IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 26, 2019, accepted September 9, 2019, date of publication September 13, 2019,
date of current version September 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941272

ResAttr-GAN: Unpaired Deep Residual Attributes Learning for Multi-Domain Face Image Translation

RENTUO TAO[1], ZIQIANG LI[1], RENSHUAI TAO[2], AND BIN LI[1], (Member, IEEE)
[1]CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application Systems, Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), Hefei 230027, China
[2]State Key Laboratory of Software Development Environment, Department of Computer Science and Technology, Beihang University (BUAA), Beijing 100191, China

Corresponding author: Bin Li (binli@ustc.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 61836011 and Grant 61473271, in part by the Fundamental Research Funds for the Central Universities under Grant WK2090090023, and in part by the GPU Computing Cluster of the Data Center, School of Information Science and Technology, University of Science and Technology of China.

ABSTRACT Facial attributes edit can be seen as an image-to-image translation problem, whose goal is to transfer images from the source domain to the target domain. Specially, facial attributes edit aims at changing some semantic attributes of a given face image while keeping the contents of unrelated area unchanged. The great challenge for this problem lies on the lacking of paired data, i.e. we do not have paired face images that only differ on particular attributes. Moreover, to train a good attributes editing model, there always needs a great amount of train data which labeled by hand. If the train data amount was reduced, then the editing performance would decrease accordingly. Strong intelligent systems should be able to learn knowledge from less data samples (similar idea with few-shot learning). To mitigate this limitation, in this paper, we proposed a Siamese-Network based residual attributes learning model to learn the attributes difference in the high-level latent space. Compared to existing models that perform attributes editing based on an attributes classifier, the proposed deep residual attributes learning model utilized relatively weaker information of attribute differences for face image translation. Sufficient qualitative and quantitative experiments conducted on CelebA dataset proved the effectiveness of our proposed method, moreover, we also adopt the proposed residual attributes learning model in two state-of-the-art models under different data usage percentage to show the effectiveness of the proposed model on boosting attribute editing performance under limited data usage. The experiment results proved that the proposed method can improve data utilization efficiency and thus can boost the editing performance when the train data was limited.

INDEX TERMS Facial attributes edit, image-to-image translation, generative adversarial network, residual attributes learning.

I. INTRODUCTION

Deep neural networks (DNNs) have demonstrated its remarkable effectiveness in solving image-to-image translation problems, mapping an image from the source domain to an corresponding image in the target domain. Facial attributes edit is a specific kind of image translation task that aims at semantically manipulating the face images while keeping the attribute-unrelated area unchanged, i.e. given a face image and its corresponding attributes label, the goal is to generate desired target domain face image according to the attributes changes (mouth: open ↔ close, age: young ↔ old, hair color: black ↔ blond, etc.). The common ways are to learn a attribute classifier to make predictions on the generated face images [1]–[6], which was used to supervise the attribute edit process to get desired outputs semantically concord with the changed attributes. Moreover, for many facial attribute edit models [3]–[5], [7], they can only make single-attribute changes on the face image, which means they need to train a image translation model for each attribute,

The associate editor coordinating the review of this manuscript and approving it for publication was Dong Wang.

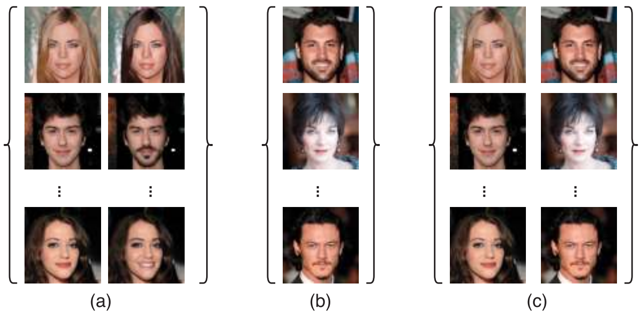


FIGURE 1. Illustration of the unpaired face images for residual attributes learning. (a) Paired face image data with different attributes. (b) Image data used for training an attributes classification network. (c) Unpaired data for deep residual attributes prediction network training.

thus less effective and cannot fulfill the multi-attributes edit simultaneously. Assume that we have got the desired face image of changed attributes, then compared to the source input, this target face image should only have difference at the edited attribute dimensions in the latent attribute space, i.e. if we map the source and target face image to the attribute space and get their corresponding latent representations, then the difference of these two representation should concord with the attributes label difference. We denote this latent difference that inspired this work as residual attributes and propose a deep residual attributes learning model which utilized a Siamese Network [8] to learn the residual attributes between the generated image and the reference image in the high-level attribute space and use errors signals backward from the residual attributes to supervise the generation process.

The proposed model mainly contains three parts, named Encoder (*Enc*), Decoder (*Dec*) and Residual Attributes Extractor (*ResAttr*) respectively. We adopt the Encoder-Decoder architecture for image generation as [6] and concatenate the conditional attributes vectors with the latent representations derived from the Encoder to get desired edited results. Due to the lacking of paired data (image pairs that differs at some semantic aspects which reflect the attributes while hold other parts the same), we cannot directly train a mapping network to fulfill the attributes edit. The concept of paired data and unpaired data was illustrated in Figure 1, where the leftmost column (a) denotes the paired trained data which can be used to train an image translation network directly, but unfortunately obtain this kind of data is very difficult and usually un-viable. The middle column (b) represents the labeled data used for training an attributes classification network and the rightmost column (c) demonstrate the randomly constructed unpaired face image groups used to train the deep residual attributes prediction network. Compared to existing methods that use the face images and its corresponding attributes annotations to train an attributes classification network in a fully-supervised way, we just select unpaired face images randomly and utilize the residual attributes information to train a residual attributes prediction network to give supervision on the image translation model.

The contributions of this work mainly lies in three folds:

- We proposed a new facial attributes editing model based on deep residual attributes learning, which was inspired by the gap between the desired output and the original input in the latent attribute space.
- We further demonstrated that the proposed model can improve the data utilization efficiency and thus can boost facial attributes edit performance when the train data amount is reduced. The experiment results show that the baseline models can achieve better performance by adopting the proposed residual attributes learning part under several data usage percentage.
- Sufficient qualitative and quantitative experiments were conducted to evaluate the performance of the proposed model. Moreover, we also proposed a simple but effective method for constructing unpaired training groups for residual attributes learning by flipping image batches.

The rest of the paper is organized as follows. In section II, we give a general introduction to the related works, which contains Generative Adversarial Networks (GAN), image-to-image translation and facial attributes edit. The model architecture and the train objective of the proposed ResAttr-GAN was explained in Section III, moreover, we also give the train algorithm of the proposed model in this section. Then we give the implementation details in section IV, which contains the train dataset information and the network architecture details, along with the hyper-parameters settings. The qualitative and quantitative experiments were presented in section V, which contains comparative single attribute editing, multiple attributes edit, residual attributes learning under different data usage percentage and attributes edit at a higher resolution. At the end, we give the conclusions and some ideas of the future works in section VI.

II. RELATED WORKS

A. GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks (GAN) was a kind of generative model that aims at learning the distribution of a train dataset by adversarial training, which was first introduced by Ian Goodfellow [9]. The idea of GAN was inspired by the zero-sum game (also called a two-player game or min-max problem) in the game theory, which means that the total gain was fixed and each player intended to achieve more by updating their strategy adversely upon each other. The two player in GAN were called generator (G) and discriminator (D) respectively, the generator aimed at generating high-fidelity samples that can fool the discriminator to make wrong predictions while the discriminator wants to clearly distinguish true or fake between the generated fake samples and true data samples. This min-max problem and the loss function of each player can be formulated as below:

$$min_G max_D E_{x\sim p_{data},z\sim p_z}\{\log D(x)+\log(1-D(G(z)))\} \quad (1)$$

$$G_{loss} = E_{z\sim p_z}\{\log(1-D(G(z)))\} \quad (2)$$

$$D_{loss} = E_{x\sim p_{data},z\sim p_z}\{-\log(D(x))-\log(1-D(G(z)))\} \quad (3)$$

where p_{data} and p_z represent the train data and random noise distribution respectively. In this work, we adopted WGAN_GP [10] as the backbone, which used a Wasserstein-1 distance and a gradient penalty item to stabilize the GAN training process, the loss functions of the generator and the discriminator in WGAN_GP are formulated as below:

$$G_{loss} = -E_{z\sim p_z}\{D(G(z))\} \tag{4}$$

$$\begin{aligned} D_{loss} = E_{x\sim p_{data}, z\sim p_z}\{&D(G(z)) - D(x) \\ &+ \lambda(\|(\nabla D(G(z)))\|_2 - 1)^2\} \end{aligned} \tag{5}$$

B. IMAGE-TO-IMAGE TRANSLATION

Image-to-Image translation was a class of computer vision problems whose goal is to get a model that can map images from the source domain to the target domain. Isola *et al.* [11] utilized a conditional GAN model to learn the mapping by paired training data. Zhu *et al.* [7] proposed an approach called CycleGAN to learn the image translation mapping in the absence of paired examples (here paired examples means the the correspondence of source domain image and target domain image exists) by a cycle-consistent adversarial network. Liu *et al.* [12] combines VAE [13] and CoGAN [14] together to learn the joint distribution of images in cross domain by two generator that share weights. Taeksoo *et al.* [15] also utilized the cycle-consistency loss to discover cross-domain relations for image translation. However, the above mentioned models can only deal with translations between two domains, to overcome this constraint, instead of learning a fixed translation, Yunjey *et al.* [1] proposed a model that utilize a generator to take both image and its corresponding domain label as input and learns to flexibly translate it into the target domain. Moreover, Lee *et al.* [16] proposed to boost image translation performance without paired train data by disentangling the image onto two spaces: a domain-invariant content space and a domain-specific attribute space. Chen *et al.* [17] also proposed a method for unpaired image-to-image translation that based on latent space interpolation. Chen *et al.* [18] applied a reversible generative network based framework for makeup transfer, which can also be seen as an image-to-image translation task.

