Weakly Supervised Facial Attribute Manipulation via Deep Adversarial Network

Yilin Wang[1] Suhang Wang[1] Guojun Qi[2] Jiliang Tang[3] Baoxin Li[1]
[1]Department of Computer Science, Arizona State Univerity
[2]Laboratory for MAchine Perception and LEarning, University of Central Florida
[3]Department of Computer Science and Engineering, Michigan State University
{yilinwang,suhang.wang,baoxin.li}@asu.edu guojun.qi@ucf.edu tangjili@msu.edu

Abstract

Automatically manipulating facial attributes is challenging because it needs to modify the facial appearances, while keeping not only the person's identity but also the realism of the resultant images. Unlike the prior works on the facial attribute parsing, we aim at an inverse and more challenging problem called attribute manipulation by modifying a facial image in line with a reference facial attribute. Given a source input image and reference images with a target attribute, our goal is to generate a new image (i.e., target image) that not only possesses the new attribute but also keeps the same or similar content with the source image. In order to generate new facial attributes, we train a deep neural network with a combination of a perceptual content loss and two adversarial losses, which ensure the global consistency of the visual content while implementing the desired attributes often impacting on local pixels. The model automatically adjusts the visual attributes on facial appearances and keeps the edited images as realistic as possible. The evaluation shows that the proposed model can provide a unified solution to both local and global facial attribute manipulation such as expression change and hair style transfer. Moreover, we further demonstrate that the learned attribute discriminator can be used for attribute localization.

1. Introduction

Facial attributes describing various semantic aspects of facial images, such as "male," "beard," "smiling," have been extensively explored due to its wide applications to face recognition [2, 6, 17], expression parsing [23, 24], and facial image search [33, 14]. Facial attributes can be used in binary settings (i.e., whether or not a visual attribute exists) [17, 2] or in relative settings (i.e., stronger or weaker presence of attributes) [43, 26]. It has been shown [26, 14] that relative attributes are equally or more useful than binary attributes in zero-shot learning and image search.

The success of facial attributes in various applications lies in the representational power of these attributes in describing rich semantic variations of a person's look. A person may look dramatically different by changing their facial attributes to have different hair colors/styles with beard or no-beard. A person's facial image taken in childhood can be totally different from that taken in adulthood because the "Age" changes as one of important facial attributes. Thus, deliberately manipulating facial attributes is an important task for various applications. For example, it can help people design their fashion styles by showing how they look like under different facial attributes. In addition, it can also help facial image search and face recognition by providing more precise facial images of different ages.

Figure 1. Face attribute manipulation results. Top row is original image, bottom row is our results. From left to right, we changed the facial attribute from "small eye" to "large eye", "no beard" to "goatee beard", "no smile" to "smile" and "hair" to "bald". Please view these examples in color and zoom in for details.

However, manipulating facial attributes with existing image editing tools is still very challenging primarily for two reasons: (1) the majority of existing image editing tools ignore the realism of generated images, and thus cannot support image attribute manipulation automatically; (2) for most of us, learning a simple manipulation in computer-aided tools like the Photoshop will take a very long time, let alone changing multiple facial attributes simultaneously. Therefore, it is very important to investigate the problem of how to automatically generate authentic facial images

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00019

for a given person with different attributes. Figure 1 illustrates some results from the manipulation of various facial attributes. Note that only the facial attribute we want to manipulate changes without affecting the realism of the photo or the other parts of these images.

Automatically manipulating facial images to generate photo-realistic and high quality images is very challenging. Though there are a variety of methods developed for image modeling and generation, none of them can give satisfactory results for automatically manipulating facial attributes. The CNN-based models have been studied for face generation and 3D chair generation in [15, 5], but these methods need a large number of images for training and can only generate faces/chairs with different poses. In [42], a deep variational auto-encoder network was proposed to combine a reference image with the desired attribute to generate a target image. More recently, [4, 8, 28, 45] have shown visually impressive results with the generative adversarial neural network (GAN) by creating photo-realistic images. However, these methods cannot be effective in reality. The images generated by CNN-based methods, while impressive, still look less realistic (with artifacts and low resolution). On the other hand, the generative adversarial network typically starts from a random vector, allowing little semantic controls on the generated images. Therefore, these methods cannot be directly applied to user controlled semantic image manipulation. Recently Variational Autoencoder (VAE) [18] emerged as a powerful tool capable of generating images from latent representations. However, passing images through the encoder-decoder pipeline often generates low-quality images, and it cannot manipulate the generated images for desired attributes without changing other content.

To address theses challenges, we propose a novel deep generative model with (1) a feed-forward neural network to learn latent representations of an image instead of noise and we view the manipulation as a transformation from the source image to the target image; (2) a local discriminator which aims to recognize and manipulate the desired attribute area and; and (3) global discriminator for keeping image realism and high quality. Specifically, in order to modify the attribute, a pairwise attribute loss in the local discriminator is minimized to manipulate the original and generated attribute region. This pairwise attribute loss is implemented with a spatial transform network [11], which serves as an attribute detector to help the generator identify the desired attribute. It is worth noting that the ground truth of the target image is unavailable, and thus the model needs to perform attribute manipulation in a weakly supervised fashion through artifact suppression, while keeping the target image photo-realistic. The main contributions in this paper are summarized below.

