from $\mathcal{Z}$ to $\mathcal{W}$, and then to $\mathcal{W}^+$, the error between input and reconstructed images significantly

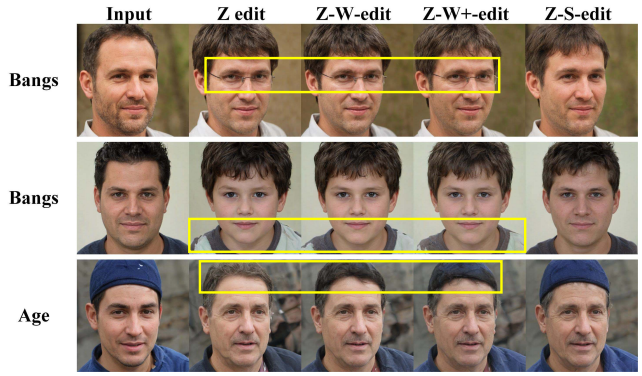

Fig. 8. Comparison of results obtained by performing the proposed method in different latent spaces. The style space provides the highest level of spatial disentanglement, and disentangled image changes are highlighted by yellow bounding boxes. Zoom in for a better view of image details.

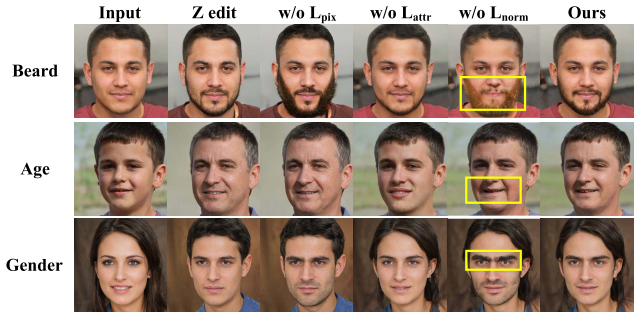

Fig. 9. Results of ablating different loss terms on translating the attribute 'Beard', 'Age', and 'Gender'. Ghosting artifacts caused by removing $\mathcal{L}_{norm}$ is highlighted by yellow boxes. Zoom in for a better view of image details.

reduces. However, artifacts and entangled changes could still be observed in the interpolation results (e.g., face skin color in the first row). On the contrary, with image-level supervision imposed, our method could largely suppress spatially entangled modifications while still achieve the target attribute change. Please also note how the reconstruction error is accumulated to the manipulation results (e.g, the background cluster), and potential solutions to this problem is discussed in Section V.

3) Comparison With Benchmark Methods: Fig. 12 shows the comparison result between our method and InterFaceGAN, StyleSpace, as well as Latent-Transformer on synthetic face images. We also include the result of simple interpolation in $\mathcal{Z}$ to present the entangled attributes (marked in parentheses), which are eliminated in InterFaceGAN via orthogonalization. To be objective, instead of manually identifying such attributes, publicly available Face++ online APIs are used and we only conduct experiments on four facial attributes that could be recognized by the toolkit (results for 'Mouth Open' is not included as it largely overlaps with 'Smile').

It is clear that simply interpolating in $\mathcal{Z}$-space would produces spatially entangled modifications to the input image. Although InterFaceGAN largely solves the problem of semantic entanglement, as shown in Fig. 12 (b) and (d), the results still suffer from undesirable changes of non-target 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 as in InterFaceGAN. On the contrary, StyleSpace could better restrict image changes within the target region, but it fails to render realistic translation of complex attributes (e.g., 'Age') which are intrinsically associated with entangled feature-map channels [50]. Latent-Transformer solves such problem by applying a more complex transformation to latent codes compared to linear interpolation, but it causes obvious changes in non-target regions (Fig. 12 (b) and (c)). Our method integrates the advantage of latent code editing in both $\mathcal{Z}$ (realism) and $\mathcal{S}$ (disentanglement), and handles the trade-off by the intervention coefficient Λ.

