

RelGAN: Multi-Domain Image-to-Image Translation via Relative Attributes

Po-Wei Wu¹ Yu-Jing Lin¹ Che-Han Chang² Edward Y. Chang^{2,3} Shih-Wei Liao¹

¹National Taiwan University ²HTC Research & Healthcare ³Stanford University

maya6282@gmail.com r06922068@ntu.edu.tw chehan.chang@htc.com echang@cs.stanford.edu liao@csie.ntu.edu.tw

Abstract

Multi-domain image-to-image translation has gained increasing attention recently. Previous methods take an image and some target attributes as inputs and generate an output image with the desired attributes. However, such methods have two limitations. First, these methods assume binary-valued attributes and thus cannot yield satisfactory results for fine-grained control. Second, these methods require specifying the entire set of target attributes, even if most of the attributes would not be changed. To address these limitations, we propose RelGAN, a new method for multi-domain image-to-image translation. The key idea is to use relative attributes, which describes the desired change on selected attributes. Our method is capable of modifying images by changing particular attributes of interest in a continuous manner while preserving the other attributes. Experimental results demonstrate both the quantitative and qualitative effectiveness of our method on the tasks of facial attribute transfer and interpolation.

1. Introduction

Multi-domain image-to-image translation aims to translate an image from one domain into another. A domain is characterized by a set of attributes, where each attribute is a meaningful property of an image. Recently, this image-to-image translation problem received considerable attention following the emergence of generative adversarial networks (GANs) [1] and its conditional variants [2]. While most existing methods [3, 4, 5, 6] focus on image-to-image translation between two domains, several multi-domain methods are proposed recently [7, 8, 9], which are capable of changing multiple attributes simultaneously. For example, in the application of facial attribute editing, one can change hair color and expression simultaneously.

Despite the impressive results of recent multi-domain methods [7, 8], they have two limitations. First, these methods assume binary attributes and therefore are not designed for attribute interpolation. Although we can feed real-valued attributes into their generators, we found that their

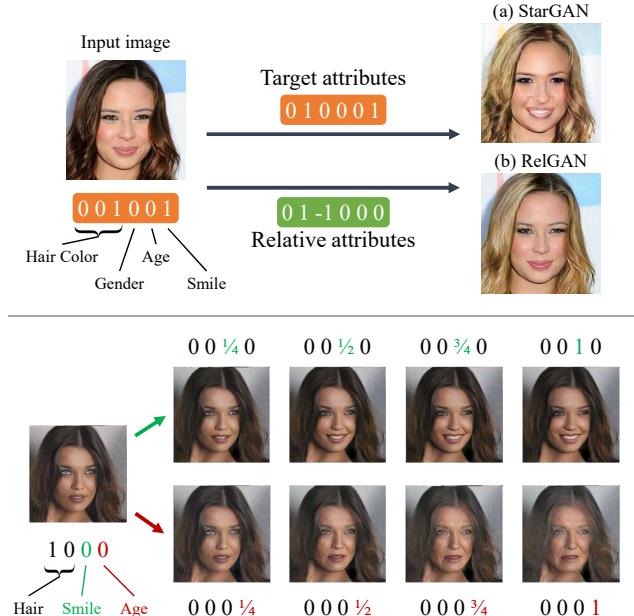


Figure 1. Top: Comparing facial attribute transfer via relative and target attributes. (a) Existing target-attribute-based methods do not know whether each attribute is required to change or not, thus could over-emphasize some attributes. In this example, StarGAN changes the hair color but strengthens the degree of smile. (b) RelGAN only modifies the hair color and preserves the other attributes (including smile) because their relative attributes are zero. Bottom: By adjusting the relative attributes in a continuous manner, RelGAN provides a realistic interpolation between before and after attribute transfer.

interpolation quality is unsatisfactory because their models trained on binary-valued attributes. (Our model remedies this shortcoming by training on real-valued relative attributes with additional discriminators.) A smooth and realistic interpolation between before and after editing is important because it enables fine-grained control over the strength of each attribute (e.g., the percentage of brown vs. blonde hair color, or the degree of smile/happiness).

Second, these methods require a complete attribute representation to specify the target domain, even if only a subset of attributes is manipulated. In other words, a user has to

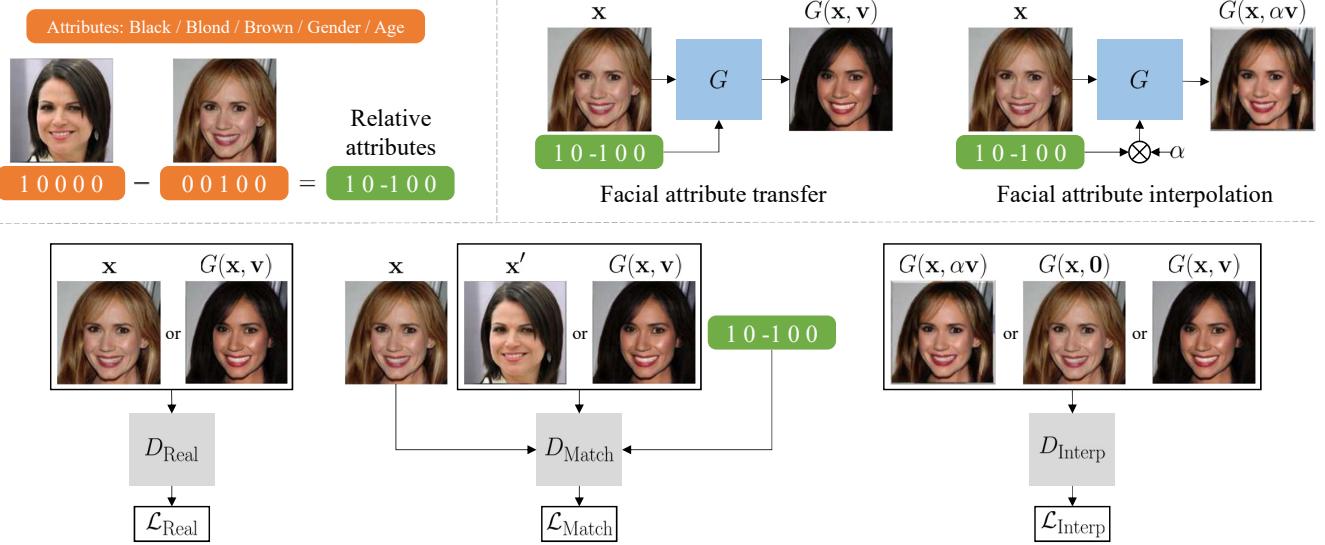


Figure 2. **RelGAN.** Our model consists of a single generator G and three discriminators $D = \{D_{\text{Real}}, D_{\text{Match}}, D_{\text{Interp}}\}$. G conditions on an input image and relative attributes (top left), and performs facial attribute transfer or interpolation (top right). During training, G aims to fool the following three discriminators (bottom): D_{Real} tries to distinguish between real images and generated images. D_{Match} aims to distinguish between real triplets and generated/wrong triplets. D_{Interp} tries to predict the degree of interpolation.

not only set the attributes of interest to the desired values but also identify the values of the unchanged attributes from the input image. This poses a challenge for fine-grained control, because a user does not know the underlying real value of each unchanged attribute.

To overcome these limitations, our key idea is that, unlike previous methods which take as input a pair $(\mathbf{x}, \hat{\mathbf{a}})$ of the original image \mathbf{x} and target attributes $\hat{\mathbf{a}}$, we take (\mathbf{x}, \mathbf{v}) , where \mathbf{v} is the *relative attributes* defined as the difference between the original attributes \mathbf{a} and the target ones $\hat{\mathbf{a}}$, i.e., $\mathbf{v} \triangleq \hat{\mathbf{a}} - \mathbf{a}$. The values of the relative attributes directly encode how much each attribute is required to change. In particular, non-zero values correspond to attributes of interest, while zero values correspond to unchanged attributes. Figure 1 illustrates our method with examples of facial attribute transfer and interpolation.

In this paper, we propose a relative-attribute-based method, dubbed RelGAN, for multi-domain image-to-image translation. RelGAN consists of a single generator G and three discriminators $D = \{D_{\text{Real}}, D_{\text{Match}}, D_{\text{Interp}}\}$, which are respectively responsible for guiding G to learn to generate (1) realistic images, (2) accurate translations in terms of the relative attributes, and (3) realistic interpolations. Figure 2 provides an overview of RelGAN.

Our contributions can be summarized as follows:

1. We propose RelGAN, a relative-attribute-based method for multi-domain image-to-image translation. RelGAN is based on the change of each attribute, and avoids the need to know the full attributes of an input image.
2. To learn a generator conditioned on the relative attributes,

we propose a matching-aware discriminator that determines whether an input-output pair matches the relative attributes.

3. We propose an interpolation discriminator to improve the interpolation quality.
4. We empirically demonstrate the effectiveness of RelGAN on facial attribute transfer and interpolation. Experimental results show that RelGAN achieves better results than state-of-the-art methods.

2. Related Work

We review works most related to ours and focus on conditional image generation and facial attribute transfer. Generative adversarial networks (GANs) [1] are powerful unsupervised generative models that have gained significant attention in recent years. Conditional generative adversarial networks (cGANs) [2] extend GANs by conditioning both the generator and the discriminator on additional information.

Text-to-image synthesis and image-to-image translation can be treated as a cGAN that conditions on text and image, respectively. For text-to-image synthesis, Reed et al. [10] proposed a matching-aware discriminator to improve the quality of generated images. Inspired by this work, we propose a matching-aware conditional discriminator. StackGAN++ [11] uses a combination of an unconditional and a conditional loss as its adversarial loss. For image-to-image translation, pix2pix [3] is a supervised approach based on cGANs. To alleviate the problem of acquiring paired data for supervised learning, unpaired image-to-image transla-

tion methods [4, 5, 6, 12] have recently received increasing attention. CycleGAN [5], the most representative method, learns two generative models and regularizes them by the cycle consistency loss.