- We propose a unified weakly-supervised framework for manipulating facial attributes. To this end, we propose a novel deep generative model, which combines a feed-forward neural network with two adversarial losses.
- We further demonstrate that the learned local pairwise discriminator, consisting of a spatial transform network and pairwise attribute loss network, is able to localize the most relevant image regions to manipulate visual attributes.
- Experimental results show impressive results on manipulating various facial attributes.

In the rest of the paper, we first review related works in Section 2. Then the proposed framework and the training algorithm are presented in Section 3. Experiment results are demonstrated in Section 4, with further discussions in Section 5.

2. Related Work

Visual attribute and attribute localization: Visual attributes which serve as a mid-level informative representation for visual analysis have been well studied in recent years. In early works [16, 17, 29, 6, 26, 44, 37, 38] attribute detection relied on the hand-crafted features such SIFT , GIST and HOG features with machine learning techniques[20, 10, 22]. Recently, [33, 35, 7, 32, 39, 40] deep learning and convolution neural network based attribute detection has achieved superior performance than hand craft based methods. Comparing to binary visual attribute, relative attribute detection is more challenging, because the pre-trained model may not be able to directly measure the strength of the attribute. In [41], authors proposed a spatial extent based methods for relative discovery. [35] introduces a more accurate relative attribute detector via pairwise siamese deep network. Compared with these methods, the goal of our model is to generate attributes on the images. Moreover, we demonstrate that the learned pairwise discriminator can also be used for accurate attribute localization.

Image editing and generation: Image editing has been extensively studied in human computer interaction and computer graphics. For example [19] provides a general solution to change the color of an image. More advanced editing such as image structure editing was proposed in [1]. However, most of current image editing methods focus on the low-level visual features, and more importantly, they fail to produce realistic images while editing the high level image content. Recently, there is a large body of research studies on image generation. [42] proposes an image generation method via deep auto encoder. Several CNN architectures have also been developed for image generation [25, 21, 5]. However, these methods need lots of labeled images to train CNNs. General adversarial network (GAN),