To measure the performance of our method on realistic images, we test our method on CelebA-HQ [43] and also compare the result with StarGAN [41] and AttGAN [42]. For methods based on pre-trained StyleGAN generators, we project real input images into the $\mathcal{W}^+$ space and thus InterFaceGAN and Latent-Transformer could be directly applied. Our method and StyleSpace could take style codes computed from the corresponding projection result as input, and produce manipulated images accordingly. From Fig. 13, it could be observed that all six methods are able to perform attribute manipulation with semantic disentanglement. However, closer inspect would reveal that results of StarGAN suffer from clear ghosting artifacts, e.g., the background pixels in Fig. 13 (b) are undesirably modified. AttGAN performs better in restricting image modifications within the target region, but unnatural image changes could still be observed in the editing results of 'Gender'. Clearly, compared to StarGAN and AttGAN, methods built with the pre-trained StyleGAN generator could perform attribute editing on face images with higher resolution with better visual quality. However, the extend InterFaceGAN also suffers from the previously discussed problem similar as shown in Fig. 12. Compared to StyleSpace and Latent-Transformer, our method could render more photo-realistic image translations and achieve better spatial disentanglement, respectively.

D. Quantitative Results

1) On Synthetic Images: TABLE IV shows the quantitative results on synthetic images. it is clear that our method outperforms benchmark approaches or achieves comparable performance in all cases, demonstrating the effectiveness of the proposed method in rendering precise translations of local regions on various levels of abstraction. This is desirable in numerous practical applications, such as portrait image editing, where the accurate manipulation of individual object is required. Notably, the high performance of StyleSpace in editing the attribute 'Age' actually results from the little change in corresponding manipulation results (see Fig. 13), indicating the incapability of StyleSpace in dealing with such attributes.

2) On Realistic Images: Results of quantitative comparison on CelebA-HQ are shown in Table V. Since minimizing the pixel-level error in non-target area is one of the objective

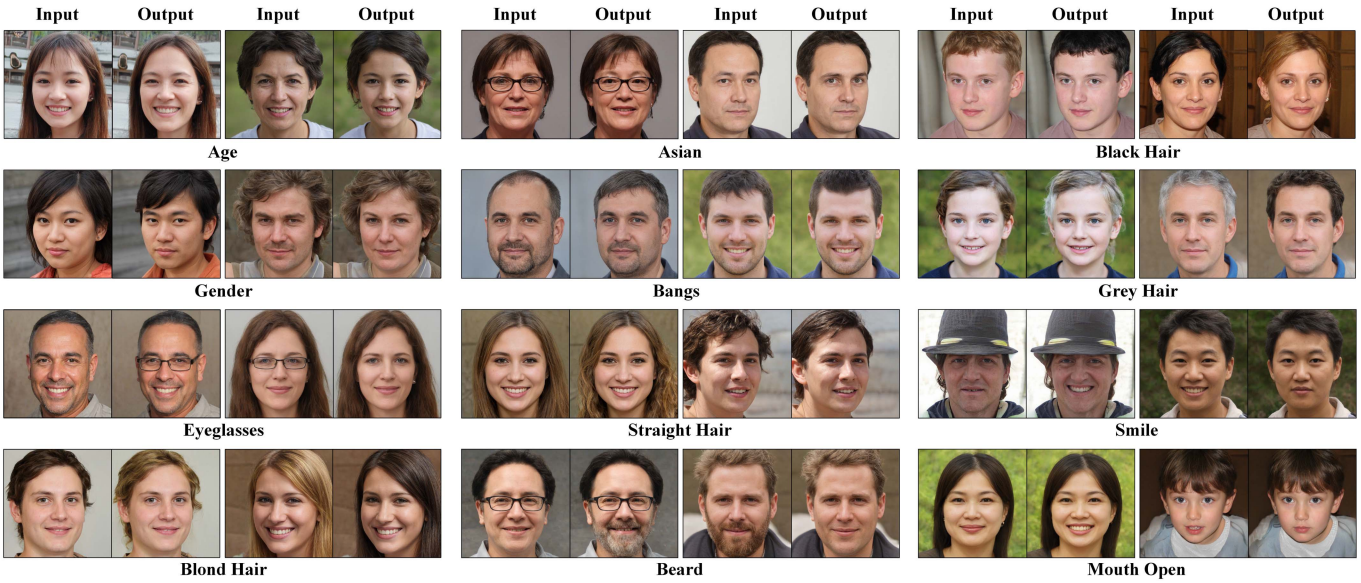

Fig. 10. Results of image manipulation on synthetic images for all 12 annotated facial attributes. For each attribute, semantic changes in both directions are provided. All images are generated by the StyleGANv2 generator pre-trained on FFHQ with size 1024×1024 (zoom in for better details). Please note that image content irrelated to the target semantic is largely preserved in output images.