Recent methods for facial attribute transfer [13, 14, 7, 8, 9, 15] formulate the problem as unpaired multi-domain image-to-image translation. IcGAN [13] trains a cGAN and an encoder, and combines them into a single model that allows manipulating multiple attributes. StarGAN [7] uses a single generator that takes as input an image and the target attributes to perform multi-domain image translation. AttGAN [8], similar to StarGAN, performs facial attribute transfer based on the target attributes. However, AttGAN uses an encoder-decoder architecture and treats the attribute information as a part of the latent representation, which is similar to IcGAN. ModularGAN [9] proposes a modular architecture consisting of several reusable and composable modules. GANimation [15] trains its model on facial images with real-valued attribute labels and thus can achieve impressive results on facial expression interpolation.

StarGAN [7] and AttGAN [8] are two representative methods in multi-domain image-to-image translation. RelGAN is fundamentally different from them in three aspects. First, RelGAN employs a relative-attribute-based formulation rather than a target-attribute-based formulation. Second, both StarGAN and AttGAN adopt auxiliary classifiers to guide the learning of image translation, while RelGAN’s generator is guided by the proposed matching-aware discriminator, whose design follows the concept of conditional GANs [2] and is tailored for relative attributes. Third, we take a step towards continuous manipulation by incorporating an interpolation discriminator into our framework.

3. Method

In this paper, we consider that a domain is characterized by an n -dimensional attribute vector $\mathbf{a} = [a^{(1)}, a^{(2)}, \dots, a^{(n)}]^T$, where each attribute $a^{(i)}$ is a meaningful property of a facial image, such as age, gender, or hair color. Our goal is to translate an input image \mathbf{x} into an output image \mathbf{y} such that \mathbf{y} looks realistic and has the target attributes, where some user-specified attributes are different from the original ones while the other attributes remain the same. To this end, we propose to learn a mapping function $(\mathbf{x}, \mathbf{v}) \mapsto \mathbf{y}$, where \mathbf{v} is the relative attribute vector that represents the desired change of attributes. Figure 2 gives an overview of RelGAN. In the following subsections, we first introduce relative attributes, then we describe the components of the RelGAN model.

3.1. Relative Attributes

Consider an image \mathbf{x} , its attribute vector \mathbf{a} as the original domain, and the target attribute vector $\hat{\mathbf{a}}$ as the target

domain. Both \mathbf{a} and $\hat{\mathbf{a}}$ are n -dimensional vectors. We define the relative attribute vector between \mathbf{a} and $\hat{\mathbf{a}}$ as

$$\mathbf{v} \triangleq \hat{\mathbf{a}} - \mathbf{a}, \quad (1)$$

which naturally represents the desired change of attributes in modifying the input image \mathbf{x} into the output image \mathbf{y} .

We argue that expressing the user’s editing requirement by the relative attribute representation is straightforward and intuitive. For example, if image attributes are binary-valued (0 or 1), the corresponding relative attribute representation is three-valued ($-1, 0, 1$), where each value corresponds to a user’s action to a binary attribute: turn on (+1), turn off (-1), or unchanged (0). From this example, we can see that relative attributes encode the user requirement and have an intuitive meaning.

Next, facial attribute interpolation via relative attributes is rather straightforward: to perform an interpolation between \mathbf{x} and $G(\mathbf{x}, \mathbf{v})$, we simply apply $G(\mathbf{x}, \alpha\mathbf{v})$, where $\alpha \in [0, 1]$ is an interpolation coefficient.

3.2. Adversarial Loss

We apply adversarial loss [1] to make the generated images indistinguishable from the real images. The adversarial loss can be written as:

$$\min_G \max_{D_{\text{Real}}} \mathcal{L}_{\text{Real}} = \mathbb{E}_{\mathbf{x}} [\log D_{\text{Real}}(\mathbf{x})] + \mathbb{E}_{\mathbf{x}, \mathbf{v}} [\log(1 - D_{\text{Real}}(G(\mathbf{x}, \mathbf{v})))], \quad (2)$$

where the generator G tries to generate images that look realistic. The discriminator D_{Real} is unconditional and aims to distinguish between the real images and the generated images.

3.3. Conditional Adversarial Loss

We require not only that the output image $G(\mathbf{x}, \mathbf{v})$ should look realistic, but also that the difference between \mathbf{x} and $G(\mathbf{x}, \mathbf{v})$ should match the relative attributes \mathbf{v} . To achieve this requirement, we adopt the concept of conditional GANs [2] and introduce a conditional discriminator D_{Match} that takes as inputs an image and the conditional variables, which is the pair (\mathbf{x}, \mathbf{v}) . The conditional adversarial loss can be written as:

$$\min_G \max_{D_{\text{Match}}} \mathcal{L}_{\text{Match}} = \mathbb{E}_{\mathbf{x}, \mathbf{v}, \mathbf{x}'} [\log D_{\text{Match}}(\mathbf{x}, \mathbf{v}, \mathbf{x}')] + \mathbb{E}_{\mathbf{x}, \mathbf{v}} [\log(1 - D_{\text{Match}}(\mathbf{x}, \mathbf{v}, G(\mathbf{x}, \mathbf{v})))]. \quad (3)$$

From this equation, we can see that D_{Match} takes a triplet as input. In particular, D_{Match} aims to distinguish between two types of triplets: real triplets $(\mathbf{x}, \mathbf{v}, \mathbf{x}')$ and fake triplets $(\mathbf{x}, \mathbf{v}, G(\mathbf{x}, \mathbf{v}))$. A real triplet $(\mathbf{x}, \mathbf{v}, \mathbf{x}')$ is comprised of two real images $(\mathbf{x}, \mathbf{x}')$ and the relative attribute vector $\mathbf{v} = \mathbf{a}' - \mathbf{a}$, where \mathbf{a}' and \mathbf{a} are the attribute vector of \mathbf{x}' and \mathbf{x} respectively. Here, we would like to emphasize that

our training data is unpaired, i.e., \mathbf{x} and \mathbf{x}' are of *different identities with different attributes*.

Inspired by the matching-aware discriminator [10], we propose to incorporate a third type of triplets: *wrong triplet*, which consists of two real images with mismatched relative attributes. By adding wrong triplets, D_{Match} tries to classify the real triplets as +1 (real and matched) while both the fake and the wrong triplets as -1 (fake or mismatched). In particular, we create wrong triplets using the following simple procedure: given a real triplet expressed by $(\mathbf{x}, \mathbf{a}' - \mathbf{a}, \mathbf{x}')$, we replace one of these four variables by a new one to create a wrong triplet. By doing so, we obtain four different wrong triplets. Algorithm 1 shows the pseudo-code of our conditional adversarial loss.

Algorithm 1 Conditional adversarial loss

```

1: function MATCH\_LOSS( $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ )
2:    $\mathbf{v}_{12}, \mathbf{v}_{32}, \mathbf{v}_{13} \leftarrow \mathbf{a}_2 - \mathbf{a}_1, \mathbf{a}_2 - \mathbf{a}_3, \mathbf{a}_3 - \mathbf{a}_1$ 
3:    $s_r \leftarrow D_{\text{Match}}(\mathbf{x}_1, \mathbf{v}_{12}, \mathbf{x}_2)$  {real triplet}
4:    $s_f \leftarrow D_{\text{Match}}(\mathbf{x}_1, \mathbf{v}_{12}, G(\mathbf{x}_1, \mathbf{v}_{12}))$  {fake triplet}
5:    $s_{w_1} \leftarrow D_{\text{Match}}(\mathbf{x}_3, \mathbf{v}_{12}, \mathbf{x}_2)$  {wrong triplet}
6:    $s_{w_2} \leftarrow D_{\text{Match}}(\mathbf{x}_1, \mathbf{v}_{32}, \mathbf{x}_2)$  {wrong triplet}
7:    $s_{w_3} \leftarrow D_{\text{Match}}(\mathbf{x}_1, \mathbf{v}_{13}, \mathbf{x}_2)$  {wrong triplet}
8:    $s_{w_4} \leftarrow D_{\text{Match}}(\mathbf{x}_1, \mathbf{v}_{12}, \mathbf{x}_3)$  {wrong triplet}
9:    $\mathcal{L}_{\text{Match}}^D \leftarrow (s_r - 1)^2 + s_f^2 + \sum_{i=1}^4 s_{w_i}^2$ 
10:   $\mathcal{L}_{\text{Match}}^G \leftarrow (s_f - 1)^2$ 
11:  return  $\mathcal{L}_{\text{Match}}^D, \mathcal{L}_{\text{Match}}^G$ 

```

3.4. Reconstruction Loss

By minimizing the unconditional and the conditional adversarial loss, G is trained to generate an output image $G(\mathbf{x}, \mathbf{v})$ such that $G(\mathbf{x}, \mathbf{v})$ looks realistic and the difference between \mathbf{x} and $G(\mathbf{x}, \mathbf{v})$ matches the relative attributes \mathbf{v} . However, there is no guarantee that G only modifies those attribute-related contents while preserves all the other aspects from a low level (such as background appearance) to a high level (such as the identity of a facial image). To alleviate this problem, we propose a cycle-reconstruction loss and a self-reconstruction loss to regularize our generator.