C. FACIAL ATTRIBUTES EDIT

Facial attributes edit was a special kind of image-to-image translation problem which aims at manipulating face images in the attributes space. Given an face image with its corresponding attributes annotation, the attributes edit model wants to output a face image with changed attributes while keep the attributes-unrelated area unchanged compared to the original input. The facial attributes editing methods can be grouped to latent interpolation methods, attributes-invariant representation disentangling methods and fully-supervised methods that rely on the attributes prediction network.

1) LATENT INTERPOLATION METHODS

The idea of this kind of methods was to train a model or Encoder to map the face images to the latent space and used the derived latent representations to find the attribute manipulating direction. For a given attribute, the dataset can be seperated to two groups that with or without the attribute, then the manipulating direction can be derived by a minus operation on the average latent representation of the positive group and the negative group. Larsen *et al.* [19] proposed a model that combines GAN [9] and VAE [13] together and learns an embedding where high-level abstract visual features can be modified by feature interpolation. Sun *et al.* [20] proposed a Mask-Adversarial AutoEncoder that can manipulate attributes by modifying the encoded features by a relative value. Upchurch *et al.* [21] proposed a deep feature interpolation method for facial attributes edit through a transformed output image reconstruction by reversely mapping the latent representation into pixel space.

2) ATTRIBUTES-INVARIANT FEATURE DISENTANGLING

This kind of methods aims at learning latent representations that are invariant with respect to the attributes, which can be concatenated with the attributes labels and pass through a decoder to generate desired outputs. Lample *et al.* [22] proposed a model named Fader Networks that incorporate this constrains through an adversarial process: learning an Encoder such that the discriminator is unable to identify the right attributes from the encoded latent representations. Creswell *et al.* [23] utilized the same idea with [22] to factor out single attribute from the rest representation. Perarnau *et al.* [24] forward the edited image through the Encoder and minimize the gap between the derived latent representation of the original input and the edited image, thus the encoded latent representations has no relation with the attributes. Xiao *et al.* [25], [27] and Zhou *et al.* [26] split the latent codes to attribute-relevant part and attribute-irrelevant part by utilizing a discriminator to make comparison on images come from different attributes swapping operations. Besides, Yan *et al.* [28] proposed a model named Attribute2Image to model images as a composite of foreground and background and learn the disentangled representations by VAE [13], which shares a similar idea with attributes-invariant methods.

3) METHODS RELY ON ATTRIBUTES CLASSIFIER

This kind of methods use the train images and its corresponding attributes to learn a attributes classifier in a fully-supervised manner. He *et al.* [6] proposed a model named AttGAN which utilize a attributes classifier to predict the attributes of the edited image and thus can supervise the generation process. Shen and Liu [5] proposed a facial attributes edit model by residual image learning. The model contains a dual residual image generation network and an attributes classification network which was used to supervise the residual image generation for each attributes. Zhang *et al.* [4]

proposed a facial attributes edit model with spatial attention, which incorporate the attributes classifier to supervise the spatial attention generation. Li *et al.* [3] utilized a two-class attribute prediction network to supervise a mask generator and a transform network to obtain the desired output by element-wise operation on the generated mask and transformed image. Liu *et al.* [29] proposed a flow-based model for controllable image synthesis, where an attribute classifier was adopted in the latent space. Liu *et al.* [30] also proposed a attribute editing model by selectively taking the difference between target and source attribute vectors as input. The idea of difference attribute vector is quite similar with residual attributes, but Liu *et al.* [30] take the difference vector as input other that output compared to our work. Pumarola *et al.* [2] also proposed a facial animation model from a single image which also adopted an attributes classifier to supervise the color mask and attention mask generation.

Apart from the methods categorized above, there are also some facial attribute transfer works based on geometry information like facial key-point landmarks [31], [32], moreover, Han et.al proposed a weekly supervised GAN model (WS-GAN) to do facial attribute editing from relative pairwise attribute order comparison made by an Elo rating network that can learn relative attribute strengths. Chen *et al.* [33] also proposed a face attribute manipulation model by decomposing the facial attributes into multiple semantic components. Compared to existing methods, we proposed a new idea of solving the facial attributes editing problem. We utilize a Siamese network to predict the attribute difference in the latent space, which can guide the translation model to a right direction for multi-domain face image manipulation. Details of the proposed model will be presented in the next sections.

III. DEEP RESIDUAL ATTRIBUTES GAN

The proposed model mainly contain three parts, named Encoder (Enc), Decoder (Dec) and Residual Attributes Extractor ($ResAttr$) respectively. The Encoder was used to map the input face image to a latent representation, which can be forwarded to the Decoder along with the conditional attribute vector, thus get the decoded face image. The residual attributes extractor aims at making correct residual attributes predictions on the input face image groups, moreover, it also contains a Discriminator to distinguish the data source of the inputs. The Encoder and Decoder together construct the Generator, which can be trained adversely with the Discriminator. The detailed information of different parts of the proposed model are given below.

A. PROBLEM FORMULATION

Facial attributes edit is a class of computer vision problem where the goal is to change some semantic aspects (like mouth close to open, hair color change and old to young, etc.) of a given face image while preserving the identity. Given a face image dataset $\{\mathcal{X}, \mathcal{Y}\}$, $\mathcal{X} = \{X_1, X_2, \ldots, X_M\}$, $\mathcal{Y} = \{Y_1^n, Y_2^n, \ldots, Y_M^n\}$, where X_i and Y_i^n represent the i-th face image and its corresponding binary attributes label respectively, M and n denote the number of the train samples and the attributes label dimension respectively. A facial attributes edit system aims at getting a face image mapping G from the original attribute domain to the target attribute domain, i.e. for a given face image X_i and its attributes label Y_i (n was removed for simplicity here), we flip the k-th element of Y_i and get a new attribute label Y_i', G was used to map X_i to X_i', where X_i' and X_i are with the same identity but with different attribute that represented by the k-th attribute element. The problem can be formulated as below:

$$\begin{aligned} Y_i &= (y_1, \ldots, y_k, \ldots, y_n) \\ Y_i' &= (y_1, \ldots, 1 - y_k, \ldots, y_n) \quad (6) \\ X_i' &= G(X_i, Y_i') \quad (7) \end{aligned}$$

The above equations give the formulation of single attribute editing process, and the multiple attributes editing process can also be derived similarly.

B. ENCODER-DECODER GENERATOR

Here we adopt the Encoder-Decoder architecture as our generator G, i.e. the facial attributes transfer model. The generator network can be seen in Figure 4, where the encoder (Enc) was used to mapping the input image X^a (we remove the coordinate index in the next sections for simplicity) to a latent feature space and get the latent representation Z^a. The decoder (Dec) was used to decode face image from the latent representation Z^a along with the attributes label Y^a or Y^b. The generation process can be formulated as:

$$\begin{aligned} Z^a &= Enc(X^a) \quad (8) \\ \tilde{X}^a &= Dec(Enc(X^a), Y^a) \quad (9) \\ \tilde{X}^b &= Dec(Enc(X^a), Y^b) \quad (10) \end{aligned}$$

where $\tilde{X}^a$ and $\tilde{X}^b$ denote the desired editing output and the reconstructed image respectively.

C. DEEP RESIDUAL ATTRIBUTES LEARNING

For the reason that we do not have paired data (same person with different attributes), thus we cannot learn residual attributes of two face images with same identity. To overcome this constraint, we instead to train a residual attributes extractor in a deep feature space using un-paired data. The deep residual attributes extraction network D was illustrated in Figure 4, which can be divided into three stages (Siamese Network, data source classifier D_2 and residual attributes extractor D_3): Siamese Network aims at mapping the input face image to a high-level semantic feature space, D_2 was to make predictions on the data sources, i.e. distinguish real face images from generated fake ones, D_3 was used to extract the residual attributes of two input face images according to the output features of Siamese Network, here we directly use minus operation as the '**op**' demonstrated in Figure 4. The process are formulated as below:

