AttGAN: Facial Attribute Editing by Only Changing What You Want

Zhenliang He, Wangmeng Zuo, *Senior Member, IEEE*, Meina Kan, *Member, IEEE*, Shiguang Shan, *Senior Member, IEEE*, and Xilin Chen, *Fellow, IEEE*

***Abstract*— Facial attribute editing aims to manipulate single or multiple attributes on a given face image, i.e., to generate a new face image with desired attributes while preserving other details. Recently, the generative adversarial net (GAN) and encoder–decoder architecture are usually incorporated to handle this task with promising results. Based on the encoder–decoder architecture, facial attribute editing is achieved by decoding the latent representation of a given face conditioned on the desired attributes. Some existing methods attempt to establish an attribute-independent latent representation for further attribute editing. However, such attribute-independent constraint on the latent representation is excessive because it restricts the capacity of the latent representation and may result in information loss, leading to over-smooth or distorted generation. Instead of imposing constraints on the latent representation, in this work, we propose to apply an *attribute classification constraint* to the generated image to just guarantee the correct change of desired attributes, i.e., to change what you want. Meanwhile, the *reconstruction learning* is introduced to preserve attribute-excluding details, in other words, to only change what you want. Besides, the *adversarial learning* is employed for visually realistic editing. These three components cooperate with each other forming an effective framework for high quality facial attribute editing, referred as *AttGAN*. Furthermore, the proposed method is extended for *attribute style manipulation* in an unsupervised manner. Experiments on two wild datasets, CelebA and LFW, show that the proposed method outperforms the state-of-the-art on realistic attribute editing with other facial details well preserved.**

Manuscript received July 25, 2018; revised January 12, 2019; accepted May 2, 2019. Date of publication May 20, 2019; date of current version August 22, 2019. This work was supported in part by the National Key R&D Program of China under Contract 2017YFA0700800, and in part by the Natural Science Foundation of China under Contract 61671182, Contract 61772496, and Contract 61732004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sen-Ching Samson Cheung. *(Corresponding author: Shiguang Shan.)*

Z. He, M. Kan, and X. Chen are with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhenliang.he@vipl.ict.ac.cn; kanmeina@ict.ac.cn; xlchen@ict.ac.cn).

W. Zuo is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: cswmzuo@gmail.com).

S. Shan is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing 100190, China, also with the University of Chinese Academy of Sciences, Beijing 100049, China, also with the CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai 200031, China, and also with the Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: sgshan@ict.ac.cn).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The material includes a PDF containing additional results of facial attribute editing.

Digital Object Identifier 10.1109/TIP.2019.2916751

***Index Terms*— Facial attribute editing, attribute style manipulation, adversarial learning.**

I. Introduction

THIS work investigates the facial attribute editing task, which aims to edit a face image by manipulating single or multiple attributes of interest (e.g., hair color, expression, mustache and age). For conventional face recognition [1], [2] and facial attribute prediction [3], [4] tasks, significant advances have been made along with the development of deep convolutional neural networks (CNNs) and large scale labeled datasets. However, it is difficult or even impossible to collect labeled images of a same person with varying attributes, thus supervised learning is generally inapplicable for facial attribute editing. Therefore, researchers turn to generative models such as variational autoencoder (VAE) [5] and generative adversarial network (GAN) [6], and make considerable progress on facial attribute editing [7]–[16].

Some existing methods [9]–[12] use different editing models for different attributes, therefore one has to train numerous models for handling various attribute editing subtasks, which is difficult for real deployment. For this problem, the encoder-decoder architecture [7], [8], [13]–[15] seems to be an effective solution for using a *single* model for *multiple* attribute manipulation. Therefore, we also focus on the encoder-decoder architecture and develop an effective method for high quality facial attribute editing.

With the encoder-decoder architecture, facial attribute editing is achieved by decoding the latent representation from the encoder conditioned on the expected attributes. Based on such framework, the key issue of facial attribute editing is **how to model the relation between the attributes and the face latent representation**. For this issue, VAE/GAN [7] represents each attribute as a vector, which is defined as the difference between the mean latent representations of the faces with and without this attribute. Then, by adding a single or multiple attribute vectors to a face latent representation, the decoded face image from the modified representation is expected to own those attributes. However, such attribute vector contains highly correlated attributes, thus inevitably leading to unexpected changes of other attributes, e.g., adding blond hair always makes a male become a female because most blond hair objects are female in the training set. In IcGAN [8], the latent representation is sampled from a normal distribution independent of the attributes. In Fader

1057-7149 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

Networks [13], an adversarial process is introduced to force the latent representation of an autoencoder to be invariant to the attributes. However, the attributes portray the characteristics of a face image, which implies the relation between the attributes and the face latent representation is highly complex and closely dependent. Therefore, simply imposing the attribute-independent constraint on the latent representation not only restricts its capacity but also may result in information loss, which is harmful to the attribute editing. From a theoretical perspective, 1) the attribute-independent constraint amounts to *minimizing* the mutual information between the attributes and the latent representation, 2) in contrast, the autoencoder objective amounts to *maximizing* the mutual information between the input image (including its attributes) and the latent representation [17]. Therefore, the attribute-independent constraint and the autoencoder objective are conflictive resulting a compromise performance.

With the above limitation of existing methods in mind, we argue that the invariance of the latent representation to the attributes is excessive, and what we need is just the correct editing of attributes no matter whether the latent representation is invariant to the attributes or not. To this end, instead of imposing the attribute-independence constraint on the latent representation [8], [13], we apply an attribute classification constraint to the generated image, just requiring the correct attribute manipulations, i.e., to "change what you want". Therefore in comparison with IcGAN [8] and Fader Networks [13], the latent representation in our method is constraint free, which guarantees its capacity and flexibility for further attribute editing. Besides, we introduce the reconstruction learning for the preservation of the attribute-excluding details[1], i.e., we aim to "only change" the expected attributes while keeping the other details unchanged. Moreover, the adversarial learning is employed for visually realistic editing. In our design, the classification constraint and the reconstruction leaning are respectively applied on two separate branches. Therefore, the classification constraint (for correct editing) and the reconstruction leaning (for detail preservation) are not necessarily conflictive unlike the methods with attribute-independent constraint, while working collaboratively with each other.

Our method, referred as AttGAN, can generate visually more pleasing results with fine facial details (see Fig. 1) in comparison with the state-of-the-arts. Moreover, our AttGAN is naturally extended for attribute style manipulation. To sum up, the contribution of this work lies in three folds:

- Properly considering the relation between the attributes and the face latent representation under the principle of just satisfying the correct editing objective. Our AttGAN removes the strict attribute-independent constraint from the latent representation, and just applies the attribute classification constraint to the generated image to guarantee the correct change of the attributes.
- Incorporating the attribute classification constraint, the reconstruction learning and the adversarial learning into a unified framework for high quality facial attribute editing, i.e., the attributes are correctly edited, the attribute-excluding details are well preserved and the whole image is visually realistic.
- Promising results of multiple facial attribute editing using a single model. AttGAN outperforms the state-of-the-arts with better perceptual quality for facial attribute editing. Moreover, our method is naturally extended for attribute style manipulation.

[1]attribute-excluding details mean the other details of a face image except for the expected attributes, such as face identity, illumination and background.

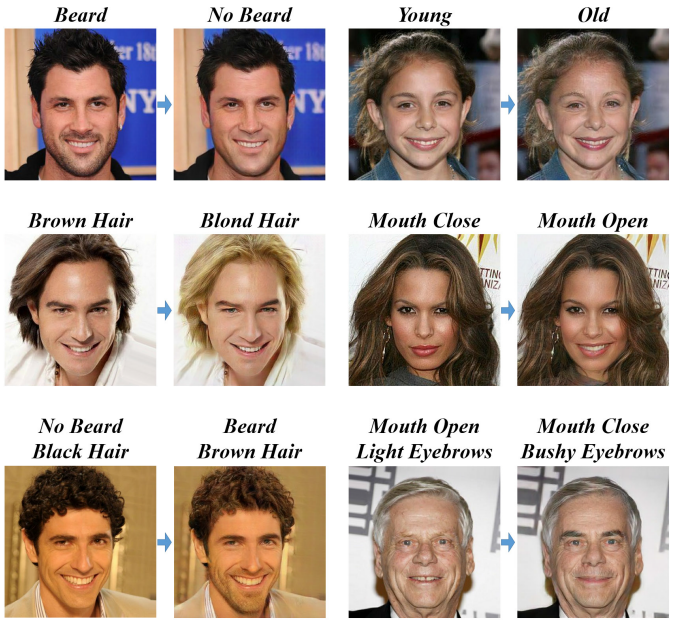


Fig. 1. Facial attribute editing results from our AttGAN. Best viewed in color and higher resolution (by zooming in).

II. Related Work

A. Facial Attribute Editing

There are two types of methods for facial attribute editing, the optimization based ones [18], [19] and the learning based ones [7]–[14], [16]. Optimization based methods include CNAI [18] and DFI [19]. To change a given face to a new face with the expected attributes, CNAI [18] defines an attribute loss as the CNN feature difference between the given face and a set of faces with the expected attributes, and then minimizes this loss with respect to the given face. Based on the assumption that CNN linearizes the manifold of the natural images into an Euclidean feature subspace [20], DFI [19] first linearly moves the deep feature of the input face along the direction vector between the faces with and without the expected attributes. Then the facial attribute editing is achieved by optimizing the input face to match its deep feature with the moved feature. Optimization based methods need to conduct several or even many optimization iterations for each testing image, which are usually time-consuming and unfriendly for real world applications.

More popular methods are learning based ones. Li et al. [9] present to train a deep identity-aware attribute transfer model to add/remove an attribute to/from a face image by employing

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

an adversarial attribute loss and a deep identity feature loss. Shen and Liu [10] adopt the dual residual learning strategy to simultaneously train two networks for respectively adding and removing a specific attribute. Specializing in makeup attribute, PairedCycleGAN [21] employs a pair of asymmetric networks for applying and removing makeup. GeneGAN [12] swaps a specific attribute between two given images by recombining the information of their latent representation. These methods [9]–[12], however, train different models for different attributes (or attribute combinations), leading to large number of models which are also unfriendly for real world applications.

