Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-20062-9

HyperplaneGAN: a unified consistent translation framework for facial attribute editing

Defang Li[1,3,4] · Huiqi Deng[5] · Peng Qin[3,4] · Weifu Chen[2,3,4] · Guocan Feng[3,4]

Received: 6 October 2023 / Revised: 18 June 2024 / Accepted: 8 August 2024

Abstract

Facial attribute editing has been obtaining remarkable progress as the rapid development in deep generative models. Existing algorithms can be roughly grouped into two distinct categories: attribute-guided models and exemplar-guided models. These models achieve impressive facial attribute editing results, however, there are some limitations. For example, images generated by current attribute-guided models are lack of diversity and attribute styles are not controllable. For exemplar-guided models, low transfer precision and fidelity of generated images are commonly complained issues. In order to generate high-quality attribute-controllable facial images, we propose a novel unified translation framework called HyperplaneGAN which has following advantages: (1) the proposed model can do both attribute-guided facial editing and exemplar-guided facial editing; (2) by employing latent unit swapping and linear separation constraint for learning pair-wise linearly separable disentangled representations, the model can do flexible and controllable translation; (3) cycle-consistency loss and residual attribute vectors are used to guide the model to manipulate specific attributes precisely while other attributes are kept intact. Substantial experimental results demonstrate that HyperplaneGAN outperforms state-of-the-art models on both attribute-guided facial editing and exemplar-guided facial editing, in terms of quantitative evaluation and qualitative evaluation.

Keywords Facial attribute editing · Hyperplane · Generative adversarial networks · Attribute disentanglement

✉ Weifu Chen
weifuchen@gzmtu.edu.cn

[1] College of General Education, Guangzhou Vocational College of Technology and Business, Guangzhou, China

[2] Department of Computer Science, Guangzhou Maritime University, Guangzhou, China

[3] School of Mathematics, Sun Yat-sen University, Guangzhou, China

[4] Guangdong Province Key Laboratory, Sun Yat-sen University, Guangzhou, China

[5] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, China

Published online: 22 August 2024

1 Introduction

Attribute editing has attracted many interests in computer vision, not only because of its important role for improving the performance of various face detection and recognition algorithms [1], but also its promising applications in the media and entertainment industry. Facial attribute editing aims to edit a target facial image or video by manipulating specific attributes, such as facial expression [2], hair color, hair style [3] or age[4], etc, while preserve attribute-excluding details, such as identity, background, etc [5–7]. Combining prior knowledge, early works focus on designing exclusive algorithms for specific tasks, for instance, hair generation [8], expression change [9, 10], beautification/de-beautification [11, 12], aging [13, 14], etc. However, these algorithms are difficult to be transferred to new editing tasks.

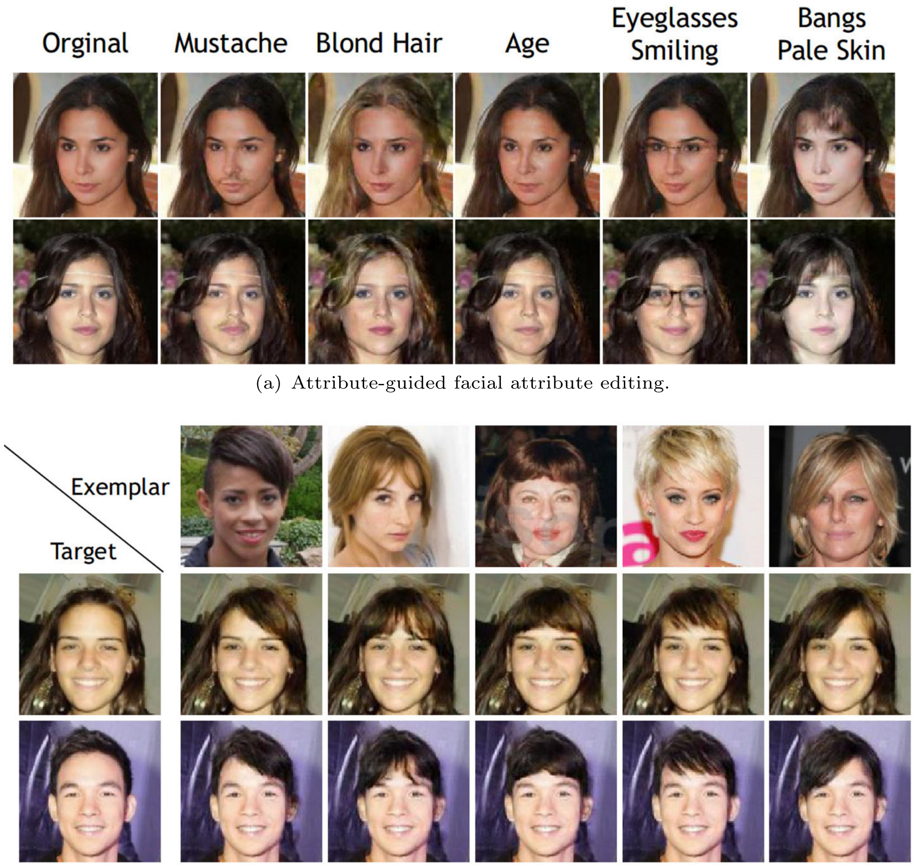
Recently, more attention has been paid to multi-attribute editing through one single model. Due to the rapid development of deep generative models, especially Generative Adversarial Nets (GANs) [15], many deep models have emerged and significantly boosted the performance of facial image editing. These methods can be broadly categorized into two groups: attribute-guided models and exemplar-guided models. Figure 1 illustrates examples of these two types of facial editing. Attribute-guided models [16–20] focus on semantically transferring attributes by making the target images possess generic attribute styles perceptually, which is guided by given attribute labels. In attribute-guided models, generation is guided by attribute label vectors, enabling attribute transfers at a semantic level. Each element of an attribute label vector typically takes a binary value of either 1 or 0. A "1" indicates that the corresponding attribute should be present in the target image, while a "0" indicates its absence. However, these models face limitations when it comes to controlling specific styles within attributes. For instance, when adding bangs, these models cannot determine the desired style of bangs to generate but the generic style generation, resulting in a restricted ability to produce diverse outcomes.

On the other hand, exemplar-guided models [21–23] concentrate on instance-level attribute transfer, involving the extraction and preservation of exemplar-dependent attributes, which are then transferred to the target images. These models rely on the availability of exemplars for guidance. Without exemplar guidance, exemplar-based methods struggle to manipulate target images, even when attempting to generate generic attribute styles. Moreover, low transfer precision and fidelity in the generated images have been commonly reported issues with these models.

It is natural to ask whether we can do the attribute-guided editing and exemplar-guided editing in a single model. To the best of our knowledge, most of the existing facial editing models only perform well in one kind of the editing. To address the need for a unified multi-attribute transfer model for facial image editing, we propose a novel framework named HyperplaneGAN. This framework caters to both attribute-guided editing and exemplar-guided editing without the need for pairwise training samples, making it highly flexible for application.

Specifically, HyperplaneGAN has an encoder-decoder architecture. The encoder is designed for learning the latent representation, and the decoder is used for reconstructing images from latent representations. Unlike conventional encoder-decoder models, our framework assumes that the latent representation can be separated into an attribute-relevant part and an attribute-irrelevant part, with each attribute-relevant part consisting of multiple units, each corresponding to a single attribute. Thus, attributes can be modified by directly manipulating the corresponding units.

In addition, we assume that facial images with and without specific attribute are linearly separable in the latent space, meaning that the latent unit of images with certain attribute can be distinguished from that of images without this attribute by a hyperplane. We explain



(a) Attribute-guided facial attribute editing.

(b) Exemplar-guided facial attribute editing examples of transferring bang styles.

Fig. 1 Examples of facial attribute editing generated by our proposed model. (a) Attribute-guided facial attribute editing examples. No exemplar was involved. The leftmost column is the original images, attributes expected to be modified are listed in the top row. (b) Exemplar-guided facial attribute editing examples. The target images are shown in the leftmost column and the exemplars are listed in the top row. The goal of this group of examples is to demonstrate the results of transferring bang styles from the exemplars to the target images

why this assumption is reasonable and easily achievable in Sec. 3.5. When moving latent unit along the normal vector of one separating hyperplane, the corresponding attribute of the target images will be enhanced or weakened. Furthermore, we can assume that these separating hyperplanes for the different attributes are pairwise orthogonal so that changing one attribute along the corresponding normal vector won’t affect other attributes. This setting allows us to modify attributes continuously without giving exemplars. Based on these assumptions, the proposed model aims to achieve following goals:

- Given exemplars, the model can exactly transfer specific attribute styles owned by the exemplars to target images while attribute-excluding details can be well preserved.
- Without exemplar images involved, the model can perceptually generate target images by modifying specific attributes.

To generate sharp and accurate target images, two additional techniques are employed in the model. The first technique is known as reconstruction cycle consistency. By taking two facial images, exchanging the editing attributes, and utilizing the decoder, we can generate two images with swapped attributes. Subsequently, these two generated images are fed into the encoder to learn the latent representations, followed by attribute swapping once again to produce images that resemble the originals. Since the editing attributes have been swapped twice, the final generated images should theoretically be identical to the original ones. The second technique employed in the model involves incorporating residual attribute vectors during the generation process. Using a complete attribute vector is unnecessary and may even have adverse effects on the editing. Instead, residual attribute vectors serve as switches, indicating to the decoder which attributes should be modified and which should be retained. The experimental results discussed in Sec. 4 validate that the utilization of these two techniques results in higher-quality and more realistic generations.

Major contributions of the proposed model include:

- We propose a unified facial editing model for both attribute-guided editing and exemplar-guided editing, which is more powerful in facial editing and will bring much more convenience for applications.
- The proposed model introduces several novel techniques to generate sharp and accurate target images, including linearly separation by hyperplanes to disentangle the attributes, latent unit swapping constraint, reconstruction cycle consistency etc. These techniques can be easily applied to other editing tasks.
- The results of extensive experiments show that the proposed model is capable of generating images for complex facial editing.

The paper is structured as follows. Sec. 2 provides an overview of the related works. In Sec. 3, we present the unified attribute transfer framework. Sec. 4 showcases extensive experimental results, while Sec. 5 presents the conclusion.

2 Related works

2.1 Generative adversarial networks

Generative Adversarial Networks (GANs) [15] are powerful latent variable models that can be used to learn complex real-world distributions. Especially for images, GANs have emerged as one of the dominant approaches for generating images of surprising complexity and realism. Typically GAN consists of a Generator ($\mathcal{G}$) and a Discriminator ($\mathcal{D}$) that compete in a two-player minimax game, where $\mathcal{G}$ tries to synthesize fake samples from random noises based on a prior distribution, whereas $\mathcal{D}$ is trained to distinguish these synthetic images from real ones. The two players combat each other and can theoretically reach an equilibrium when the distribution p_g of the synthetic images converges to the distribution p_{data} of real images. Once reaching the equilibrium, the generator is able to produce indistinguishable fake images.