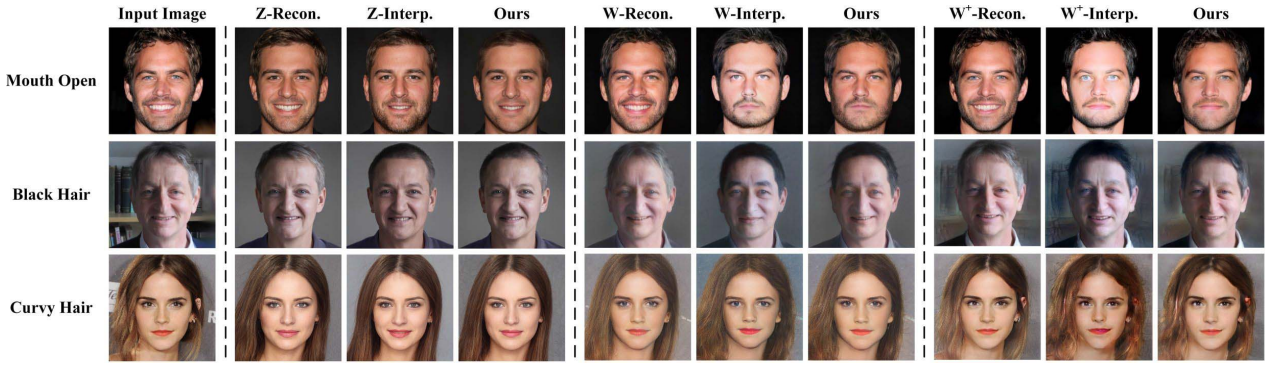

Fig. 11. Sample results of reconstruction (Recon.), interpolation (Interp.), and manipulation of real celebrity images in different latent spaces. The target facial attribute for each row is labeled on the left. Please note the limited reconstruction accuracy obtained by the optimization-based GAN inversion method. Zoom in for a better view of image details.

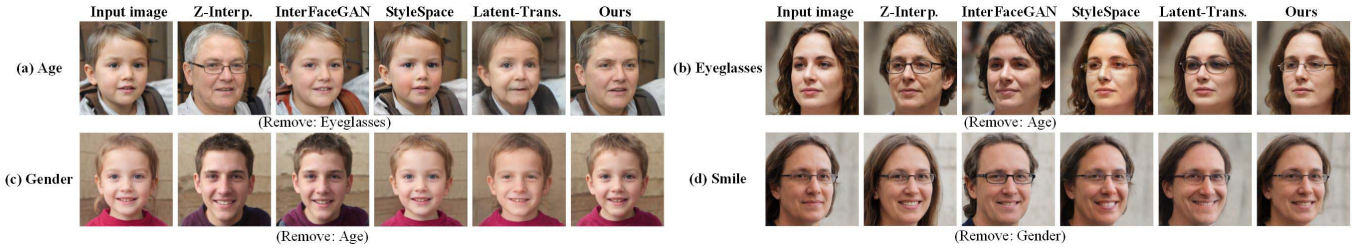

Fig. 12. Comparison between our method and editing in $\mathcal{Z}$-space (denoted as Z-interp.), InterFaceGAN, StyleSpace, Latent Transformer (denoted as Latent-Trans.) on synthetic images. Target attributes to be manipulated are labeled on the left for each set of results. In InterFaceGAN, the interference of unwanted attribute changes is eliminated by orthogonalization in the latent space, which are detected by Face++ APIs and marked in parentheses.

functions of our method, it could be seen that our method achieves the best or equally competitive results in MSE all attributes. Moreover, the proposed method also shows the best performance in SSIM and ID for 'Black Hair' and 'Blond Hair'. However, as for attributes involving multiple facial components, such as 'Gender' and 'Age', the SSIM scores obtained by conditional GAN-based methods, i.e., AttGAN and StarGAN, are higher than ours. This is caused by the structural change made within the target area (see result of editing 'Gender' in Fig. 13), where Style Intervention does not impose pixel-level consistency in those regions but StarGAN and AttGAN do (L1 reconstruction loss computed over the