$$F^\alpha, F^\beta = Siamese(X^\alpha), Siamese(X^\beta) \quad (11)$$

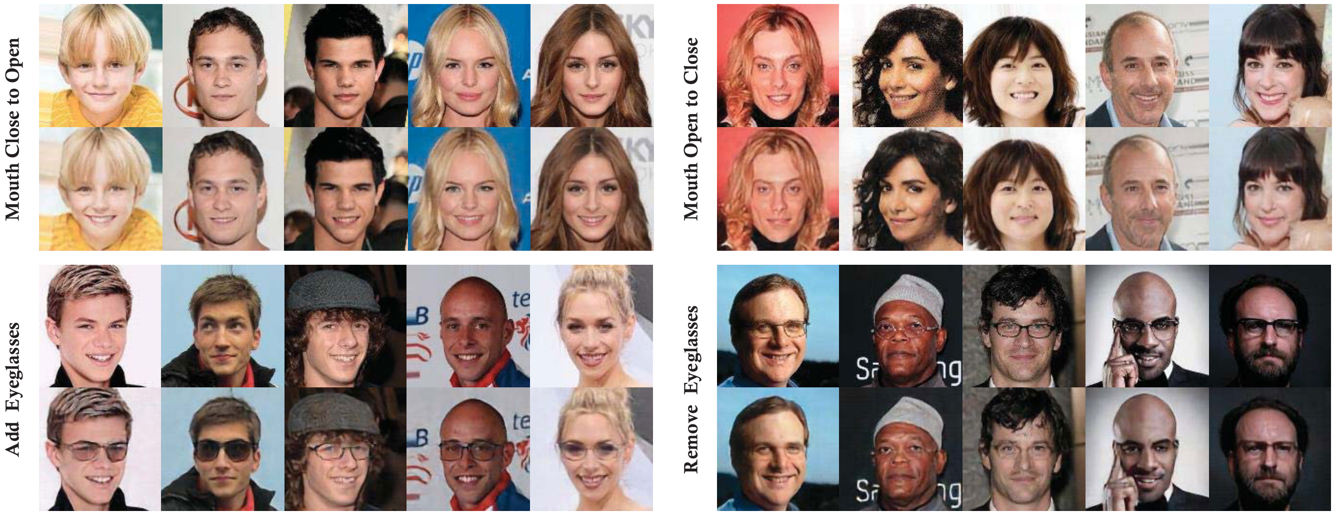


FIGURE 2. Demonstration of the edited results of mouth 'open' ↔ 'close' translation (top part), adding and removing eyeglasses (bottom part) based on ResAttr-GAN. The first row and the second row of different parts denote the original input and the corresponding edited results respectively.

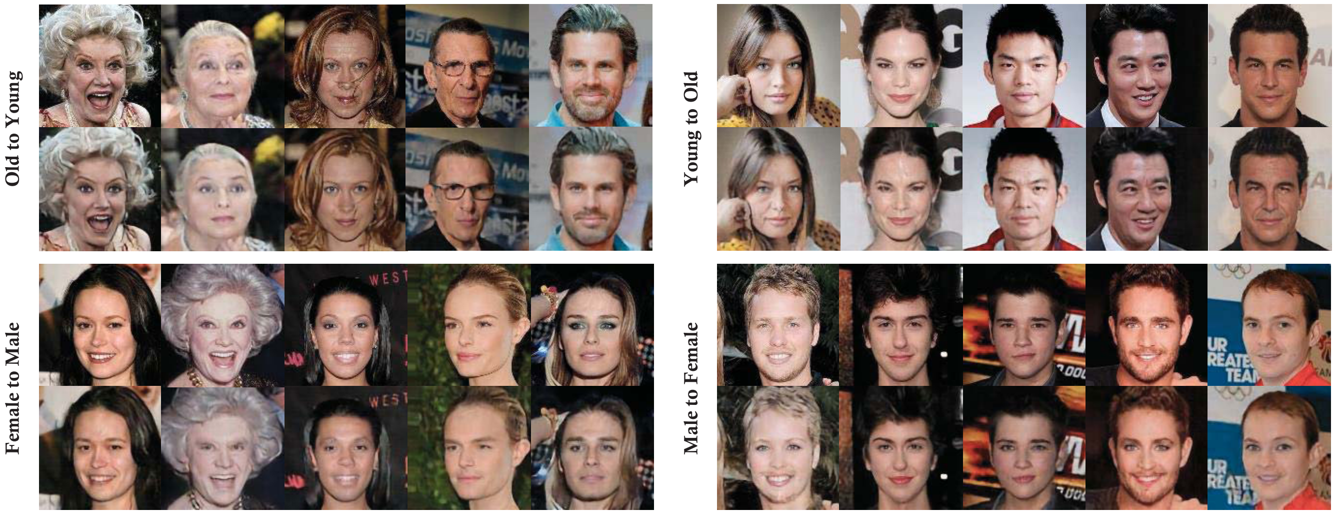


FIGURE 3. Demonstration of the edited results of mouth 'old' ↔ 'young' translation (top part) and 'male' ↔ 'female' translation (bottom part) based on ResAttr-GAN. The first row and the second row of different parts denote the original input and the corresponding edited results respectively.

$$S^{\alpha}, S^{\beta} = D_2(F^{\alpha}), D_2(F^{\beta}) \tag{12}$$

$$Res\_Attr\_\alpha\beta = D_3(F^{\alpha} - F^{\beta}) \tag{13}$$

The above equations can also be summarized as:

$$S^{\alpha}, S^{\beta}, Res\_Attr\_\alpha\beta = D(X^{\alpha}, X^{\beta}) \tag{14}$$

where X^{α} and X^{β} denote unpaired face images from the train data set with attributes α and β, and $F^{\alpha}, F^{\beta}, S^{\alpha}, S^{\beta}$ represent the high-level semantic features and the data source of image X^{α} and X^{β} respectively. $Res\_Attr\_\alpha\beta$ was the extracted residual attributes of X^{α} and X^{β}. There need to notice that X^{α} and X^{β} do not need to have the same identity, i.e. the residual attributes extractor was trained by unpaired face image data. In Figure. 1, we give an illustration of the concept of unpaired data.

D. RESATTR-GAN

In this section, we will give the train objective of the proposed model in detail. For a given input face image X^a with attribute a, the difference between the desired output and the source input image should be concord with the attribute difference. Moreover, the generated image should keep other attribute unrelated areas unchanged compared to the input image X^a. From equation (9) and (10) we can get the reconstruction $\tilde{X}^a$ and edited result $\tilde{X}^b$ of X^a. The main difference between $\{\tilde{X}^a, \tilde{X}^b\}$ and $\{X^{\alpha}, X^{\beta}\}$ is that $\{\tilde{X}^a, \tilde{X}^b\}$ are generated images and $\{X^{\alpha}, X^{\beta}\}$ are images from train data set. We rewrite equation (14) for $\{\tilde{X}^a, \tilde{X}^b\}$ and $\{X^{\alpha}, X^{\beta}\}$ as below (forward process):

$$S^a, S^b, Res\_Attr\_ab = D(\tilde{X}^a, \tilde{X}^b) \tag{15}$$

$$S^{\alpha}, S^{\beta}, Res\_Attr\_\alpha\beta = D(X^{\alpha}, X^{\beta}) \tag{16}$$

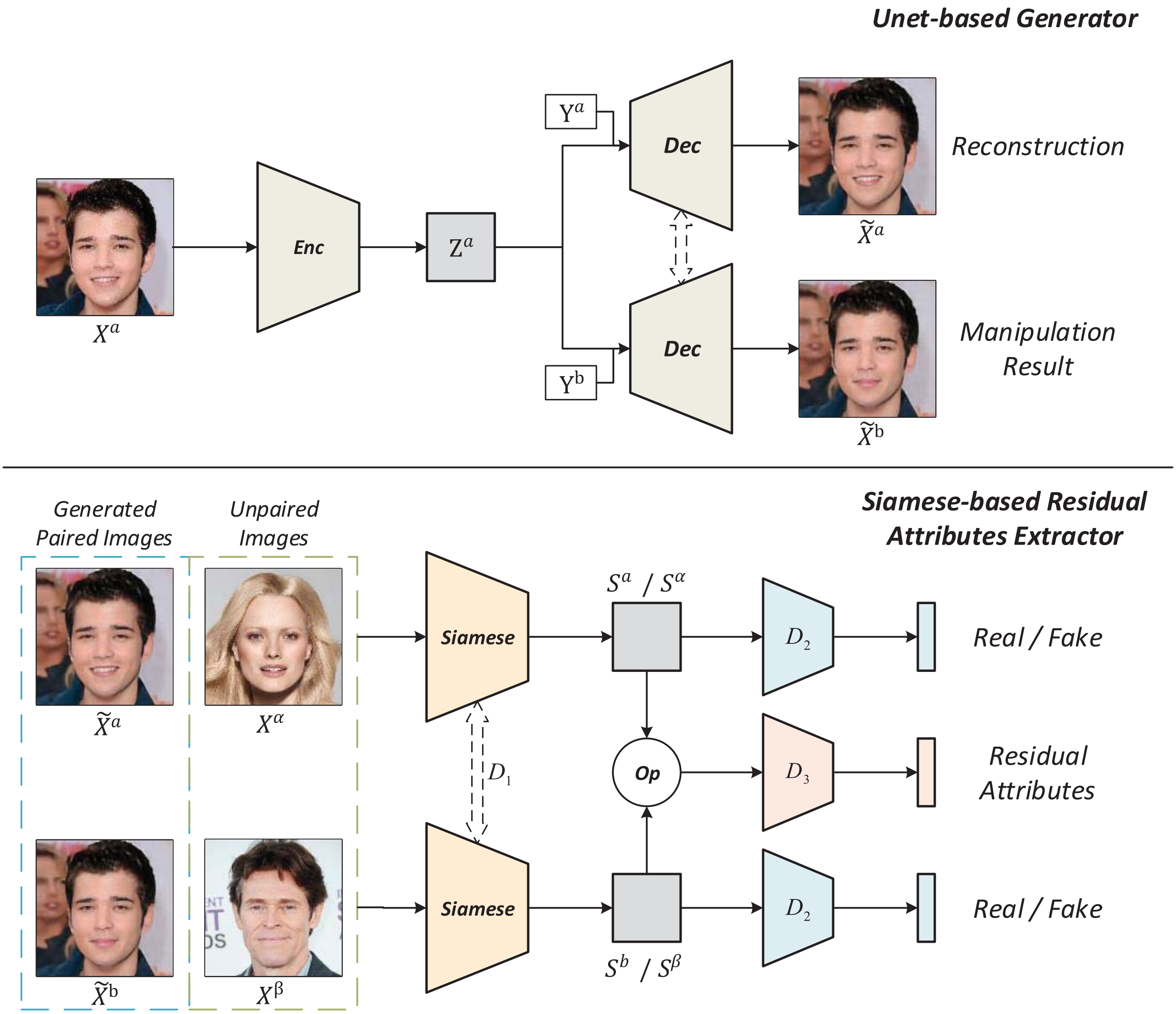


FIGURE 4. Illustration of the ResAttr-GAN network architecture. The top part denote the U-net based generator model, which composed of an encoder and a decoder, where the encoder was used to map the input to a latent space and the decoder was used to reconstruct face images from the latent representation along with the attribute vector. The bottom part denote the Siamese Network based discriminator model, which composed of a residual attributes extractor and a domain source classifier. The dashed line between two Siamese model represents "weights sharing". The meaning of the math symbols can are explained in Section III.