Due to its ability to produce sharp and realistic images, GAN has been widely used in image generation. However, some researches show that GAN is easily suffered from model collapse which leads to instability in optimization [19, 24]. Many efforts have been made to improve the stability of GAN. WGAN [25] and WGAN-GP [26] are two GAN-based methods that use the Wasserstein distance rather than the Kullback-Leibler distance to train the models, which has been proven that the Wasserstein distance can make the GAN training process more stable. In this work, we choose WGAN-GP [26] as the backbone of the proposed framework, since the Wasserstein-1 distance and the gradient penalty item used in WGAN-GP are more effective to stabilize the training process.

2.2 Facial attribute editing

Currently, state-of-the-art facial image attribute editing approaches are mainly based on GAN [16–19, 22, 27, 28]. Those works can be roughly classified into two categories based on different tasks: semantic-level attribute transfer (aka attribute-guided facial editing) and instance-level attribute transfer (aka exemplar-guided facial editing).

2.2.1 Semantic-level facial attribute editing

Guiding by the attribute labels, semantic-level attribute transfer directly manipulates facial attributes of target images. Many recent works have achieved impressive results in this direction [16–19, 29, 30]. Conditional GAN (cGAN) [31] extends GAN by assuming that both the generator and the discriminator are conditional on some extra information. Since GAN does not have the capability to map a real image to its latent representation, IcGAN [29] first trains the encoders to approximate the inverse mappings from the real image to the latent representation and the extra information respectively. Then it modifies and combines the extra information with the latent representation to train the decoder to generate images like cGAN. Hence, IcGAN allows to synthesize images conditioned on arbitrary conditional representation.

Fader Networks [16] employs adversarial training to learn attribute-invariant latent representation. StarGAN [17] trains a single generator that learns mappings among multiple domains and introduces an auxiliary classifier that allows a single discriminator to control multiple domains. AttGAN [18] removes the strict attribute-independent constraint from the latent representation and applies the attribute-classification constraint to the generated images to guarantee the attributes are changed correctly. STGAN [19] further improves AttGAN by incorporating selective transfer units with an encoder-decoder structure for simultaneously improving the attribute manipulation ability and the image quality. SaGAN [32] applies global spatial attention on the target images to explicitly specify areas where attribute editing will be conducted, and other irrelevant regions will be preserved.

In [33], multi-path consistency loss is introduced to evaluate the differences between direct and indirect translations to regularize the training. UGAN [34] employs a source classifier in the discriminator to determine whether the translated images still hold the features in the source domain, and remove other irrelevant features. To get the best advantages of both Bayesian inference and adversarial training, LSA-VAE [35] incorporates the ideas of cVAE [36] and cGAN [31] and imposes an adversarial training scheme on the encoder and the decoder to achieve both facial attribute editing and facial image synthesis. ClsGAN [20] introduces upper convolution residual networks(Tr-resnet) to selectively extract information from source images and target labels to improve the quality of target images and the accuracy of generated attributes.

In these models, attribute labels are directly input to the networks to guide the editing. Although these models have significantly improved the facial editing results, these methods suffer some limitations. For instance, these models cannot generate images with specific attribute styles and synthesized attributes are lacking diverse styles. This is mainly because the attribute label vectors used in the models are binary and don’t contain sufficient information about attribute styles.

2.2.2 Instance-level facial attribute editing

Instance-level attribute editing transfers specific attribute styles from exemplars to target images, which is a more challenging task. There are less studies that discuss this kind of methods. GeneGAN [21] transfers a desired attribute from a reference image to target images by constructing disentangled attribute subspaces from weakly labeled data, but it can not be extended to multi-attribute editing with a single model. DNA-GAN [37] and ELEGANT [22] encode attributes into disentangled representations and generate (residual) images with specified attributes by swapping corresponding parts of latent encodings. ST-GAN [38] uses a spatial transformer network to transform external objects into the correct positions before superimposing them onto facial images. GeoGAN [23] addresses the problem of instance-level facial attribute editing without using paired training samples by geometry-aware flow which serves as a well-suited representation for modeling the transformation between instance-level facial attributes. In detail, GeoGAN uses facial landmarks as geometric guidance to automatically learn differentiable flows, despite there exist large pose gaps. MulGAN [27] and Multi-attribute transfer [28] adopt similar latent representation swapping strategies like DNA-GAN and ELEGANT and introduce auxiliary classifiers to improve the quality of synthesized images. StarGANv2 [5] introduces style encoder to extract style code from the reference image rather than domain labels used in StarGAN [17] and utilizes it to manipulate source image, but the translations often involve unnecessary manipulations such as inconsistency in facial identity and background. HiSD [39] improves StarGANv2 by organizing the labels into a hierarchical tree structure and carefully design modules to guarantee style disentanglement, however, it is limited by the uninterpretablity of style codes. VecGAN [40, 41] uses similar hierarchical labels defined by HiSD, and learns style translation directions in a linear fashion in the latent space with orthogonality constraints and disentanglement losses, but there are still issues of artifacts and entanglement(e.g. hair color versus mustache).

Recently, StyleGAN [42, 43] has emerged as a highly successful model for generating realistic and high-quality faces. As a result, many studies have focused on exploring the latent space for real image editing which is referred as StyleGAN inversion [44–47]. However, these approaches are not trained end-to-end and often face challenges in balancing reconstruction and editability. Moreover, there is no guarantee that the resulting latent space will successfully disentangle interesting attributes.

Different from above algorithms, we propose a unified model for both semantic-level attribute transfer and instance-level attribute transfer through disentangling latent representation. The proposed model can disentangle and manipulate multiple facial attributes simultaneously and generates images with high quality.

3 Methodology

We first introduce some notations. Let $\mathcal{X}$ be a multi-attribute facial image set, and each image $\mathbf{x} \in \mathcal{X}$ is associated with an n-dimensional binary attribute label vector $\mathbf{y} = [y_1, \ldots, y_n]^T$ where $y_i \in \{0, 1\}$. If y_i is positive, it means that the image x has the corresponding attribute, otherwise x doesn't have the attribute. We denote images having an attribute as the positive set of the attribute, and images not having the attribute as the negative set. Denote $\mathcal{Y}$ as the label vector set.

3.1 Motivation

Although there are many models that have been proposed for facial attribute editing, most of them are designed for either semantic-guided attribute editing or exemplar-guided editing. Meanwhile, transferring multiple attributes may easily affect other attributes or even distort the appearances. Motivated by ELEGANT [22] and InterfaceGAN [6], we propose a unified consistent framework to generate high-quality images for both semantic-level attribute editing and instance-level attribute editing. In order to preserve appearances, the latent representation of a facial image is divided into two parts: attribute-relevant part and attribute-irrelevant part. Manipulation of facial images is only imposed on the attribute-relevant part so that the attribute-irrelevant part (e.g, identity) won't be changed during the manipulating. In order to do semantic-level attribute transfer, the attribute-relevant latent feature vector should be disentangled so that if we want to change some facial attributes, we only need to manipulate the corresponding components. Furthermore, to transfer an attribute from the negative set to the positive set precisely, we assume that there exits a hyperplane separating the negative set from the positive set. When moving the attribute unit from the negative set to the positive set along the normal vector of the hyperplane, the facial image gradually owns the attribute. Furthermore, if the attribute hyperplanes are orthogonal, i.e., the normal vectors are orthogonal, then attribute manipulations along the normal vectors are independent.

Figure 2 shows the architecture of our proposed model, which consists of one encoder $\mathcal{G}_{enc}$, one decoder $\mathcal{G}_{dec}$ and two discriminators $\mathcal{D}_{adv}$ and $\mathcal{D}_{cls}$. As can be seen, all the modules share the same encoder and the same decoder, which can greatly reduce the number of parameters. $\mathcal{D}_{adv}$ acts like the discriminator for adversarial training and $\mathcal{D}_{cls}$ is for attribute classification. For convenience, we use $\mathcal{G}$ to represent $\{\mathcal{G}_{enc}$,$\mathcal{G}_{dec}\}$. Details of designing principles and loss functions are introduced below.

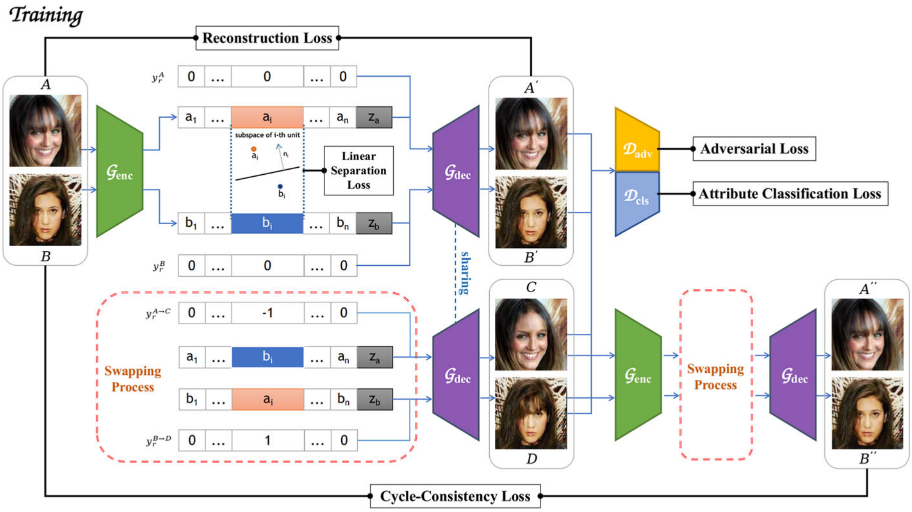


Fig. 2 Training phase of the proposed model, which consists of four losses for training: the reconstruction loss, the adversarial loss, the attribute classification loss and the cycle-consistency loss

3.2 Disentangled representation

Suppose that (A, B) is a pair of images for training the i-th attribute, where A is from the positive set and B is from the negative set. The attribute label vectors of A and B should be $\mathbf{y}^A = [y_1^A, \cdots, 1_i, \cdots, y_n^A]^T$ and $\mathbf{y}^B = [y_1^B, \cdots, 0_i, \cdots, y_n^B]^T$ respectively. Note that A and B are not needed to be paired, that is, A and B are not required to belong to the same person. Let Z_A and Z_B be the latent representations of A and B mapped by $\mathcal{G}_{enc}$. As described in Sec. 3.1, Z_A and Z_B are split into attribute-relevant parts and attribute-irrelevant parts, which are denoted as

$$\begin{aligned} Z_A &= \mathcal{G}_{enc}(A) = [\mathbf{a}_1, \cdots, \mathbf{a}_i, \cdots, \mathbf{a}_n, \mathbf{z_a}]^T \\ Z_B &= \mathcal{G}_{enc}(B) = [\mathbf{b}_1, \cdots, \mathbf{b}_i, \cdots, \mathbf{b}_n, \mathbf{z_b}]^T \end{aligned} \tag{1}$$

where $[\mathbf{a}_1, \cdots, \mathbf{a}_i, \cdots, \mathbf{a}_n]$ and $[\mathbf{b}_1, \cdots, \mathbf{b}_i, \cdots, \mathbf{b}_n]$ are attribute-relevant parts of A and B, and $\mathbf{z_a}$ and $\mathbf{z_b}$ are the attribute-irrelevant parts. $\mathbf{a}_i$ and $\mathbf{b}_i$ are d_i-dimensional representing the i-th attribute. Without loss of generality, we assume that $d_1 = d_2 = \cdots = d_n = d$, that is, each attribute embedding can be represented by a d-dimensional vector. The attribute-irrelevant parts $\mathbf{z_a}$ and $\mathbf{z_b}$ are designed to preserve necessary information for the recovery of facial details, such as identity, background etc., which should keep unchanged during attribute manipulations.