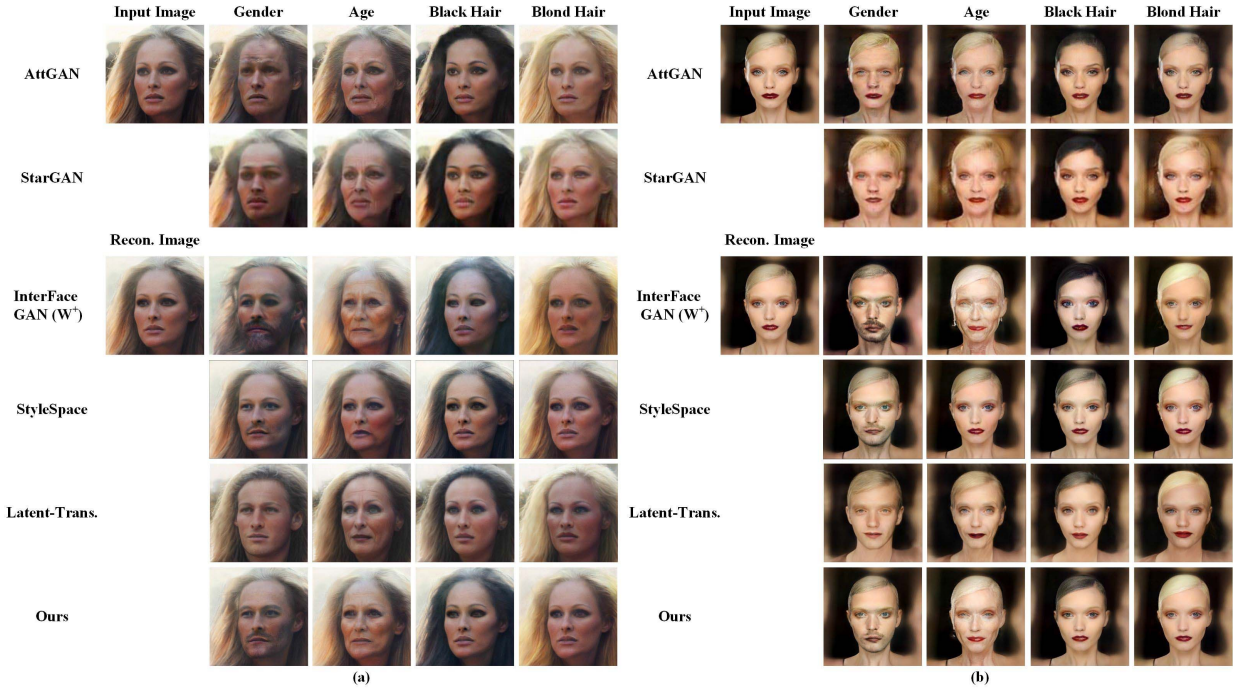

Fig. 13. Comparison between our method and StarGAN, AttGAN, InterFaceGAN extended to the $\mathcal{W}^+$ space (denoted as InterFaceGAN (W^+)), StyleSpace, as well as Latent Transformer (denoted as Latent-Trans.) on CelebA-HQ. Image reconstruction is performed in the $\mathcal{W}^+$ space via optimization for fair comparison. Zoom in for a better view of image details.

TABLE IV

COMPARISON OF QUANTITATIVE RESULTS ON SYNTHETIC IMAGES. THE METRIC 'ID' IS THE CONFIDENCE SCORE OF FACE VERIFICATION OBTAINED BY THE FACE++ ONLINE ANALYSIS TOOLKIT. '$\mathcal{Z}$ / $\mathcal{W}$ / $\mathcal{W}^+$ - INTERP.' REFERS TO LINEAR INTERPOLATING IN $\mathcal{Z}$ / $\mathcal{W}$ / $\mathcal{W}^+$ SPACES WITH NORMALIZED DISTANCES, RESPECTIVELY