1) ADVERSARIAL LOSS

As similar to the vanilla-GAN, the generator wants to confuse the discriminator to predict true on the generated fake samples while the discriminator aims at clearly distinguish the fake samples from true. Thus the adversarial objective of the proposed ResAttr-GAN can be formulated as:

$$min_{Enc,Dec}max_D\Big\{E_{X^\alpha,X^\beta\sim p_{data}}\{\log S^\alpha+\log S^\beta\} + E_{X^a\sim p_{data},b\sim p_{attr}}\{\log(1-S^a)+\log(1-S^b)\}\Big\} \quad (17)$$

Thus we can derive the adversarial loss for the generator and discriminator respectively from the above min-max problem:

$$\mathcal{L}_{adv}\{Enc, Dec\} = E_{X^a\sim p_{data},b\sim p_{attr}}\{\log(1-S^a)+\log(1-S^b)\} \quad (18)$$

$$\mathcal{L}_{adv}\{D\} = -E_{X^\alpha,X^\beta\sim p_{data}}\{\log S^\alpha+\log S^\beta\} - E_{X^a\sim p_{data},b\sim p_{attr}}\{\log(1-S^a) + \log(1-S^b)\} \quad (19)$$

2) RECONSTRUCTION LOSS

If the generator can work well, i.e. the decoder can get correct output from the latent representations of the input face image combine with the attribute label input. Then $\tilde{X}^a$ should be

TABLE 1. Percentage (%) of attributes under different data usage.

Data Usage	Bangs	Black Hair	Blond Hair	Brown Hair	Bushy Eyebrows	Eyeglasses	Male	Mouth Open	Mustache	No Beard	Young
1	15.11	23.59	14.95	20.78	14.36	6.51	42.01	48.22	4.19	83.29	77.55
$\frac{1}{16}$	14.84	24.35	14.49	20.46	14.50	6.77	42.12	47.78	3.73	83.34	77.92
$\frac{1}{32}$	14.70	24.34	14.52	20.43	14.17	6.45	42.18	47.65	3.94	82.94	77.93
$\frac{1}{64}$	15.23	24.02	14.35	20.47	14.63	6.61	41.96	48.01	3.66	83.47	77.45
$\frac{1}{128}$	15.13	24.28	14.78	20.06	14.36	6.05	42.22	49.40	3.31	82.90	76.43

nearly the same with X^a, which can be formulated as:

$$\mathcal{L}_{reons}\{Enc, Dec\} = \|\tilde{X}^a - X^a\|_l \tag{20}$$

where l was denoted as the norm factor and we set $l = 1$ (L1-norm) here to get more smooth reconstruction.

3) RESIDUAL ATTRIBUTES LOSS

As for the residual attributes, the discriminator wants to correctly predict the residual attributes of images come from the true dataset while the generator wants to generate realistic images which can have the right residual attributes compared to the reference image. The residual attributes extraction loss of the generator and the discriminator can be written as:

$$\mathcal{L}_{res}\{D\} = \|Res\_Attr\_\alpha\beta - (\alpha - \beta)\|_l \tag{21}$$

$$\mathcal{L}_{res}\{Enc, Dec\} = \|Res\_Attr\_ab - (a - b)\|_l \tag{22}$$

where we use MSE-loss ($l = 2$) to train the residual attributes extractor and the generator.

4) OVERALL OBJECTIVE

From the above derivation, we can finally get the overall train objective for our proposed ResAttr-GAN model:

$$\begin{aligned}\mathcal{L}\{Enc, Dec\} &= \lambda_{adv} \cdot \mathcal{L}_{adv}\{Enc, Dec\} \\ &\quad + \lambda_{recons} \cdot \mathcal{L}_{recons}\{Enc, Dec\} \\ &\quad + \lambda_{res} \cdot \mathcal{L}_{res}\{Enc, Dec\} \qquad (23) \\ \mathcal{L}\{D\} &= \lambda_{adv} \cdot \mathcal{L}_{adv}\{D\} + \lambda_{res} \cdot \mathcal{L}_{res}\{D\} \qquad (24)\end{aligned}$$

where λ_{adv}, λ_{recons}, λ_{res} are the coefficients for different loss items. The train algorithm of the proposed model can be seen in Algorithm 1.

IV. IMPLEMENTATION DETAILS

A. CELEBA DATASET

CelebFaces Attributes Dataset (CelebA [34]) is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 binary attribute annotations. The face images were all align cropped to 128*128 for most experiments in this paper. Moreover, we choose 11 less visually ambiguous attributes to evaluate the effectiveness of our proposed model, which include 'Bangs', 'Black Hair', 'Blond Hair', 'Brown Hair', 'Bushy Eyebrows', 'Eyeglasses', 'Male', 'Mouth Slightly Open', 'Mustache', 'No Beard' and 'Young'. Besides, we use the officially separated training and validation set as our training data and use the test set for evaluation. In Table. 1, we give the statistical information of the percentage of different attributes under different amount of data usage (range from 1 to 1/128), from which we can see there exist attributes imbalance problem in the CelebA dataset.

B. IMAGE GROUPS FOR RESIDUAL ATTRIBUTES LEARNING

Grouped face image are needed to train the deep residual attributes extractor. Suppose the number of train images is M, then the number of possible groups is $\binom{M}{2} \approx 1.5e^{10}$, which is a extraordinarily large number for model training (time and resource consuming). Instead, we proposed a simple but effective method to choose image groups by just flip the image order in each batch:

$$\{X_i, Y_i\}_{i=1,2,\ldots,m} \longrightarrow \{X_i, Y_i\}_{i=m,m-1,\ldots,1}$$

thus we can get m unpaired face image groups:

$$\{X_1, X_m\}, \{X_2, X_{m-1}\}, \ldots, \{X_{m-1}, X_2\}, \{X_m, X_1\}$$

Actually, traing group $\{X_i, X_{m-i}\}$ and $\{X_{m-i}, X_i\}$ are same groups for residual attributes learning for the reason that $Tanh()$ is an odd function, i.e. $Tanh(Z_i - Z_{m-i}) = -Tanh(Z_{m-i} - Z_i)$. Training batches are randomly chosen at each iteration thus adequate image groups can be derived by the proposed image group construction scheme.

C. NETWORK ARCHITECTURE AND TRAINING DETAILS

Table 2 and Table 3 give the detailed network architecture of the encoder-decoder Generator and the residual attributes extractor respectively. The encoder network and the decoder network were composed of 5 convolution and deconvolution (here used the transposed convolution) layers respectively. Batch normalization and Leaky-ReLU activation were used in the Encoder while ReLU activation units were used in decoder. Moreover, we also incorporate a skip-connection like Unet [35] in the generator for better decoding performance. As for the residual attributes extractor, it was composed of 5 convolution layers append with an instance normalization layer and a leaky-ReLU activation layer, and

Algorithm 1 Training Algorithm of ResAttr-GAN

```
Input: Dataset $\{X, Y\}$, initialized model $Enc, Dec, D$ and loss weights $\lambda_{adv}, \lambda_{recons}, \lambda_{res}$
Output: Trained models $Enc, Dec, D$
for number of epochs do
    for number of iterations do
        Sample mini-batch of m samples $\{X_i, Y_i\}_{i=1,2,\ldots,m}$
        Random permutation on attributes and get $\{Y'_i\}_{i=1,2,\ldots,m}$
        Flip the data along the batch size dimension and get $\{X_i^{flip}, Y_i^{flip}\}_{i=1,2,\ldots,m}$
        Forward Pass:
            $X'_i \leftarrow Dec(Enc(X_i), Y'_i)$
            $\tilde{X}_i \leftarrow Dec(Enc(X_i), Y_i)$
            $\tilde{S}_i, S'_i, Res\_Attr\_G \leftarrow D(\tilde{X}_i, X'_i)$
            $S_i, S_i^{flip}, Res\_Attr\_D \leftarrow D(X_i, X_i^{flip})$
        Losses Calculation:
            $\mathcal{L}_{recons}\{Enc, Dec\} \leftarrow \sum_{i=1}^{m} \|\tilde{X}_i - X_i\|_l$
            $\mathcal{L}_{adv}\{Enc, Dec\} \leftarrow \sum_{i=1}^{m} \{\log(1 - \tilde{S}_i) + \log(1 - S'_i)\}$
            $\mathcal{L}_{adv}\{D\} \leftarrow \sum_{i=1}^{m} \{\log \tilde{S}_i + \log S'_i + \log(1 - S_i) + \log(1 - S_i^{flip})\}$
            $\mathcal{L}_{res}\{Enc, Dec\} \leftarrow \sum_{i=1}^{m} \|Res\_Attr\_G - (Y_i - Y'_i)\|_l$
            $\mathcal{L}_{res}\{D\} \leftarrow \sum_{i=1}^{m} \|Res\_Attr\_D - (Y_i - Y_i^{flip})\|_l$
        Model Update:
        for number of k do
            Update the Discriminator weights by descending the gradient:
                $\theta_D \leftarrow \theta_D - \nabla_{\theta_D}\{\lambda_{adv} \cdot \mathcal{L}_{adv}\{D\}, \lambda_{res} \cdot \mathcal{L}_{res}\{D\}\}$
        end for
        Update the Encoder and Decoder weights by descending the gradient:
            $\{\theta_{Enc}, \theta_{Dec}\} \leftarrow \{\theta_{Enc}, \theta_{Dec}\} - \nabla_{\{\theta_{Enc},\theta_{Dec}\}}\{\lambda_{adv} \cdot \mathcal{L}_{adv}\{Enc, Dec\}$
                $+ \lambda_{recons} \cdot \mathcal{L}_{recons}\{Enc, Dec\} + \lambda_{res} \cdot \mathcal{L}_{res}\{Enc, Dec\}\}$
    end for
end for
```

followed by a two-sibling network, one for data source classification and the other for residual attributes prediction. Besides, we use Adam [36] with learning rate of 0.0001, $\beta_1 = 0.5$ and $\beta_2 = 0.999$ for training, and the coefficients for different loss items in equation (23) and (24) are set to: $\lambda_{adv} = 1$, $\lambda_{gp} = 10$, $\lambda_{recons} = 100$, $\lambda_{res} = 100$. During the training process, we update the residual attributes extractor 5 times for each update of the generator.

V. EXPERIMENTS

In this section, we will give the experiment results. We conducted several facial attributes edit experiments (including comparative single-attribute editing, multiple attributes editing, attributes editing on higher resolution face images) to evaluate the effectiveness of the proposed model qualitatively and quantitatively. Moreover, we also give the comparative quantitative evaluation results to show that the proposed

TABLE 2. Generator network architecture.

Part	Input → Output Shape	Layer Information
Encoder	$(h,w,3) \rightarrow (\frac{h}{2},\frac{w}{2},64)$	CONV-(N64,K4x4,S2,P1), BN, Leaky ReLU
	$(\frac{h}{2},\frac{w}{2},64) \rightarrow (\frac{h}{4},\frac{w}{4},128)$	CONV-(N128,K4x4,S2,P1), BN, Leaky ReLU
	$(\frac{h}{4},\frac{w}{4},128) \rightarrow (\frac{h}{8},\frac{w}{8},256)$	CONV-(N256,K4x4,S2,P1), BN, Leaky ReLU
	$(\frac{h}{8},\frac{w}{8},256) \rightarrow (\frac{h}{16},\frac{w}{16},512)$	CONV-(N512,K4x4,S2,P1), BN, Leaky ReLU
	$(\frac{h}{16},\frac{w}{16},512) \rightarrow (\frac{h}{32},\frac{w}{32},1024)$	CONV-(N1024,K4x4,S2,P1), BN, Leaky ReLU
Decoder	$(\frac{h}{32},\frac{w}{32},1024+n_attr) \rightarrow (\frac{h}{16},\frac{w}{16},1024)$	DECONV-(N1024,K4x4,S2,P1), BN, ReLU
	$(\frac{h}{16},\frac{w}{16},64) \rightarrow (\frac{h}{8},\frac{w}{8},64)$	DECONV-(N512,K4x4,S2,P1), BN, ReLU
	$(\frac{h}{8},\frac{w}{8},128) \rightarrow (\frac{h}{4},\frac{w}{4},64)$	DECONV-(N256,K4x4,S2,P1), BN, ReLU
	$(\frac{h}{4},\frac{w}{4},256) \rightarrow (\frac{h}{2},\frac{w}{2},64)$	DECONV-(N128,K4x4,S2,P1), BN, ReLU
	$(\frac{h}{2},\frac{w}{2},512) \rightarrow (h,w,3)$	DECONV-(N3,K4x4,S2,P1), BN, ReLU

TABLE 3. Residual Attributes Extractor network architecture.

Part	Input → Output Shape	Layer Information
$InputLayer$	$(h,w,3*2) \rightarrow (\frac{h}{2},\frac{w}{2},64*2)$	CONV-(N64,K4x4,S2,P1), IN, Leaky ReLU
ResAttr (D_Stage_I)	$(\frac{h}{2},\frac{w}{2},64*2) \rightarrow (\frac{h}{4},\frac{w}{4},128*2)$	CONV-(N128,K4x4,S2,P1), IN, Leaky ReLU
	$(\frac{h}{4},\frac{w}{4},128*2) \rightarrow (\frac{h}{8},\frac{w}{8},256*2)$	CONV-(N256,K4x4,S2,P1), IN, Leaky ReLU
	$(\frac{h}{8},\frac{w}{8},256*2) \rightarrow (\frac{h}{16},\frac{w}{16},512*2)$	CONV-(N512,K4x4,S2,P1), IN, Leaky ReLU
	$(\frac{h}{16},\frac{w}{16},512*2) \rightarrow (\frac{h}{32},\frac{w}{32},1024*2)$	CONV-(N1024,K4x4,S2,P1), IN, Leaky ReLU
Output Layer (D_Stage_II)	$(\frac{h}{32},\frac{w}{32},1024) \rightarrow (1,1,1024)$	CONV-(N1024,K4x4,S1,P0), IN, Leaky ReLU
	$(1,1,1024) \rightarrow (1,1,13)$	FC(13), Tanh
Output Layer (D_Stage_III)	$(\frac{h}{32},\frac{w}{32},1024*2) \rightarrow (1,1,1024*2)$	CONV-(N1024,K4x4,S1,P0), IN, Leaky ReLU
	$(1,1,1024*2) \rightarrow (1,1,2)$	FC(1), Sigmoid

residual attributes learning model can boost attributes editing performance when the train data amount was reduced.

A. EVALUATION METRICS

1) ATTRIBUTES EDITING ACCURACY

The most direct evaluation metric was to check whether the output edited results accompany with the attributes input, i.e. predict the attributes of the output and measure the editing accuracy can reflect the model performance.

2) FRECHET INCEPTION DISTANCE (FID [37])

FID is supposed to improve the Inception Score [38] by comparing the statistics of generated samples to real samples, instead of evaluating generated samples in a vacuum. Lower FID is better, corresponding to more closer distance between the generated and real data distributions.

3) STRUCTURAL SIMILARITY (SSIM [39])

The Structural SIMilarity (SSIM) index is a method for measuring the similarity between two images (value range from 0 to 1, high value represent more similarity). The SSIM index can be viewed as a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality.

B. COMPARATIVE MODELS

We adopt IcGAN [24], AEGAN [19], FaderNetworks [22], StarGAN [1] and AttGAN [6] as our comparative models.

TABLE 4. FID and SSIM evaluation results on different comparative models.

Metrics	IcGAN [24]	AEGAN [19]	FaderNet [22]	StarGAN [1]	AttGAN [6]	ResAttr-GAN
FID	49.54	33.13	43.80	14.60	14.73	18.86
SSIM	0.23	0.39	0.78	0.84	0.84	0.81

TABLE 5. Facial attributes edit accuracy (%) of different comparative models.

Methods	Bangs	Black Hair	Blond Hair	Brown Hair	Bushy Eyebrows	Eye Glasses	Male	Mouth Open	Mus-tache	No Beard	Young
IcGAN [24]	71.04	50.31	42.87	**60.52**	31.64	**96.19**	74.99	85.49	5.69	22.44	31.20
AEGAN [19]	45.64	30.49	29.01	29.38	29.47	65.57	39.84	45.58	10.37	14.34	22.05
FaderNet [22]	22.93	20.82	9.99	20.50	20.22	54.69	**83.57**	81.96	4.14	42.61	50.14
StarGAN [1]	**83.22**	63.46	59.85	41.53	23.37	91.31	52.85	**90.18**	26.78	38.94	50.73
AttGAN [6]	79.26	**85.45**	**68.45**	57.46	35.52	85.15	65.58	83.71	23.52	**57.77**	55.97
ResAttr-GAN	61.36	57.74	64.49	41.64	**44.32**	52.05	46.70	85.59	**27.05**	37.00	**58.79**