Cycle-reconstruction loss. We adopt the concept of cycle consistency [5] and require that $G(:, \mathbf{v})$ and $G(:, -\mathbf{v})$ should be the inverse of each other. Our cycle-reconstruction loss is written as

$$\min_G \mathcal{L}_{\text{Cycle}} = \mathbb{E}_{\mathbf{x}, \mathbf{v}} [\|G(G(\mathbf{x}, \mathbf{v}), -\mathbf{v}) - \mathbf{x}\|_1]. \quad (4)$$

Self-reconstruction loss. When the relative attribute vector is a zero vector $\mathbf{0}$, which means that no attribute is changed, the output image $G(\mathbf{x}, \mathbf{0})$ should be as close as possible to \mathbf{x} . To this end, we define the self-reconstruction loss as:

$$\min_G \mathcal{L}_{\text{Self}} = \mathbb{E}_{\mathbf{x}} [\|G(\mathbf{x}, \mathbf{0}) - \mathbf{x}\|_1], \quad (5)$$

where G degenerates into an auto-encoder and tries to reconstruct \mathbf{x} itself. We use L1 norm in both reconstruction losses.

3.5. Interpolation Loss

Our generator interpolates between an image \mathbf{x} and its translated one $G(\mathbf{x}, \mathbf{v})$ via $G(\mathbf{x}, \alpha\mathbf{v})$, where α is an interpolation coefficient. To achieve a high-quality interpolation, we encourage the interpolated images $G(\mathbf{x}, \alpha\mathbf{v})$ to appear realistic. Specifically, inspired by [16], we propose a regularizer that aims to make $G(\mathbf{x}, \alpha\mathbf{v})$ indistinguishable from the non-interpolated output images, i.e., $G(\mathbf{x}, \mathbf{0})$ and $G(\mathbf{x}, \mathbf{v})$. To this end, we introduce our third discriminator D_{Interp} to compete with our generator G . The goal of D_{Interp} is to take an generated image as input and predict its *degree of interpolation* $\hat{\alpha}$, which is defined as $\hat{\alpha} = \min(\alpha, 1 - \alpha)$, where $\hat{\alpha} = 0$ means no interpolation and $\hat{\alpha} = 0.5$ means maximum interpolation. By predicting $\hat{\alpha}$, we resolve the ambiguity between α and $1 - \alpha$.

The interpolation discriminator D_{Interp} minimizes the following loss:

$$\begin{aligned} \min_{D_{\text{Interp}}} \mathcal{L}_{\text{Interp}}^D = & \mathbb{E}_{\mathbf{x}, \mathbf{v}, \alpha} [\|D_{\text{Interp}}(G(\mathbf{x}, \alpha\mathbf{v})) - \hat{\alpha}\|^2 \\ & + \|D_{\text{Interp}}(G(\mathbf{x}, \mathbf{0}))\|^2 \\ & + \|D_{\text{Interp}}(G(\mathbf{x}, \mathbf{v}))\|^2], \end{aligned} \quad (6)$$

where the first term aims at recovering $\hat{\alpha}$ from $G(\mathbf{x}, \alpha\mathbf{v})$. The second and the third term encourage D_{Interp} to output zero for the non-interpolated images. The objective function of G is modified by adding the following loss:

$$\min_G \mathcal{L}_{\text{Interp}}^G = \mathbb{E}_{\mathbf{x}, \mathbf{v}, \alpha} [\|D_{\text{Interp}}(G(\mathbf{x}, \alpha\mathbf{v}))\|^2], \quad (7)$$

where G tries to fool D_{Interp} to think that $G(\mathbf{x}, \alpha\mathbf{v})$ is non-interpolated. In practice, we find empirically that the following modified loss stabilizes the adversarial training process:

$$\begin{aligned} \min_{D_{\text{Interp}}} \mathcal{L}_{\text{Interp}}^D = & \mathbb{E}_{\mathbf{x}, \mathbf{v}, \alpha} [\|D_{\text{Interp}}(G(\mathbf{x}, \alpha\mathbf{v})) - \hat{\alpha}\|^2 \\ & + \|D_{\text{Interp}}(G(\mathbf{x}, \mathbb{I}[\alpha > 0.5]\mathbf{v}))\|^2], \end{aligned} \quad (8)$$

where $\mathbb{I}[\cdot]$ is the indicator function that equals to 1 if its argument is true and 0 otherwise. Algorithm 2 shows the pseudo-code of $\mathcal{L}_{\text{Interp}}^D$ and $\mathcal{L}_{\text{Interp}}^G$.

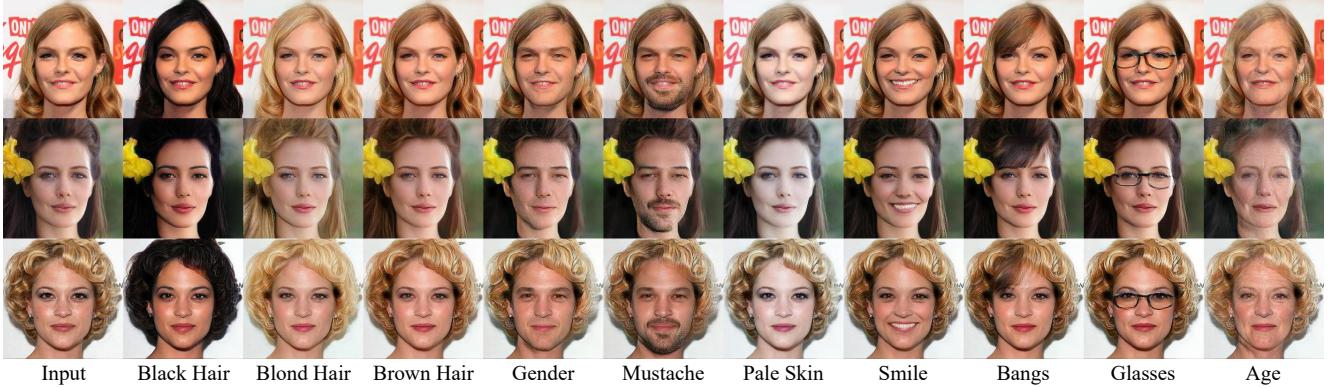


Figure 3. Facial attribute transfer results of RelGAN on the CelebA-HQ dataset.

Algorithm 2 Interpolation loss

```

1: function INTERP_LOSS( $\mathbf{x}, \mathbf{v}$ )
2:    $\alpha \sim U(0, 1)$ 
3:    $y_0 \leftarrow D_{\text{Interp}}(G(\mathbf{x}, \mathbf{0}))$  {non-interpolated image}
4:    $y_1 \leftarrow D_{\text{Interp}}(G(\mathbf{x}, \mathbf{v}))$  {non-interpolated image}
5:    $y_\alpha \leftarrow D_{\text{Interp}}(G(\mathbf{x}, \alpha\mathbf{v}))$  {interpolated image}
6:   if  $\alpha \leq 0.5$  then
7:      $\mathcal{L}_{\text{Interp}}^D \leftarrow y_0^2 + (y_\alpha - \alpha)^2$ 
8:   else
9:      $\mathcal{L}_{\text{Interp}}^D \leftarrow y_1^2 + (y_\alpha - (1 - \alpha))^2$ 
10:     $\mathcal{L}_{\text{Interp}}^G \leftarrow y_\alpha^2$ 
11:   return  $\mathcal{L}_{\text{Interp}}^D, \mathcal{L}_{\text{Interp}}^G$ 

```

3.6. Full Loss

To stabilize the training process, we add orthogonal regularization [17] into our loss function. Finally, the full loss function for $D = \{D_{\text{Real}}, D_{\text{Match}}, D_{\text{Interp}}\}$ and for G are expressed, respectively, as

$$\min_D \mathcal{L}^D = -\mathcal{L}_{\text{Real}} + \lambda_1 \mathcal{L}_{\text{Match}}^D + \lambda_2 \mathcal{L}_{\text{Interp}}^D \quad (9)$$

and

$$\begin{aligned} \min_G \mathcal{L}^G &= \mathcal{L}_{\text{Real}} + \lambda_1 \mathcal{L}_{\text{Match}}^G + \lambda_2 \mathcal{L}_{\text{Interp}}^G \\ &\quad + \lambda_3 \mathcal{L}_{\text{Cycle}} + \lambda_4 \mathcal{L}_{\text{Self}} + \lambda_5 \mathcal{L}_{\text{Ortho}}, \end{aligned} \quad (10)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, and λ_5 are hyper-parameters that control the relative importance of each loss.

4. Experiments

In this section, we perform extensive experiments to demonstrate the effectiveness of RelGAN. We first describe the experimental settings (Section 4.1, 4.2, and 4.3). Then we show the experimental results on the tasks of facial

attribute transfer (Section 4.4), facial image reconstruction (Section 4.5), and facial attribute interpolation (Section 4.6). Lastly, we present the results of the user study (Section 4.7).

4.1. Dataset

CelebA. The CelebFaces Attributes Dataset (CelebA) [18] contains 202,599 face images of celebrities, annotated with 40 binary attributes such as hair colors, gender, age. We center crop these images to 178×178 and resize them to 256×256 .

CelebA-HQ. Karras et al. [19] created a high-quality version of CelebA dataset, which consists of 30,000 images generated by upsampling the CelebA images with an adversarially trained super-resolution model.

FFHQ. The Flickr-Faces-HQ Dataset (FFHQ) [20] consists of 70,000 high quality facial images at 1024×1024 resolution. This dataset has a larger variation than the CelebA-HQ dataset.

4.2. Implementation Details

The images of the three datasets are center cropped and resized to 256×256 . Our generator network, adapted from StarGAN [7], is composed of two convolutional layers with a stride of 2 for down-sampling, six residual blocks, and two transposed convolutional layers with a stride of 2 for up-sampling. We use switchable normalization [21] in the generator. Our discriminators $D = \{D_{\text{Real}}, D_{\text{Match}}, D_{\text{Interp}}\}$ have a shared feature sub-network comprised of six convolutional layers with a stride of 2. Each discriminator has its output layers added onto the feature sub-network. Please see the supplementary material for more details about the network architecture.