Moreover, several learning based methods for multiple facial attribute editing with one model are proposed. In VAE/GAN [7], GAN [6] and VAE [5] are combined to learn a latent representation and a decoder. Then the attribute editing is achieved by modifying the latent representation to own the information of expected attributes and then decoding it. Specializing in hair style editing, H-GAN [22] is another hybrid model of VAE and GAN with a latent recognizer and a hair style recognizer. The hair style editing is achieved by, first adding the mean hair style vector to the face latent representation, and then decoding the modified latent representation conditioned on the given hair style. IcGAN [8] separately trains a cGAN [23] and an encoder, requiring that the latent representation is sampled from a uniform distribution and therefore independent of the attributes. Then the attribute editing is performed by first encoding an image into the latent representation and then decoding the representation conditioned on the given attributes. Fader Networks [13] employs an adversarial process on the latent representation of an autoencoder to learn the attribute-invariant representation. Then, the decoder takes such representation and arbitrary attribute vector as input to generate the edited result. However, the attribute-independent constraint on the latent representation in IcGAN and Fader Networks is excessive, because it harms the representation ability and may result in information loss, leading to unexpected distortion on the generated images (e.g., over smoothing). Kim et al. [14] define different blocks of the latent code as the representations of different attributes, and swap several latent code blocks between two given images to achieve multiple attribute swapping. DNA-GAN [15] also swap attribute relevant latent blocks between a given pair of images to make "crossbreed" images. Both Kim et al. [14] and DNA-GAN [15] can be viewed as extensions of GeneGAN [12] for multiple attributes. StarGAN [16] trains a conditional attribute transfer network via attribute classification loss and cycle consistency loss. StarGAN and our AttGAN are concurrently and independently proposed[2] and share some similar objective functions. Main differences between StarGAN and AttGAN are in two folds: 1) StarGAN uses cycle consistency loss while our AttGAN uses reconstruction learning for direct constraint without any cyclic process, 2) StarGAN trains a conditional attribute transfer network and does not involve any latent representation while AttGAN uses an encoder-decoder architecture and models the relation between the latent representation and the attributes.

Image translation task is closely related to facial attribute editing, which focuses on transforming images among different domains, e.g., converting a real scene photo to Monet style painting [24]. Therefore, the main difference between image translation task and facial attribute editing task is that, image translation specializes in domain level manipulation while facial attribute editing specializes in attribute level manipulation within the face domain. For image translation, CycleGAN [24] trains two bidirectional transfer models between two image domains by employing the cycle consistency loss and two domain specific adversarial learning processes. Lu et al. [25] employ a conditional CycleGAN to generate a high resolution face image for the low resolution input that satisfies the given attributes. UNIT [11] learns to encode the images of two different domains into a common latent space, and then decode the latent representation to the expected domain via the domain specific decoder. Although the image translation methods do not specialize in facial attribute editing, one can adapt some of these methods for our task, e.g., CycleGAN can be used for facial attribute editing by regarding face images with and without the expected attributes as two different domains. Another closely related task is attribute-to-image, i.e., to generate an image with specific attributes from the random noise. Different from facial attribute editing, attribute-to-image task focuses on generating images from random noise rather than given images. For this task, Attribute2Image [26] employs a disentangling conditional VAE with a layered representation.

Our AttGAN is a learning based method for single or multiple facial attribute editing, which is mostly motivated by the encoder-decoder based methods VAE/GAN [7], IcGAN [8] and Fader Networks [13]. We mainly focus on the disadvantages of these three methods on modeling the relation between the latent representation and the attributes, and propose a novel method to solve such problem.

B. Generative Adversarial Networks

Denote by $p_{data}(\mathbf{x})$ the distribution of the real image $\mathbf{x}$, and $p_{\mathbf{z}}(\mathbf{z})$ the distribution of the input. Generative adversarial net (GAN) [6] is a special generative model to learn a generator $G(\mathbf{z})$ to capture the distribution p_{data} via an adversarial process. Specifically, a discriminator D is introduced to distinguish the generated images from the real ones, while the generator $G(\mathbf{z})$ is updated to confuse the discriminator. The adversarial process is formulated as a minimax game as

$$\min_G \max_D \ \mathbb{E}_{\mathbf{x}\sim p_{data}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))]. \tag{1}$$

Theoretically, when the adversarial process reaches the Nash equilibrium, the minimax game attains its global optimum $p_{G(\mathbf{z})} = p_{data}$ [6].

GAN is notorious for its unstable training and mode collapse. DCGAN [27] uses CNN and batch normalization [28] for stable training. Subsequently, to avoid mode collapse and further enhance the training stability, WGAN [29] minimizes

[2]StarGAN first appears on 2017.11.24 - http://arxiv.org/abs/1711.09020, and our AttGAN first appears on 2017.11.29 - http://arxiv.org/abs/1711.10678.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

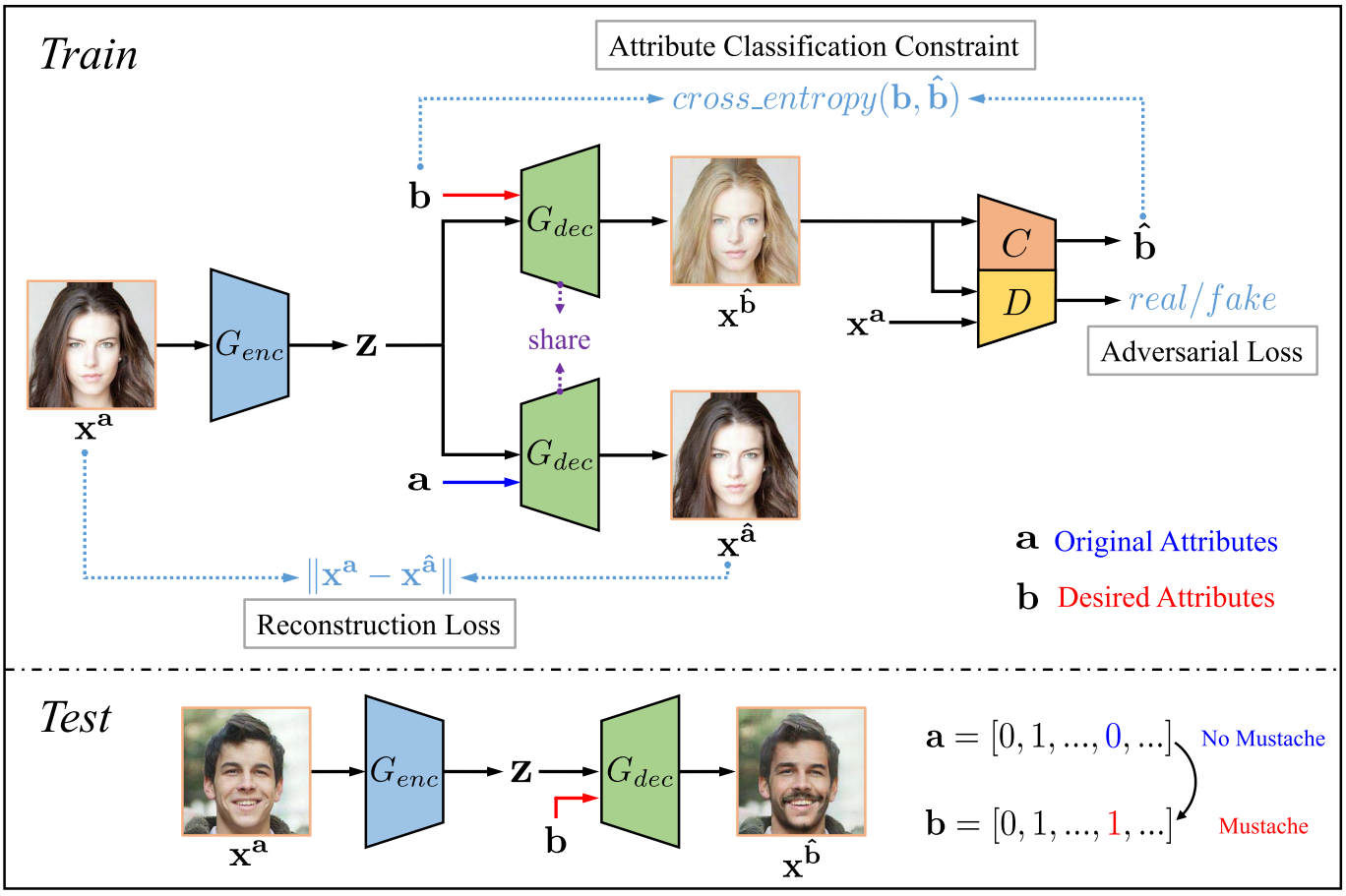


Fig. 2. Overview of our AttGAN, which contains three main components at training: the attribute classification constraint, the reconstruction learning and the adversarial learning. The attribute classification constraint guarantees the correct attribute manipulation on the generated image. The reconstruction learning aims at preserving the attribute-excluding details. The adversarial learning is employed for visually realistic generation.

the Wasserstein-1 distance between the generated distribution and the real distribution as

$$\min_{G} \max_{\|D\|_L \leq 1} \mathbb{E}_{\mathbf{x} \sim p_{data}}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}}[D(G(\mathbf{z}))], \quad (2)$$

where D is constrained to be the 1-Lipschitz function implemented by weight clipping. Furthermore, WGAN-GP [30] improves WGAN on the implementation of Lipschitz constraint by imposing a gradient penalty on the discriminator instead of weight clipping. In this work, we adopt WGAN-GP for the adversarial learning.

Several works have been developed for the conditional generation with given attributes or class labels [23], [31]–[33]. Employing an auxiliary classifier or regressor, both AC-GAN [32] and InfoGAN [33] learn the conditional generation by mapping the generated images back to the conditional signals. Since AC-GAN generates images from random noise, it cannot be directly used for the attribute editing task and it is not trivial to design an AC-GAN variant for a satisfactory attribute editing performance. In this work, we smartly incorporate the AC-GAN spirit to form the attribute classification constraint, which corporates with the other components for effective facial attribute editing.