Method	MSE (↓)				SSIM (↑)				ID (↑)			
	Gender	Age	Eye glasses	Smile	Gender	Age	Eye glasses	Smile	Gender	Age	Eye glasses	Smile
$\mathcal{Z}$ - Interp.	0.121	0.053	0.078	0.095	0.51	0.61	0.59	0.61	75.78	86.60	83.81	92.83
$\mathcal{W}$ - Interp.	0.064	0.044	0.042	0.072	0.60	0.64	0.64	0.59	79.17	80.89	90.65	84.06
$\mathcal{W}^+$ - Interp.	0.062	0.040	0.039	0.064	**0.61**	0.64	0.66	0.65	79.82	85.10	92.78	89.64
InterFaceGAN	0.142	0.046	0.046	0.053	0.55	0.63	0.64	0.62	77.83	89.73	93.67	93.05
StyleSpace	0.059	**0.010**	0.012	0.025	0.60	**0.72**	0.66	0.62	80.44	**96.36**	92.37	90.25
Latent-Transformer	0.064	0.011	0.010	0.028	0.59	0.71	0.69	0.70	44.80	49.26	83.38	79.19
Ours	**0.053**	0.015	**0.008**	**0.022**	**0.61**	0.66	**0.72**	**0.71**	**80.47**	89.37	**94.99**	**95.21**

TABLE V

COMPARISON OF QUANTITATIVE RESULTS MEASURED ON CELEBA-HQ BETWEEN OUR METHOD AND OTHER BENCHMARKS. THE METRIC 'ID' IS THE CONFIDENCE SCORE OF FACE VERIFICATION OBTAINED BY THE FACE++ ONLINE ANALYSIS TOOLKIT

Method	MSE (↓)				SSIM (↑)				ID (↑)			
	Gender	Age	Black Hair	Blond Hair	Gender	Age	Black Hair	Blond Hair	Gender	Age	Black Hair	Blond Hair
StarGAN	0.021	0.021	0.064	0.074	**0.69**	**0.69**	0.63	0.62	**89.26**	89.27	92.43	92.62
AttGAN	0.014	0.011	0.017	0.011	0.68	**0.69**	0.63	0.63	89.05	92.68	94.28	94.39
InterFaceGAN ($\mathcal{W}^+$)	0.021	0.020	0.014	0.014	0.49	0.51	0.57	0.59	81.76	82.10	91.16	91.84
StyleSpace	0.013	**0.007**	0.015	0.017	0.55	0.63	0.63	**0.65**	83.50	**94.76**	95.53	95.50
Latent-Transformer	0.018	0.008	0.021	0.031	0.55	**0.69**	0.63	0.55	53.63	60.55	92.47	85.68
Ours	**0.011**	0.009	**0.013**	**0.009**	0.57	0.60	**0.66**	**0.65**	84.80	92.50	**95.84**	**96.01**

entire image). This increases the chance of allowing unnecessary modification to image content within the target region. Other methods based on the pre-trained StyleGAN generator, i.e., InterFaceGAN, StyleSpace, and Latent-Transformer, all suffer from the same problem. Nevertheless, this does not lower the visual quality and naturalness of editing results obtained.

3) Analysis of Time and Memory Cost: We also measure the cost of time and GPU memory at inference stage, and the results are reported in TABLE VI. Clearly, the cost of

TABLE VI
COMPARISON BETWEEN OUR METHOD AND BENCHMARK METHODS IN TERMS OF IMAGE RESOLUTION, MAXIMUM GPU MEMORY COST, AND TIME COST. THE TIME COST IS MEASURED AT INFERENCE TIME, WHICH IS COMPUTED BY AVERAGING THE TIME FOR TRANSLATING 1,000 SYNTHETIC IMAGES SAMPLED AT RANDOM. MEMORY COST IS MEASURED BY THE ACTUAL GPU OCCUPATION RETURNED BY THE SYSTEM CALL

Method	Image Resolution	Maximum GPU Memory Cost (MB)	Time Cost (s)
StarGAN	128×128	755	$5.8e^{-3} \pm 6.9e^{-5}$
AttGAN	256×256	1417	$2.4e^{-3} \pm 2.0e^{-4}$
StyleSpace	1024×1024	2800	$1.8e^{-1} \pm 7.8e^{-4}$
InterFaceGAN ($\mathcal{W}^+$)	1024×1024	2575	$1.4e^{1} \pm 2.4e^{-1}$
Latent-Transformer	1024×1024	35030*	$2.9e^{-1} \pm 1.2e^{-6}$
Ours	1024×1024	2389	$3.5e^{0} \pm 5.0e^{-2}$

* We observsed a dramatic increasement in GPU memory usage returned by 'nvidia-smi' when testing with the officially released code.

testing with StarGAN and AttGAN is much lower than other methods using large-scale pre-trained generators. However, these methods could only manipulate images at 128×128 or 256×256, and could hardly be efficiently applied to high-resolution (HR) images.