3.3 Swapping corresponding units

Although we have allocated a specific unit for the corresponding attribute, we need to train the model to map the attribute into the specific unit. The training process is implemented by swapping corresponding units introduced in the following. As shown in Fig. 2, suppose we have got Z_A and Z_B for the input images A and B, then we swap the units $\mathbf{a}_i$ and $\mathbf{b}_i$ for the i-th attribute and obtain new latent representations $Z_{(A \to B)_i} = [\mathbf{a}_1, \cdots, \mathbf{b}_i, \cdots, \mathbf{a}_n, \mathbf{z_a}]$ and $Z_{(B \to A)_i} = [\mathbf{b}_1, \cdots, \mathbf{a}_i, \cdots, \mathbf{b}_n, \mathbf{z_b}]$ to generate intermediate images C and D. Then the attribute label vectors for C and D should be $\mathbf{y}^C = \left[y_1^A, \cdots, 0_i, \cdots, y_n^A\right]^T$ and $\mathbf{y}^D = \left[y_1^B, \cdots, 1_i, \cdots, y_n^B\right]^T$. That is, the i-th attribute of C will be changed from $\mathbf{a}_i$ to $\mathbf{b}_i$ but other attributes of A will be preserved. Similarly, the i-th attribute of D will be changed from $\mathbf{b}_i$ to $\mathbf{a}_i$ but other attributes of B will be preserved.

We use a scheme like autoencoder to train the encoder $\mathcal{G}_{enc}$ and the decoder $\mathcal{G}_{dec}$. However, for each attribute, we have six paths for the encoder-decoder process: $A \to A^{'}$, $B \to B^{'}$, $A \xrightarrow{swap_i(A,B)} C$, $B \xrightarrow{swap_i(B,A)} D$, $C \xrightarrow{swap_i(C,D)} A^{''}$, $D \xrightarrow{swap_i(D,C)} B^{''}$, where, for example, $swap_i(A, B)$ is defined as the swapping operator that returns $Z_{(A \to B)_i}$. Combining with other techniques described in the following sections, the model can be trained to faithfully map attributes into corresponding units, which we will testify in experiments.

3.4 Residual attribute

In order to exactly tell the decoder whether an attribute should be changed or preserved in the generation, we introduce auxiliary vectors called residual attribute vectors of the attribute label vectors [19, 48] into the generation to guide the model. Following the above examples, the residual attribute vector for $A \xrightarrow{swap_i(A,B)} C$ is computed as

$$\mathbf{y}^r_{(A\rightarrow C)_i} = \mathbf{y}^C - \mathbf{y}^A = (0, ..., -1_i, ..., 0) \tag{2}$$

and the residual attribute vector for $B \xrightarrow{swap_i(B,A)} D$ is computed as

$$\mathbf{y}^r_{(B\rightarrow D)_i} = \mathbf{y}^D - \mathbf{y}^B = (0, ..., 1_i, ..., 0). \tag{3}$$

If a component is 0, it means that the corresponding attribute should be preserved. Otherwise, the attribute will be changed. Then, the decoding process can be summarized as $C = \mathcal{G}_{dec}\left(Z_{A\rightarrow B}, \mathbf{y}^r_{(A\rightarrow C)_i}\right)$ and $D = \mathcal{G}_{dec}\left(Z_{B\rightarrow A}, \mathbf{y}^r_{(B\rightarrow D)_i}\right)$. Figure 2 demonstrates this swapping and generation process for swapping i-th pair of units. We perform this swapping process with respect to each single attribute iteratively until all attributes concerned are involved, so as to implement multi-attribute translation.

Once finishing training, given an exemplar, we can do facial attribute editing through this swapping process. Figure 3(a) demonstrates the swapping process of "Eyeglasses" attribute from the exemplar to the target image.

3.5 Defining hyperplanes

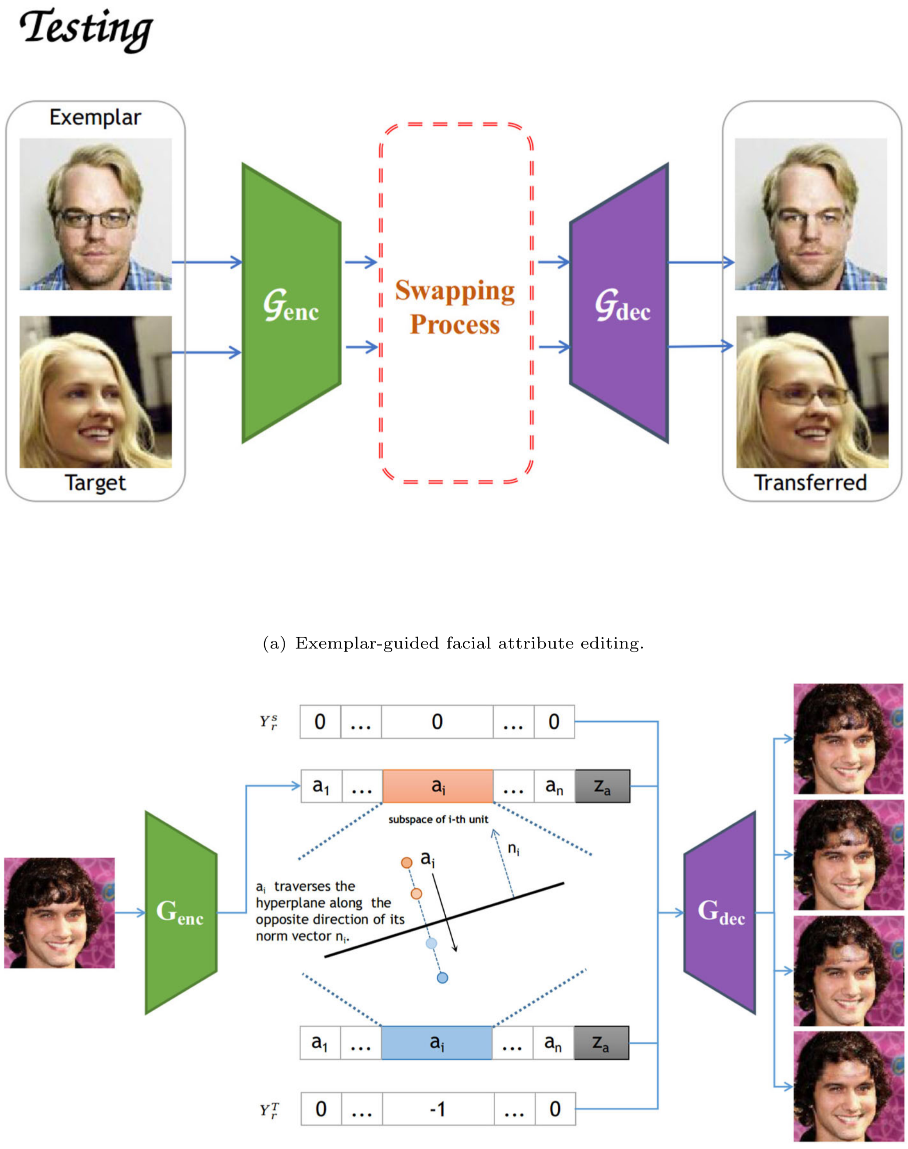
In Sec. 3.3, we have introduced an attribute editing technique by swapping semantic units. This approach can be used for exemplar-guided attribute editing. In order to capacitate the model to do the attribute-guided editing, we assume that latent representations of the positive set and the negative set of each attribute are linear separable. That is, there exits a hyperplane separating the latent codes of the positive samples (w.r.t. the corresponding attribute) from those of the negative samples. Therefore, when moving the attribute unit across the boundary of the hyperplane along its normal vector, the attribute will turn into the opposite.

We notice that in InterfaceGAN [6] the authors also introduced the hyperplane concept in interpreting the disentangled face representation learned by GANs. Our hyperplane used in the proposed model differs from theirs in the following three aspects

- The hyperplane in InterfaceGAN was learned from the latent codes by linear SVMs, while the hyperplane can be set arbitrarily in our model and latent representations can be learned automatically to fit the hyperplane assumption to guarantee attribute disentanglement.
- For multiple attribute editing, InterfaceGAN assumes that the latent codes are sampled from the standard normal distribution, and different entries of the latent semantic scores are disentangled if and only if the normal vectors of the learned hyperplanes are orthogonal. However, their algorithm cannot guarantee the orthogonality, that is, some semantics may entangle with others in InterfaceGAN. In contrast, since the hyperplanes are manually set in our model, we can directly set them orthogonal.
- Since InterfaceGAN is defined on other state-of-the-art GAN models, it can be thought as a two-stage GAN model. In contrast, the proposed hyperplane-based model is an end-to-end deep learning model.

In details, the definition of hyperplane is given as

Definition 1 Given $\mathbf{n} \in \mathbb{R}^d$ with $\mathbf{n} \neq \mathbf{0}$, the set $\left\{\mathbf{m} \in \mathbb{R}^d : \mathbf{n}^T\mathbf{m} = 0\right\}$ defines a hyperplane in $\mathbb{R}^d$, and $\mathbf{n}$ is called the normal vector.



(a) Exemplar-guided facial attribute editing.

(b) Attribute-guided facial attribute editing.

Fig. 3 Testing phase of the proposed model. Our model can complete two categories of facial attribute editing: (a) demonstrates transferring a specific attribute (here, eyeglasses) from the exemplar to the target image, which is by swapping the corresponding pair of units in the latent representations to achieve exact attribute style transfer. Note that the swapping process is the same as in Fig. 2 (b) demonstrates facial attribute editing by manipulating latent units along the direction of the normal vectors in the attribute-relevant latent space. In this kind of tasks, it does not require exemplars to guide the model to generate novel images with specific attributes (here, morphing of bangs removal)

Given a hyperplane H_i with a normal vector $\mathbf{n}_i$ associated with the i-th attribute, without loss of generality, we assume that the positive set of the attribute lies in the half space with $\mathbf{n}_i^T\mathbf{m} > 0$ ($\mathbf{m} \in \mathbb{R}^d$), and the negative set locates in the other half space with $\mathbf{n}_i^T\mathbf{m} < 0$. Then, the inner product between $\mathbf{m}$ and $\mathbf{n}_i$ can be used as a metric to indicate which category the i-th attribute of a sample belongs to. Note that the inner product is not a distance since it will be negative.