For $\mathcal{L}_{\text{Real}}$ (Equation 2) and $\mathcal{L}_{\text{Match}}$ (Equation 3), we use LSGANs-GP [22] for stabilizing the training process. For the hyper-parameters in Equation 9 and 10, we use $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = \lambda_4 = 10$, and $\lambda_5 = 10^{-6}$. We use the

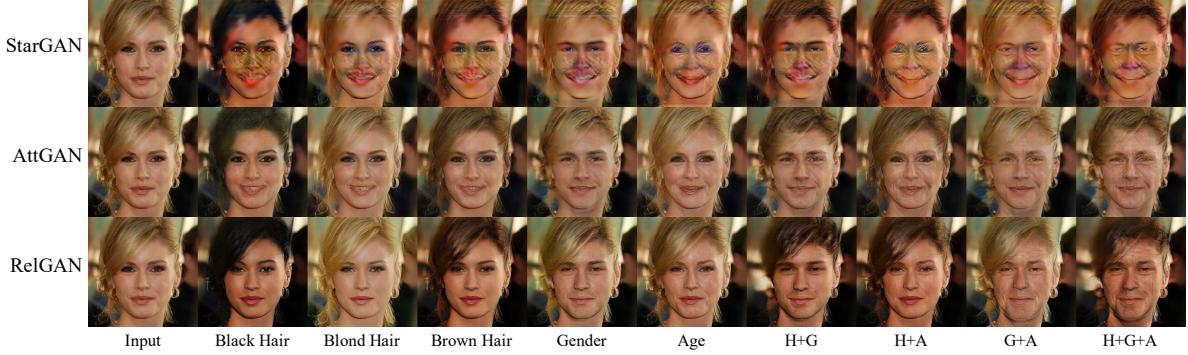


Figure 4. Facial attribute transfer results of StarGAN, AttGAN, and RelGAN on the CelebA-HQ dataset. Please zoom in for more details.

| Training set | <i>n</i> | Test set | StarGAN | AttGAN | RelGAN |
|--------------|----------|-----------|---------|--------|--------------|
| CelebA | 9 | CelebA | 10.15 | 10.74 | 4.68 |
| CelebA-HQ | 9 | CelebA-HQ | 13.18 | 11.73 | 6.99 |
| CelebA-HQ | 17 | CelebA-HQ | 49.28 | 13.45 | 10.35 |
| CelebA-HQ | 9 | FFHQ | 34.80 | 25.53 | 17.51 |
| CelebA-HQ | 17 | FFHQ | 69.74 | 27.25 | 22.74 |

Table 1. **Visual quality comparison.** We use Fréchet Inception Distance (FID) for evaluating the visual quality (lower is better). *n* is the number of attributes used in training. RelGAN achieves the lowest FID score among the three methods in all the five settings.

Adam optimizer [23] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We train RelGAN from scratch with a learning rate of 5×10^{-5} and a batch size of 4 on the CelebA-HQ dataset. We train for 100K iterations, which is about 13.3 epochs. Training RelGAN on a GTX 1080 Ti GPU takes about 60 hours.

4.3. Baselines

We compare RelGAN with StarGAN [7] and AttGAN [8], which are two representative methods in multi-domain image-to-image translation. For both methods, we use the code released by the authors and train their models on the CelebA-HQ dataset with their default hyper-parameters.

4.4. Facial Attribute Transfer

Visual quality comparison. We use Fréchet Inception Distance (FID) [24] (lower is better) as the evaluation metric to measure the visual quality. We experimented with three different training sets: CelebA with 9 attributes, CelebA-HQ with 9 attributes, and CelebA-HQ with 17 attributes. Table 1 shows the FID comparison of StarGAN, AttGAN, and RelGAN. We can see that RelGAN consistently outperforms StarGAN and AttGAN for all the three training sets. Additionally, we experimented with training on the CelebA-HQ dataset while testing on the FFHQ dataset to evaluate the generalization capability. Still, RelGAN achieves better FID scores than the other methods.

| Images | Hair | Gender | Bangs | Eyeglasses |
|-----------|--------------|--------------|--------------|--------------|
| CelebA-HQ | 92.52 | 98.37 | 95.83 | 99.80 |
| StarGAN | 95.48 | 90.21 | 96.00 | 97.08 |
| AttGAN | 89.43 | 94.44 | 92.49 | 98.26 |
| RelGAN | 91.08 | 96.36 | 94.96 | 99.20 |
| Images | Mustache | Smile | Pale Skin | Average |
| CelebA-HQ | 97.90 | 94.20 | 96.70 | 96.47 |
| StarGAN | 89.87 | 90.56 | 96.56 | 93.68 |
| AttGAN | 95.35 | 90.30 | 98.23 | 94.07 |
| RelGAN | 94.57 | 92.93 | 96.79 | 95.13 |

Table 2. The classification accuracy (percentage, higher is better) on the CelebA-HQ images and the generated images of StarGAN, AttGAN, and RelGAN. For each attribute, the highest accuracy among the three methods is highlighted in bold.

Classification accuracy. To quantitatively evaluate the quality of image translation, we trained a facial attribute classifier on the CelebA-HQ dataset using a Resnet-18 architecture [25]. We used a 90/10 split for training and testing. In Table 2, We report the classification accuracy on the test set images and the generated images produced by StarGAN, AttGAN, and RelGAN. The accuracy on the CelebA-HQ images serves as an upper-bound. RelGAN achieves the highest average accuracy and rank 1st in 3 out of the 7 attributes.

Qualitative results. Figure 3 and 4 show the qualitative results on facial attribute transfer. Figure 3 shows representative examples to demonstrate that RelGAN is capable of generating high quality and realistic attribute transfer results. Figure 4 shows a visual comparison of the three methods. StarGAN's results contain notable artifacts. AttGAN yields blurry and less detailed results compared to RelGAN. Conversely, RelGAN is capable of preserving unchanged attributes. In the case of changing hair color, RelGAN preserves the smile attribute, while both StarGAN and AttGAN make the woman's mouth open due to their target-attribute-based formulation. More qualitative results can be found in

| $\mathcal{L}_{\text{Real}}$ | $\mathcal{L}_{\text{Match}}$ | $\mathcal{L}_{\text{Cycle}} + \mathcal{L}_{\text{Self}}$ | Results | | | | | | |
|-----------------------------|------------------------------|--|---------|--|--|--|--|--|--|
| ✓ | ✓ | | | | | | | | |
| ✓ | | ✓ | | | | | | | |
| | ✓ | ✓ | | | | | | | |
| ✓ | ✓ | ✓ | | | | | | | |

Table 3. **Ablation study.** From left to right: input, black hair, blond hair, brown hair, gender, mustache, pale skin, and smile.

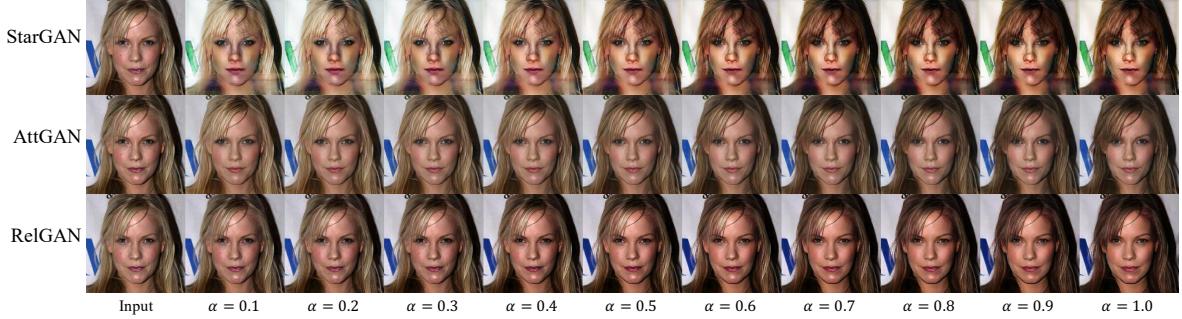


Figure 5. Facial attribute interpolation results of StarGAN, AttGAN, and RelGAN on the CelebA-HQ dataset.

| Method | $\mathcal{L}_{\text{Cycle}}$ | $\mathcal{L}_{\text{Self}}$ | L1 | L2 | SSIM |
|---------|------------------------------|-----------------------------|---------------|-----------------|--------------|
| StarGAN | ✓ | | 0.1136 | 0.023981 | 0.567 |
| AttGAN | | ✓ | 0.0640 | 0.008724 | 0.722 |
| RelGAN | ✓ | | 0.1116 | 0.019721 | 0.731 |
| RelGAN | | ✓ | 0.0179 | 0.000649 | 0.939 |
| RelGAN | ✓ | ✓ | 0.0135 | 0.000463 | 0.947 |

Table 4. **Facial image reconstruction.** We measure the reconstruction error using L1 and L2 distance (lower is better), and SSIM (higher is better).

| Method | Hair | Age | Gender |
|--|---------------|---------------|---------------|
| AttGAN | 0.0491 | 0.0449 | 0.0426 |
| StarGAN | 0.0379 | 0.0384 | 0.0375 |
| RelGAN w/o $\mathcal{L}_{\text{Interp}}$ | 0.0363 | 0.0308 | 0.0375 |
| RelGAN | 0.0170 | 0.0278 | 0.0167 |

Table 5. **Facial attribute interpolation.** We measure the interpolation quality using the standard deviation of SSIM (Equation 11, lower is better).

the supplementary material.

In Table 3, we show an ablation study of our loss function. We can see that: (1) Training without $\mathcal{L}_{\text{Cycle}} + \mathcal{L}_{\text{Self}}$ (1st row) cannot preserve identity. (2) Training without $\mathcal{L}_{\text{Match}}$ (2nd row) only learns to reconstruct the input image. (3) Training without $\mathcal{L}_{\text{Real}}$ (3rd row) gives reasonable results. (4) Training with full loss (4th row) yields the best results.

4.5. Facial Image Reconstruction

One important advantage of RelGAN is preserving the unchanged attributes, which is a desirable property for facial attribute editing. When all the attributes are unchanged, i.e., the target attribute vector is equal to the original one, the facial attribute transfer task reduces to a reconstruction task. Here, we evaluate the performance of facial image reconstruction as a proxy metric to demonstrate that RelGAN better preserves the unchanged attributes.

To perform facial image reconstruction, we respectively apply StarGAN and AttGAN by taking the original attributes as the target attributes, and apply RelGAN by taking a zero vector as the relative attributes. We measure L1, L2 norm, and SSIM similarity [26] between the input and

| Method | Hair | Bangs | Eyeglasses | Gender | Pale Skin | Smile | Age | Mustache | Reconstruction | Interpolation |
|---------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----------------|---------------|
| StarGAN | 0.00 | 0.74 | 1.11 | 1.11 | 0.74 | 2.21 | 1.11 | 0.74 | 1.77 | 6.05 |
| AttGAN | 27.71 | 34.32 | 19.19 | 28.78 | 20.66 | 52.76 | 42.44 | 32.84 | 7.82 | 54.98 |
| RelGAN | 72.29 | 64.94 | 79.70 | 64.65 | 78.60 | 45.02 | 56.46 | 66.42 | 97.12 | 66.42 |

Table 6. The voting results of the user study (percentage, higher is better).

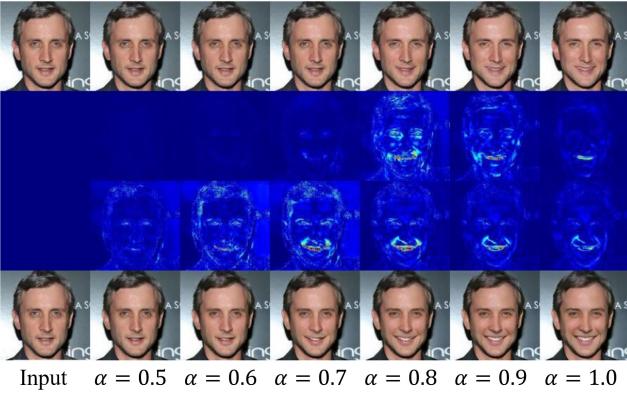


Figure 6. We use heat maps to visualize the difference between two adjacent images. **Top two rows:** interpolation without $\mathcal{L}_{\text{Interp}}$ gives an inferior interpolation due to an abrupt change between $\alpha = 0.7$ and $\alpha = 0.8$. **Bottom two rows:** interpolation with $\mathcal{L}_{\text{Interp}}$ gives better results since the appearance change is more evenly distributed across the image sequence.

the output images. As shown in Table 4, StarGAN uses a cycle-reconstruction loss only while AttGAN uses a self-reconstruction loss only. We evaluate three variants of RelGAN to uncover the contribution of $\mathcal{L}_{\text{Cycle}}$ and $\mathcal{L}_{\text{Self}}$. The results show that RelGAN without $\mathcal{L}_{\text{Cycle}}$ already outperforms StarGAN and AttGAN in terms of all the three metrics. RelGAN further improves the results.

4.6. Facial Attribute Interpolation

We next evaluate RelGAN on the task of facial attribute interpolation. For both StarGAN and AttGAN, their interpolated images are generated by $G(\mathbf{x}, \alpha \mathbf{a} + (1 - \alpha) \hat{\mathbf{a}})$, where \mathbf{a} and $\hat{\mathbf{a}}$ are the original and the target attribute vector, respectively. Our interpolated images are generated by $G(\mathbf{x}, \alpha \mathbf{v})$.

Qualitative results. As can be seen from Figure 5, StarGAN generates a non-smooth interpolation that the appearance change mainly happens between $\alpha = 0.4$ and $\alpha = 0.6$. Both StarGAN and AttGAN have an abrupt change between the input and the result with $\alpha = 0.1$. In particular, the blond hair attribute is not well preserved by both methods. RelGAN achieves the most smooth-varying interpolation.

Quantitative evaluation. We use the following metric to evaluate the interpolation quality. Given an input image \mathbf{x}_0 , an output image \mathbf{x}_m , and a set of interpolated im-

ages $\{\mathbf{x}_1, \dots, \mathbf{x}_{m-1}\}$, a high quality and smoothly-varying interpolation implies that the appearance changes steadily from \mathbf{x}_0 to \mathbf{x}_m . To this end, we compute the standard deviation of the SSIM scores between \mathbf{x}_{i-1} and \mathbf{x}_i , i.e.,

$$\sigma(\{\text{SSIM}(\mathbf{x}_{i-1}, \mathbf{x}_i) | i = 1, \dots, m\}), \quad (11)$$

where $\sigma(\cdot)$ computes the standard deviation. We use $m = 10$ in this experiment. A smaller standard deviation indicates a better interpolation quality. As shown in Table 5, StarGAN is comparable to RelGAN without $\mathcal{L}_{\text{Interp}}$. RelGAN with $\mathcal{L}_{\text{Interp}}$ effectively reduces the standard deviation, showing that our interpolation is not only realistic but also smoothly-varying. Figure 6 shows a visual comparison of RelGAN with and without $\mathcal{L}_{\text{Interp}}$.

4.7. User Study

We conducted a user study to evaluate the image quality of RelGAN. We consider 10 tasks, where eight are the facial attribute transfer tasks (Section 4.4), one is the facial image reconstruction (Section 4.5), and one is the facial attribute interpolation (Section 4.6). 302 users were involved in this study. Each user is asked to answer 40 questions, each of which is generated from randomly sampling a Celeba-HQ image and a task, and then applying StarGAN, AttGAN, and RelGAN to obtain their result respectively. For the attribute transfer tasks, users are asked to pick the best result among the three methods. For the other two tasks, we allow users to vote for multiple results that look satisfactory. The results of the user study are summarized in Table 6. RelGAN obtains the majority of votes in all the tasks except the smile task.

5. Conclusion

In this paper, we proposed a novel multi-domain image-to-image translation model based on relative attributes. By taking relative attributes as input, our generator learns to modify an image in terms of the attributes of interest while preserves the other unchanged ones. Our model achieves superior performance over the state-of-the-art methods in terms of both visual quality and interpolation. Our future work includes using more advanced adversarial learning methods [27, 28, 29] and mask mechanisms [15, 9, 30, 31] for further improvement.

References

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014. [1](#), [2](#), [3](#)
- [2] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014. [1](#), [2](#), [3](#)
- [3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *CVPR*, 2017. [1](#), [2](#)
- [4] Z. Yi, H. Zhang, P. Tan, and M. Gong, “DualGAN: Unsupervised dual learning for image-to-image translation,” in *ICCV*, 2017. [1](#), [3](#)
- [5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, 2017. [1](#), [3](#), [4](#)
- [6] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *NIPS*, 2017. [1](#), [3](#)
- [7] 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 *CVPR*, 2018. [1](#), [3](#), [5](#), [6](#)
- [8] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, “AttGAN: Facial attribute editing by only changing what you want,” *IEEE Transactions on Image Processing*, 2019. [1](#), [3](#), [6](#)
- [9] B. Zhao, B. Chang, Z. Jie, and L. Sigal, “Modular generative adversarial networks,” in *ECCV*, 2018. [1](#), [3](#), [8](#)
- [10] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” in *ICML*, 2016. [2](#), [4](#)
- [11] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, “StackGAN++: Realistic image synthesis with stacked generative adversarial networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. [2](#)
- [12] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, “Multi-modal unsupervised image-to-image translation,” in *ECCV*, 2018. [3](#)
- [13] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Alvarez, “Invertible conditional GANs for image editing,” in *NIPS Workshop on Adversarial Training*, 2016. [3](#)
- [14] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer *et al.*, “Fader networks: Manipulating images by sliding attributes,” in *NIPS*, 2017. [3](#)
- [15] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, “GANimation: Anatomically-aware facial animation from a single image,” in *ECCV*, 2018. [3](#), [8](#)
- [16] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, “Understanding and improving interpolation in autoencoders via an adversarial regularizer,” in *ICLR*, 2019. [4](#)
- [17] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in *ICLR*, 2019. [5](#)
- [18] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *ICCV*, 2015. [5](#)
- [19] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in *ICLR*, 2018. [5](#)
- [20] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *CVPR*, 2019. [5](#)
- [21] P. Luo, J. Ren, Z. Peng, R. Zhang, and J. Li, “Differentiable learning-to-normalize via switchable normalization,” in *ICLR*, 2019. [5](#)
- [22] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “On the effectiveness of least squares generative adversarial networks,” *PAMI*, 2018. [5](#)
- [23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *ICLR*, 2015. [5](#)
- [24] 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 *NIPS*, 2017. [6](#)
- [25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016. [6](#)
- [26] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, 2004. [7](#)
- [27] A. Jolicoeur-Martineau, “The relativistic discriminator: a key element missing from standard GAN,” in *ICLR*, 2019. [8](#)
- [28] T. Miyato and M. Koyama, “cGANs with projection discriminator,” in *ICLR*, 2018. [8](#)
- [29] C.-H. Chang, C.-H. Yu, S.-Y. Chen, and E. Y. Chang, “KG-GAN: Knowledge-guided generative adversarial networks,” *arXiv preprint arXiv:1905.12261*, 2019. [8](#)
- [30] G. Zhang, M. Kan, S. Shan, and X. Chen, “Generative adversarial network with spatial attention for face attribute editing,” in *ECCV*, 2018. [8](#)
- [31] R. Sun, C. Huang, J. Shi, and L. Ma, “Mask-aware photorealistic face attribute manipulation,” *arXiv preprint arXiv:1804.08882*, 2018. [8](#)

Appendix A. Network Architecture

Figure 7 shows the schematic diagram of RelGAN. Table 7 and 8 show the network architecture of RelGAN.

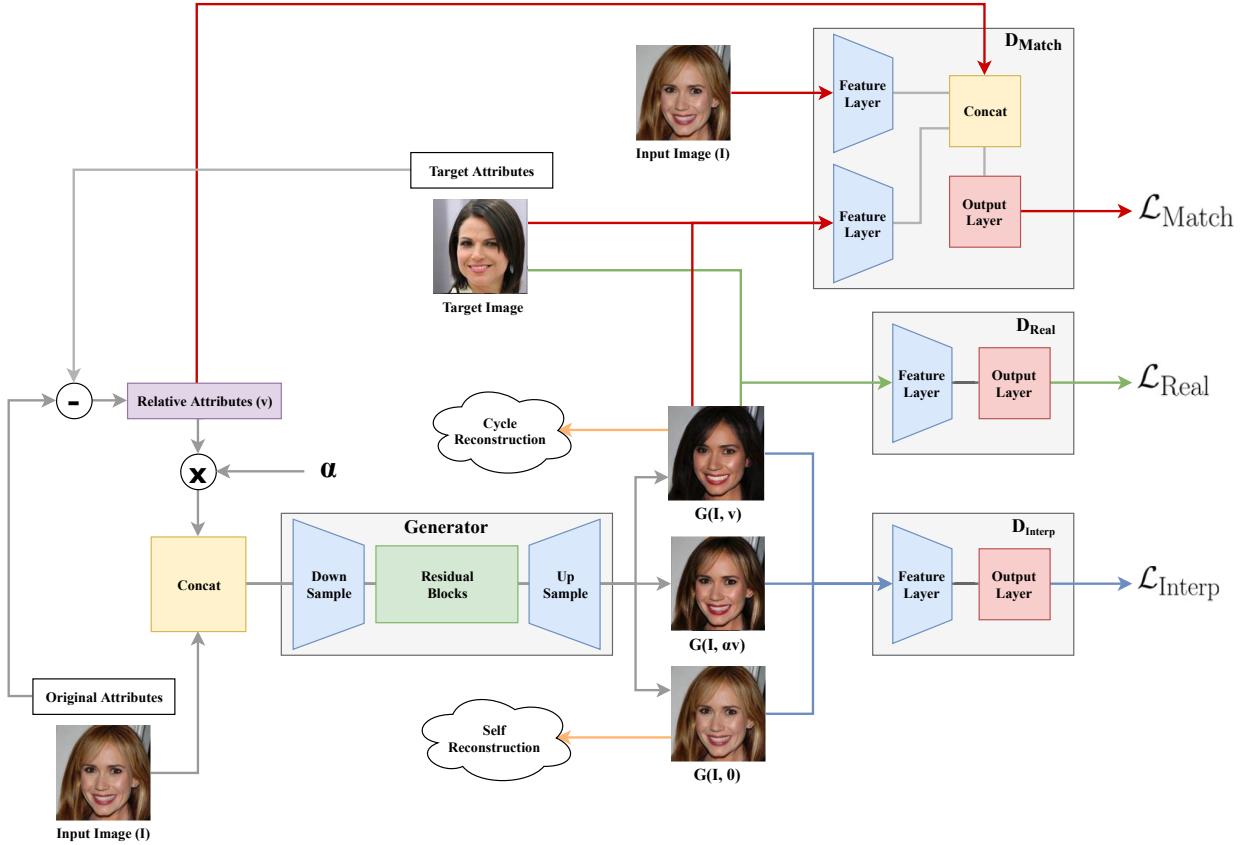


Figure 7. Detailed schematic diagram of RelGAN. D_{Real} , D_{Match} , and D_{Interp} share the weights of the feature layers.

| Component | Input → Output Shape | Layer Information |
|-----------------|---------------------------------------|--|
| Down Sample | $(h, w, 3+n) \rightarrow (h, w, 64)$ | Conv-(F=64,K=7,S=1,P=3),SN,ReLU |
| | $(h, w, 64) \rightarrow (h, w, 128)$ | Conv-(F=128,K=4,S=2,P=1),SN,ReLU |
| | $(h, w, 128) \rightarrow (h, w, 256)$ | Conv-(F=256,K=4,S=2,P=1),SN,ReLU |
| Residual Blocks | $(h, w, 256) \rightarrow (h, w, 256)$ | Residual Block: Conv-(F=256,K=3,S=1,P=1),SN,ReLU |
| | $(h, w, 256) \rightarrow (h, w, 256)$ | Residual Block: Conv-(F=256,K=3,S=1,P=1),SN,ReLU |
| | $(h, w, 256) \rightarrow (h, w, 256)$ | Residual Block: Conv-(F=256,K=3,S=1,P=1),SN,ReLU |
| | $(h, w, 256) \rightarrow (h, w, 256)$ | Residual Block: Conv-(F=256,K=3,S=1,P=1),SN,ReLU |
| | $(h, w, 256) \rightarrow (h, w, 256)$ | Residual Block: Conv-(F=256,K=3,S=1,P=1),SN,ReLU |
| | $(h, w, 256) \rightarrow (h, w, 256)$ | Residual Block: Conv-(F=256,K=3,S=1,P=1),SN,ReLU |
| Up Sample | $(h, w, 256) \rightarrow (h, w, 128)$ | Conv-(F=128,K=4,S=2,P=1),SN,ReLU |
| | $(h, w, 128) \rightarrow (h, w, 64)$ | Conv-(F=64,K=4,S=2,P=1),SN,ReLU |
| | $(h, w, 64) \rightarrow (h, w, 3)$ | Conv-(F=3,K=7,S=1,P=3),Tanh |

Table 7. **Generator network architecture.** We use switchable normalization, denoted as SN, in all layers except the last output layer. n is the number of attributes. F is the number of filters. K is the filter size. S is the stride size. P is the padding size. The number of trainable parameters is about 8M.

| Component | Input → Output Shape | Layer Information |
|------------------------------------|---|--------------------------------------|
| Feature Layers | $(h, w, 3) \rightarrow (h/2, w/2, 64)$ | Conv-(F=64,K=4,S=2,P=1),Leaky ReLU |
| | $(h/2, w/2, 64) \rightarrow (h/4, w/4, 128)$ | Conv-(F=128,K=4,S=2,P=1),Leaky ReLU |
| | $(h/4, w/4, 128) \rightarrow (h/8, w/8, 256)$ | Conv-(F=256,K=4,S=2,P=1),Leaky ReLU |
| | $(h/8, w/8, 256) \rightarrow (h/16, w/16, 512)$ | Conv-(F=512,K=4,S=2,P=1),Leaky ReLU |
| | $(h/16, w/16, 512) \rightarrow (h/16, w/16, 1024)$ | Conv-(F=1024,K=4,S=2,P=1),Leaky ReLU |
| | $(h/32, w/32, 1024) \rightarrow (h/64, w/64, 2048)$ | Conv-(F=2048,K=4,S=2,P=1),Leaky ReLU |
| D_{Real} : Output Layer | $(h/64, w/64, 2048) \rightarrow (h/64, w/64, 1)$ | Conv-(F=1,K=1) |
| D_{Interp} : Output Layer | $(h/64, w/64, 2048) \rightarrow (h/64, w/64, 64)$ | Conv-(F=64,K=1) |
| | $(h/64, w/64, 64) \rightarrow (h/64, w/64, 1)$ | Mean(axis=3) |
| D_{Match} : Output Layer | $(h/64, w/64, 4096 + n) \rightarrow (h/64, w/64, 2048)$ | Conv-(F=1,K=1),Leaky ReLU |
| | $(h/64, w/64, 2048) \rightarrow (h/64, w/64, 1)$ | Conv-(F=1,K=1) |

Table 8. **Discriminator network architecture.** We use Leaky ReLU with a negative slope of 0.01. n is the number of attributes. F is the number of filters. K is the filter size. S is the stride size. P is the padding size. The number of trainable parameters is about 53M.

Appendix B. Additional Results

Figure 8 and 9 show comparison results between StarGAN, AttGAN, and RelGAN on the hair color tasks. The residual heat maps show that RelGAN preserves the smile attribute while the other methods strengthen the smile attribute. Figure 10 and 11 show more comparison results. Figure 12, 13 show additional results on facial attribute transfer. Figure 14, 15, 16, and 17 show additional results on facial attribute interpolation. All the input images are from the CelebA-HQ dataset.

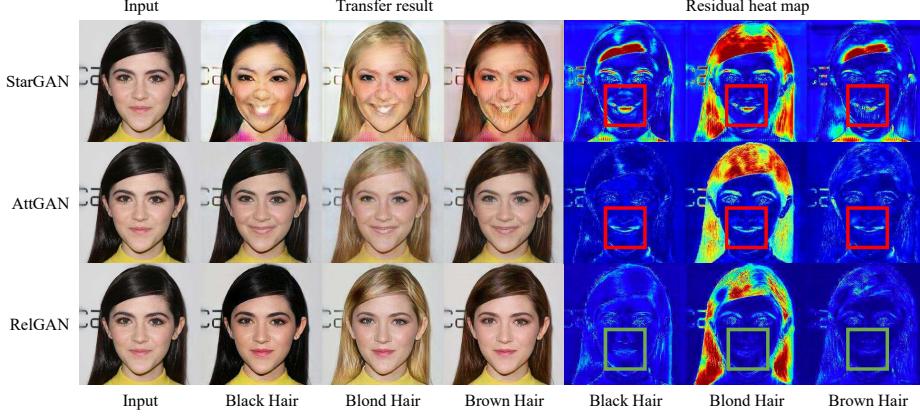


Figure 8. Hair color transfer results of StarGAN, AttGAN, and RelGAN. The residual heat maps visualize the differences between the input and the output images.

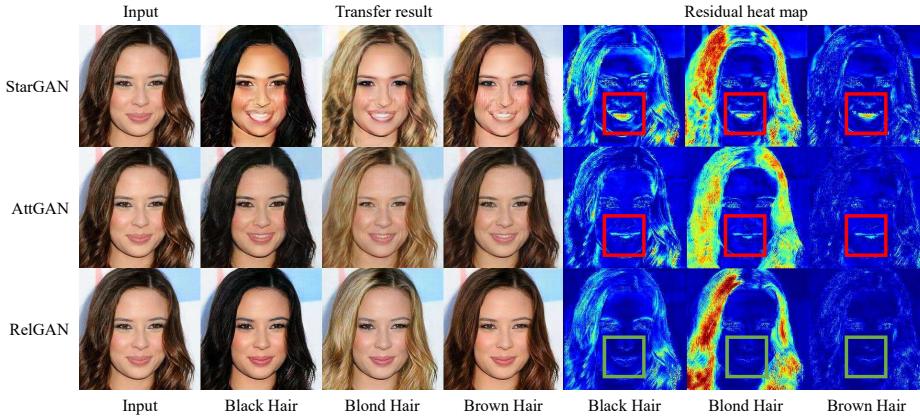


Figure 9. Hair color transfer results of StarGAN, AttGAN, and RelGAN. The residual heat maps visualize the differences between the input and the output images.

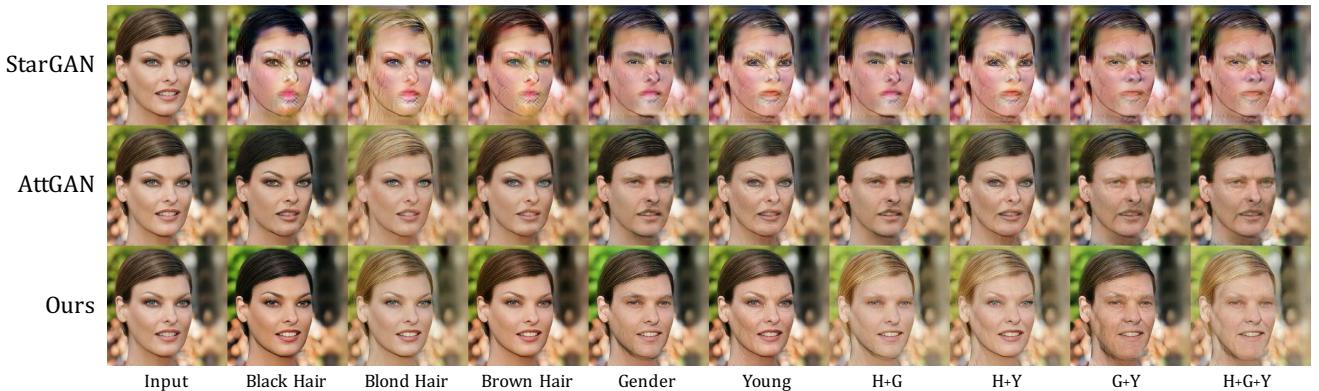


Figure 10. Facial attribute transfer results of StarGAN, AttGAN, and RelGAN. Please zoom in for more details. In the case of changing hair color, RelGAN preserves the smile attribute, while both StarGAN and AttGAN make the woman look unhappy due to their target-attribute-based formulation.

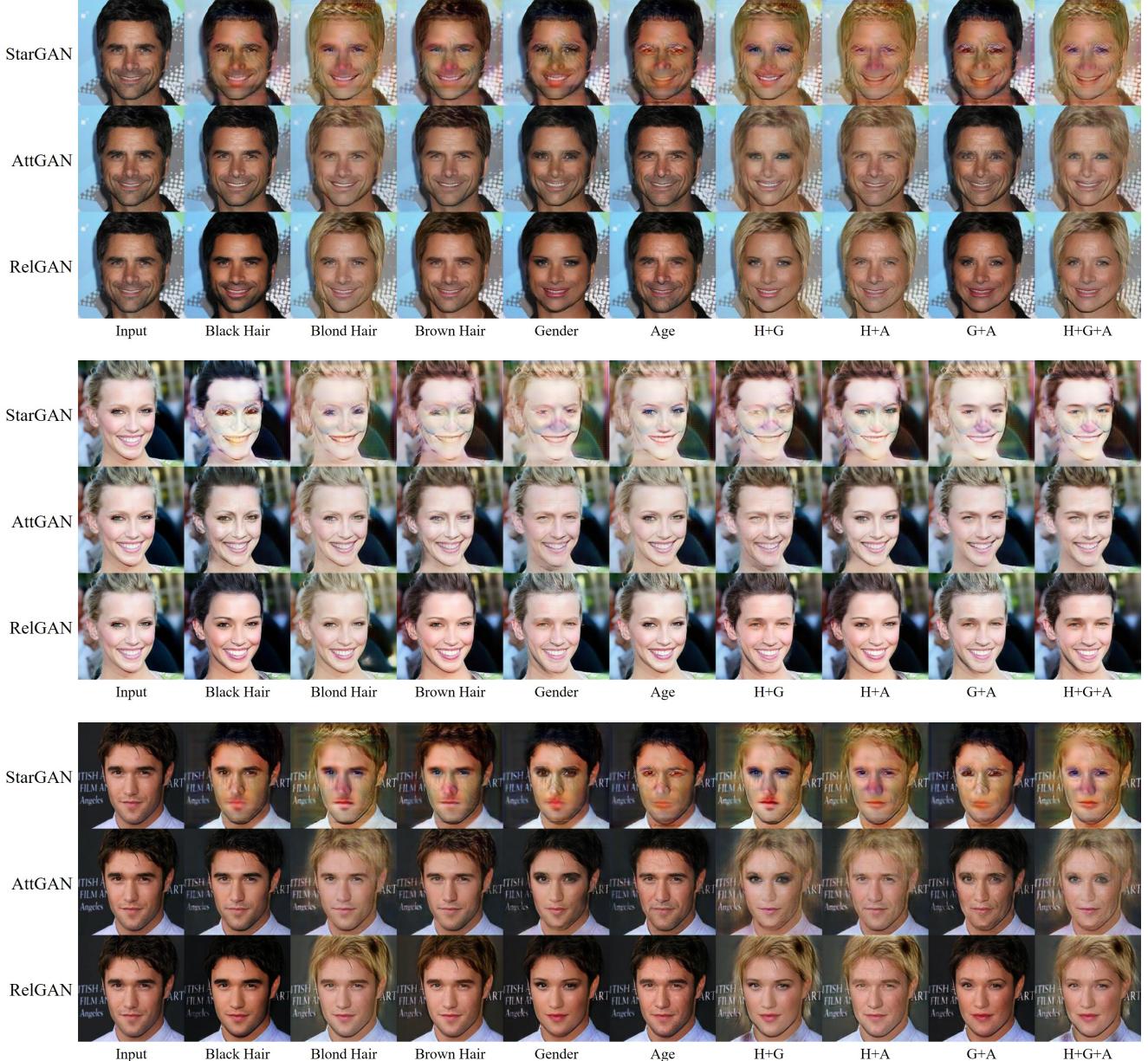


Figure 11. Facial attribute transfer results of StarGAN, AttGAN, and RelGAN. Please zoom in for more details.



Figure 12. Single attribute transfer results. From left to right: input, black hair, blond hair, brown hair, gender, and mustache.

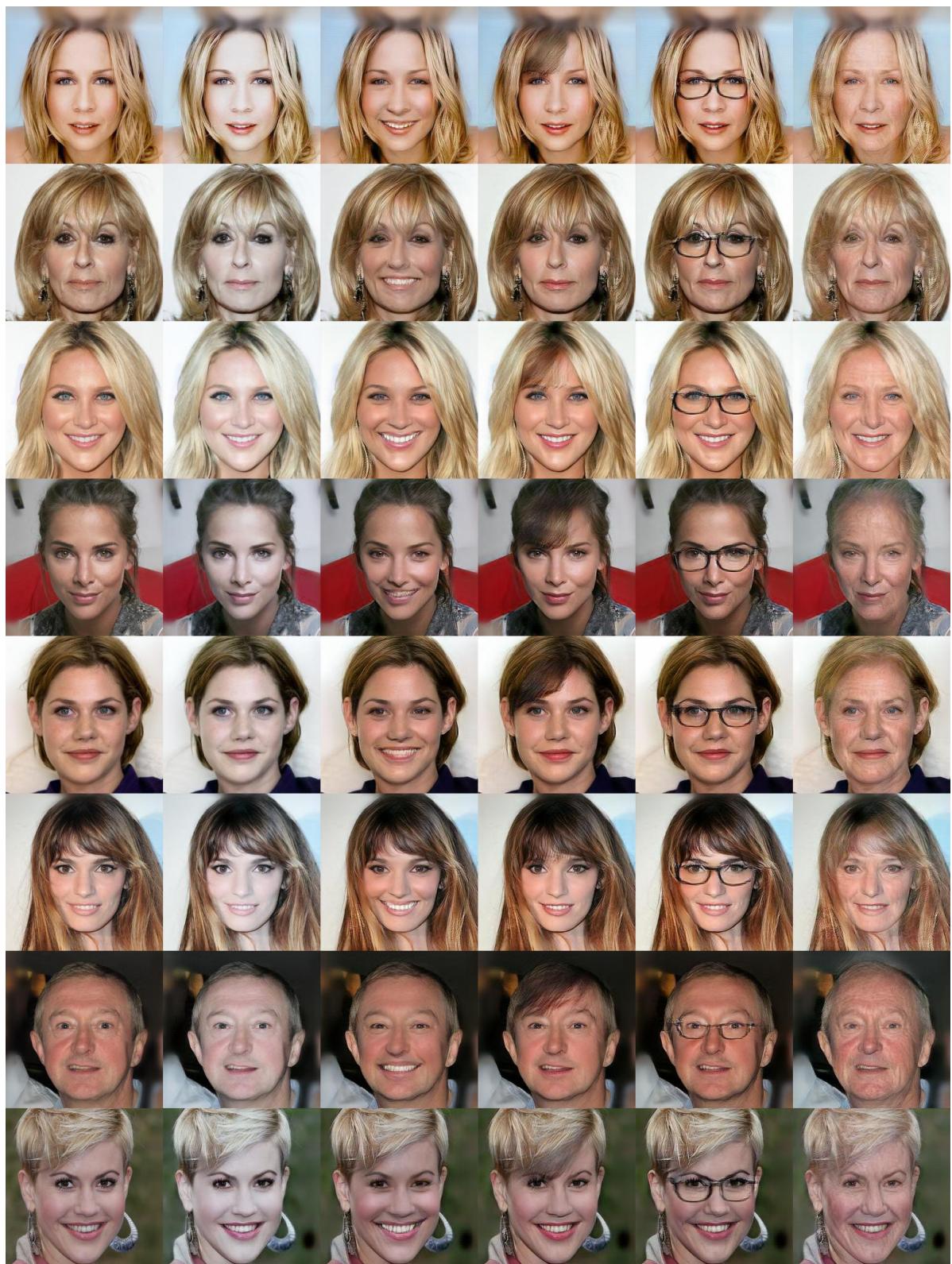


Figure 13. Single attribute transfer results. From left to right: input, pale skin, smiling, bangs, glasses, and age.

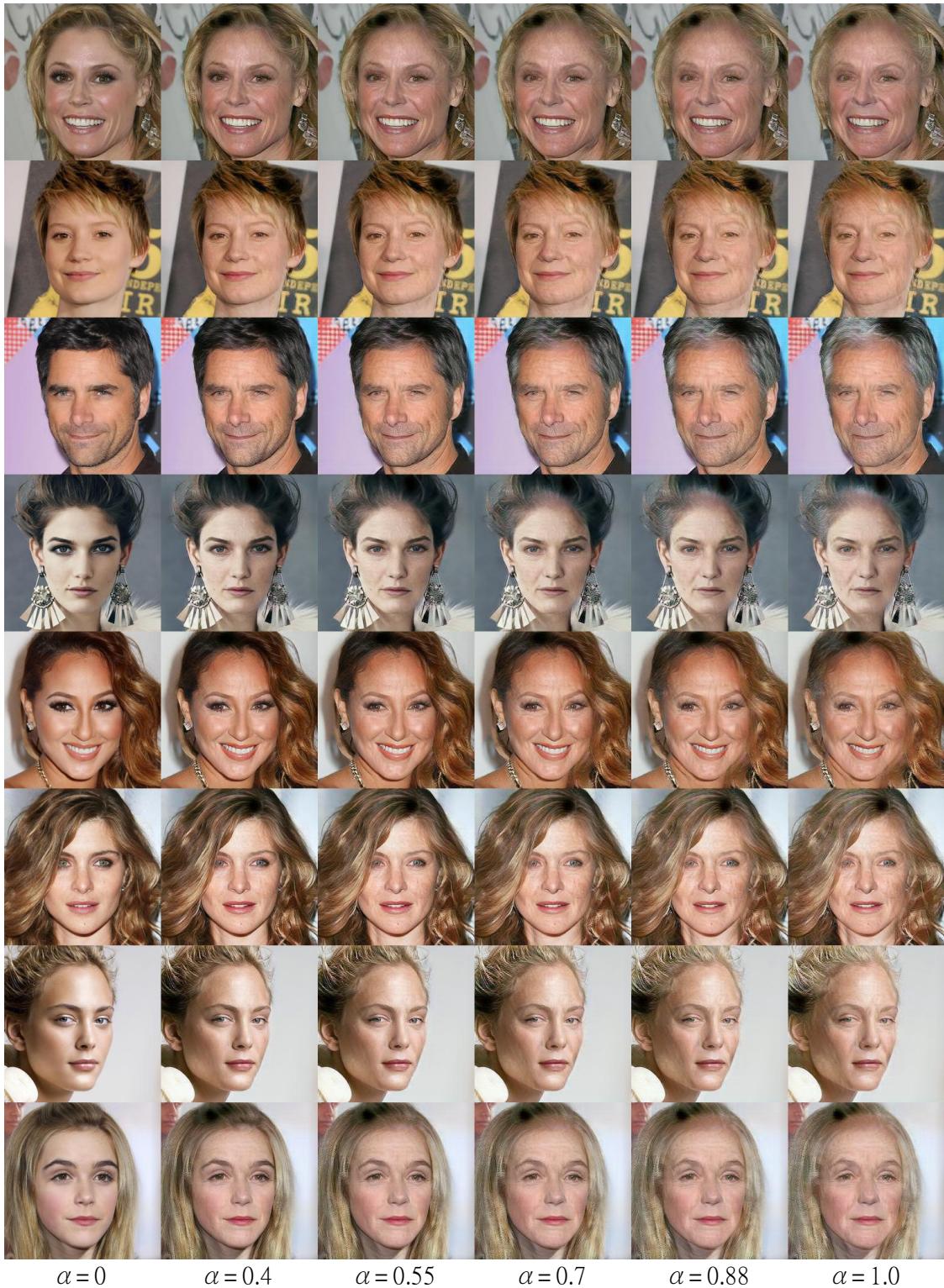


Figure 14. Single attribute interpolation results (age). RelGAN generates different levels of attribute transfer by varying α .



Figure 15. Single attribute interpolation results (gender). RelGAN generates different levels of attribute transfer by varying α .

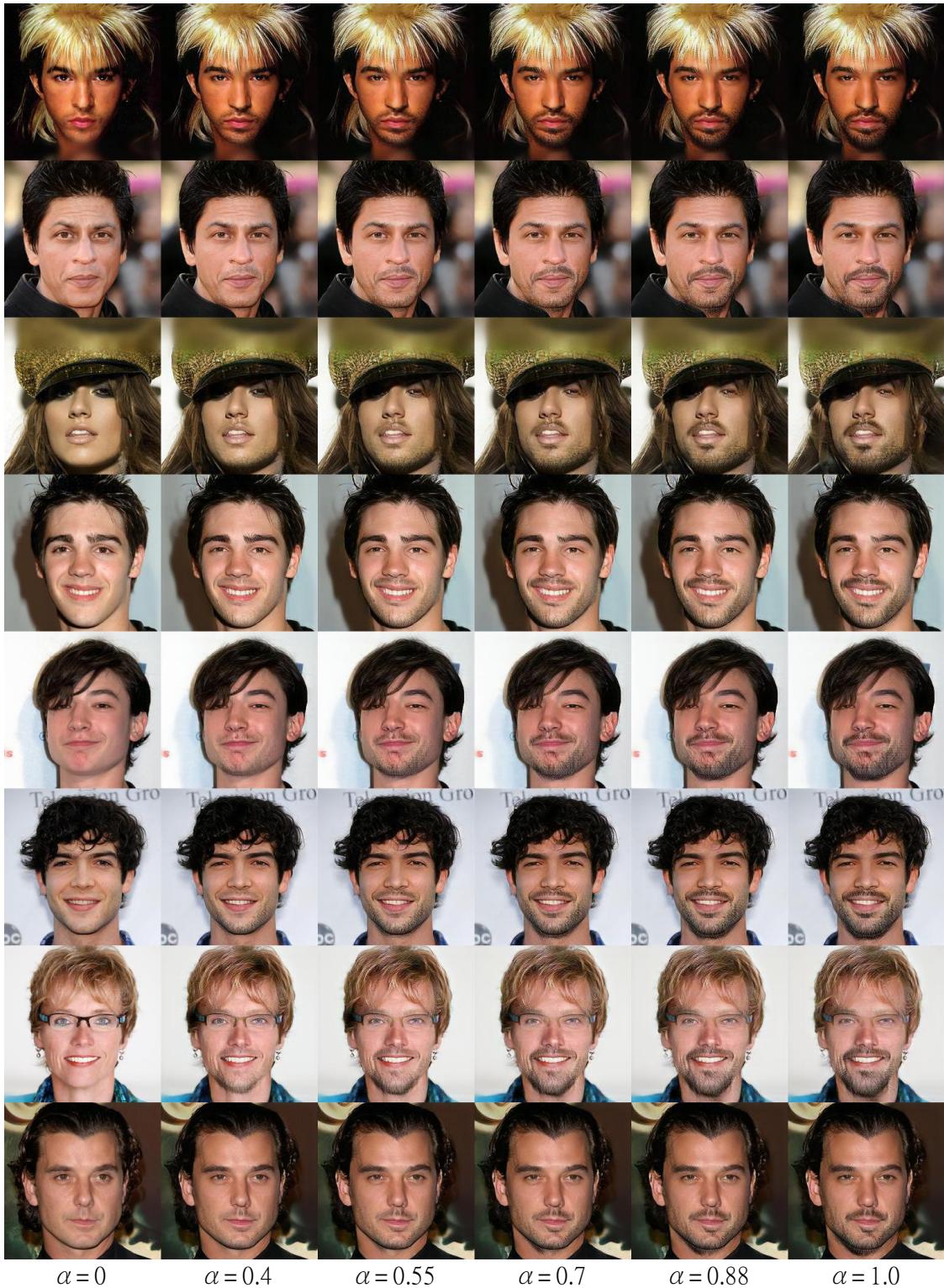


Figure 16. Single attribute interpolation results (mustache). RelGAN generates different levels of attribute transfer by varying α .



Figure 17. Single attribute interpolation results (smile). RelGAN generates different levels of attribute transfer by varying α .