III. Attribute GAN (AttGAN)

This section introduces the AttGAN approach for the editing of binary facial attributes, i.e., each attribute is represented by 1/0 for with/without it and all attributes are represented by a 1/0 sequence. As shown in Fig. 2, our AttGAN is comprised of two basic subnetworks, i.e., an encoder G_{enc} and a decoder G_{dec}, together with an attribute classifier C and a discriminator D. In the following, we describe the design principles of AttGAN and introduce the objectives for training these modules. Then we present an extension of AttGAN for attribute style manipulation, e.g., to edit the "Eyeglasses" attribute to sunglasses, black rim glasses or thin rim glasses.

A. Testing Formulation

Given a face image $\mathbf{x}^{\mathbf{a}}$ with n binary attributes $\mathbf{a} = [a_1, \ldots, a_n]$, the encoder G_{enc} is used to encode $\mathbf{x}^{\mathbf{a}}$ into the latent representation, denoted as

$$\mathbf{z} = G_{enc}(\mathbf{x}^{\mathbf{a}}). \quad (3)$$

Then the process of editing the attributes of $\mathbf{x}^{\mathbf{a}}$ to another attributes $\mathbf{b} = [b_1, \ldots, b_n]$ is achieved by decoding $\mathbf{z}$ conditioned on $\mathbf{b}$, i.e.,

$$\mathbf{x}^{\hat{\mathbf{b}}} = G_{dec}(\mathbf{z}, \mathbf{b}), \quad (4)$$

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

where $\mathbf{x}^{\hat{\mathbf{b}}}$ is the edited image expected to own the attribute $\mathbf{b}$. Thus the whole editing process is formulated as

$$\mathbf{x}^{\hat{\mathbf{b}}} = G_{dec}(G_{enc}(\mathbf{x}^{\mathbf{a}}), \mathbf{b}). \tag{5}$$

B. Training Formulation

It can be seen from Eq. (5) that the attribute editing problem can be formally defined as the learning of the encoder G_{enc} and decoder G_{dec}. This learning problem is unsupervised, because the ground truth of the editing, i.e. $\mathbf{x}^{\mathbf{b}}$, is unavailable.

On the one hand, the editing on the given face image $\mathbf{x}^{\mathbf{a}}$ is expected to produce a realistic image with attributes $\mathbf{b}$. For this purpose, an **attribute classifier** is used to constrain the generated image $\mathbf{x}^{\hat{\mathbf{b}}}$ to correctly own the desired attributes, i.e., the attribute prediction of $\mathbf{x}^{\hat{\mathbf{b}}}$ should be $\mathbf{b}$. Meanwhile, the **adversarial learning** is employed on $\mathbf{x}^{\hat{\mathbf{b}}}$ to ensure its visual reality.

On the other hand, an eligible attribute editing should only change those desired attributes, while keeping the other details unchanged. To this end, the **reconstruction learning** is introduced to 1) make the latent representation $\mathbf{z}$ conserve enough information for the later recovery of the attribute-excluding details, 2) enable the decoder G_{dec} to restore the attribute-excluding details from $\mathbf{z}$. Specifically, for the given $\mathbf{x}^{\mathbf{a}}$, the generated image conditioned on its own attributes $\mathbf{a}$, i.e.,

$$\mathbf{x}^{\hat{\mathbf{a}}} = G_{dec}(\mathbf{z}, \mathbf{a}) \tag{6}$$

should approximate $\mathbf{x}^{\mathbf{a}}$ itself, i.e., $\mathbf{x}^{\hat{\mathbf{a}}} \rightarrow \mathbf{x}^{\mathbf{a}}$.

In summary, *the relation between the attributes* $\mathbf{a}$/$\mathbf{b}$ *and the latent representation* $\mathbf{z}$ *is implicitly modeled* in two aspects: 1) the interaction between $\mathbf{z}$ and $\mathbf{b}$ in the decoder should produce an realistic image $\mathbf{x}^{\hat{\mathbf{b}}}$ with correct attributes, and 2) the interaction between $\mathbf{z}$ and $\mathbf{a}$ in the decoder should produce an image $\mathbf{x}^{\hat{\mathbf{a}}}$ approximating the input $\mathbf{x}^{\mathbf{a}}$ itself.

1) Attribute Classification Constraint: As mentioned above, it is required that the generated image $\mathbf{x}^{\hat{\mathbf{b}}}$ should correctly own the new attributes $\mathbf{b}$. Therefore, we employ an attribute classifier C to constrain the generated image $\mathbf{x}^{\hat{\mathbf{b}}}$ to own the desired attributes, i.e., $C(\mathbf{x}^{\hat{\mathbf{b}}}) \rightarrow \mathbf{b}$, formulated as follows,

$$\min_{G_{enc}, G_{dec}} \mathcal{L}_{cls_g} = \mathbb{E}_{\mathbf{x}^{\mathbf{a}} \sim p_{data}, \mathbf{b} \sim p_{attr}}[\ell_g(\mathbf{x}^{\mathbf{a}}, \mathbf{b})], \tag{7}$$

$$\ell_g(\mathbf{x}^{\mathbf{a}}, \mathbf{b}) = \sum_{i=1}^{n} -b_i \log C_i(\mathbf{x}^{\hat{\mathbf{b}}}) - (1-b_i)\log(1-C_i(\mathbf{x}^{\hat{\mathbf{b}}})), \tag{8}$$

where p_{data} and p_{attr} indicate the distribution of real images and the distribution of attributes, $C_i(\mathbf{x}^{\hat{\mathbf{b}}})$ indicates the prediction of the i^{th} attribute, and $\ell_g(\mathbf{x}^{\mathbf{a}}, \mathbf{b})$ is the summation of binary cross entropy losses of all attributes.

The attribute classifier C is trained on the input images with their original attributes, by the following objective,

$$\min_{C} \mathcal{L}_{cls_c} = \mathbb{E}_{\mathbf{x}^{\mathbf{a}} \sim p_{data}}[\ell_r(\mathbf{x}^{\mathbf{a}}, \mathbf{a})], \tag{9}$$

$$\ell_r(\mathbf{x}^{\mathbf{a}}, \mathbf{a}) = \sum_{i=1}^{n} -a_i \log C_i(\mathbf{x}^{\mathbf{a}}) - (1-a_i)\log(1-C_i(\mathbf{x}^{\mathbf{a}})). \tag{10}$$

2) Reconstruction Loss: Furthermore, the reconstruction learning aims for preservation of attribute-excluding details. To this end, the decoder should learn to reconstruct the input image $\mathbf{x}^{\mathbf{a}}$ by decoding the latent representation $\mathbf{z}$ conditioned on the original attributes $\mathbf{a}$. The learning objective is formulated as

$$\min_{G_{enc}, G_{dec}} \mathcal{L}_{rec} = \mathbb{E}_{\mathbf{x}^{\mathbf{a}} \sim p_{data}}[\|\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\hat{\mathbf{a}}}\|_1], \tag{11}$$

where we use the ℓ_1 loss rather than ℓ_2 loss to suppress the blurriness.

3) Adversarial Loss: The adversarial learning between the generator (including the encoder and decoder) and discriminator is introduced to make the generated image $\mathbf{x}^{\hat{\mathbf{b}}}$ visually realistic. Following WGAN [29], the adversarial losses for the the discriminator and generator are formulated as below,

$$\min_{\|D\|_L \leq 1} \mathcal{L}_{adv_d} = -\mathbb{E}_{\mathbf{x}^{\mathbf{a}} \sim p_{data}} D(\mathbf{x}^{\mathbf{a}}) + \mathbb{E}_{\mathbf{x}^{\mathbf{a}} \sim p_{data}, \mathbf{b} \sim p_{attr}} D(\mathbf{x}^{\hat{\mathbf{b}}}), \tag{12}$$

$$\min_{G_{enc}, G_{dec}} \mathcal{L}_{adv_g} = -\mathbb{E}_{\mathbf{x}^{\mathbf{a}} \sim p_{data}, \mathbf{b} \sim p_{attr}}[D(\mathbf{x}^{\hat{\mathbf{b}}})], \tag{13}$$

where D is the discriminator described in Eq. (2). The adversarial losses are optimized via WGAN-GP [30].

4) Overall Objective: By combining the attribute classification constraint, the reconstruction loss and the adversarial loss, an unified attribute GAN (AttGAN) is obtained, which can edit the desired attributes with the attribute-excluding details well preserved. Overall, the objective for the encoder and decoder is formulated as below,

$$\min_{G_{enc}, G_{dec}} \mathcal{L}_{enc,dec} = \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{cls_g} + \mathcal{L}_{adv_g}, \tag{14}$$

and the objective for the discriminator and the attribute classifier is formulated as below,

$$\min_{D,C} \mathcal{L}_{dis,cls} = \lambda_3 \mathcal{L}_{cls_c} + \mathcal{L}_{adv_d}, \tag{15}$$

where the discriminator and the attribute classifier share most layers, λ_1, λ_2 and λ_3 are the hyperparameters for balancing the losses.