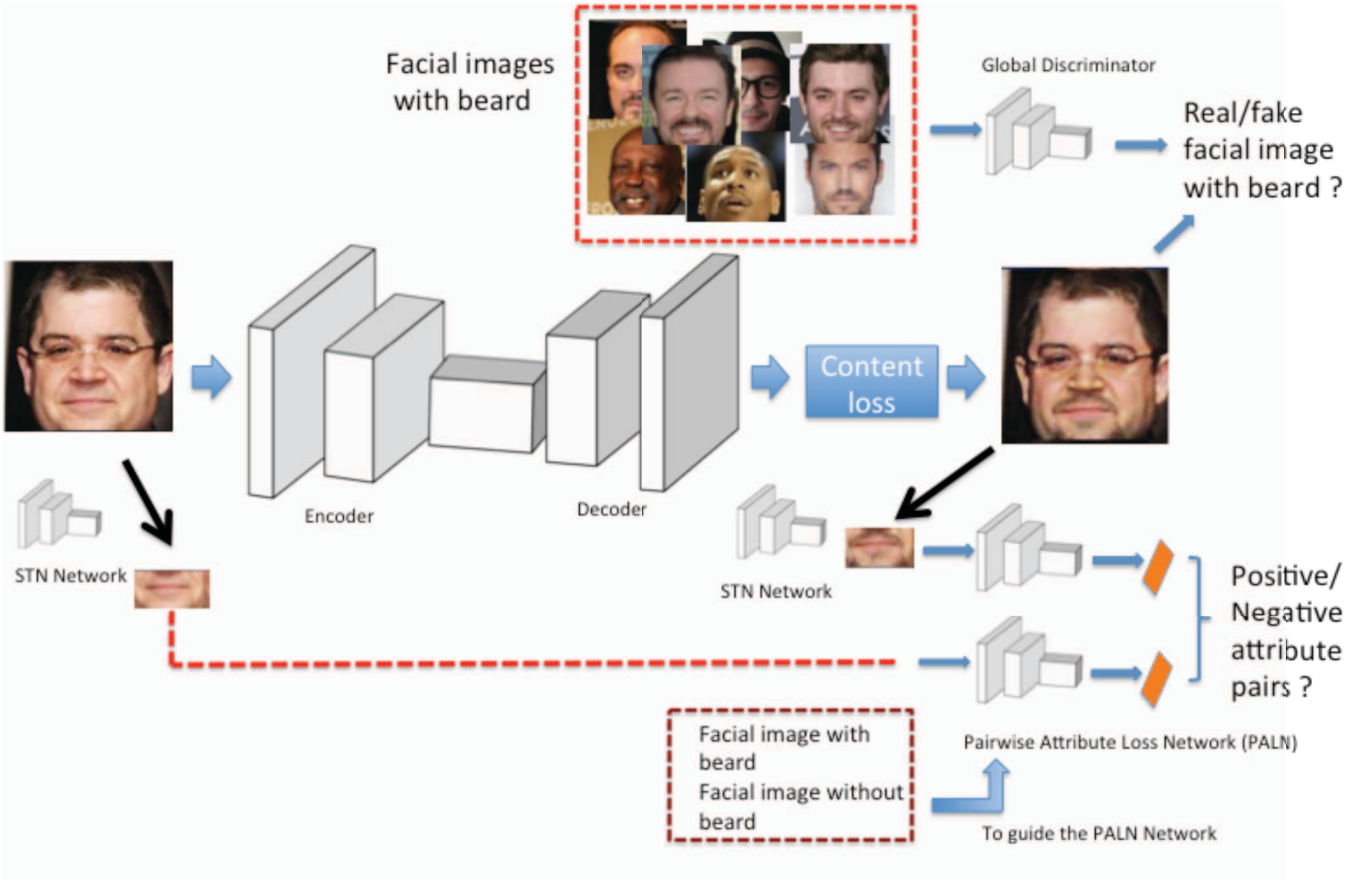

Figure 2. The architecture of the proposed method. It contains one generator and two discriminators. The reference images are those having a common desired facial attribute, e.g., "beard". The pairwsie attribute loss network is guided by training image and reference image pair or training image and generated image pair. More details can be seen in Sec3.2.

proposed by [8], aims to learn generative networks in a min-max fashion via a second adversarial network. The general process of GAN is that the discriminator tries to distinguish between real samples and generated samples, while the generator tries to fool the discriminator by generating more real-like images. In [4] and [28], authors suggested that higher quality image can be generated via Laplacian pyramid and CNN based features. Unfortunately, the GAN based methods [27, 8] usually start from random vectors and do not provide a way for user to control the semantics of the generated content. In [45], a user-image interactive way was proposed to control the image generation process. However, it only focuses on the shape and colors and cannot enhance the semantic visual attributes in the images.

3. The Proposed Method

In this section, we describe the proposed method for visual attribute manipulation. Given an input facial image, our goal is to generate a photo-realistic version with the same content but with a desired new attribute. Figure 2 shows the proposed network, which consists of one generator, one content loss and two adversarial losses. More details will be discussed in the following subsections.

3.1. Background

The GAN consists of two parts: (a) a generator neural network $G(z, \theta_g)$ that maps a random vector $z \in \mathcal{Z}$ to a natural image $x \in \mathcal{R}^{H \times W \times C}$, where $\mathcal{Z}$ denotes an embedding latent space; (b) a discriminator neural network $D(x, \theta_d)$ that predicts if an image is real or fake (generated image). The objective loss function of the GAN is to optimize G and D in a min-max [8] fashion such that G tries to fool D by generating images that look real while D tries to distinguish the fake from the real. For simplicity, we denote the generator and discriminator by $D(z)$ and $G(x)$, dropping out their parameters from their notations.

Given the nature of the GAN that it is designed to generate realistic images, we adopt it to manipulate the image attributes. However, the input of the GAN is a random vector $z \in \mathcal{Z}$ and we do not have much control over what image attributes and content would be created. On the contrary, our goal is to generate a realistic facial image with certain

Table 1. Architecture of the encoder and decoder for image generation.

Layer	Activation Size
Input Image	$128 \times 128 \times 3$
$11 \times 11 \times 32$ conv, pad 5, stride 1	$128 \times 128 \times 3$
$3 \times 3 \times 64$ conv, pad 1, stride 2	$128 \times 128 \times 32$
$3 \times 3 \times 64$ conv, pad 1, stride 2	$64 \times 64 \times 64$
$3 \times 3 \times 128$ conv, pad 1, stride 4	$8 \times 8 \times 64$
$3 \times 3 \times 256$ conv, pad 1, stride 2	$4 \times 4 \times 256$
Residual block 256 filters	$4 \times 4 \times 256$
Residual block 256 filters	$4 \times 4 \times 256$
$3 \times 3 \times 64$ dconv, pad 1, stride 2	$8 \times 8 \times 64$
$3 \times 3 \times 64$ dconv, pad 1, stride 4	$64 \times 64 \times 64$
$3 \times 3 \times 32$ dconv, pad 1, stride 2	$128 \times 128 \times 32$
$11 \times 11 \times 3$ dconv, pad 1, stride 1	$128 \times 128 \times 3$

attributes by manipulating a source image (see example in Figure 1). Thus, the GAN cannot be used directly for attribute manipulation.

3.2. The Generator

Here we discuss how to create desired attributes by manipulating an input image.

Generator: We assume that the image lies in a low-dimensional manifold. Thus, given a real image I_* with the feature vector x_*, we first train a feedfoward neural network P as the encoder projecting x_* to a low-dimensional space as $P(x_*, \theta_P)$, where θ_P is the parameter of P. We then add a decoder neural network G with $P(x_*, \theta_P)$ as input to generate the modified image. The goal of the encoder and the decoder is to generate an image with the similar content but having desired attribute. We will explain how to make the image realistic and possessing desired attributes in the next subsection. The objective function of training the encoder and the decoder can be written as follows,

$$G_{loss}(\theta_P, \theta_G) = \min_{\Theta} \sum_{i=1}^{n} \mathcal{L}_{content}(G(P(x_i, \theta_P), \theta_G), x_i) \tag{1}$$

where x_i denotes the ith image in the training set and $\Theta = \{\theta_P, \theta_G\}$ represents the parameters. The architectures of P and G are symmetric as summarized in Table 3.2.