StyleSpace could achieve good efficiency in both time and memory on HR images, as translation results are obtained by simply interpolating the latent code and then forwarding through the generator. The time cost of InterFaceGAN ($\mathcal{W}^+$) is the largest as it takes much time to detect entangled facial attributes with 3rd party APIs, which would further increase as more facial attribute are taken into consideration. If user interaction is involved to choose the facial semantic to be removed, the time cost would become hard to estimated and has large variation across users. Although Latent-Transformer takes much less time for editing a single image, we observe a dramatic increasement in memory usage using the officially released code. Theoretically, Latent-Transformer trains 18 fully connected networks to manipulate the latent code in $\mathcal{W}^+$ space, which would inevitably increase both the storage and memory cost.

Our method has a lower memory cost compared to Latent-Transformer and is more time-efficient compared to InterFaceGAN. Although the memory cost of our method is slightly lower compared to StyleSpace, the time needed is longer due to the optimization process. Possible solutions to this problem will be discussed in Section V.

V. LIMITATION AND DISCUSSION

Although experimental results have demonstrated the inherent disentanglement of style space $\mathcal{S}$ and the capability of 'Style Intervention' in generating translation results with rich textural details, there is still much room for further improvement.

Specifically, although our method is lightweight (i.e., no deep network needs to be trained) and has low memory cost at test phase, the time efficiency of our method is relatively lower compared to some benchmark methods (e.g., StyleSpace [50], see TABLE VI). This is because an optimization process needs to be performed for each input image, which aims to locate the component of style code to be manipulated. Therefore, possible solutions to this problem include proposing a more efficient approach to find the style channels that are closely related to target semantics [64], as well as exploring to generalize latent directions computed based on one input image to other data samples [65].

Another limitation of our method is that the performance on realistic images is relatively lower to that on synthetic faces, and one possible reason is the poor quality of latent embeddings obtained by the vanilla GAN inversion technique. Specifically, the source of error is two-fold: 1) *the inherent reconstruction error* which would inevitably be accumulated to the manipulation results, and 2) *the limited editability of embedded representations* [54], [62] which could introduce extra ghosting artifacts when traversing through the latent space. To solve these problems, many GAN inversion methods [10], [54], [62], [66], [67] have been proposed to improve both the reconstruction accuracy and editability of embedded latent codes. Therefore, adopting a more advanced GAN inversion for real image embedding could potentially improve the performance of our method.

VI. CONCLUSION

Due to the intrinsically entangled nature of existing latent spaces of style-based generators, image translations obtained by recent studies are spatially entangled and thus it is undesirable in practical applications. In this paper, we take a close inspection of the network structure and propose to intervene the style code directly for the precise manipulation of individual feature map. Furthermore, a lightweight and flexible optimization-based algorithm is proposed based on the in-depth observation of internal mechanism of style-based generators. Extensive experimental results have demonstrated the ability of our method in achieving spatially disentangled translation of both real and synthesized face images with high resolutions.

REFERENCES

[1] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.
[2] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.
[3] W. Jiang et al., "PSGAN: Pose and expression robust spatial-aware GAN for customizable makeup transfer," in *Proc. CVPR*, 2020, pp. 5194–5202.
[4] Y. Men, Y. Mao, Y. Jiang, W.-Y. Ma, and Z. Lian, "Controllable person image synthesis with attribute-decomposed GAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5084–5093.
[5] J. Wang et al., "Neural pose transfer by spatially adaptive instance normalization," in *Proc. CVPR*, 2020, pp. 5831–5839.
[6] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "SEAN: Image synthesis with semantic region-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5104–5113.
[7] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "StarGAN v2: Diverse image synthesis for multiple domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8188–8197.
[8] C. Fu, Y. Hu, X. Wu, G. Wang, Q. Zhang, and R. He, "High-fidelity face manipulation with extreme poses and expressions," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2218–2231, 2021.