C. Why Are Attribute-Excluding Details Preserved?

The above AttGAN design can be viewed as a multi-task leaning of attribute editing task with classification loss and face reconstruction task with reconstruction loss, which share the entire encoder-decoder network. However, AttGAN only conducts the reconstruction learning on the generated image conditioned on the original attributes $\mathbf{a}$, why the preservation ability of attribute-excluding details can be generalized to the generation conditioned on another attributes $\mathbf{b}$? We suggest the reason is that, AttGAN transfers the detail preservation ability from the face reconstruction task to the attribute editing task. Since these two tasks share the same input domain and output domain, they are very similar tasks with tiny *transferability*

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

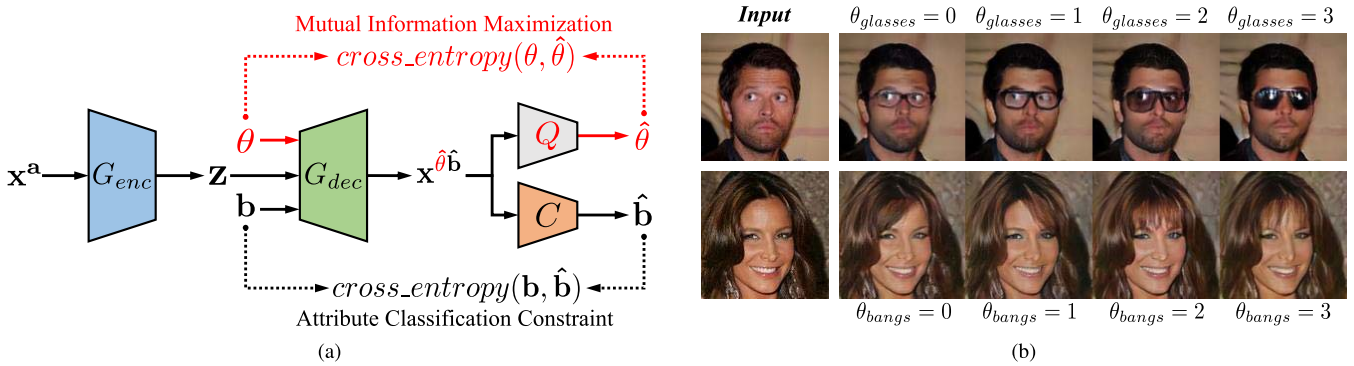


Fig. 3. Illustration of AttGAN extension for attribute style manipulation. (a) shows the extended framework based on the original AttGAN. θ denotes the style controllers and Q denotes the style predictor. (b) shows the visual effect of changing attribute style by varying θ.

gap [36] between them. Therefore, the detail preservation ability learned from the face reconstruction task can be easily transfered to the attribute editing task. Besides, these two tasks are learned simultaneously, therefore such transfer is dynamic and the attribute editing learning does not flush the ability of facial detail reconstruction.

D. Extension for Attribute Style Manipulation

In Sec. III-A and III-B, the attributes are binary represented, i.e., "with" or "without", which is stiff for real world applications. However, for example, in most cases what one is interested in is adding a certain style of eyeglasses such as sunglasses or thin rim glasses, rather than just with/without eyeglasses. This problem is more difficult because the labeled data with attribute style is unavailable. Therefore, we attempt to find an unsupervised way to make the generative model $G = (G_{enc}, G_{dec})$ be controlled by a style controller θ in an interpretable aspect, e.g., θ controls the "Eyeglasses" to be thin rim, black rim or sunglasses. The solution used in this work is to maximize the mutual information between θ and the generated image $\mathbf{x}^{\theta} \sim G_{dec}(G_{enc}(\mathbf{x}^{\mathbf{a}}), \theta * \mathbf{b})$ in order to make them highly correlated [31], [33], i.e.,

$$\max_{G=(G_{enc},G_{dec})} I(\theta; \mathbf{x}^{\theta}). \tag{16}$$

According to [33], the mutual information can be obtained by

$$I(\theta; \mathbf{x}^{\theta}) = \max_{Q} \mathbb{E}_{\theta\sim p(\theta), \mathbf{x}^{\theta}\sim p_G(\mathbf{x}^{\theta}|\theta)}[\log Q(\theta|\mathbf{x}^{\theta})] + H(\theta), \tag{17}$$

where $H(\theta)$ is a constant and $Q(\theta|\mathbf{x}^{\theta})$ is an auxiliary posterior distribution which can be viewed as a style predictor here. We substitute Eq. (17) into Eq. (16) and abandon the constant $H(\theta)$, and get the objective as below,

$$\max_{G} \max_{Q} \mathbb{E}_{\theta\sim p(\theta), \mathbf{x}^{\theta}\sim p_G(\mathbf{x}^{\theta}|\theta)}[\log Q(\theta|\mathbf{x}^{\theta})]. \tag{18}$$

where the inner '$\max_{Q}$' estimates the mutual information, while the outer '$\max_{G}$' maximizes the mutual information. In our setting, θ indicates the category of the style, therefore $Q(\theta|\mathbf{x}^{\theta})$ is a categorical distribution implemented by the softmax output of a neural network, i.e.,

$$Q(\theta = i|\mathbf{x}^{\theta}) = \frac{\exp(w_i^T f(\mathbf{x}^{\theta}) + b_i)}{\sum_j \exp(w_j^T f(\mathbf{x}^{\theta}) + b_j)}. \tag{19}$$

Thus, substituting Eq. (19) into Eq. (18) and rewriting the optimization problem in the form with loss function, we finally get the objective as below,

$$\min_{G} \min_{w,b,f} Loss = \mathbb{E}_{\theta\sim p(\theta), \mathbf{x}^{\theta}\sim p_G(\mathbf{x}^{\theta}|\theta)}[-\sum_i \delta(\theta = i) \cdot \log \frac{\exp(w_i^T f(\mathbf{x}^{\theta}) + b_i)}{\sum_j \exp(w_j^T f(\mathbf{x}^{\theta}) + b_j)}], \tag{20}$$

where these two 'min's are done iteratively. Fig. 3 shows our AttGAN extension with style controller θ and a style predictor Q.

IV. Implementation Details

Our AttGAN is implemented by the machine learning system Tensorflow [37] and the code is publicly available at https://github.com/LynnHo/AttGAN-Tensorflow.

A. Network Architecture

Table I and Table II shows the detailed network architectures of our AttGAN. The discriminator D is a stack of convolutional layers followed by fully connected layers, and the classifier C has a similar architecture and shares all convolutional layers with D. The encoder G_{enc} is a stack of convolutional layers and the decoder G_{dec} is a stack of transposed convolutional layers. We also employ the U-Net [38] like symmetric skip connections between the encoder and decoder, which have been shown to produce high quality results on the image translation task [39]. Architectures for 64×64 images are used in the comparisons with VAE/GAN [7], IcGAN [8] and StarGAN [16], and architectures for 128×128 images are used in the comparisons with Fader Networks [13], Shen et al. [10] and CycleGAN [24]. 384×384 images are shown in other experiments for better visual effect.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

TABLE I
NETWORK ARCHITECTURES OF ATTGAN FOR $128+^2$ IMAGES

Encoder (G_{enc})	**Decoder** (G_{dec})	**Discriminator** (D)	**Classifier** (C)
Conv(64,4,2), BN, Leaky ReLU	DeConv(1024,4,2), BN, ReLU	Conv(64,4,2), LN/IN, Leaky ReLU	
Conv(128,4,2), BN, Leaky ReLU	DeConv(512,4,2), BN, ReLU	Conv(128,4,2), LN/IN, Leaky ReLU	
Conv(256,4,2), BN, Leaky ReLU	DeConv(256,4,2), BN, ReLU	Conv(256,4,2), LN/IN, Leaky ReLU	
Conv(512,4,2), BN, Leaky ReLU	DeConv(128,4,2), BN, ReLU	Conv(512,4,2), LN/IN, Leaky ReLU	
Conv(1024,4,2), BN, Leaky ReLU	DeConv(3,4,2), Tanh	Conv(1024,4,2), LN/IN, Leaky ReLU	
		FC(1024), LN, Leaky ReLU	FC(1024), LN, Leaky ReLU
		FC(1)	FC(13), Sigmoid

TABLE II
NETWORK ARCHITECTURES OF ATTGAN FOR 64^2 IMAGES

Encoder (G_{enc})	**Decoder** (G_{dec})	**Discriminator** (D)	**Classifier** (C)
Conv(64,5,2), BN, Leaky ReLU	DeConv(512,5,2), BN, ReLU	Conv(64,3,1), LN/IN, Leaky ReLU	
Conv(128,5,2), BN, Leaky ReLU	DeConv(256,5,2), BN, ReLU	Conv(64,5,2), LN/IN, Leaky ReLU	
Conv(256,5,2), BN, Leaky ReLU	DeConv(128,5,2), BN, ReLU	Conv(128,5,2), LN/IN, Leaky ReLU	
Conv(512,5,2), BN, Leaky ReLU	DeConv(64,5,2), BN, ReLU	Conv(256,5,2), LN/IN, Leaky ReLU	
	DeConv(3,5,1), Tanh	Conv(512,5,2), LN/IN, Leaky ReLU	
		Conv(512,3,1), LN/IN, Leaky ReLU	
		FC(1024), LN, Leaky ReLU	FC(1024), LN, Leaky ReLU
		FC(1)	FC(13), Sigmoid

* Conv(d,k,s) and DeConv(d,k,s) denote the convolutional layer and transposed convolutional layer with d as dimension, k as kernel size and s as stride. BN is batch normalization [28], LN is layer normalization [34] and IN is instance normalization [35].

B. Training Details

The model is trained by Adam optimizer [41] ($\beta_1 = 0.5$, $\beta_2 = 0.999$) with the batch size of 32 and the learning rate of 0.0002. The coefficients for the losses in Eq. (14) and Eq. (15) are set as: $\lambda_1 = 100$, $\lambda_2 = 10$, and $\lambda_3 = 1$, which aims to make the loss values be in the same order of magnitude.

V. EXPERIMENTS

Dataset: We evaluate the proposed AttGAN on CelebA [3] and LFW [40] dataset. CelebA contains two hundred thousand images, each of which has annotation of 40 binary attributes (with/without). Thirteen attributes with strong visual impact are chosen in all our experiments, including "Bald", "Bangs", "Black Hair", "Blond Hair", "Brown Hair", "Bushy Eyebrows", "Eyeglasses", "Gender", "Mouth Open", "Mustache", "No Beard", "Pale Skin" and "Age", which cover most attributes used in the existing works. Officially, CelebA is separated into training set, validation set and testing set. We use the training set and validation set together to train our model while using the testing set for evaluation. Besides, for more comprehensive comparisons to illustrate the robustness of our AttGAN, the entire LFW dataset with 13,233 images is used as another testing set.