The following proposition states that when considering only one CONV-BN-LeakyReLU module (see Sec. 4.1 for the configuration), the latent representations of random samples with attribute A from a normal distribute are very likely to locate close a given hyperplane, which generalizes the result in [6].

Proposition 1 *Suppose* $\mathbf{X}_A \in \mathbb{R}^{p_1}$ *is a multivariate random vector to attribute A, and suppose* $\mathbf{X}_A \backsim \mathcal{N}(\mu_A, \Sigma_A)$. *Let y be the ground truth label of sample x and ŷ be its predicted label given by a classifier. Denote the output of a CONV-BN-LeakyReLU module as* $\mathbf{z}_A \in \mathbb{R}^{p_2}$, *where the LeakyReLU is with hyperparameter* α, $0 < \alpha < 1$. *Then, given a predefined hyperplane H with normal vector* $\mathbf{n}$, *we have* $Pr(|\mathbf{n}^T\mathbf{z}_A| \leq 2a\sqrt{\frac{p_2}{p_2-2}}) \geq (1-3e^{-cp_2})(1-\frac{2}{a}e^{-a^2/2})$ *for any* $a \geq 1$ *and* $p_2 \geq 4$. *Here c is a is fixed constant positive number.*

Proof 1. Suppose the convolutional operator of CONV $\mathbf{W} \in \mathbb{R}^{p_2 \times p_1}$ is a convolutional matrix, and let $\tilde{\mathbf{X}}_A = \mathbf{W}\mathbf{X}_A$, $\tilde{\mu}_A = \mathbf{W}\mu_A$, $\tilde{\Sigma}_A = \mathbf{W}\Sigma_A\mathbf{W}^T$), then it is easy to prove that $\tilde{\mathbf{X}}_A \backsim \mathcal{N}(\tilde{\mu}_A, \tilde{\Sigma}_A)$.

2. Suppose the output of BN is $\hat{X}_A$, then the distribution of $\hat{X}_A$ converges to standard normal distribution, i.e., $\hat{X}_A \backsim \mathcal{N}(\mathbf{0}, \mathbf{I}_{p_2})$.
3. Given an output of BN $\hat{\mathbf{x}}_A$, define a diagonal matrix $\mathbf{I}_{p_2}$ as

$$(I_{p_2})_{ii} = \begin{cases} 1 \text{ if } (\hat{\mathbf{x}}_A)_i \geq 0 \\ \alpha \text{ if } (\hat{\mathbf{x}}_A)_i < 0 \end{cases}.$$

   Then the output of LeakyReLU at $\hat{\mathbf{x}}_A$ can be written as $\mathbf{z}_A = LeakyReLU(\hat{\mathbf{x}}_A) = \mathbf{I}_{p_2}\hat{\mathbf{x}}_A$.
4. Without loss of generality, we fix $\mathbf{n}$ to be $[1, 0, \cdots, 0]^T$. Then $\mathbf{n}^T\mathbf{z}_A = (\mathbf{z}_A)_1$. Hence, if $(\hat{\mathbf{x}}_A)_1 \leq 2a\sqrt{\frac{p_2}{p_2-2}}$, $(\mathbf{z}_A)_1 \leq 2a\sqrt{\frac{p_2}{p_2-2}}$, which means

$$Pr\left(|\mathbf{n}^T\mathbf{z}_A| \leq 2a\sqrt{\frac{p_2}{p_2-2}}\right) \geq Pr\left(|\mathbf{n}^T\hat{\mathbf{x}}_A| \leq 2a\sqrt{\frac{p_2}{p_2-2}}\right)$$

   According to **Property 2** in [6], since $Pr(|\mathbf{n}^T\hat{\mathbf{x}}_A| \leq 2a\sqrt{\frac{p_2}{p_2-2}}) \geq (1-3e^{-cp_2})(1-\frac{2}{a}e^{-a^2/2})$, we have $Pr(|\mathbf{n}^T\mathbf{z}_A| \leq 2a\sqrt{\frac{p_2}{p_2-2}}) \geq (1-3e^{-cp_2})(1-\frac{2}{a}e^{-a^2/2})$, which ends the proof. □

When minimizing the linear separation loss defined in Sec. 3.7.4, the imposing constraint drives the encoder to learn latent representations that can be separated by the hyperplane. The evaluation results of InterFaceGAN [6] on the assumption that any binary attribute can be separated by a hyperplane in the latent space also evidence such latent representations can be learnt. In particular, when n orthogonal normal vectors are preset, the encoder can be trained to learn attribute-relevant latent representations which can be separated by the corresponding hyperplanes. In Sec. 4, experimental results also confirm that this assumption is reasonable and the latent codes of positive samples and negative samples are distinguishable. Figure 4 demonstrates the learning process.

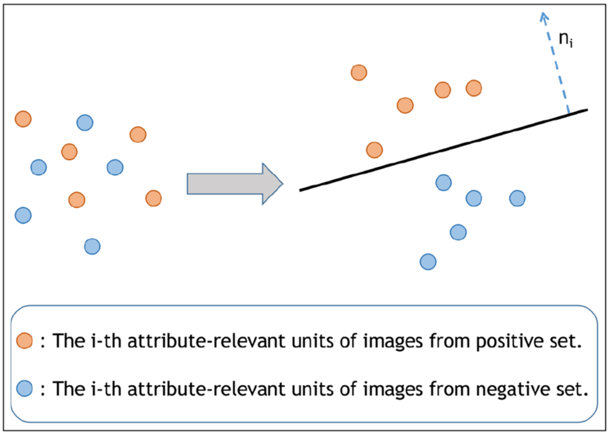

n_i
● : The i-th attribute-relevant units of images from positive set.
● : The i-th attribute-relevant units of images from negative set.

Fig. 4 Without introducing a hyperplane, pairs of samples from the positive set and the negative set of the i-th attribute may be inseparable in the latent space (Left). Given a hyperplane (or a normal vector), the encoder can be trained to learn the latent representations of the positive samples and the negative samples that can be separated by the hyperplane (Right)

After fixing the normal vectors and training the proposed model, modifying an attribute without using exemplars is simple. We just need to move the semantic unit along the corresponding normal vector, either forward or backward, until it crosses the separating hyperplane. At this point, the attribute category of the image will be changed accordingly. Thanks to the orthogonality between normal vectors, changing some attributes along their respective vectors does not affect other attributes. This disentanglement of latent representations provides considerable convenience for editing multiple attributes simultaneously.

3.6 Adversarial training

To improve the effectiveness of facial editing, adversarial learning between $\mathcal{G}_{enc}$, $\mathcal{G}_{dec}$ and $\mathcal{D}_{adv}$ is introduced to make the generated images look visually realistic. The attribute classifier $\mathcal{D}_{cls}$ is used to ensure that the swapped units do change their corresponding attributes properly.

3.7 Loss functions

In order to achieve the desired goals, we choose five loss functions to form the objective function for training the modules of the proposed model. The following loss functions are defined based on the example in Fig. 2 manipulating the i-th attribute.

3.7.1 Adversarial loss

Since the target images in the generation paths $A \xrightarrow{swap_i(A,B)} C$ and $B \xrightarrow{swap_i(B,A)} D$ in Fig. 2 usually do not exist, we employ the adversarial loss [15] for restricting the generated images having the i-th attribute that cannot be distinguished from the true ones. In this work, we leverage WGAN [25] with gradient penalty [26] as the adversarial loss for training $\mathcal{G}_{enc}$, $\mathcal{G}_{dec}$ and $\mathcal{D}_{adv}$:

$$\begin{aligned} L^{\mathcal{D}}_{adv} = & -\mathbb{E}[\mathcal{D}_{adv}(A)] - \mathbb{E}[\mathcal{D}_{adv}(B)] + \mathbb{E}\left[\mathcal{D}_{adv}\left(C\right)\right] \\ & + \mathbb{E}\left[\mathcal{D}_{adv}\left(D\right)\right] + \frac{\lambda_{gp}}{2}\left\{\mathbb{E}_{\hat{p}}\left[\left(\left\|\nabla_{\hat{p}}\mathcal{D}_{adv}(\hat{p})\right\|_2 - 1\right)^2\right]\right. \\ & \left. +\mathbb{E}_{\hat{q}}\left[\left(\left\|\nabla_{\hat{q}}\mathcal{D}_{adv}(\hat{q})\right\|_2 - 1\right)^2\right]\right\} \end{aligned} \tag{4}$$

$$L^{\mathcal{G}}_{adv} = -\mathbb{E}\left[\mathcal{D}_{adv}\left(C\right)\right] - \mathbb{E}\left[\mathcal{D}_{adv}\left(D\right)\right] \tag{5}$$

where $\hat{p}$ and $\hat{q}$ are uniformly sampled along straight lines between pairs of (A, C) and (B, D), respectively. That is, $\hat{p} = \epsilon A + (1-\epsilon)C, \epsilon \sim \mathcal{U}(0, 1)$ and $\hat{q} = \epsilon B + (1-\epsilon)D, \epsilon \sim \mathcal{U}(0, 1)$. When $\mathcal{G}_{enc}$ and $\mathcal{G}_{dec}$ are fixed, we can optimize $\mathcal{D}_{adv}$ by minimizing $L^{\mathcal{D}}_{adv}$. Alternatively, when $\mathcal{D}_{adv}$ is fixed, we can optimize $\mathcal{G}_{enc}$ and $\mathcal{G}_{dec}$ by minimizing $L^{\mathcal{G}}_{adv}$.

3.7.2 Attribute classification loss

Remember that the attribute label vectors of A and B are $\mathbf{y}^A = [y^A_1, \cdots, 1_i, \cdots, y^A_n]^T$ and $\mathbf{y}^B = [y^B_1, \cdots, 0_i, \cdots, y^B_n]^T$. After swapping the i-th attribute, the attribute label vectors of C and D should $\mathbf{y}^C = [y^A_1, \cdots, 0_i, \cdots, y^A_n]^T$ and $\mathbf{y}^D = [y^B_1, \cdots, 1_i, \cdots, y^B_n]^T$. In order to train the generator to generate images with proper latent attribute-relevant units as indicating in $\mathbf{y}^C$ and $\mathbf{y}^D$, the classifier $\mathcal{D}_{cls}$ is introduced to predict whether a generated image is owning the i-th attribute. We denote the posterior probability as $\mathcal{D}_{cls}(\cdot \mid x)$, where x is a generated image. Based on $\mathcal{D}_{cls}$, the attribute classification loss is defined as the cross-entropy loss

$$L^{\mathcal{G}}_{cls} = \mathbb{E}\left[-\log\mathcal{D}_{cls}(\hat{\mathbf{y}}^C \mid C) - \log\mathcal{D}_{cls}(\hat{\mathbf{y}}^D \mid D)\right], \tag{6}$$

where $\hat{\mathbf{y}}$ is a predicted attribute label vector. In addition, we can also define cross-entropy loss for the original images A and B

$$L^{\mathcal{D}}_{cls} = \mathbb{E}\left[-\log\mathcal{D}_{cls}(\hat{\mathbf{y}}^A \mid A) - \log\mathcal{D}_{cls}(\hat{\mathbf{y}}^B \mid B)\right] \tag{7}$$

3.7.3 Reconstruction loss

To train the encoder-decoder architecture and learn good latent representation, we employ the reconstruction loss. For the examples in Fig. 2:$A \rightarrow A^{'}$ and $B \rightarrow B^{'}$, the reconstructions are $A^{'} = \mathcal{G}_{dec}(\mathcal{G}_{enc}(A), \mathbf{y}^r_A)$ and $B^{'} = \mathcal{G}_{dec}(\mathcal{G}_{enc}(B), \mathbf{y}^r_B)$, where $\mathbf{y}^r_A = \mathbf{y}^{A^{'}} - \mathbf{y}^A = [0, 0, \cdots, 0]^T$ and $\mathbf{y}^r_B = \mathbf{y}^{B^{'}} - \mathbf{y}^B = [0, 0, \cdots, 0]^T$. Hence, the reconstruction loss can be defined as

$$L^{\mathcal{G}}_{\text{rec}} = \left\|A - A^{'}\right\|_1 + \left\|B - B^{'}\right\|_1. \tag{8}$$

Here we used ℓ_1 norm rather than ℓ_2 norm to measure the reconstruction error, because L_1 norm can drive the model to generate more realistic images with sharp features.

3.7.4 Linear separation loss

As discussed in Sec. 3.5, we hope that the latent representations of the positive set and the negative set of one attribute can be separated by a hyperplane. Rather than to learn the hyperplanes, we preset the hyperplanes (i.e., the normal vectors of the hyperplanes) and train the encoder and the decoder to learn latent representations satisfying the linear separation constraints. Mathematically, given n orthogonal unit vectors $[\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_n]$ as the normal vectors, suppose A is from the positive set of the i-th attribute and B is from the negative set, $\mathbf{a}_i$ and $\mathbf{b}_i$ are the corresponding i-th latent semantic units of A and B. Without loss of generality, assume that $\mathbf{a}_i$ is labeled as "+1", and $\mathbf{b}_i$ is labeled as "-1". The the linear separation loss is defined as the hinge loss:

$$L_{\text{ls}}^{\mathcal{G}} = \big[\max\big(0, (1 - (+1)(\mathbf{n}_i^T \mathbf{a}_i + b_i))\big) \\ + \max\big(0, (1 - (-1)(\mathbf{n}_i^T \mathbf{b}_i + b_i))\big)\big] \tag{9}$$

where b_i is a biased term.

3.7.5 Cycle-consistency loss

For the examples in Fig. 2, if we swap the i-th attribute of A and B twice, that is, for the paths $A \xrightarrow{swap_i(A,B)} C \xrightarrow{swap_i(C,D)} A''$ and $B \xrightarrow{swap_i(B,A)} D \xrightarrow{swap_i(D,C)} B''$, it is reasonable to require that A and A'', B and B'' should be identical. We employ a cycle consistency loss [17, 49, 50] to train the generator so that good latent representations can be learned and exactly disentangled by the encoder to fit the preset sematic units, and useful information can be preserved in the representations so that facial images can be recovered from the latent codes by the decoder:

$$L_{cycle}^{\mathcal{G}} = \left\| A - A'' \right\|_1 + \left\| B - B'' \right\|_1 \tag{10}$$

3.7.6 Full objective

Finally, we combine above losses to form the objective function of the proposed model to optimize the generators and discriminators alternatively

$$L_{\mathcal{G}} = \lambda_1 L_{rec}^{\mathcal{G}} + \lambda_2 L_{cycle}^{\mathcal{G}} + \lambda_3 L_{cls}^{\mathcal{G}} + \lambda_4 L_{ls}^{\mathcal{G}} + L_{adv}^{\mathcal{G}}. \tag{11}$$

$$L_{\mathcal{D}} = \lambda_5 L_{cls}^{\mathcal{D}} + L_{adv}^{\mathcal{D}} \tag{12}$$

where $\lambda_1, \cdots, \lambda_5$ are tuning hyperparameters.

4 Experiments

Extensive experiments were conducted to evaluate the performance of the proposed model. First, we describe the model configurations in Section 4.1. Then, we introduce the benchmark dataset and the baselines for comparison in Sections 4.2 and 4.3, respectively. The experimental results on exemplar-guided editing and attribute-guided editing are presented in Sections 4.4 and 4.5. Additionally, quantitative comparison results are provided in Section 4.6. Finally, an ablation study is performed in Section 4.7.

4.1 Model configurations

The encoder $\mathcal{G}_{enc}$ is equipped with four down-sampling layers of Conv-Norm-LeakyReLU block, where "Conv" represents the convolution operation, "Norm" means the batch normalization [51], "LeakyReLU" is the LeakyReLU activation function. The decoder $\mathcal{G}_{dec}$ has four layers of Deconv-Norm-LeakyReLU block, which recovers the image back to its original size and “Deconv" represents the transposed convolution operation. For the discriminators, the binary classifier $\mathcal{D}_{adv}$ uses four down-sampling layers of Conv-Norm-LeakyReLU block followed by two layers of MLP-LeakyReLU, where MLP means fully connected layers, and the attribute classifier $\mathcal{D}_{cls}$ shares all convolutional layers with $\mathcal{D}_{adv}$, but owns two new layers of MLP-LeakyReLU block.

The model was trained with Adam optimizer [52], setting $\beta_1 = 0.5$, $\beta_2 = 0.999$, with batch size of 16 and a fixed learning rate of 0.0002 during 20 epochs. The coefficients in Eq. (11), Eq. (12) and Eq. (4) are set as: $\lambda_1 = 100$, $\lambda_2 = 1$, $\lambda_3 = 10$, $\lambda_4 = 1$, $\lambda_5 = 1$ and $\lambda_{gp} = 10$. All experiments were performed in a Pytorch platform with a single NVIDIA GTX TITAN X Pascal Graphic Card.

4.2 Benchmark dataset

All the algorithms were evaluated on CelebFaces Attributes Dataset (CelebA) [53], which contains 202599 face images of size 218 × 178 from 10177 celebrities. Each image was annotated with forty binary labels describing facial attributes like hair color, gender and age. In this work, we considered manipulating eight attributes, including “Bangs", “Eyeglasses", “Mouth Slightly Open", “Smiling", “Mustache", “Blond Hair", “Pale Skin" and “Young", due to their certain representativeness in appearance. In the experiments, data were performed preprocessing as follows: all the images were cropped in the central 170 × 170 region and scaled down to 128 × 128, and the intensity value of each pixel was normalized to $[-1, 1]$.

4.3 Baselines

We compared the proposed model with several state-of-the-art approaches: ELEGANT [22], StarGAN [17], AttGAN [18] and STGAN [19], where ELEGANT [22] is a representative instance-level attribute transfer model that is designed to transfer attributes along with corresponding styles between exemplars and target images, StarGAN [17], AttGAN [18] and STGAN [19] are semantic-level attribute transfer models that manipulate different attributes of target images by using different target attribute labels to guide the manipulation. For fair comparison, all baselines are retrained on CelebA dataset using the public released codes by the authors with default hyperparameters. Table 1 summarizes the capability of these models in handling multi-attribute editing, attribute-guided editing and exemplar-guided editing.

4.4 Exemplar-guided facial editing

In this set of experiments, we compared the performance of the models on exemplar-guided attribute editing tasks. Given an exemplar, the task is to transfer specific attributes of the

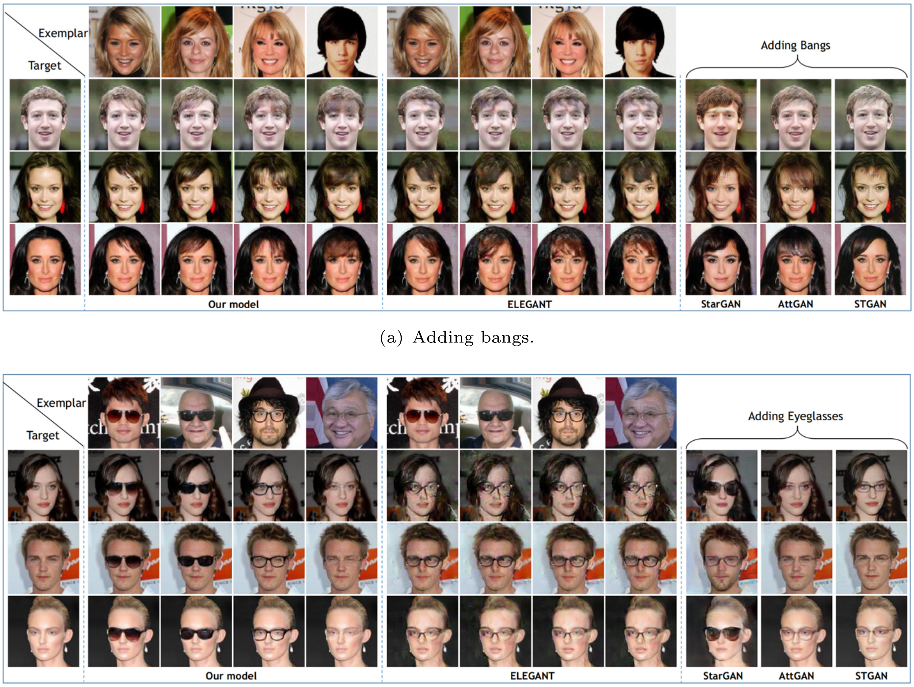
Table 1 Capability comparison with baselines and the proposed model in facial attribute editing

	StarGAN	AttGAN	STGAN	ELEGANT	Our model
Multi-attribute	✓	✓	✓	✓	✓
Attribute-guided	✓	✓	✓		✓
Exemplar-guided				✓	✓

exemplar to a target image. The transfer process of the proposed model is demonstrated in Fig. 3(a). Since Stargan [17], Attgan [18] and STGAN [19] are not designed for exemplar-guided editing, specific attribute labels are required for these three baselines.

The comparison results of transferring bangs and eyeglasses are shown in Fig. 5(a) and (b). For each task, three target images and four exemplars with different bang styles or eyeglasses were chosen, which are shown in the top row and the leftmost column in each figure, respectively. The generated results of the proposed model are shown in Column 2-5, the results generated by ELEGANT are shown in Column 6-9, and those given by Stargan, Attgan and STGAN are shown in the last three columns.