[9] E. Collins, R. Bala, B. Price, and S. Susstrunk, "Editing in style: Uncovering the local semantics of GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5771–5780.

[10] E. Richardson et al., "Encoding in style: A StyleGAN encoder for image-to-image translation," in *Proc. CVPR*, 2021, pp. 2287–2296.

[11] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Proc. ECCV*, 2020, pp. 319–345.

[12] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y. Yang, "HoloGAN: Unsupervised learning of 3D representations from natural images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7588–7597.

[13] I. Goodfellow et al., "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.

[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. ICLR*, 2016, pp. 1–16.

[15] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. ICLR*, 2018, pp. 1–35.

[16] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1501–1510.

[17] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2337–2346.

[18] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "GANSpace: Discovering interpretable GAN controls," 2020, *arXiv:2004.02546*.

[19] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1532–1540.

[20] A. Voynov and A. Babenko, "Unsupervised discovery of interpretable directions in the GAN latent space," 2020, *arXiv:2002.03754*.

[21] J. Zhu et al., "Low-rank subspaces in GANs," in *Proc. NeurIPS*, vol. 34, 2021, pp. 1–11.

[22] O. K. Yuksel, E. Simsar, E. G. Er, and P. Yanardag, "LatentCLR: A contrastive learning approach for unsupervised discovery of interpretable directions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14263–14272.

[23] R. Abdal, P. Zhu, N. Mitra, and P. Wonka, "StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows," 2020, *arXiv:2008.02401*.

[24] D. Bau et al., "Semantic photo manipulation with a generative image prior," 2020, *arXiv:2005.07727*.

[25] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9243–9252.

[26] H. Yang, L. Chai, Q. Wen, S. Zhao, Z. Sun, and S. He, "Discovering interpretable latent space directions of GANs beyond binary attributes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12177–12185.

[27] Y. Alharbi and P. Wonka, "Disentangled image generation through structured noise injection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5134–5142.

[28] S. Hong, M. Arjovsky, D. Barnhart, and I. Thompson, "Low distortion block-resampling with spatially stochastic networks," in *Proc. NeurIPS*, 2020, pp. 4441–4452.

[29] R. Suzuki, M. Koyama, T. Miyato, T. Yonetsuji, and H. Zhu, "Spatially controllable image synthesis with internal representation collaging," 2018, *arXiv:1811.10153*.

[30] A. Tewari et al., "StyleRig: Rigging StyleGAN for 3D control over portrait images," in *Proc. CVPR*, 2020, pp. 6142–6151.

[31] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6541–6549.

[32] D. Bau et al., "GAN dissection: Visualizing and understanding generative adversarial networks," in *Proc. ICLR*, 2019, pp. 1–18.

[33] B. Peng, H. Fan, W. Wang, J. Dong, and S. Lyu, "A unified framework for high fidelity face swap and expression reenactment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3673–3684, Jun. 2021.

[34] M. Duan, K. Li, Q. Liao, and Q. Tian, "DEF-Net: A face aging model by using different emotional learnings," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 3012–3022, May 2021.

[35] Y. Wu, R. Wang, M. Gong, J. Cheng, Z. Yu, and D. Tao, "Adversarial UV-transformation texture estimation for 3D face aging," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4338–4350, Jul. 2021.

[36] L. Zhang, H. Yang, T. Qiu, and L. Li, "AP-GAN: Improving attribute preservation in video face swapping," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2226–2237, Apr. 2021.

[37] X. Tu et al., "Image-to-video generation via 3D facial dynamics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 1805–1819, May 2021.

[38] X. Shu, J. Tang, Z. Li, H. Lai, L. Zhang, and S. Yan, "Personalized age progression with bi-level aging dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 905–917, Apr. 2017.

[39] X. Shu, J. Tang, H. Lai, L. Liu, and S. Yan, "Personalized age progression with aging dictionary," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3970–3978.

[40] Y. Sun, J. Tang, X. Shu, Z. Sun, and M. Tistarelli, "Facial age synthesis with label distribution-guided generative adversarial network," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2679–2691, 2020.

[41] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.

[42] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "AttGAN: Facial attribute editing by only changing what you want," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5464–5478, Nov. 2019.

[43] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards diverse and interactive facial image manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5549–5558.