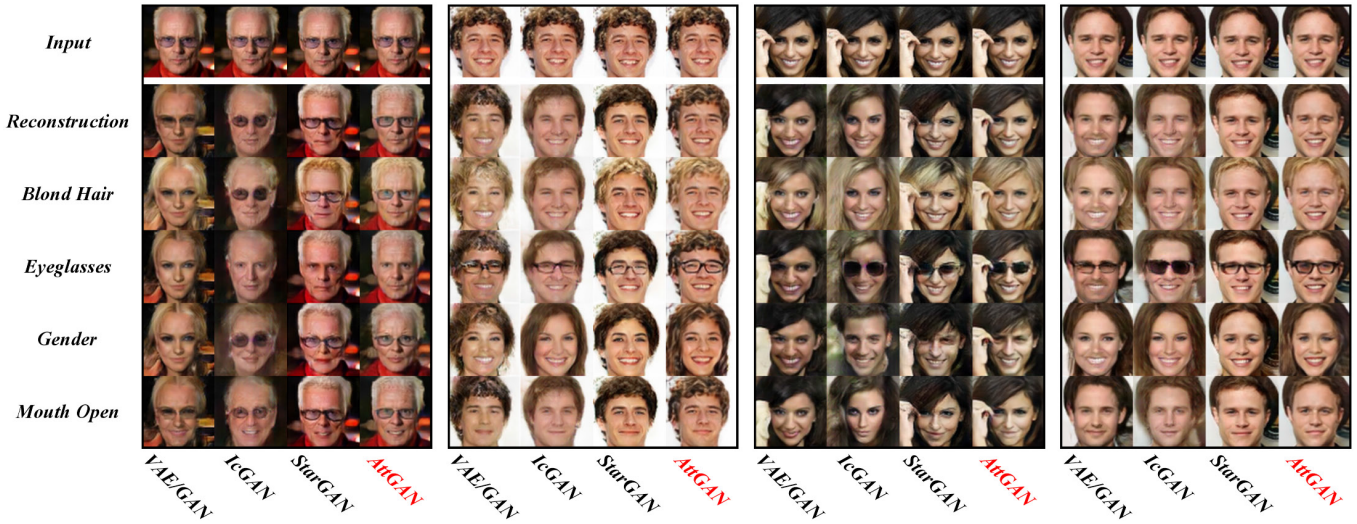
Methods: Under the same experimental settings, we compare our AttGAN with VAE/GAN [7], IcGAN [8] and StarGAN [16], all of which including our AttGAN are trained to *handle all thirteen attributes with a single model*. Besides, we compare our AttGAN with Fader Networks [13], Shen et al. [10] and CycleGAN [24]. Shen et al. and CycleGAN can handle only one attribute with one model. Although Fader Networks is capable for multiple attribute editing with one model, in practice, multiple attribute setting makes the results blurry. Therefore, for these three baselines, *each attribute has its own specific model*. VAE/GAN[3], IcGAN[4], StarGAN[5] and Fader Networks[6] are trained by their official code, while Shen et al. and CycleGAN are implemented by ourself.

A. Visual Analysis

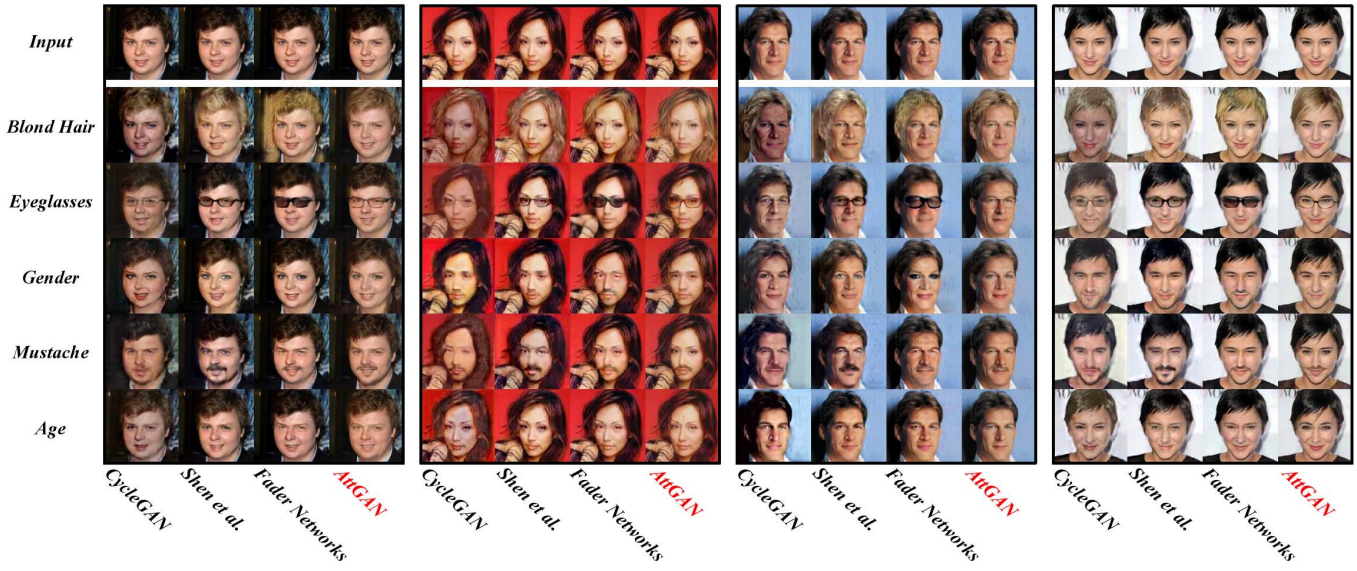
1) Single Facial Attribute Editing: Firstly, on CelebA [3] testing set, we compare the proposed AttGAN with VAE/GAN [7], IcGAN [8] and StarGAN [16] in terms of single facial attribute editing, shown in Fig. 4a. As can be seen, in some cases VAE/GAN produces unexpected changes of other attributes, for example, all three male inputs become female in VAE/GAN when editing the blond hair attribute. This phenomenon happens because the attribute vectors used for editing in VAE/GAN contains highly correlated attributes such as blond hair and female. Therefore, some other unexpected but highly correlated attributes are also involved when using such attribute vectors for editing. IcGAN performs better on accurately editing attributes, however, it seriously changes other attribute-excluding details especially the face identity.

[3] VAE/GAN: https://github.com/andersbll/autoencoding_beyond_pixels
[4] IcGAN: https://github.com/Guim3/IcGAN
[5] StarGAN: https://github.com/yunjey/StarGAN
[6] Fader Networks: https://github.com/facebookresearch/Fader Networks

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.



(a)



(b)

Fig. 4. Results on CelebA [3] of single facial attribute editing. For each specified attribute, the facial attribute editing here is to **invert** it, e.g., to edit female to male, male to female, mouth open to mouth close, and mouth close to mouth open etc. (a) Comparisons with VAE/GAN [7], IcGAN [8] and StarGAN [16] on editing (inverting) specified attributes. These methods including our AttGAN employ single model for all attributes. (b) Comparisons with CycleGAN [24], Shen et al. [10] and Fader Networks [13] on editing (inverting) specified attributes. These three methods employ different models for different attributes. Best viewed in color and higher resolution (by zooming in).

This is mainly because IcGAN imposes attribute-independent constraint and normal distribution constraint on the latent representation, which harms its capacity and results in loss of attribute-excluding information. Compared to VAE/GAN and IcGAN, our AttGAN accurately edits both local attributes (bangs, eyeglasses and mouth open) and global attributes (gender), credited to the attribute classification constraint which guarantees the correct change of the attributes. Although StarGAN also accurately edit the attributes, but the results contain undesired changes, e.g., the skin colors of the first two objects change. In contrast, AttGAN well preserves the attribute-excluding details such as face identity, illumination, and background, credited to that 1) the latent representation is constraint free, which guarantees its capacity for conserving the attribute-excluding information, 2) the reconstruction learning explicitly enable the encoder-decoder to preserve the attribute-excluding details on the generated images.

Comparisons on CelebA [3] with CycleGAN [24], Shen et al. [10] and Fader Networks [13] are shown in Fig. 4b. The results of Fader Networks especially on adding eyeglasses are blurry, which is very likely caused by the strict attribute-independent constraint on the latent representation. The results of Shen et al. and CycleGAN contain noise and artifacts. Another observation is that, adding "Mustache" makes the female (the second and fourth input in Fig. 4b) become male in Shen et al. and CycleGAN. In the

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

Fig. 5. Comparisons on LFW [40] of single facial attribute editing with CycleGAN [24], Shen et al. [10], Fader Networks [13] and StarGAN [16]. Best viewed in color and higher resolution (by zooming in).

opposite, our AttGAN naturally add the mustache keeping the female's characteristic well although the model rarely (or never) sees the female with mustache in the training set, which reflects the AttGAN's superior ability to disentangle attributes (such as male and mustache) and preserve details.

Comparisons on LFW [40] are shown in Fig. 5. As shown, the competitors produce artifacts to some extent, e.g., females become males when editing the 'Mustache' attribute in CycleGAN and Shen et al., and Fader Networks produces blurriness on 'Blond Hair' and 'Eyeglasses' attribute. As for our AttGAN, more accurate attribute editing result with very few artifacts are achieved benefited from the elaborate design of our mechanism for facial attribute editing.

2) Multiple Facial Attribute Editing: All of VAE/GAN [7], IcGAN [8], StarGAN [16] and our AttGAN can simultaneously edit multiple attributes with a single model, and thus we investigate these three methods in terms of multiple facial attribute editing for more comprehensive comparison. Fig. 6 shows the results of simultaneously editing two or three attributes.

Similar to single attribute editing, some generated images from VAE/GAN contain undesired changes of other attributes since VAE/GAN cannot decorrelate highly correlated attributes. As for IcGAN, distortion of face details and over smoothing become even more severe, because its constrained latent representation lead to worse performance in the more complex multiple attribute editing task. By contrast, our method still performs well under complex combinations of attributes, benefited from the appropriate modeling of the relation between the attributes and the latent representation.