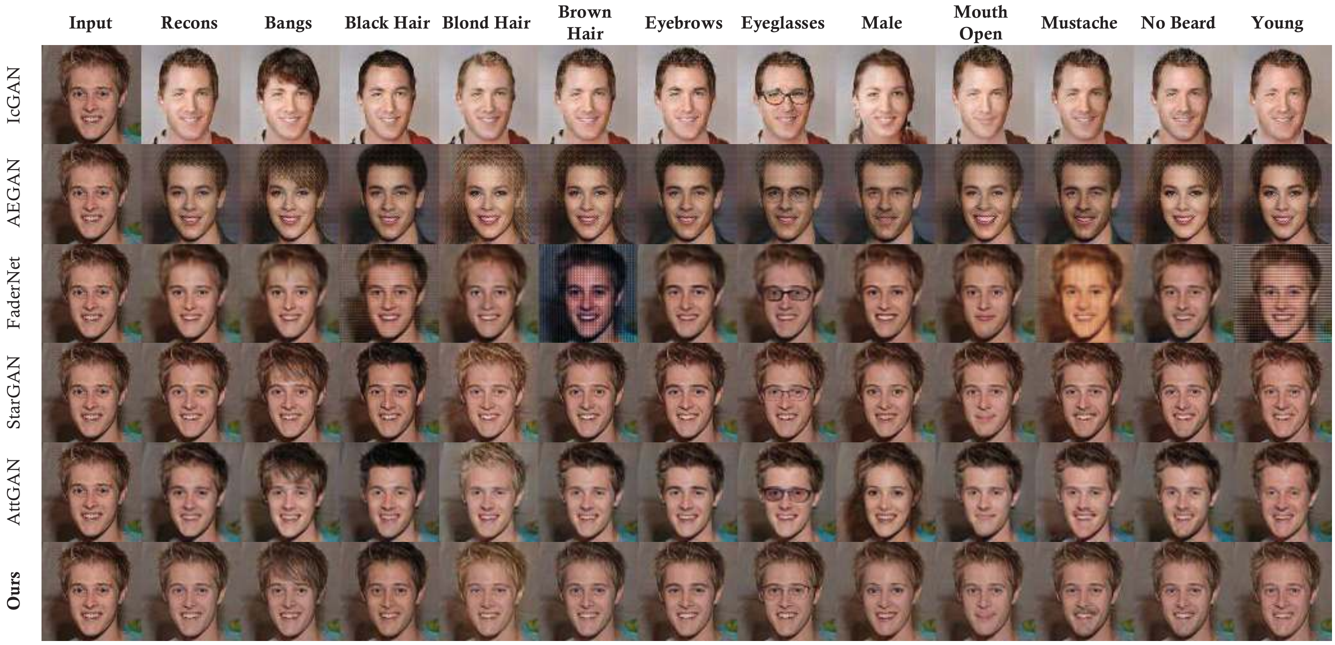


FIGURE 5. Single attribute edit results of different models. The leftmost column denote the original input and the second column denote the corresponding reconstruction results. The swapped editing results ('Male' ↔ 'Female' etc.) of different attributes were shown from the third column to the last column.

IcGAN [24] perform image editing by combining an encoder (mapping a real image into a latent space and a conditional representation) with a cGAN and the official implementation can be found at https://github.com/Guim3/IcGAN. AEGAN [19] perform attribute manipulating by interpolation in the latent space (for each attribute, it compute the mean vector for images with the attribute and without the attribute), the corresponding official implementation can be found at https://github.com/andersbll/autoencoding_beyond_pixels. FaderNetworks [22] do the task by disentangling the salient information of the image and the values of attributes directly in the latent space, however, for each attribute, it need to train a corresponding model, i.e. FaderNetworks cannot use one trained model for different attributes editing task. The official implementation can be found at https://github.com/facebookresearch/FaderNetworks. The latter two models StarGAN [1] and AttGAN [6] all perform multi-domain facial attributes transfer by utilizing a discriminator (make predictions on the data source and attributes domain of the inputs) which trained in a fully-supervised manner. The official implementation of these two models can be found at: StarGAN: https://github.com/yunjey/stargan,

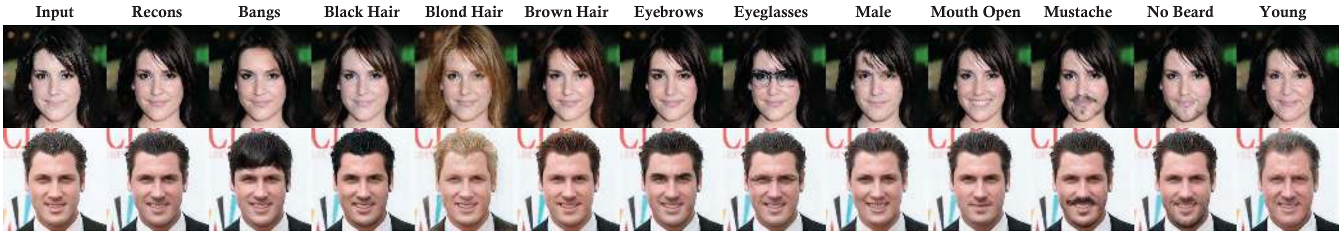


FIGURE 6. Single attribute editing results of ResAttr-GAN on image resolution 256 × 256. The leftmost column denote the original input and the second column denote the corresponding reconstruction results. The swapped editing results ('Male' ↔ 'Female' etc.) of different attributes were shown from the third column to the last column.

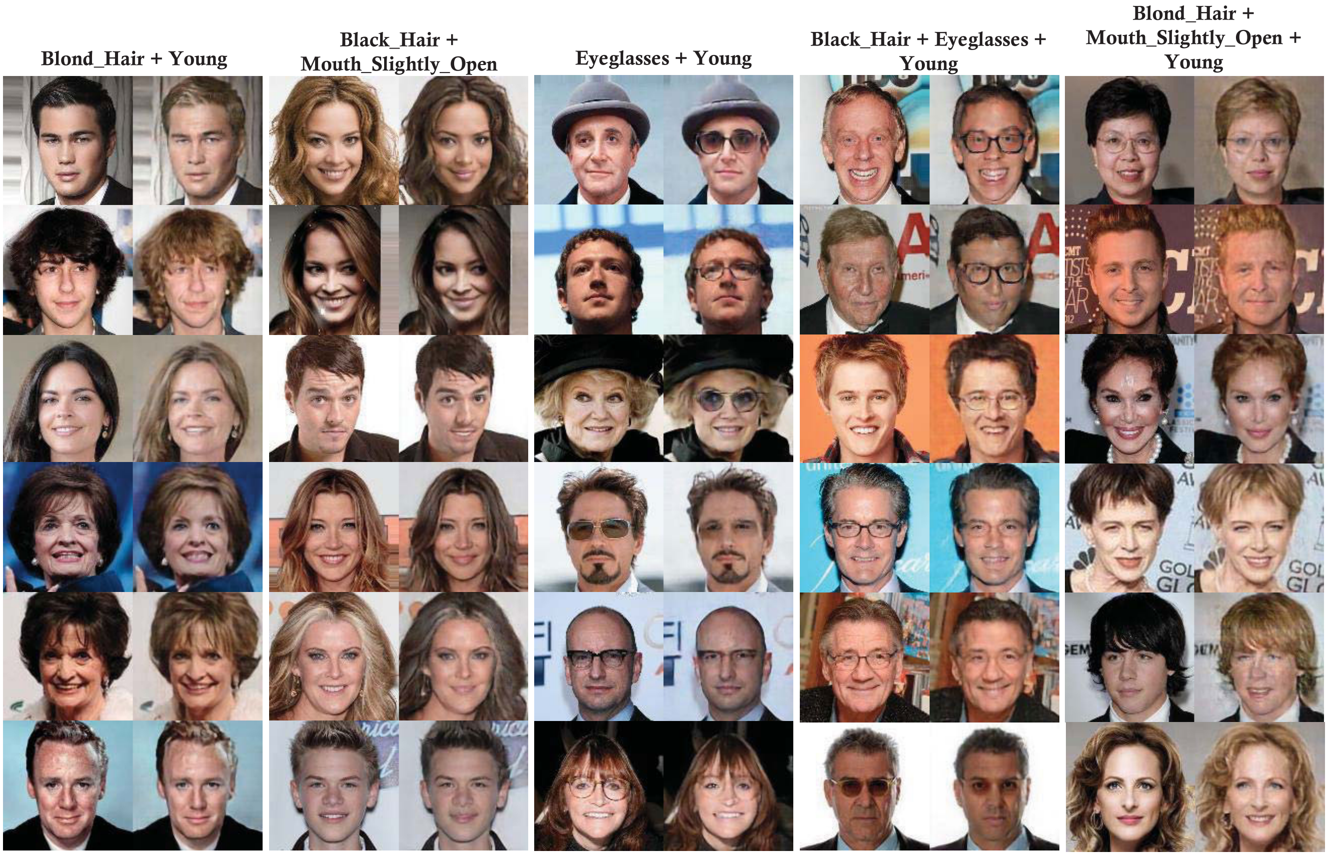

Blond_Hair + Young
Black_Hair + Mouth_Slightly_Open
Eyeglasses + Young
Black_Hair + Eyeglasses + Young
Blond_Hair + Mouth_Slightly_Open + Young

FIGURE 7. Demonstration of the multiple attributes editing results based on ResAttr-GAN. For each part of the figure from left to right, the first column denote the original face image and the second column denote the edited results. The edited attributes can be seen at the top of each part, for hair colors, the original face images all translated to the targeted hair color domain and for other attributes like eyeglass or young, this demonstration contains both positive and negative translations, if the eyeglass value of the original input was true, then the translation will remove the eyeglasses, otherwise will add the glasses.

AttGAN: https://github.com/LynnHo/AttGAN-Tensorflow respectively.

C. SINGLE ATTRIBUTE EDIT

To evaluate the effectiveness of the proposed model on facial attributes transfer, we firstly give the results of single attribute edit. In Figure. 2, we show the edit results of mouth 'open' ↔ 'close' translation and, add or remove 'eyeglasses'. From the demonstration results we can see that the proposed method can manipulate attribute 'mouth slightly open' and 'eyeglasses' effectively and can add various style eyeglasses while some small artifacts may occur when removing eyeglasses. Similarly, Figure. 3 gives the the results of 'old' ↔ 'young' and 'male' ↔ 'female' translation. More attributes transfer results (hair color change, beard add/ remove, bushy eyebrows add/remove etc.) can be seen in Appendix. Besides, we also provide qualitatively comparative results in Figure 5, from which we can see that IcGAN give low reconstruction quality and AEGAN tend to produce blur edited outputs. AttGAN can handle attribute editing effectively but some

TABLE 6. FID and SSIM evaluation on StarGAN with / without ResAttr.

Metrics	StarGAN/16	(S+**ResAttr**)/16	StarGAN/32	(S+**ResAttr**)/32	StarGAN/64	(S+**ResAttr**)/64	StarGAN/128	(S+**ResAttr**)/128
FID	27.51	28.90	33.31	32.10	36.30	32.40	42.68	46.01
SSIM	0.735	0.745	0.734	0.746	0.731	0.747	0.729	0.729

TABLE 7. FID and SSIM evaluation on AttGAN with / without ResAttr.

Metrics	AttGAN/16	(A+**ResAttr**)/16	AttGAN/32	(A+**ResAttr**)/32	AttGAN/64	(A+**ResAttr**)/64	AttGAN/128	(A+**ResAttr**)/128
FID	24.89	23.53	37.80	38.79	57.14	57.25	110.24	105.11
SSIM	0.777	0.741	0.718	0.731	0.685	0.683	0.582	0.578

TABLE 8. Attributes edit accuracy (%) comparison with / without ResAttr.

Data Usage	Methods	Average Accuracy	Bangs	Black Hair	Blond Hair	Brown Hair	Bushy Eyebrows	Eye Glasses	Male	Mouth Open	Mus-tache	No Beard	Young
$\frac{1}{16}$	StarGAN	50.89	70.69	52.23	36.63	34.16	32.21	81.69	61.64	90.99	12.98	32.96	53.57
	S+ResAttr	**52.90**	**78.26**	**58.16**	**44.20**	**40.31**	**32.65**	73.84	**64.91**	89.75	**14.90**	**33.11**	51.80
	AttGAN	56.34	65.76	61.26	64.24	33.60	65.69	96.25	40.95	92.76	18.77	33.36	47.06
	A+ResAttr	**56.65**	**69.60**	60.97	59.38	29.87	**66.96**	**96.41**	38.47	**96.03**	**22.33**	**35.06**	**48.05**
$\frac{1}{32}$	StarGAN	48.60	66.32	49.42	33.19	39.58	30.46	81.72	64.49	85.50	10.33	31.29	42.32
	S+ResAttr	**53.43**	**72.40**	**60.90**	**49.28**	**41.22**	25.17	80.54	**68.11**	**92.10**	**14.47**	**31.41**	**52.15**
	AttGAN	44.84	46.71	35.96	22.88	23.58	64.40	96.32	27.63	86.57	21.21	28.33	39.63
	A+ResAttr	**50.56**	**60.59**	**39.06**	22.60	**25.56**	61.72	**96.62**	**57.96**	**91.72**	**24.48**	**35.07**	**40.83**
$\frac{1}{64}$	StarGAN	47.51	60.65	58.62	36.53	34.46	30.07	66.69	65.60	89.16	9.43	30.03	41.34
	S+ResAttr	**51.32**	**67.44**	54.58	**38.74**	**35.22**	**36.44**	**76.57**	62.88	**92.37**	**11.31**	**31.85**	**57.13**
	AttGAN	37.08	35.18	25.62	12.74	19.42	46.29	94.83	23.62	72.66	18.63	23.29	35.59
	A+ResAttr	**51.13**	**57.09**	**37.87**	**24.33**	**25.02**	**62.64**	**96.05**	**58.64**	**93.00**	**19.96**	**42.18**	**45.65**
$\frac{1}{128}$	StarGAN	39.40	45.35	38.93	23.52	26.04	17.83	66.85	53.71	80.87	10.15	25.38	44.82
	S+ResAttr	**40.81**	**45.58**	33.62	21.89	**26.92**	**18.41**	**75.87**	**55.27**	**85.86**	**10.91**	**28.86**	**45.69**
	AttGAN	27.01	21.34	19.32	9.08	17.03	29.81	86.56	19.03	48.00	6.11	15.24	25.55
	A+ResAttr	**28.38**	**35.19**	**20.35**	**9.51**	**17.42**	27.48	86.39	**20.92**	**50.54**	**6.40**	13.37	24.63

attribute-unrelated area was easily influenced during the editing process. From the qualitative single attribute translation results demonstrated in the above mentioned figures, we can see that the proposed model can handle facial attributes editing effectively and can achieve comparable editing performance. Besides, we also evaluate the proposed model on a higher resolution (256 × 256), the demonstration editing results can be seen in Figure 6.

D. EFFECTS OF THE RESIDUAL ATTRIBUTES EXTRACTOR

To evaluate the effectiveness of the proposed residual attributes learning model (denoted as **ResAttr** in Table 6, Table 7 and Table 8) on boosting facial attributes editing performance, we adopt the residual attributes learning part in two state-of-the-art baseline models: StarGAN [1] and AttGAN [6]. We reduce the train data usage percentage to 1/16, 1/32, 1/64, 1/128 respectively, i.e. the number of train images was reduced to different level. Then we add the proposed **ResAttr** to StarGAN and AttGAN respectively (denoted as **S** + **ResAttr** and **A** + **ResAttr**) and compare the attributes editing performance of various models. From Table 6, Table 7 and Table 8, we can see that the model with **ResAttr** can achieve higher attribute editing accuracy with comparable FID and SSIM scores compare to baseline models under different data usage percentage. From Table 8 we can see that the average attributes editing accuracy were improved by utilizing residual attributes learning part, which means the proposed 'ResAttr' can boost facial attributes editing accuracy effectively. Another thing need to mention is that we only flip the training images once, which means we only use M (batch size) group of unpaired images in one iteration. In fact, we can construct more groups of unpaired data through randomly permutation, we believe that the editing accuracy can still be improved if we can leverage more unpaired data.

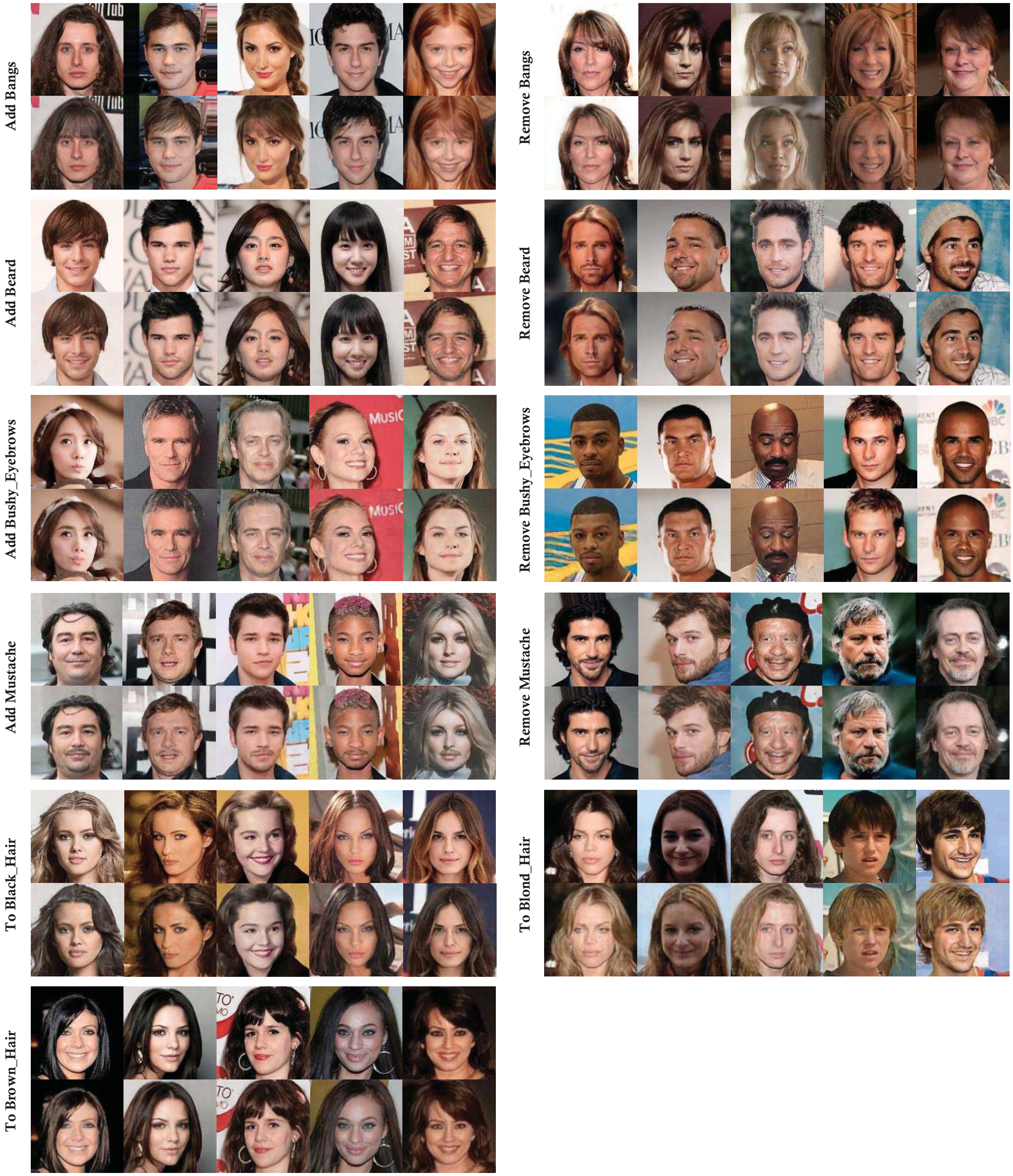


FIGURE 8. Other attributes edited results of ResAttr-GAN. From top to bottom: Add/Remove 'Bangs'; Add/Remove 'Beard'; Add/Remove 'Bushy Eyebrows'; Add/Remove 'Mustache'; To Black/Blond/Brown Hair.

E. QUANTITATIVE EVALUATION OF RESATTR-GAN

To furthermore evaluate the performance of the proposed model quantitatively, we first follow the criteria as He *et al.* [6] to calculate the attribute editing accuracy, which was fulfilled by utilizing a pre-trained attributes classifier to predict the attributes of the edited image generated from different facial attributes transfer models. Moreover, we also adopt FID [37] and SSIM [39] to measure the quality of generated samples (manipulation results) and reconstruction. The quantitatively evaluation results can be seen in Table 4 and Table 5. As for the quality of the generated samples and reconstruction, we can see the evaluation results in Table 4, from which we can see that ResAttr-GAN can achieve comparable performance with StarGAN [1] and AttGAN [6], which is much higher than IcGAN [24], AEGAN [19] and FaderNetworks [22] both in FID (the lower the better) and SSIM (the higher the better). And for the editing accuracy, we trained an attributes classifier based on resnet-101 [40] network and achieved classification accuracies of [94.79, 87.55, 95.10, 86.65, 91.29, 99.39, 97.40, 92.72, 96.57, 94.75, 85.77]% for attributes 'Bangs', 'Black Hair', 'Blond Hair', 'Brown Hair', 'Bushy Eyebrows', 'Eyeglasses', 'Male', 'Mouth Open', 'Mustache', 'No Beard' and 'Young' respectively. The pretrained attributes classifier was used to calculate the attribute editing accuracy of different comparative models. The editing accuracy of different models were show in Table 5, from which we can see that the proposed model can achieve comparable editing performance with these models. However, for the reason that we utilized a much weaker attributes difference information rather than an attributes classifier to supervise the editing process, the editing accuracy were not that high compared to the state-of-the-art methods that based on attributes classifier.

F. MULTIPLE ATTRIBUTE EDIT

To furthermore evaluate the performance of the proposed model on multiple attributes editing, we conducted several experiments of multiple attributes translation, which include (a) "Black Hair" + "Mouth Slightly Open", (b) "Blond Hair" + "Young", (c) "Eyeglasses" + "Young", (d) "Blond Hair" + "Mouth Slightly Open" + "Young" and (e) "Black Hair" + "Eyeglasses" + "Young" respectively. The multiple attributes editing results were demonstrated in Figure 7, where for each column, the left part denotes the original source image and the right part shows the corresponding multiple attributes manipulation results. For each multiple-attributes edit, we give bi-direction translation demonstrations, like in (a), we give "Black Hair" + "Young to Old" translation and "Black Hair" + "Old to Young" translation in the top three rows and bottom three rows respectively. From those demonstrated multiple attributes editing results we can see that the proposed model can simultaneously handle multiple attributes editing effectively.

VI. CONCLUSION

In this paper, we proposed a new facial attributes edit model that based on deep residual attributes learning, compared to existing methods that rely on an attributes classification network trained in a fully-supervising manner, we propose to learn the attributes difference of unpaired face images and can achieved comparable facial attributes editing results. Moreover, we demonstrated that when the train data amount was reduced, the proposed deep residual learning model can improve the data utilization efficiency and thus boost the editing performance. The experiment results demonstrated the effectiveness of our proposed method in both single and multiple attributes editing. However, there are still some obstacles need to overcome in facial attributes edit, like keeping attributes-unrelated area unchanged and robustness when dealing with extreme cases like heavy-hair to bald translation and side face editing etc.. Besides, there also exist attributes unbalance problem in the dataset, which can be seen from Table 1, thus we believe that better results can be achieved by adopting data balance methods in future works.

APPENDIX

ADDITIONAL FACIAL ATTRIBUTES EDITING RESULTS

See Figure 8.

REFERENCES

[1] 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.*, Jun. 2018, pp. 8789–8797.

[2] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "GANimation: Anatomically-aware facial animation from a single image," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 818–833.

[3] M. Li, W. Zuo, and D. Zhang, "Deep identity-aware transfer of facial attributes," 2016, *arXiv:1610.05586*. [Online]. Available: https://arxiv.org/abs/1610.05586

[4] G. Zhang, M. Kan, S. Shan, and X. Chen, "Generative adversarial network with spatial attention for face attribute editing," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 417–432.

[5] W. Shen and R. Liu, "Learning residual images for face attribute manipulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1225–1233.

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

[7] 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.*, Oct. 2017, pp. 2223–2232.

[8] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, vol. 2, 2015, pp. 1–30.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5769–5779.

[11] 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.*, Jul. 2017, pp. 1125–1134.

[12] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 700–708.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: https://arxiv.org/abs/1312.6114

[14] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 469–477.

[15] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1857–1865.

[16] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 35–51.

[17] Y.-C. Chen, X. Xu, Z. Tian, and J. Jia, "Homomorphic latent space interpolation for unpaired image-to-image translation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2408–2416.

[18] H.-J. Chen, K.-M. Hui, S.-Y. Wang, L.-W. Tsao, H.-H. Shuai, and W.-H. Cheng, "Beautyglow: On-demand makeup transfer framework with reversible generative network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 10042–10050.

[19] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," 2015, *arXiv:1512.09300*. [Online]. Available: https://arxiv.org/abs/1512.09300

[20] R. Sun, C. Huang, J. Shi, and L. Ma, "Mask-aware photorealistic face attribute manipulation," 2018, *arXiv:1804.08882*. [Online]. Available: https://arxiv.org/abs/1804.08882

[21] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger, "Deep feature interpolation for image content changes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7064–7073.

[22] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. A. Ranzato, "Fader networks: Manipulating images by sliding attributes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5967–5976.

[23] A. Creswell, Y. Mohamied, B. Sengupta, and A. A. Bharath, "Adversarial information factorization," 2017, *arXiv:1711.05175*. [Online]. Available: https://arxiv.org/abs/1711.05175

[24] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional GANs for image editing," 2016, *arXiv:1611.06355*. [Online]. Available: https://arxiv.org/abs/1611.06355

[25] T. Xiao, J. Hong, and J. Ma, "DNA-GAN: Learning disentangled representations from multi-attribute images," 2017, *arXiv:1711.05415*. [Online]. Available: https://arxiv.org/abs/1711.05415

[26] S. Zhou, T. Xiao, Y. Yang, D. Feng, Q. He, and W. He, "GeneGAN: Learning object transfiguration and attribute subspace from unpaired data," 2017, *arXiv:1705.04932*. [Online]. Available: https://arxiv.org/abs/1705.04932

[27] T. Xiao, J. Hong, and J. Ma, "Elegant: Exchanging latent encodings with GAN for transferring multiple face attributes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 168–184.

[28] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2Image: Conditional image generation from visual attributes," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 776–791.

[29] R. Liu, Y. Liu, X. Gong, X. Wang, and H. Li, "Conditional adversarial generative flow for controllable image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7992–8001.

[30] M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo, and S. Wen, "STGAN: A unified selective transfer network for arbitrary image attribute editing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3673–3682.

[31] G. Dorta, S. Vicente, N. D. F. Campbell, and I. Simpson, "The GAN that warped: Semantic attribute editing with unpaired data," 2018, *arXiv:1811.12784*. [Online]. Available: https://arxiv.org/abs/1811.12784

[32] W. Yin, Z. Liu, and C. C. Loy, "Instance-level facial attributes transfer with geometry-aware flow," 2018, *arXiv:1811.12670*. [Online]. Available: https://arxiv.org/abs/1811.12670

[33] Y.-C. Chen, X. Shen, Z. Lin, X. Lu, I.-M. Pao, and J. Jia, "Semantic component decomposition for face attribute manipulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9859–9867.

[34] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3730–3738.

[35] 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.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.

[38] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

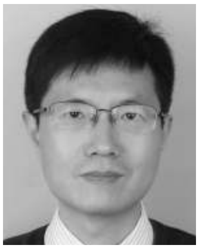
[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

RENTUO TAO received the B.E. degree from the Hefei University of Technology (HFUT), Hefei, China, in 2013. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China (USTC), Hefei. His research interests include deep generative models, machine learning, and computer vision.

ZIQIANG LI received the B.E. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2019, where he is currently pursuing the master's degree. His research interests include medical image segmentation, deep generative models, deep learning, and computer vision.

RENSHUAI TAO received the B.E. degree from Beihang University (BUAA), Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include machine learning and computer vision.



BIN LI (M'07) received the B.Sc. degree from the Hefei University of Technology, Hefei, China, in 1992, the M.Sc. degree from the Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, in 1995, and the Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, in 2001, where he is currently a Professor with the School of Information Science and Technology. He has authored or coauthored over 40 refereed publications. His current research interests include evolutionary computation, pattern recognition, and human–computer interaction. He is also the Founding Chair of the Hefei Chapter of the IEEE Computational Intelligence Society, a Counselor of the IEEE USTC Student Branch, a Senior Member of the Chinese Institute of Electronics (CIE), and a member of the Technical Committee of the Electronic Circuits and Systems Section of CIE.