As can be seen, for different bang styles and eyeglasses, the proposed model could learn the representations of these attributes and transfer them to the target images precisely while all other attributes in the target images were kept intact. For example, skin color, background, face identity and illumination of the generated images were perfectly preserved with high visual quality. In contrast, although ELEGANT could generate images owning the exemplars' specific attributes, it failed to accurately transfer the specific attribute styles to the target images. For instance, ELEGANT transferred different types of eyeglasses in the exemplars to black-rimmed glasses in the target images (see Column 6-9 of Fig. 5(b)), and didn't achieve desirable attribute styles for bangs neither (see Column 6-9 of Fig. 5(a)). In addition,



(a) Adding bangs.

(b) Adding eyeglasses.

Fig. 5 Results of single-attribute facial editing. The results of our model and ELEGANT [22] were generated by the guidance of exemplars and those of StarGAN [17], AttGAN [18] and STGAN [19] were generated by the guidance of given attribute label vectors. (a) and (b) demonstrate editing results on attributes of "bangs" and "eyeglasses". In each figure, the leftmost column is target images and the topmost row are exemplars with specific attribute styles which were expected to be transferred to the target images by HyperplaneGAN (our model) and ELEGANT. Zoom in for better resolution.

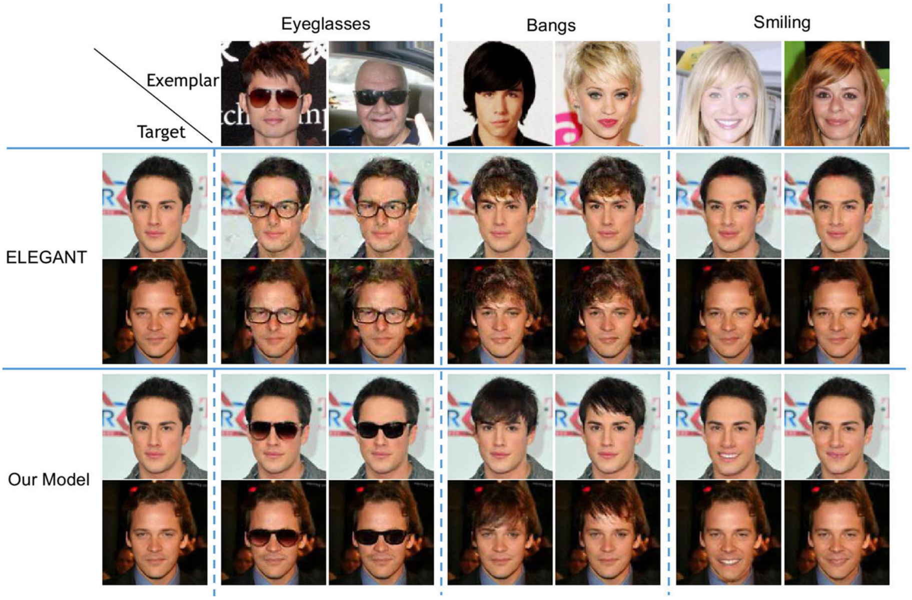


Fig. 6 Comparison exemplar-guided results of the proposed model and ELEGANT [22]. The leftmost column are target images and the topmost row of images are exemplars with specific attribute styles which are expected to be transferred to the target images.

ELEGANT also suffered from distortion in facial details. More comparison results of the proposed model and ELEGANT are displayed in Fig. 6. Notice that the exemplars' attribute styles are black glasses, left or right bangs, open-mouth or close-mouth smiling. Hyperplane-GAN could precisely transfer the exemplars' specific attributes and their attribute styles to the target images, while ELEGANT failed to transfer these attribute details to the target images again. For example, ELEGANT generated images with close-mouth smiling no matter the exemplars were open-mouth smiling or close-mouth smiling.

As discussed above, Stargan, Attgan and STGAN cannot learn the attribute label vectors from the exemplars automatically. Therefore, we need to manually provide the attribute label vectors for them. From Fig. 5(a) and (b), we can see that all these three models succeeded in generating images with specific attributes and preserving the identities, but the styles of the attributes are unpredictable. Besides, there still exist other shortcomings in the results, for example, artifacts in StarGAN, variation of hair color in StarGAN and AttGAN, variations of skin color and image brightness in STGAN. The reason for the variation in StarGAN and AttGAN is that the attributes may be enhanced or weakened by the generators to ensure that images generated under the guidance of attribute labels can be correctly classified by the discriminators. For instance, in the first example of Fig. 5(a), Zuckerberg's image was labeled as brown hair, when adding bangs, the hair color of the image generated by StarGAN was dark brown hair even though the hair color of the original image was light brown, while in the second example of Fig. 5(a), the hair color of the image generated by AttGAN was lighter than its original.

For multiple facial attributes editing by exemplars, we demonstrate the results in Fig. 7, where images in the leftmost column are exemplars and the descriptions on the left of the exemplars are the attributes which were expected to transfer to target images in the topmost

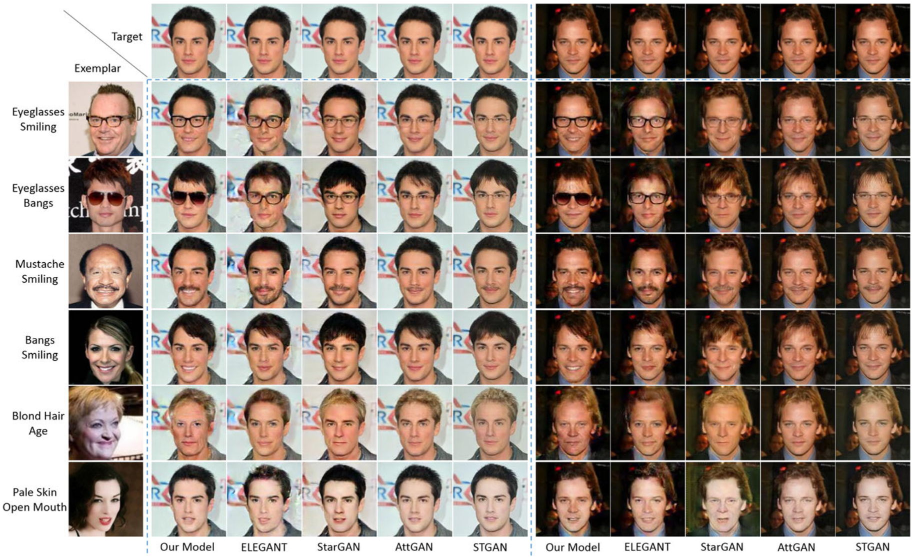


Fig. 7 Results of multiple-attribute facial editing. The results of the proposed model and ELEGANT [22] were generated by the guidance of exemplars and those of StarGAN [17], AttGAN [18] and STGAN [19] were generated by the guidance of attribute label vectors. The leftmost column are the exemplar images and their attributes which were expected to guide multiple-attribute editing. The topmost row are target images. Zoom in for better resolution.

row. Similar to single attribute editing, distortion of facial details became more severe for ELEGANT, and it failed to transfer some attribute styles such as eyeglasses, mustache, smiling, to target images. As for the other three models, some generated images were not manipulated as expected, for instance, AttGAN and STGAN failed to edit age attribute. The shortcomings of these models mentioned above still existed. In contrast, our method always performed well for multi-attribute editing and all the attributes could be transferred precisely.

4.5 Attribute-guided facial editing

One of the very important features of the proposed model is that it is capable of manipulating images by given specific attributes without exemplars involved, which benefits from the hyperplanes we have defined in the attribute-relevant latent space. Consequently, the proposed model can also be used for semantic-level attribute editing as StarGAN, AttGAN and STGAN. Given an input image, the proposed model manipulates the latent codes corresponding to the attributes we concern. In detail, when considering editing the i-th attribute, we change the i-th attribute-relevant unit $\mathbf{a}_i$ along the direction of the normal vector $\mathbf{n}_i$ of the corresponding hyperplane, that is, $\mathbf{a}_i \leftarrow \mathbf{a}_i + \alpha \mathbf{n}_i$, where α is the step size. Then the new latent representation and the residual attribute vector are concatenated as the input for the generator. Once the sign of the distance from $\mathbf{a}_i$ to the i-th hyperplane is changed, which means $\mathbf{a}_i$ passes through the hyperplane, it is expected that the i-th attribute of the generated images should be changed from the negative set to the positive set or vice versa. Since ELEGANT cannot do the attribute-guided facial editing, we only compared the proposed model with StarGAN, AttGAN and STGAN for this task. The comparison results are presented in Fig. 8

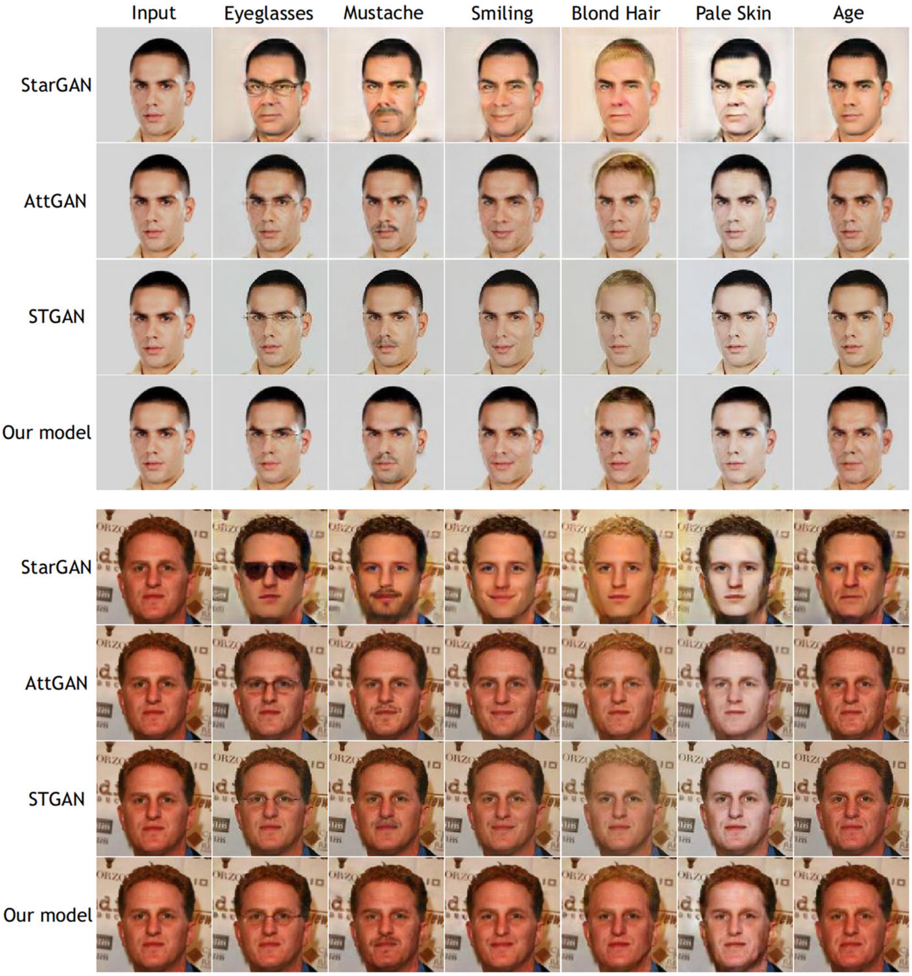


Fig. 8 Comparison results of semantic-level facial attribute editing. The proposed model were compared with other three attribute-guided models: StarGAN [17], AttGAN [18] and STGAN [19]. Zoom in for better resolution.

As can be seen, although the baselines were able to do semantic-level attribute editing, there still existed some limitations. For instance, there were artifacts in the images generated by StarGAN, resulting in the loss of facial details, which led to the failure of preserving identity. As for AttGAN, there was a halo above the head while editing hair color and the editing of age was also unsatisfactory (see the first example in Fig. 8). STGAN was able to preserve personal identity well, but it still could not successful modify age attribute (see the first example in Fig. 8). In contrast, the proposed model could modify the attributes more precisely and realistically, and both the identity and the facial details were well preserved.

Figure 9 demonstrates the morphing effect of the proposed model. As we can see, the attribute morphing of the generated images are smooth and natural. When $\alpha > 0$ and the new $\mathbf{a}_i$ located on the positive side of the hyperplane, the i-th attribute of the generated images became obvious; when $\alpha < 0$ and $\mathbf{a}_i$ located on the negative side of the hyperplane, the i-th attribute faded. This generation procedure is illustrated in Fig. 3(b).

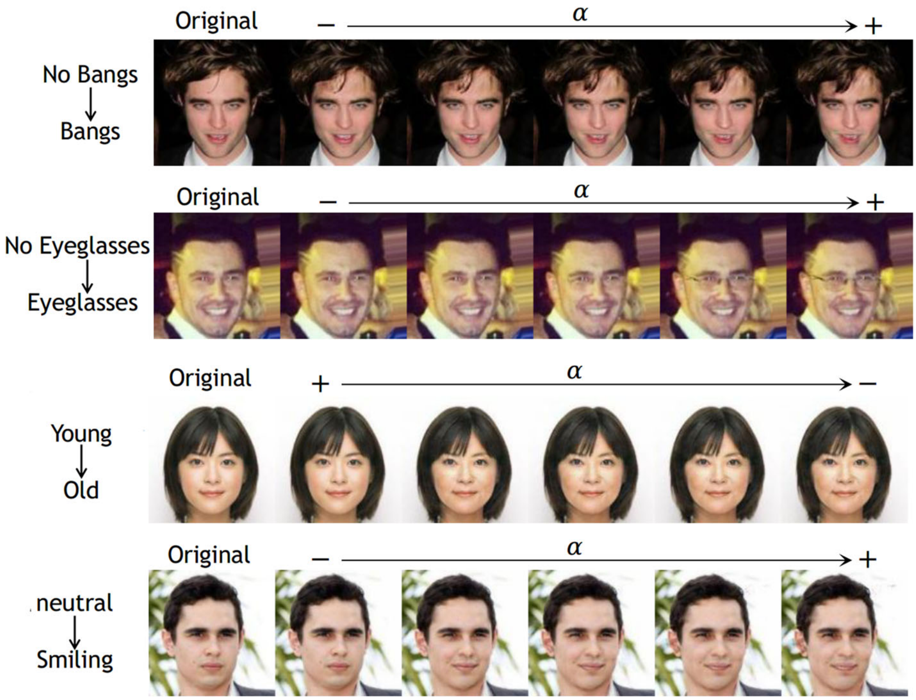


Fig. 9 Attribute morphing results of hyperplane-based facial attribute editing by the proposed model. For each group, the leftmost texts describe how the attributes were changing from the original images as the step size α was increasing or decreasing.

4.6 Quantitative analysis

As it is well known, the quality of transferred images is closely related to the quality of image reconstruction. For instance, Fig. 10 showcases the reconstructed images of three facial images produced by the proposed model and the baselines. To facilitate comparison, the original images are displayed in the top row in each group.

Upon closer examination, it can be observed that the reconstructions generated by StarGAN and AttGAN were less convincing compared to other algorithms. Attributes of the manipulated images generated by guidance of attribute labels are excessively emphasized, for instance, brown hair for input A, mustache for input B and blond hair for input C. This partially explains why StarGAN and AttGAN yielded less satisfactory results. STGAN achieved better reconstructions than StarGAN and AttGAN, which sometimes exhibited increased sharpness but enhanced illumination. In contrast, our proposed model and ELEGANT are able to preserve facial details throughout the reconstruction process.

To evaluate the image quality quantitatively, three metrics were used in this work which were listed in Table 2, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [54] and Learned Perceptual Image Patch Similarity (LPIPS) [55]. For PSNR and SSIM, the higher value means more similar between source images and their reconstructed images. For LPIPS, which evaluates similarity on deep features of images by feeding the reconstructions to pre-trained networks, such as VGG [56] or AlexNet [57], the lower value means the more similar. In addition, in order to demonstrate the complexity

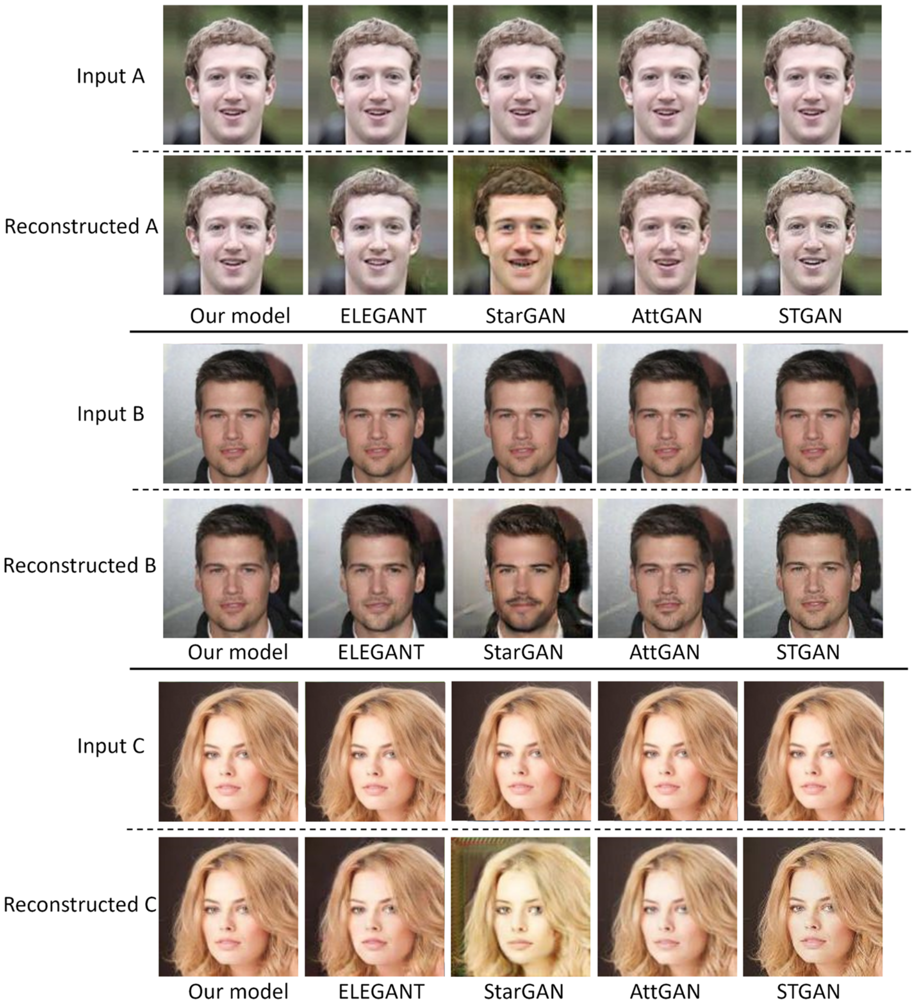


Fig. 10 Results of image reconstruction by the proposed model, ELEGANT [22], StarGAN [17], AttGAN [18] and STGAN [19]. Zoom in for better resolution.

Table 2 Quantitative comparison results for the quality of the reconstructions given by the proposed model and the baselines, in terms of PSNR, SSIM, and LPIPS. The best results are in bold font

Model	PSNR	SSIM	LPIPS	#params($\mathcal{G}$)
StarGAN	21.66	0.78	0.119	8.5M
AttGAN	30.81	0.91	0.031	43.4M
STGAN	21.72	0.66	0.061	225.1M
ELEGANT	27.75	0.88	0.041	10.2M
Our Model	**37.87**	**0.96**	**0.009**	**3.1M**

Numbers of the parameters of the generators are also listed in the right-most column

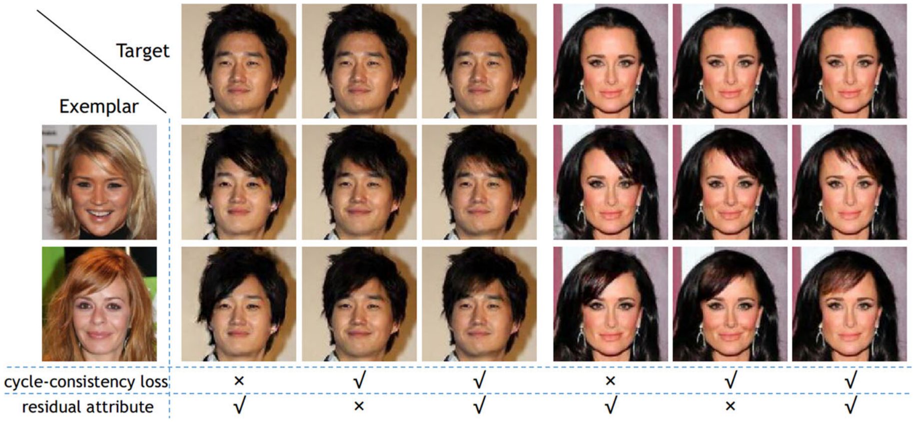

Target
Exemplar
cycle-consistency loss × √ √ × √ √
residual attribute √ × √ √ × √

Fig. 11 Ablation study on the function of the cycle-consistency loss and the residual attribute. Two groups of samples were generated by HyperplaneGAN and its two variants, transferring bangs from the exemplars to the target images.

of model architecture, numbers of parameters of the generators are also listed in the table. Benefited from the constraints on reconstruction, cycle-consistency and residual attribute, the proposed model achieved the best performance in term of all the metrics. Note that the proposed model also has the smallest number of parameters for the generator.

4.7 Ablation study

We studied the function of the cycle-consistency loss and the residual attribute during the generation. We introduced two variants of HyperplaneGAN: one without the cycle-consistency loss and one without the residual attribute, in comparison with the original configuration. The comparison results of these models are demonstrated in Fig. 11.

Without the cycle-consistency loss, although the variant could generate images with specific attributes according to the exemplars, it also brought some unwanted changes and detail loss. For example, we wanted to change the bangs of examples in Fig. 11, but the variant also changed the hair styles. The possible reason is that without the cycle-consistency constraint, the generator tends to ignore details of original image so that the generated images are not consistent with the original. On the other hand, without introducing the residual attribute vectors, although the variant could achieve attribute transferring from the exemplars, but it tended to generate images that pander to the attribute classifier, resulting in the intensification of smiling attribute. This phenomenon of attribute intensification also happens in StarGAN and AttGAN. In comparison, the original configuration of the proposed model could overcome these problems and generated more realistic and consistent results in the experiments.

5 Conclusion

To integrate the concepts of attribute-guided facial attribute editing and exemplar-guided facial attribute editing, learning a disentangled representation and understanding its inherent characteristics in the latent space is a key factor for the success of an editing model. With this insight, we proposed an encoder-decoder model based on the adversarial learning, with assumption that facial images attributes are linearly separable in the latent space. We also designed novel modules and a training pipeline to form a unified and consistent transfer model for both types of facial editing. Extensive experiments demonstrated that proposed model is competent to accurately transfer specific attributes to the target images for both kinds of editing and generated more convincing results. However, our model was trained only on the low-resolution version of the CelebA dataset, and the performance of the model on other datasets, including non-face datasets, has not been verified. For future work, in addition to low-resolution static images, we aim to strengthen the ability of the proposed method in dealing with high-resolution images. We can also consider how to add temporal constraints for video facial editing.

Acknowledgements This work is supported by the NSFC under grants Nos. 61272338, 61703443, 61673018 and Guangzhou Science and Technology Founding Committee under grant No.201804010255 and Guangdong Province Key Laboratory of Computational Science at Sun Yat-sen University (No. 2020B1212060032) and 2022 Scientific Research Platforms and Projects of Guangdong Provincial Education Department(No.2022KTSCX302) and Guangzhou Vocational College of Technology & Business(No.2022YB01).

Data Availability Statement Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Confict of interest The authors certify that there is no confict of interest in the subject matter discussed in this manuscript.

References

1. Zheng X, Guo Y, Huang H, Li Y, He R (2020) A survey of deep facial attribute analysis. Int J Comput Vision 128:2002–2034
2. Xie S, Hu H, Chen Y (2020) Facial expression recognition with two-branch disentangled generative adversarial network. IEEE Trans Circuits Syst Video Technol 31(6):2359–2371
3. Kim T, Chung C, Kim Y, Park S, Kim K, Choo J (2022) Style your hair: Latent optimization for pose-invariant hairstyle transfer via local-style-aware hair alignment. In: European conference on computer vision. Springer, pp 188–203
4. Wu Y, Wang R, Gong M, Cheng J, Yu Z, Tao D (2021) Adversarial uv-transformation texture estimation for 3d face aging. IEEE Trans Circuits Syst Video Technol 32(7):4338–4350
5. Choi Y, Uh Y, Yoo J, Ha J-W (2020) Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 8188–8197
6. Shen Y, Yang C, Tang X, Zhou B (2020) Interfacegan: Interpreting the disentangled face representation learned by gans. IEEE Trans Pattern Anal Mach Intell 44(4):2004–2018
7. Shi Y, Yang X, Wan Y, Shen X (2022) Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp 11254–11264
8. Beeler T, Bickel B, Noris G, Beardsley P, Marschner S, Sumner RW, Gross M (2012) Coupled 3d reconstruction of sparse facial hair and skin. ACM Trans Graph (ToG) 31(4):1–10
9. Yang F, Wang J, Shechtman E, Bourdev L, Metaxas D (2011) Expression flow for 3d-aware face component transfer. In: ACM SIGGRAPH. pp 1–10
10. Thies J, Zollhöfer M, Nießner M, Valgaerts L, Stamminger M, Theobalt C (2015) Real-time expression transfer for facial reenactment. ACM Trans Graph 34(6):1–14
11. Leyvand T, Cohen-Or D, Dror G, Lischinski D (2006) Digital face beautification. In: ACM Siggraph 2006 Sketches
12. Chen Y-C, Shen X, Jia J (2017) Makeup-go: Blind reversion of portrait edit. In: Proceedings of the IEEE international conference on computer vision. pp 4501–4509

13. Kemelmacher-Shlizerman I, Suwajanakorn S, Seitz SM (2014) Illumination-aware age progression. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 3334–3341
14. Zhang J, Zhou K, Luximon Y, Lee T-Y, Li P (2023) Meshwgan: Mesh-to-mesh wasserstein gan with multi-task gradient penalty for 3d facial geometric age transformation. IEEE Trans Vis Comput Graph
15. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems. pp 2672–2680
16. Lample G, Zeghidour N, Usunier N, Bordes A, Denoyer L, Ranzato M (2017) Fader networks: Manipulating images by sliding attributes. In: Advances in neural information processing systems. pp 5967–5976
17. Choi Y, Choi M, Kim M, Ha J-W, Kim S, Choo J (2018) Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 8789–8797
18. He Z, Zuo W, Kan M, Shan S, Chen X (2019) Attgan: Facial attribute editing by only changing what you want. IEEE Trans Image Process 28(11):5464–5478
19. Liu M, Ding Y, Xia M, Liu X, Ding E, Zuo W, Wen S (2019) Stgan: A unified selective transfer network for arbitrary image attribute editing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 3673–3682
20. Liu Y, Fan H, Ni F, Xiang J (2021) Clsgan: Selective attribute editing model based on classification adversarial network. Neural Netw 133:220–228
21. Zhou S, Xiao T, Yang Y, Feng D, He Q, He W (2017) Genegan: Learning object transfiguration and attribute subspace from unpaired data. arXiv:1705.04932
22. Xiao T, Hong J, Ma J (2018) Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In: Proceedings of the European Conference on Computer Vision (ECCV). pp 168–184
23. Yin W, Liu Z, Loy CC (2019) Instance-level facial attributes transfer with geometry-aware flow. In: Proceedings of the AAAI conference on artificial intelligence, vol. 33. pp 9111–9118
24. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training gans. In: Advances in neural information processing systems. pp 2234–2242
25. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: Proceedings of the 34th International conference on machine learning. pp 214–223
26. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC (2017) Improved training of wasserstein gans. In: Advances in neural information processing systems. pp 5767–5777
27. Guo J, Qian Z, Zhou Z, Liu Y (2019) Mulgan: Facial attribute editing by exemplar. arXiv:1912.12396
28. Zhang J, Huang Y, Li Y, Zhao W, Zhang L (2019) Multi-attribute transfer via disentangled representation. In: Proceedings of the AAAI conference on artificial intelligence, vol. 33. pp 9195–9202
29. Perarnau G, Van De Weijer J, Raducanu B, Álvarez JM (2016) Invertible conditional gans for image editing. arXiv:1611.06355
30. Yan X, Yang J, Sohn K, Lee H (2016) Attribute2image: Conditional image generation from visual attributes. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. Springer, pp 776–791
31. Mirza M, Osindero S (2014) Conditional generative adversarial nets. Comput Sci 2672–2680
32. Zhang G, Kan M, Shan S, Chen X (2018) Generative adversarial network with spatial attention for face attribute editing. In: Proceedings of the European Conference on Computer Vision (ECCV). pp 417–432
33. Lin J, Xia Y, Wang Y, Qin T, Chen Z (2019) Image-to-image translation with multi-path consistency regularization. In: Proceedings of the 28th International joint conference on artificial intelligence. pp 2980–2986
34. Zhu D, Liu S, Jiang W, Gao C, Wu T, Guo G (2019) Ugan: Untraceable gan for multi-domain face translation. arXiv:1907.11418
35. Li D, Zhang M, Zhang L, Chen W, Feng G (2021) A novel attribute-based generation architecture for facial image editing. Multimedia Tools Appl 80:4881–4902
36. Sohn K, Lee H, Yan X (2015) Learning structured output representation using deep conditional generative models. In: Advances in neural information processing systems. pp 3483–3491
37. Xiao T, Hong J, Ma J (2017) Dna-gan: Learning disentangled representations from multi-attribute images. arXiv:1711.05415
38. Lin C-H, Yumer E, Wang O, Shechtman E, Lucey S (2018) St-gan: Spatial transformer generative adversarial networks for image compositing. In: Proceedings of the ieee conference on computer vision and pattern recognition. pp 9455–9464
39. Li X, Zhang S, Hu J, Cao L, Hong X, Mao X, Huang F, Wu Y, Ji R (2021) Image-to-image translation via hierarchical style disentanglement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp 8639–8648
40. Dalva Y, Altındiş SF, Dundar A (2022) Vecgan: Image-to-image translation with interpretable latent directions. In: European Conference on Computer Vision. Springer, pp 153–169

41. Dalva Y, Pehlivan H, Hatipoglu OI, Moran C, Dundar A (2023) Image-to-image translation with disentangled latent vectors for face editing. IEEE Trans Pattern Anal Mach Intell 1–12. https://doi.org/10.1109/TPAMI.2023.3308102
42. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4401–4410
43. Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T (2020) Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 8110–8119
44. Roich D, Mokady R, Bermano AH, Cohen-Or D (2022) Pivotal tuning for latent-based editing of real images. ACM Trans Graph (TOG) 42(1):1–13
45. Hu X, Huang Q, Shi Z, Li S, Gao C, Sun L, Li Q (2022) Style transformer for image inversion and editing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 11337–11346
46. Alaluf Y, Tov O, Mokady R, Gal R, Bermano A (2022) Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 18511–18521
47. Pehlivan H, Dalva Y, Dundar A (2023) Styleres: Transforming the residuals for real image editing with stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 1828–1837
48. Wu P-W, Lin Y-J, Chang C-H, Chang EY, Liao S-W (2019) Relgan: Multi-domain image-to-image translation via relative attributes. In: Proceedings of the IEEE/CVF international conference on computer vision. pp 5914–5922
49. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp 2223–2232
50. Kim T, Cha M, Kim H, Lee JK, Kim J (2017) Learning to discover cross-domain relations with generative adversarial networks. In: Proceedings of the 34th International conference on machine learning. pp 1857–1865, JMLR. org
51. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR, pp 448–456
52. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv:1412.6980
53. Liu Z, Luo P, Wang X, Tang X (2016) Deep learning face attributes in the wild. In: IEEE international conference on computer vision. pp 3730–3738
54. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
55. Zhang R, Isola P, Efros AA, Shechtman E, Wang O (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 586–595
56. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
57. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp 1097–1105

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.