3) Attribute Intensity Control: Directly applicable for attribute intensity control is a side characteristic of our AttGAN. Although AttGAN is trained with binary attribute values (0/1), we find that AttGAN can be generalized for continuous attribute value in testing phase without any modification to its original design. As shown in Fig. 7, with

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

Fig. 6. Comparisons on CelebA [3] of multiple facial attribute editing among VAE/GAN [7], IcGAN [8], StarGAN [16] and our AttGAN. For each specified attribute combination, the facial attribute editing here is to **invert** each attribute in that combination.

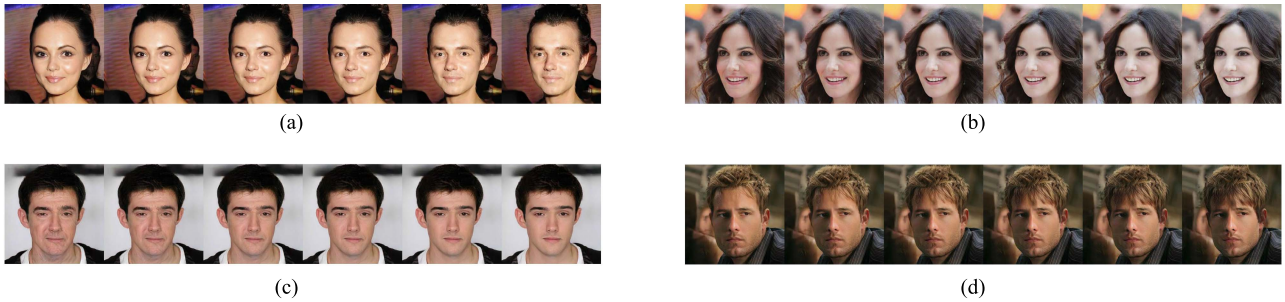


Fig. 7. Illustration of attribute intensity control. Best viewed in color and higher resolution (by zooming in). (a) Female to male. (b) No pale skin to pale skin. (c) Old to young. (d) No bangs to bangs.

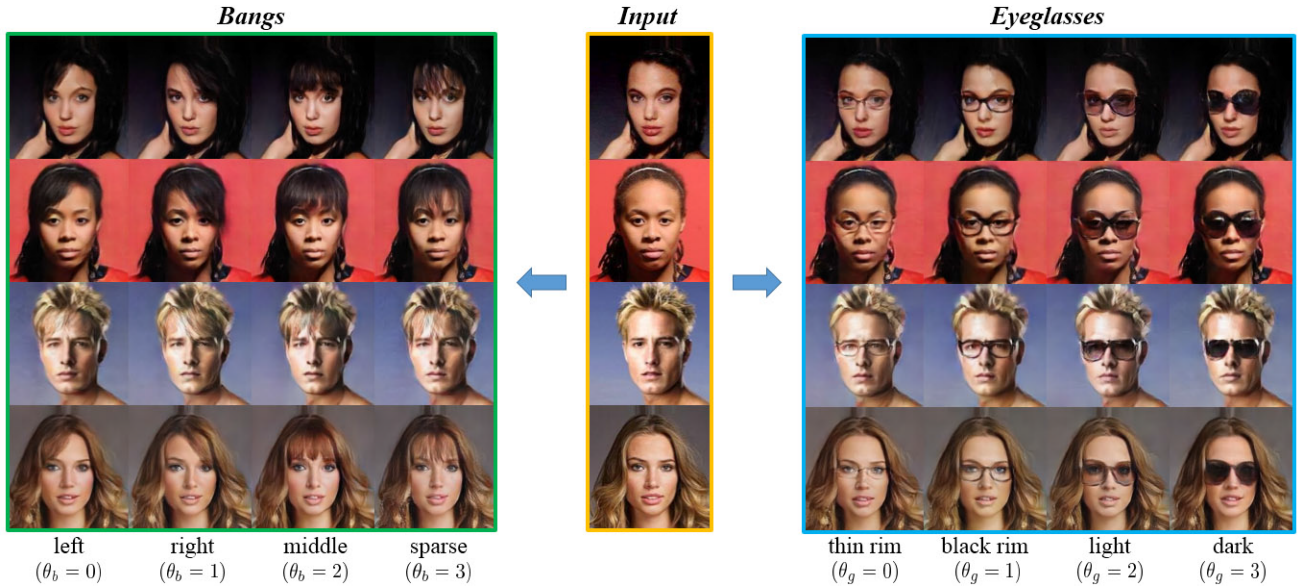


Fig. 8. Exemplar results of attribute style manipulation by using our extended AttGAN.

continuous value in [0, 1] as input, the gradual change of the generated images are smooth and natural.

4) Attribute Style Manipulation: Fig. 8 shows the results of the AttGAN extension for attribute style manipulation. As can be seen, different styles of attributes are dug out, such as different sides of bangs: left, right or middle. The extension is quite flexible and allows one to select the style he/she is interested in, rather than a stiff one.

5) High Quality Results and Failures: Fig. 14-16 in supplemental material shows additional results of high quality images with 384 × 384 resolution. Fig. 17 in supplemental material shows some failures. From observation, these failures are often

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

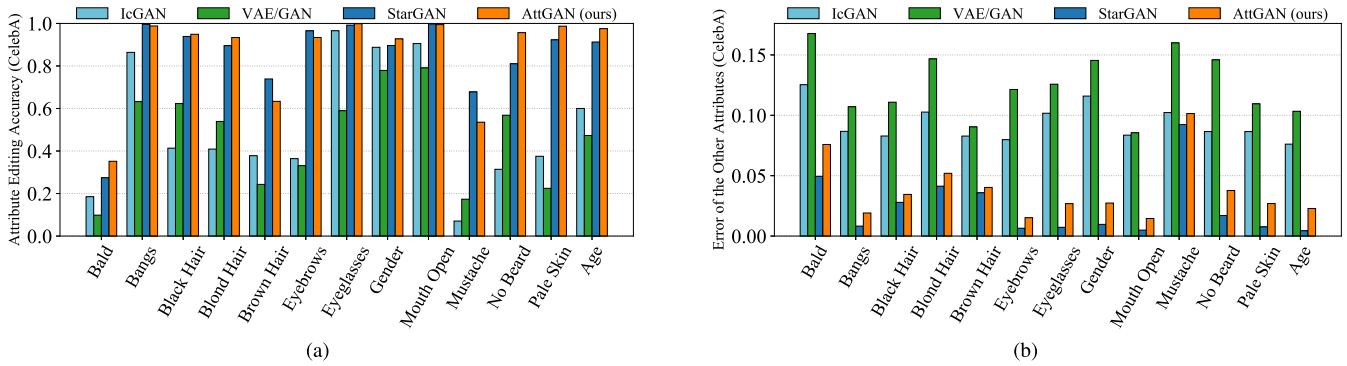


Fig. 9. Comparisons on CelebA [3] among IcGAN [8], VAE/GAN [7], StarGAN [16] and our AttGAN in terms of (a) facial attribute editing accuracy of target attribute and (b) preservation error of the rest attributes. (a) Attribute Editing Accuracy (higher the better). (b) Attribute Preservation Error (lower the better).

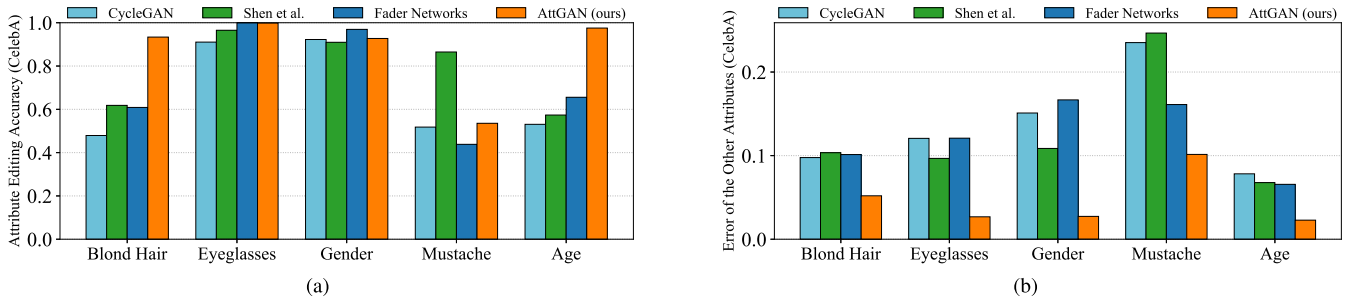


Fig. 10. Comparisons on CelebA [3] among CycleGAN [24], Shen et al. [10], Fader Networks [13] and our AttGAN in terms of (a) facial attribute editing accuracy of the target attribute and (b) preservation error of the rest attributes. (a) Attribute Editing Accuracy (higher the better). (b) Attribute Preservation Error (lower the better).

caused by the need of large appearance modification, such as editing a face with plenty of hair to "Bald" and adding "Bangs" to a bald face. Besides, another possible cause of the failures may be the imbalance data distribution, e.g., only 2.2% images in the training set have "Bald" attribute.

B. Quantitative Analysis

Facial Attribute Editing Accuracy/Error: To evaluate the facial attribute editing accuracy of our AttGAN, an attribute classifier independent of all methods is used to judge the attributes of the generated faces. This attribute classifier is trained on CelebA [3] dataset and achieves average accuracy of 90.89% per attribute on CelebA testing set. If the attribute of a generated image is predicted the same as the desired one by the classifier, it is considered a correct generation, otherwise an incorrect one. Besides, we also evaluate the average preservation error of the other attributes when editing each single attribute.

For evaluations on CelebA [3], Fig. 9a shows the attribute editing accuracy of VAE/GAN [7], IcGAN [8], StarGAN [16] and our AttGAN, all of which employ single model for multiple attributes. As can be seen, both AttGAN and StarGAN achieve much better accuracy than VAE/GAN and IcGAN, especially on "No Beard", "Pale Skin" and "Age". Moreover, the preservation errors of the rest attributes of AttGAN and StarGAN are much lower than VAE/GAN and IcGAN as shown in Fig. 9b. As for the comparisons between AttGAN and StarGAN, the attribute editing accuracies of them are comparable, but the attribute preservation error of AttGAN is a bit higher. However, the generated images of our AttGAN are more natural and realistic than StarGAN (see Fig. 4a and Fig. 6).

Furthermore, Fig. 10a and Fig. 10b show the attribute editing accuracy and preservation error of Fader Networks [13], Shen et al. [10] and CycleGAN [24], which employ one specific model for each attribute. As can be seen, all three baselines well edit the attributes which is comparable to AttGAN, but their preservation errors of the rest attributes are higher than AttGAN.

For evaluations on LFW [40], Fig 11a and Fig 11b show the attribute editing accuracy and preservation error, which is similar to the results of CelebA [3]. Both StarGAN [16] and our AttGAN achieve higher attribute editing accuracy on the target attribute, with lower attribute preservation error on the rest attributes.

C. Ablation Study: Effect of Each Component

In this part, we evaluate the necessity of the three main components: attribute classification constraint, reconstruction loss and adversarial loss. Besides, we also evaluate the disadvantage of the attribute-independent constraint. In Fig. 12, we show the results of different combinations of these components, where all experiments are based on models which learn to handle multiple attributes with one network. Row (1) contains the results of our AttGAN's original setting, which are natural and well preserve the attribute-excluding details.

Without the attribute classification constraint (row (2) of Fig. 12), the network just outputs the reconstruction images since these is no signal to force the network to generate the correct attributes. Similar phenomenon (but with some

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

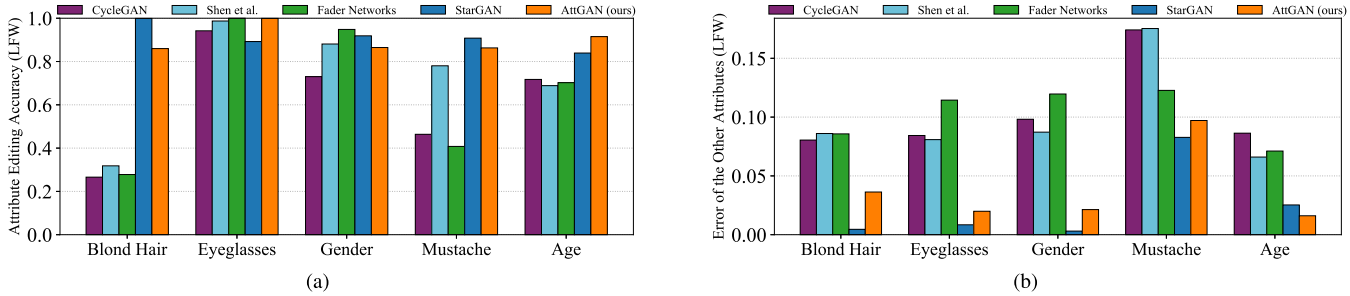


Fig. 11. Comparisons on LFW [40] among CycleGAN [24], Shen et al. [10], Fader Networks [13], StarGAN [16] and our AttGAN in terms of (a) facial attribute editing accuracy of target attribute and (b) preservation error of the rest attributes. (a) Attribute Editing Accuracy (higher the better). (b) Attribute Preservation Error (lower the better).

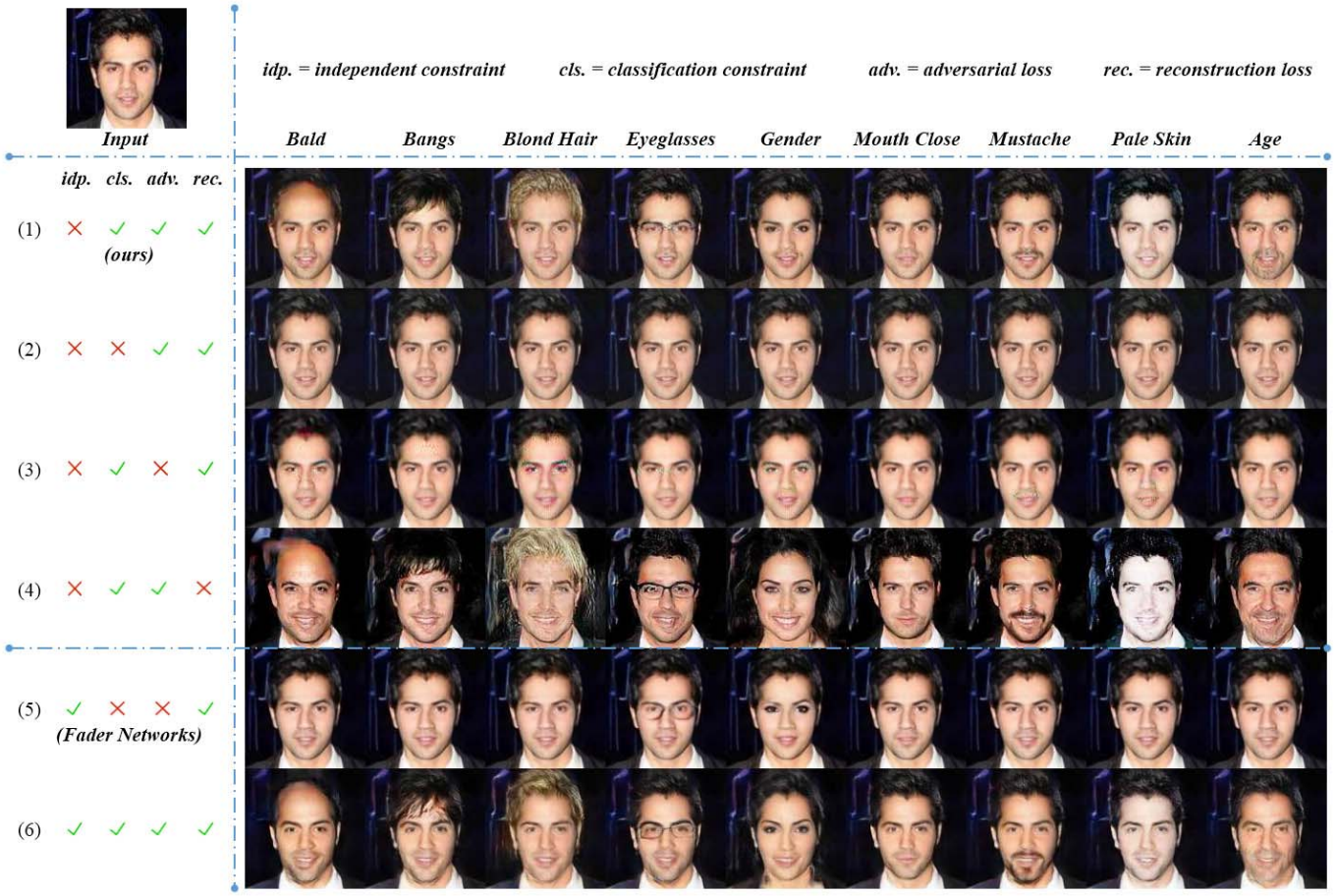


Fig. 12. Effect of different combinations of the four components.

noise) happens when we remove the adversarial loss although the classification constraint is kept (row (3)). One possible reason is that the training with classification constraint but without adversarial loss is similar to making an adversarial attack [42]. Therefore, although the classification constraint exists, the adversarial examples with incorrect attributes still fool the classifier (by the noise). In conclusion, the classification constraint does not work without the adversarial learning, or in other words, the adversarial learning helps to avoid adversarial examples. However, this is another topic needing more theoretical analysis and experiments, which is far beyond this paper.

In row (4) of Fig. 12, we present the results of AttGAN without reconstruction loss. As shown, although the resulting attributes are correct, the face identities change a lot accompanied with many artifacts. Therefore, the reconstruction loss is vital for preserving the attribute-excluding details.

Row (5) of Fig. 12 presents the results of the Fader Networks [13] like setting (attribute-independent constraint + reconstruction learning) and row (6) is AttGAN with attribute-independent constraint. As we can see in the row (5), the Fader Networks like setting works only on eyeglasses, gender and mouth open attributes with unsatisfactory performance. When we combine the AttGAN losses with the Fader Networks losses (row (6)), the attributes is correctly edited but the results contain artifacts and the attribute-excluding details change (e.g., the shape of nose and mouth). These experiments demonstrates that the attribute-independent constraint on the

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

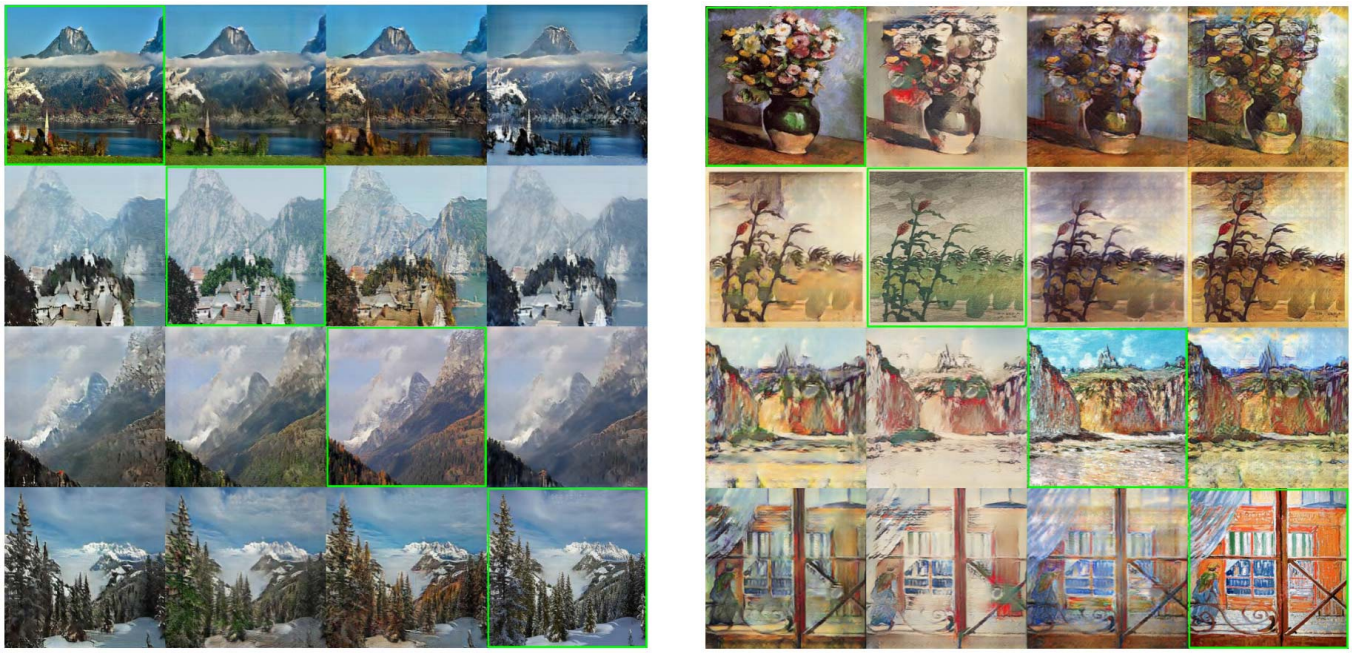


Fig. 13. Exploration of AttGAN on image style translation. The **diagonal** ones are the inputs. (a) Season translation. (b) Painting translation.

latent representation is not a favorable solution for facial attribute editing, since it constraints the capacity of the latent code resulting in information loss and degraded output images.

D. Exploration of Image Translation

Since facial attribute editing is closely related to image translation, we also try our AttGAN on the image style translation task where we define the style as a kind of attribute. We employ AttGAN on a season dataset [43] and a painting dataset [24] and the results are shown in Fig. 13. As we can see, the results of season are acceptable but the style translation of paintings is not so good accompanied with artifacts and blurriness. Compared to facial attribute editing, image style translation needs more variations on texture and color, a single model might be difficult to simultaneously handle all styles with large variation. However, AttGAN is a potential framework which deserves more explorations and extensions.

VI. Conclusion and Future Work

From the perspective of facial attribute editing, we reveal and validate the disadvantage of the attribute-independent constraint on the latent representation. Further, we properly consider the relation between the attributes and the latent representation and propose a facial attribute editing method, AttGAN, which incorporates attribute classification constraint, reconstruction learning, and adversarial learning to form an effective framework for high quality facial attribute editing. Experiments demonstrate that our AttGAN can accurately edit facial attributes, while well preserving the attribute-excluding details, with better visual effect, editing accuracy and lower editing error than the competing methods. Moreover, the AttGAN is directly applicable for attribute intensity control and can be extended to attribute style manipulation, which shows its potential for further exploration.

Although this work specializes in facial attribute editing, it has potential for attribute editing on general object, such as person attributes. However, different from facial attributes, person attributes have much larger variations which makes editing attributes of a person more challenging. For example, for "carrying handbag" attribute annotated in Market [44] and Duke [45] dataset, the "handbag" may hangs on the left arm or be carried on the right hand. In future work, it is worthwhile to investigate the attribute editing of the general object.

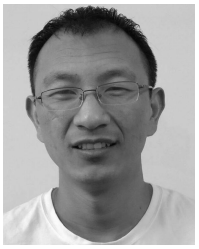
References

[1] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1891–1898.
[2] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[3] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.
[4] M. Ehrlich, T. J. Shields, T. Almaev, and M. R. Amer, "Facial attributes classification using multi-task representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun./Jul. 2016, pp. 752–760.
[5] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014. [Online]. Available: https://arxiv.org/abs/1312.6114
[6] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014, pp. 1–9.
[7] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1558–1566.
[8] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional GANs for image editing," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS) Workshops*, 2016. [Online]. Available: https://arxiv.org/abs/1611.06355
[9] M. Li, W. Zuo, and D. Zhang. (2016). "Deep identity-aware transfer of facial attributes." [Online]. Available: https://arxiv.org/abs/1610.05586

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

[10] W. Shen and R. Liu, "Learning residual images for face attribute manipulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1225–1233.
[11] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 700–708.
[12] S. Zhou, T. Xiao, Y. Yang, D. Feng, Q. He, and W. He, "GeneGAN: Learning object transfiguration and attribute subspace from unpaired data," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2017. [Online]. Available: http://arxiv.org/abs/1705.04932
[13] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, "Fader networks: Manipulating images by sliding attributes," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5967–5976.
[14] T. Kim, B. Kim, M. Cha, and J. Kim. (2017). "Unsupervised visual attribute transfer with reconfigurable generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1707.09798
[15] T. Xiao, J. Hong, and J. Ma, "DNA-GAN: Learning disentangled representations from multi-attribute images," in *Proc. Int. Conf. Learn. Represent. (ICLR) Workshops*, 2018. [Online]. Available: https://arxiv.org/abs/1711.05415
[16] 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 Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8789–8797.
[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.
[18] M. Li, W. Zuo, and D. Zhang. (2016). "Convolutional network for attribute-driven and identity-preserving human face generation." [Online]. Available: https://arxiv.org/abs/1608.06434
[19] P. Upchurch *et al.*, "Deep feature interpolation for image content changes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6090–6099.
[20] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, "Better mixing via deep representations," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 552–560.
[21] H. Chang, J. Lu, F. Yu, and A. Finkelstein, "PairedCycleGAN: Asymmetric style transfer for applying and removing makeup," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 40–48.
[22] W. Yin, Y. Fu, Y. Ma, Y.-G. Jiang, T. Xiang, and X. Xue, "Learning to generate and edit hairstyles," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1627–1635.
[23] M. Mirza and S. Osindero. (2014). "Conditional generative adversarial nets." [Online]. Available: https://arxiv.org/abs/1411.1784
[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.
[25] Y. Lu, Y.-W. Tai, and C.-K. Tang, "Attribute-guided face generation using conditional CycleGAN," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jun. 2018, pp. 282–297.
[26] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2Image: Conditional image generation from visual attributes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 776–791.
[27] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016. [Online]. Available: https://arxiv.org/abs/1511.06434
[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.
[29] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 214–223.
[30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5767–5777.
[31] T. Kaneko, K. Hiramatsu, and K. Kashino, "Generative attribute controller with conditional filtered generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7006–7015.
[32] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Adv. Neural Inf. Process. Syst. Workshops (NeurIPS)*, 2016, pp. 2642–2651.
[33] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 2172–2180.
[34] J. L. Ba, J. R. Kiros, and G. E. Hinton. (2016). "Layer normalization." [Online]. Available: https://arxiv.org/abs/1607.06450
[35] D. Ulyanov, A. Vedaldi, and V. Lempitsky. (2016). "Instance normalization: The missing ingredient for fast stylization." [Online]. Available: https://arxiv.org/abs/1607.08022
[36] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014, pp. 3320–3328.
[37] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.
[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.
[39] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.
[40] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Proc. Workshop Faces Real-Life Images, Detection, Alignment, Recognit.*, 2008, pp. 1–15.
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980
[42] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014. [Online]. Available: https://arxiv.org/abs/1312.6199
[43] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool. (2017). "ComboGAN: Unrestrained scalability for image domain translation." [Online]. Available: https://arxiv.org/abs/1712.06909
[44] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1116–1124.
[45] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, and Y. Yang. (2017). "Improving person re-identification by attribute and identity learning." [Online]. Available: https://arxiv.org/abs/1703.07220

Zhenliang He received the B.E. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2011. He is currently pursuing the Ph.D. degree with the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing. His research interests include facial landmark detection, facial attribute editing, and generative adversarial networks.



Wangmeng Zuo (M'09–SM'14) received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2007. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology. His current research interests include image enhancement and restoration, image and face editing, object detection, visual tracking, and image classification. He has published over 70 papers in top-tier academic journals and conferences. He has served as the Tutorial Organizer in ECCV 2016, and an Associate Editor for the *IET Biometrics* and the *Journal of Electronic Imaging*.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.

Meina Kan (M'14) received the B.S. degree in computer science from Shandong University in 2007, and the Ph.D. degree in computer vision from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS) in 2013. In 2011, she studied at the Centre for Multimedia and Network Technology, School of Computer Engineering, Nanyang Technological University. She is currently an Associate Professor with ICT, CAS. Her research interests mainly focus on computer vision and pattern recognition, especially on face recognition, transfer learning, deep learning, and weakly supervised learning. She has served as the Co-Chair for the ICPR18 Workshop on Deep Learning for Pattern Recognition, and the ACCV14 Workshop on Human Identification for Surveillance.

Shiguang Shan (M'04–SM'15) received the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2004, where he has been a Full Professor, since 2010, and is currently the Deputy Director of the CAS Key Lab of Intelligent Information Processing. His research interests cover computer vision, pattern recognition, and machine learning. He has published over 300 papers, with a total of over 15 000 Google Scholar citations. He served as the Area Chair for many international conferences, including the ICCV11, ICASSP14, ICPR12/14/19, ACCV12/16/18, FG13/18, BTAS18, and CVPR19. He was a recipient of China's State Natural Science Award in 2015, and China's State S&T Progress Award in 2005 for his research work. He was/is an Associate Editor of several journals, including the IEEE T-IP, *Neurocomputing*, CVIU, and PRL.

Xilin Chen (F'16) is currently a Professor with the Institute of Computing Technology, Chinese Academy of Sciences (CAS). He has authored one book and more than 200 papers in refereed journals and proceedings in the areas of computer vision, pattern recognition, image processing, and multimodal interfaces. He is a fellow of the IAPR and CCF. He served as an Organizing Committee Member for many conferences, including the General Co-Chair of FG13/FG18 and the Program Co-Chair of ICMI 2010. He is/was an Area Chair of CVPR 2017/2019 and ICCV 2019. He is currently an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, a Senior Editor of the *Journal of Visual Communication and Image Representation*, a Leading Editor of the *Journal of Computer Science and Technology*, and an Associate Editor-in-Chief of the *Chinese Journal of Computers* and the *Chinese Journal of Pattern Recognition and Artificial Intelligence*.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on March 17,2020 at 07:32:32 UTC from IEEE Xplore. Restrictions apply.