Content loss: Minimizing the content loss $\mathcal{L}_{content}$ in the above objective guarantees the generated image has the same or similar content as the input image. The two images cannot be exactly the same since they are supposed to have different attributes. Thus, we encourage them to have similar feature representations computed by a CNN-based loss network rather than forcing them to be identical in the pixel domain. We choose the squared-error loss on the CNN feature representations, yielding a perceptual content loss. In our experiments, the loss network $\phi_l(I)$ is the feature map in the l-th layer of the 16-layer VGG network [34].

$$\mathcal{L}_{content} = \sum_{l=3}^{5} \frac{1}{C_l \times H_l \times W_l} \|\phi_l(I) - \phi_l(I)^*\|_2^2 \tag{2}$$

where C, H and W define the shape of the chosen feature map. As mentioned in [12], minimizing the feature reconstruction error in an auto-encoder encourages the perceptual similarity instead of pixel domain similarity.

3.3. The Discriminator

The generator can be trained directly if we have perfect attribute-edited images. However, such ground truth is impossible to obtain in practice. To solve this problem, we use a pairwise loss to enforce that the input and the generated images should have different attributes. So if the input image is of "no beard", the corresponding output image from the generator will have "beard". This forms an adversarial pair between the input and output ends, forming a novel Pairwise Attribute Loss Network (PALN) as illustrated in Figure 3. This adversarial network is contrary to the GAN where a pair of discriminator and generator are adversaries. This PALN avoids directly modeling target attributes, and thus we do not have to collect a large number of perfectly edited images with these attributes to train the network. Below we will discuss the details of this idea.

Local attribute discriminator: The local attribute discriminator, trained by pairwise loss, is introduced to generate images with desired attributes. Since the attribute is hard to model without sufficient training examples, a straightforward supervised method is prohibited. Instead, the proposed model consists of two parts for the local attribute discriminator: (1) Spatial Transform Network (STN), which detects the most relevant image regions to a visual attribute; and (2) Pairwise Attribute Loss Network (PALN), which predicts the pairwise label of two same attribute regions output from the STN. If the output of STN from an image pair has same attribute, we treat them with pairwise label 0, otherwise is 1. Note that, *these pairs are not required to come from the same person in training process, making it very flexible to train the model.*

(1) Spatial Transformer Network (STN). Intuitively, to manipulate visual attributes, we need to localize the regions relevant to the visual attributes. We choose the STN [11] for region localization due to its advantages of estimating translation, rotation, and warping without any human annotations. In our framework, we simplify the structure of STN , which only contains three blocks: a CNN transforming the input image to an affine matrix θ (three parameters including scaling, vertical translation and horizontal translation), a grid generator creating a set of sampling grids to find the relevant image patches, and a bilinear kernel producing the final output from the sampling grids. The network structure

Table 2. Architecture of the spatial transformer network.

Layer	Activation Size
Input Image	$128 \times 128 \times 3$
$11 \times 11 \times 32$ conv, pad 5, stride 2	$64 \times 64 \times 32$
$7 \times 7 \times 64$ conv, pad 1, stride 2	$32 \times 32 \times 64$
$3 \times 3 \times 128$ conv, pad 1, stride 1	$32 \times 32 \times 128$
$3 \times 3 \times 128$ conv, pad 1, stride 2	$16 \times 16 \times 128$
Fully Connected layer with 128 hidden units	128
Fully Connected lyaer with 3 hidden units	3

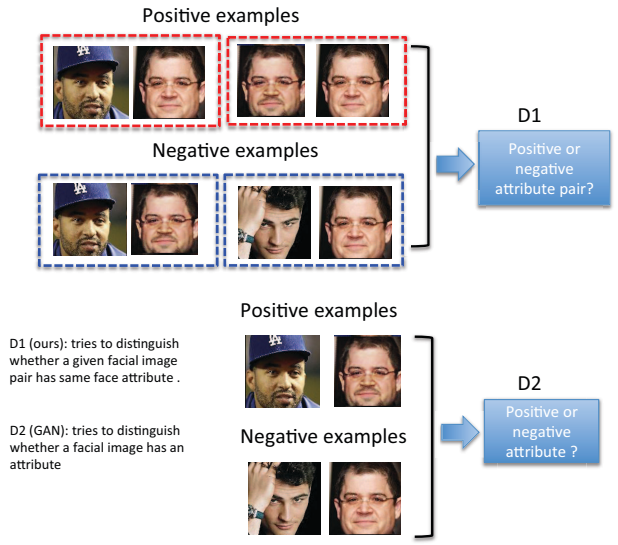

Figure 3. Difference between our discriminator and discriminator in GAN.

of STN used in our framework is shown in Table 2. The STN initially finds the object relevant regions and proceeds to search neighborhoods and coverages to find the attribute relevant image regions.

(2) Pairwise Attribute Loss Network(PALN). The PALN, as illustrated in Figure 3, takes a pair of STN output regions as input and generates a relative attribute label for the input pair, where 1 denotes that the input pair has different visual attributes, and 0 otherwise. Like the Siamese network [3], the pairwise attribute loss function of PALN computes the distances between two input feature vectors. The goal of PALN is to force the generator to produce desired attributes that are absent from the input images. This is implemented by (1) training PALN with real-world image pairs such that it is able to discriminate if they have the same attribute or not, and (2) guiding the training of the generator by penalizing G generating the wrong attributes.

For example, if we want to manipulate the attribute from "no beard" to "beard", we could use positive pairs (beard versus no beard) and negative pairs (both have "beard" or both have "no beard") to train STN and PALN. By feeding the generator with "no beard" images, the generator should produce an image with "beard". This is because the PALN will penalize the generator producing the image that do not flip the attribute (no beard) in the input image.

We use the following objective function to train the PALN,

$$\begin{aligned} D_{att}(I_1, I_2) = &- L \cdot log(D_R(I_1, I_2)) \\ &- (1 - L) \cdot log(1 - D_R(I_1, I_2)) \end{aligned} \tag{3}$$

where $D_R(I_1, I_2) = \|\phi(I_1) - \phi(I_2)\|_F^2$ represents Euclidean distance in a feature space.

Global discriminator: The global discriminator $\mathcal{D}_{global}$ determines whether an image is fake or not by encouraging the generated content to be semantically realistic. It also forces that the newly-generated attributes to be not only realistic, but also consistent with the surrounding contexts. In our framework, we define the cross entropy loss upon the output of a CNN. We make use of the structure in [28]. Specifically, the loss function is defined below,

$$\mathcal{D}_{global}(I) = -L \cdot log(D_p(I)) - (I - L) \cdot log(1 - D_P(I)) \tag{4}$$

where $\mathcal{D}_{global}$ represents the extracted CNN features. If I is a real image, $L = 1$, and $L = 0$ otherwise.

3.4. Objective function

As mentioned above, we introduce a content loss and two adversarial losses. With the generation content loss only, the generated image is the same as the original image and it tends to be blurry and smooth, because without groundtruth, the L_2 penalty encourages the output image to be similar to the input image to avoid large penalty. By using two discriminators, we employ the adversarial loss which is a reflection of how the generator can maximally fool the discriminator and how well the discriminator can distinguish between the real and the fake.

Putting the above together, we train the whole network from end to end simultaneously by the following objective function,

$$\begin{aligned} \min_{\{\theta_P, \theta_G\}} \max_{\{D_{global}, D_{att}\}} \sum_{i=1}^{N} &(G_{loss}(\theta_P, \theta_G) \\ &+ \lambda_1 D_{global} + \lambda_2 D_{att}) \end{aligned} \tag{5}$$

where λ_1 and λ_2 are the positive hyperparameters balancing the importance between different losses.

3.5. Network Training

Although the training process is scheduled in three parts, we train the network directly instead of using the curriculum strategy by gradually increasing the difficulty level and

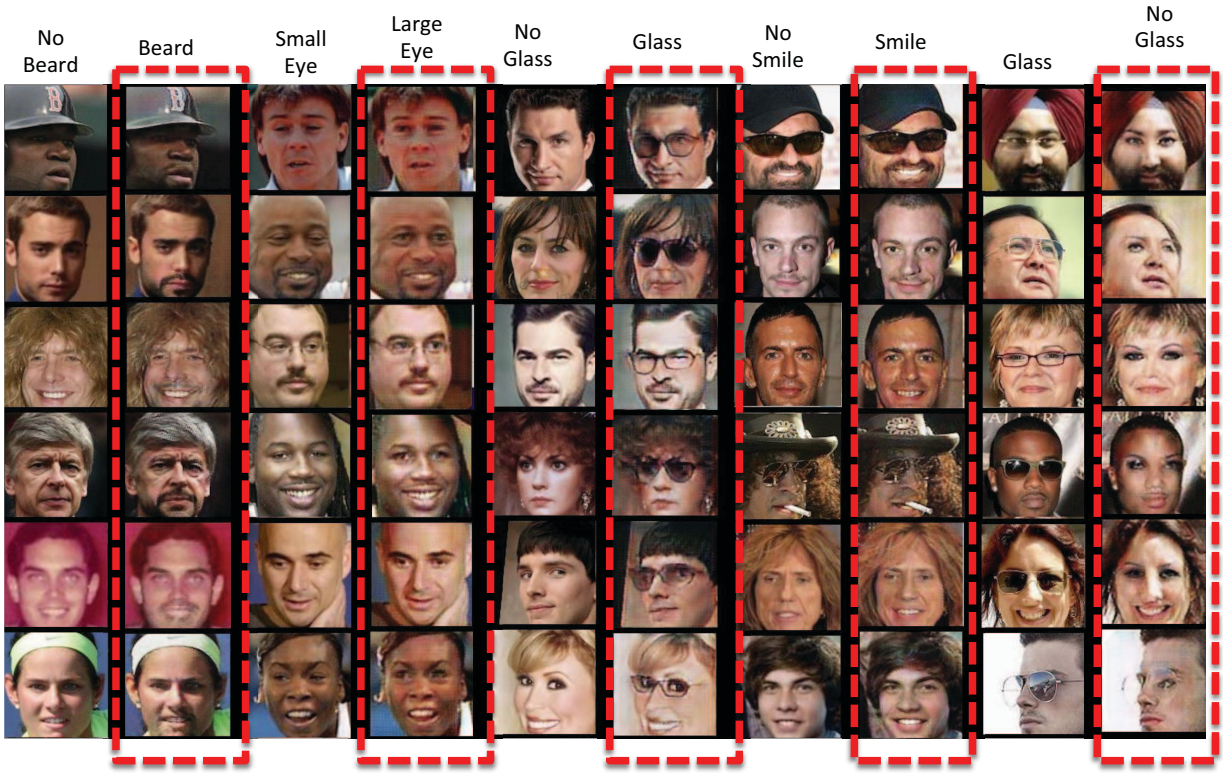

Figure 4. Local facial attribute manipulation on the Cleb A and the LFW dataset. For each sub-figure, the red bounding box mark the result from our method. Due to the space limit, we show the common attribute in CleA and LFW. Please view these examples in color and zoom in for details.

network scale. The transform network and the discriminator are alternately optimized by minimizing the objective in Eq 5. Here we apply the ADAM solver [13] to train the transform network and the discriminator with a learning rate of 0.0001. When training with the adversarial losses, we use a method similar to [28] to avoid the discriminator from becoming too strong at the beginning of the training process.

4. Experiments

In the experimental study, we aim at addressing the following questions: (1) How effective is the proposed framework in generating photo-realistic images with the desired attribute modified while the face identity kept the same; and (2) Can STN accurately locate the desired facial attribute such that PALN can manipulate it? We conduct qualitative study, subjective perceptual study and quantitative perceptual study on the generated images to answer the first question and visualize the attribute localization results to answer the second question. We begin by introducing the dataset.

4.1. Dataset

We use the ClebA dataset [23] and the Labeled Face in the Wild dataset (LFW) [9] to learn and evaluate our model. ClebA consists of 202,599 face images with 40 attributes and each face image is cropped and roughly aligned. LFW contains 13,143 images of faces with predicted annotations for 73 different attributes. All the images in these two dataset are aligned and the size of our generate image is set as 128×128. Similar to [36, 18], we test *smile, no_beard, wear_glass, wear_no_glass, hair,and hair_black* in ClebA and test *smile, no_beard, beard, eyes_open, mouth_open* in LFW. Our goal is to change these attribute thus the target negative attribute are *no smile, beard, wear_no_glass, wear_glass, bald, hair_blond* (Similar for LFW attributes). These attributes were chosen because it would be plausible for a single person to be changed into having each of those attributes. Note that there are no ground truth manipulated images for training the transformation networks since the images in the datasets are all obtained from the Internet.

4.2. Facial Attribute Manipulation

In order to more holistically evaluate the visual quality of our results, we employ two tactics. First, we present our result on local facial attribute and global facial attribute and visually compare our results with the recent state-of-the-art method DFI [36]. Note that although VAE-GAN [18] is able to change the facial attribute, it often fails to preserve the identity of a facial image and DFI already outperforms

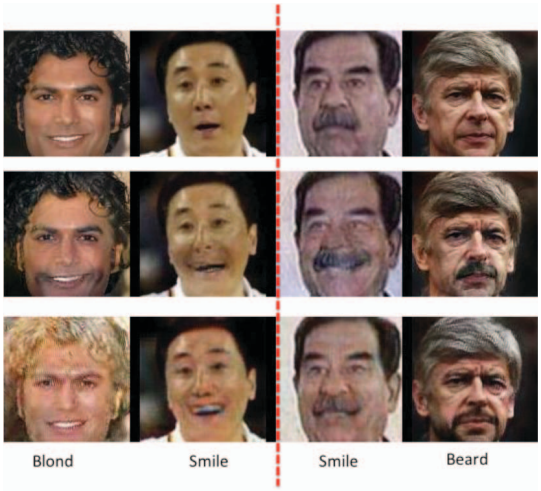

Figure 5. Visual comparison between the proposed method and DFI. Top row is the original image, second row is results from DRI and bottom row is our result. Please view these examples in color.

VAE-GAN dramatically [36]. Therefore, we choose DFI as a competitive baseline. Second, we run perceptual study. For graphics problems like photo generation, plausibility to a human observer is often the ultimate goal. Therefore, we employ Amazon Mechanical Turk, generated image reconstruction error and face recognition accuracy loss as metrics.

Qualitative Results.: Based on the area of the modified region, we present the modification of local facial attribute and global facial attribute separately in Figure 4 and Figure 6. For each tested image, our generated results is marked with a red bounding box. We show typical example with non-frontal face, e.g., examples on "glass-wear" and "glass removal", low-resolution images, e.g., examples on "adding beard" and "removing hair", and occlusion, e.g., cigarette and shadow. From the results, we can be observed that our results are consistent and realistic regardless different attribute manipulation. In Figure 5, we compare the results with DFI. Due to space limits, we choose two good cases and two bad cases from DFI and compare them with our results. Although, DFI often produce the right attribute, it does not preserve the photo-realism. This is because DFI is an interpolation approach based on images with similar attributes in the feature spaces. Therefore, when the nearest images in the space is far away from the source image, the interpolated image will lose the details in the original image. Note that , *DFI has post-processing to remove the artifacts, while ours does not need to have.*

Subjective Perceptual Study. The judgment of facial image manipulation is inherently subjective. In order to obtain an objective comparison among our model and DFI, we conducted a blind perceptual study with Amazon Mechanical Turk workers. Turkers were presented with a series of

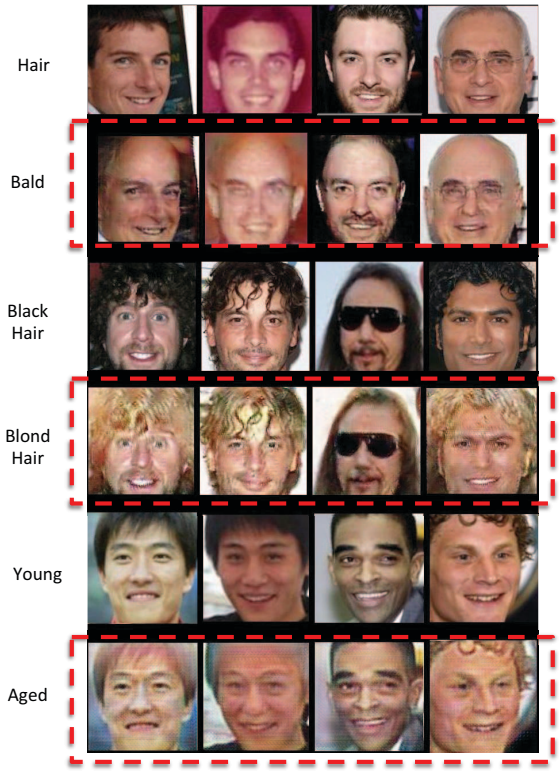

Figure 6. Face global attribute manipulation on Cleb A and LFW dataset. For each sub-figure, the red bounding box mark the result from our method. Please view these examples in color.

images and generated image pairs, which contains at least one image from our method and another one from DFI. On each trial, each image appeared for 60 seconds, after which the images disappeared and Turkers need to respond as to which was more like a real image. The results on 24 images with all the attribute manipulation totally 500 attribute pairs. We collect at least 4 judgments on each pair and find that our method is preferred to DFI with a ratio of 23:15. The most 3 preferred attributes of our method is: "beard", "wear glass" and "smile".

Quantitative Perceptual Study. In addition to AMT comparison, we want to evaluate what extent the facial information can be preserved through the modification. Therefore, we calculate three metrics on the generated images. The first one is peak signal to noise ratio (PSNR), which directly measures the pixel difference. The second is structure similarity index (SSIM), which estimates the structure distortion of two images. Lastly, we use identity distance measured by pre-trained faceNet [31] to determine high-level similarity of two faces. These three metrics are computed between the generated results and the original face image. Note that, some global attributes, e.g., Aged, aim to change whole facial appearance which is not suitable for perceptual study. Thus, we only evaluate the results from local facial attributes manipulation.

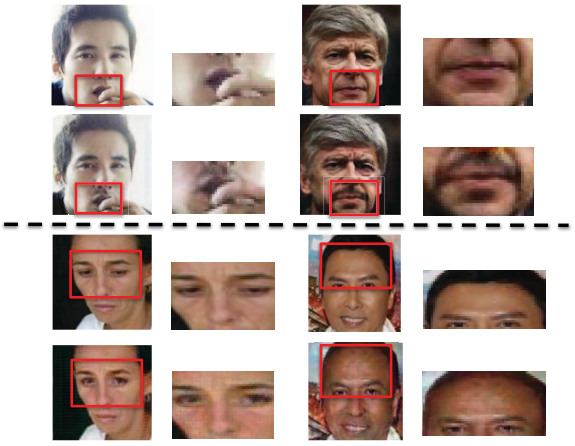

Figure 7. Visualization of attribute localization. Top row is original source image and bottom row is generated image. From the results, we can observe that the STN can locate the relevant attribute region.

To evaluate reconstruction error, we first mask 25% pixels that are related to attribute region on both original image and generated image. Then using PSNR and SSIM to evaluate the rest of the image. Note that the mask region of both images are exactly the same and won't affect the results on rest region. The average PSNR and SSIM for DFI generated images are 22.15 and 0.78, while ours is 24.20 and 0.82. These results demonstrate that our manipulation results have less impact on the pixels that are not related to the target facial attribute.

Face recognition can partly reveal the ability of preserving the identity information for the network. In order to test how much identity information can be preserved between our method and DFI, we evaluate the generated results for face recognition. We first randomly pick 50 identities from ClebA and LFW receptively. Then we randomly split the dataset into two balanced set named "probe" and "gallery". Given a modified image from "probe", the goal is to find an example of the same identity from the "gallery". The split is randomly and each identity has same number of images in each set. After 10 round tests, the top 5 accuracy of our method is 78.7% and DFI is 64.3%. It demonstrates that the importance of pairwise local attribute loss for identity preserving.

Attribute localization The STN network in our framework aims to locate the facial attributes. Figure 7 visualizes the attribute localization results on attribute "beard", "small eye" and "hair". We can see that our network correctly detect the relevant regions. However, similar to other studies on attribute detection [35, 41], the image scale has to be defined empirically as scaling is more sensitive and can transform the image dramatically. In our work, we fix the bounding box size with 40 by 60 and the image transform scale is initialized as 1/3 of the image size for local attribute except eye-related attributes and 2/3 for rest attributes.

Application Since our model is able to generate semantically plausible and visually pleasing facial attributes, the directly application of our algorithm is augmenting the existing attribute dataset via our synthetic image pairs. Data augmentation for neural network usually adopts flipping, cropping or scaling, the use of synthetic images as training data has been explored to a limited extent. Therefore, we conduct a experiment with the comparison to prior relative attribute method [26]. We use train-test split in LFW-10 dataset [30] with the three setting: **real set**: the original 600 training image pairs, **mixed set**: 75% real training image pair and 25% generated synthetic image pairs; and **augmented set**: the original training pairs with extra 600 automatically generated synthetic pairs. [1] The same rankSVM [26] is trained on these three different sets and tested on same test set. The results are real vs mixed vs augmented as 69.79% vs 71.30% vs 76.70% on five facial attributes named: smile, eyes open, bold, beard and visible forehead. The relative 7% gains demonstrate that our semantic data augmentation can help the existing methods by generating unlimited data samples.

5. Conclusion

We presented a unified deep adversarial network for facial attribute manipulation. The framework is able to produce higher quality facial images than existing state-of-the-arts. To train the network, we employ one perceptual content loss and two adversarial losses. Considering that, when compared to existing method, the proposed does not contain any post processing and result enhancement neural network, it suggests that the proposed framework can serve as a highly competitive baseline for aligned facial image manipulation.

6. Acknowledgment

Yilin and Baoxin were supported in part by ONR grants. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ONR. Guojun Qi was partly supported by NSF grant #1704309 and IARPA grant #D17PC00345. Jiliang Tang was supported by the National Science Foundation (NSF) under grant number IIS-1714741 and IIS-1715940.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG*, 28(3):24, 2009.

[1] Note that the synthetic pairs can be obtained from any face image and we use the images from clebA.

[2] L. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *CVPR*, pages 1543–1550. IEEE, 2011.

[3] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *(CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[4] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[5] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015.

[6] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, pages 3474–3481. IEEE, 2012.

[7] V. Escorcia, J. C. Niebles, and B. Ghanem. On the relationship between visual attributes and convolutional networks. In *CVPR*, pages 1256–1264. IEEE, 2015.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[9] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report.

[10] S. Huang, K. S. Candan, and M. L. Sapino. Bicp: Block-incremental cp decomposition with update sensitive refinement. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1221–1230. ACM, 2016.

[11] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[12] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016.

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, pages 2973–2980. IEEE, 2012.

[15] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.

[16] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, pages 365–372. IEEE, 2009.

[17] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958. IEEE, 2009.

[18] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *Proceedings of The 33rd ICML*, pages 1558–1566, 2016.

[19] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 689–694. ACM, 2004.

[20] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2017.

[21] M. Li, W. Zuo, and D. Zhang. Deep identity-aware transfer of facial attributes. *arXiv preprint arXiv:1610.05586*, 2016.

[22] X. Li, S. Huang, K. S. Candan, and M. L. Sapino. Focusing decomposition accuracy by personalizing tensor decomposition (ptd). In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 689–698. ACM, 2014.

[23] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[24] P. Luo, X. Wang, and X. Tang. Hierarchical face parsing via deep learning. In *CVPR 2012*, pages 2480–2487. IEEE, 2012.

[25] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[26] D. Parikh and K. Grauman. Relative attributes. In *2011 ICCV*, pages 503–510. IEEE, 2011.

[27] G.-J. Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *arXiv preprint arXiv:1701.06264*, 2017.

[28] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[29] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. In *ECCV*, pages 876–889. Springer, 2012.

[30] R. N. Sandeep, Y. Verma, and C. Jawahar. Relative parts: Distinctive parts for learning relative attributes. In *CVPR*, pages 3614–3621, 2014.

[31] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015.

[32] S. Shankar, V. K. Garg, and R. Cipolla. Deep-carving: Discovering visual attributes by carving deep neural nets. In *CVPR*, pages 3403–3412, 2015.

[33] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 801–808. IEEE, 2011.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] K. K. Singh and Y. J. Lee. End-to-end localization and ranking for relative attributes. In *European Conference on Computer Vision*, pages 753–769. Springer, 2016.

[36] P. Upchurch, J. Gardner, K. Bala, R. Pless, N. Snavely, and K. Weinberger. Deep feature interpolation for image content changes. *arXiv preprint arXiv:1611.05507*, 2016.

[37] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*,

pages 391–400. International World Wide Web Conferences Steering Committee, 2017.

[38] Y. Wang, Y. Hu, S. Kambhampati, and B. Li. Inferring sentiment from web images with joint inference on visual and social cues: A regulated matrix factorization approach. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[39] Y. Wang, S. Wang, J. Tang, H. Liu, and B. Li. Ppp: Joint pointwise and pairwise image label prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6005–6013, 2016.

[40] Y. Wang, S. Wang, J. Tang, G.-J. Qi, H. Liu, and B. Li. Clare: A joint approach to label classification and tag recommendation. In *AAAI*, pages 210–216, 2017.

[41] F. Xiao and Y. Jae Lee. Discovering the spatial extent of relative attributes. In *CVPR*, pages 1458–1466, 2015.

[42] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. *arXiv preprint arXiv:1512.00570*, 2015.

[43] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, pages 192–199, 2014.

[44] Y. Zhou and J. Luo. A practical method for counting arbitrary target objects in arbitrary scenes. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[45] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.