[44] M. Liu et al., "STGAN: A unified selective transfer network for arbitrary image attribute editing," in *Proc. CVPR*, 2019, pp. 3673–3682.

[45] Y.-J. Lin, P.-W. Wu, C.-H. Chang, E. Chang, and S.-W. Liao, "RelGAN: Multi-domain image-to-image translation via relative attributes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5914–5922.

[46] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[47] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. ECCV*, 2018, pp. 286–301.

[48] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "PULSE: Self-supervised photo upsampling via latent space exploration of generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2437–2445.

[49] X. Yao, A. Newson, Y. Gousseau, and P. Hellier, "A latent transformer for disentangled face editing in images and videos," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13789–13798.

[50] Z. Wu, D. Lischinski, and E. Shechtman, "StyleSpace analysis: Disentangled controls for StyleGAN image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12863–12872.

[51] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN: How to embed images into the StyleGAN latent space?" in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4432–4441.

[52] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN++: How to edit the embedded images?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8296–8305.

[53] J. Gu, Y. Shen, and B. Zhou, "Image processing using multi-code GAN prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3012–3021.

[54] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain GAN inversion for real image editing," in *Proc. ECCV*, 2020, pp. 592–608.

[55] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. ICLR*, 2013, pp. 1–8.

[56] Y. Xu, Y. Shen, J. Zhu, C. Yang, and B. Zhou, "Generative hierarchical features from synthesizing images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4432–4442.

[57] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[58] K. Seonghyeon. *Pytorch Implementation of Styleganv2*. Accessed: Aug. 28, 2021. [Online]. Available: https://github.com/rosinality/stylegan2-pytorch

[59] Megvii. *Face++ Research Toolkit*. Accessed: Jun. 12, 2020. [Online]. Available: http://www.faceplusplus.com

[60] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. *Official PyTorch Implementation of StarGAN*. Accessed: Aug. 28, 2021. [Online]. Available: https://github.com/yunjey/stargan

[61] E. Y.-J. Lin. *PyTorch Implementation of AttGAN*. Accessed: Aug. 28, 2021. [Online]. Available: https://github.com/elvisyjlin/AttGAN-PyTorch
[62] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an encoder for StyleGAN image manipulation," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–14, Aug. 2021.
[63] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 586–595.
[64] M. J. Chong, W.-S. Chu, A. Kumar, and D. Forsyth, "Retrieve in style: Unsupervised facial feature transfer and retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3887–3896.
[65] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler, "EditGAN: High-precision semantic image editing," in *Proc. NIPS*, vol. 34, 2021, pp. 16331–16345.
[66] Y. Alaluf, O. Tov, R. Mokady, R. Gal, and A. Bermano, "HyperStyle: StyleGAN inversion with HyperNetworks for real image editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18511–18521.
[67] T. M. Dinh, A. T. Tran, R. Nguyen, and B.-S. Hua, "HyperInverter: Improving StyleGAN inversion via hypernetwork," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11389–11398.

Yunfan Liu received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 2015, the M.S. degree in electronic engineering systems from the University of Michigan, Ann Arbor, USA, in 2017. He is currently pursuing the Ph.D. degree with the National Laboratory of Pattern Recognition, Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. His research interests include computer vision, pattern recognition, and machine learning.

Qi Li (Member, IEEE) received the B.E. degree in automation from the China University of Petroleum, Qingdao, China, in 2011, the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2016. He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Center for Research on Intelligent Perception and Computing, CASIA. His research interests include face recognition, computer vision, and machine learning.

Qiyao Deng received the B.E. degree in automation from the Beijing University of Chemical Technology, Beijing, China, in 2017, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CASIA), China, in 2022. Her research interests include biometrics, pattern recognition, computer vision, and machine learning.

Zhenan Sun (Senior Member, IEEE) received the B.S. degree in industrial automation from the Dalian University of Technology, Dalian, China, in 1999, the M.S. degree in system engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2006. Since 2006, he has been a Faculty Member with the National Laboratory of Pattern Recognition, Center for Research on Intelligent Perception and Computing, CASIA, where he is a Professor. He has authored or coauthored more than 200 technical articles. His current research interests include biometrics, pattern recognition, and computer vision. He is a fellow of the IAPR, and an Associate Editor of the IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE.