High-Fidelity and Arbitrary Face Editing

Yue Gao[1], Fangyun Wei[2], Jianmin Bao[2], Shuyang Gu[3], Dong Chen[2], Fang Wen[2], Zhouhui Lian[1*]

[1]Wangxuan Institute of Computer Technology, Peking University, China
[2]Microsoft Research Asia
[3]University of Science and Technology of China

Abstract

Cycle consistency is widely used for face editing. However, we observe that the generator tends to find a tricky way to hide information from the original image to satisfy the constraint of cycle consistency, making it impossible to maintain the rich details (e.g.*, wrinkles and moles) of non-editing areas. In this work, we propose a simple yet effective method named HifaFace to address the above-mentioned problem from two perspectives. First, we relieve the pressure of the generator to synthesize rich details by directly feeding the high-frequency information of the input image into the end of the generator. Second, we adopt an additional discriminator to encourage the generator to synthesize rich details. Specifically, we apply wavelet transformation to transform the image into multi-frequency domains, among which the high-frequency parts can be used to recover the rich details. We also notice that a fine-grained and wider-range control for the attribute is of great importance for face editing. To achieve this goal, we propose a novel attribute regression loss. Powered by the proposed framework, we achieve high-fidelity and arbitrary face editing, outperforming other state-of-the-art approaches.*

1. Introduction

Face editing is a process of editing the specific attributes or regions of an input facial image while keeping the non-editing attributes/areas unchanged. With the rapid development of Generative Adversarial Networks (GANs) [11], many recent works on face editing [19, 6, 15, 24, 33, 8] leverage the advanced conditional GANs and achieve remarkable progress. Due to the lack of paired images during training, they typically use *cycle consistency* to keep the non-editing attributes/areas unchanged. Namely, given an image $\boldsymbol{x}$, it requires $\boldsymbol{x} = G(G(\boldsymbol{x}, \boldsymbol{\Delta}), -\boldsymbol{\Delta})$, where G represents the generator and $\boldsymbol{\Delta}$ indicates the attribute that needs to be changed. However, we find that even if the cycle consistency is satisfied, images generated by G may still be blurry and lose rich details from input images.

(a) Input $\boldsymbol{x}$ (b) $\hat{\boldsymbol{y}} = \boldsymbol{x}$ + EG (c) $\hat{\boldsymbol{x}} = \hat{\boldsymbol{y}}$ - EG (d) **HifaFace**

Figure 1: The results of a representative face editing method StarGAN [6]: (a) the input image; (b) the output image synthesized by editing (a) with the attribute eyeglasses (EG); (c) the reconstructed image with (b) as input. We observe that the rich details are all missing in $\hat{\boldsymbol{y}}$, but are almost restored in $\hat{\boldsymbol{x}}$. (d) The high-fidelity face image synthesized by our HifaFace.

To demonstrate the above-mentioned problem, we take StarGAN [6] as an example. As shown in Figure 1, we feed an input face image $\boldsymbol{x}$ to StarGAN and expect it to add eyeglasses on the face. Although the output $\hat{\boldsymbol{y}}$ does wear eyeglasses, the details (*e.g.*, wrinkles and moles) are all missing. However, we observe an intriguing phenomenon as follows. When we feed $\hat{\boldsymbol{y}}$ into the StarGAN model and expect removing eyeglasses on the face, the reconstruction result $\hat{\boldsymbol{x}}$ surprisingly recovers almost all rich details, which satisfies the purpose of setting cycle consistency. This observation indicates that the generator encodes the rich details of the input image into the output image in the form of "hidden" signals, and then decodes the feature with these "hidden" signals to achieve reconstruction. The above-mentioned phenomenon is called steganography [7] and is undesirable for face editing [29].

To prevent the generator from playing this trick to satisfy cycle consistency, we propose a simple yet effective face editing method called HifaFace. We tackle this problem from two perspectives. First, we directly feed the high-frequency information of the input image to the end of the

*Zhouhui Lian is the corresponding author. This work was supported by Beijing Nova Program of Science and Technology (Grant No.: Z191100001119077).

generator to alleviate the generator's struggles for synthesizing rich details so that it gives up encoding the hidden signals. Second, we adopt an additional discriminator to constrain the generator to synthesize rich details, thus further preventing the generator from finding a trivial solution for cycle consistency.

Specifically, we adopt wavelet transformation to transform an image into multiple frequency domains. We find that almost all rich details lie in the high-frequency domains. In order to feed the high-frequency information to the generator, we adopt an encoder-decoder-like structure and design a novel module named Wavelet-based Skip-Connection to replace the original Skip-Connection.

To achieve the goal of providing a fine-grained and wider-range control for each facial attribute, we also propose a novel loss, called the attribute regression loss, which requires the generated image to explicitly describe the change on selected attributes and thus enables wider-range and controllable face editing. Furthermore, our method is able to effectively exploit large amounts of unlabeled face images for training, which can further improve the fidelity of synthesized faces in the wild. Powered by the proposed framework, we obtain high-fidelity and arbitrarily controllable face editing results.

In summary, our major contributions are threefold:

- We propose a novel wavelet-based face editing method, called **HifaFace**, for high-fidelity and arbitrary face editing.
- We revisit cycle consistency in face editing and observe that the generator learns to apply a tricky way to satisfy the constraint of cycle consistency by hiding signals in the output image. We thoroughly analyze this phenomenon and provide an effective solution to handle the problem.
- Both qualitative and quantitative results demonstrate the effectiveness of the proposed framework for improving the quality of edited face images.

2. Related Work

Generative Adversarial Networks. Generative Adversarial Networks (GANs) [11] have been widely used in the literature. A typical GAN-based model adopts a generator and a discriminator to implement adversarial training during the learning process. The capability of GANs enables a wide range of computer vision applications such as image generation [20, 21], font generation [9, 32], low-level vision [23, 34], complex distribution modeling [4, 26, 36], and so on.

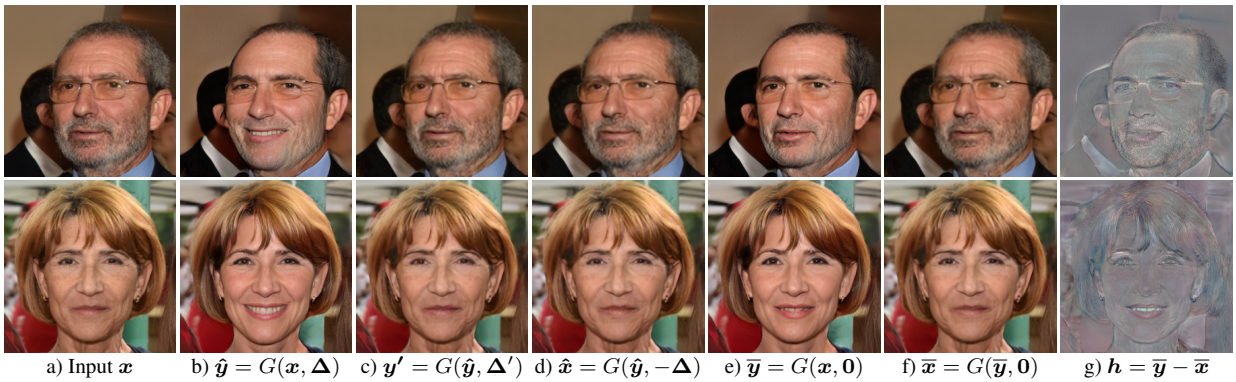
Cycle Consistency. Assessing the matches between two or more samples with cycle consistency is a commonly used technique in computer vision. It has been applied to many popular vision tasks such as image alignment [39, 40], depth estimation [37, 35], correspondence learning [38, 31] and etc. How to exploit the cyclic relation to learn the bi-directional transformation functions between different domains attracts intensive attention in recent years, since it can be used in domain adaption [17] and unpaired image-to-image translation [41, 6]. Also, some previous works [7, 27, 29] notice the problem of cycle consistency in CycleGANs [41] that the cycle consistency tends to do steganography [7] during the training process. In this paper, we concentrate on the process of steganography in the face editing framework and demonstrate its harmfulness for face editing results. Then, we propose a simple yet effective method to solve the problem of cycle consistency in the task of face editing.

Face Generation and Editing. The face is of great interest in computer vision and computer graphics circles. Thanks to the recent development of GANs, a number of methods have reported achieving the high-quality generation of face images [3, 20, 21] and the flexible manipulation of facial attributes for a given face [6, 33, 24]. Generally speaking, these methods can be classified into two groups. The first group of methods [30, 1, 2] leverages pre-trained GANs to achieve the manipulation of faces. They first extract the latent code for a given face, then manipulate it to obtain the edited results. One major drawback of those methods is that they are not able to get the perfect latent code for a given image, resulting in the edited image that loses the rich details of the original face. The second group of methods [13, 6, 33, 24, 8] utilizes image-to-image translation techniques for face editing, where the original face image is also fed to the network. However, synthesis results of these methods are far from satisfactory and their qualities are lower compared to the first type of approaches. One possible reason is the existence of the above-mentioned problem of "cycle consistency". In this paper, we explicitly investigate the reasons why image-to-image translation methods cannot obtain satisfactory results for the face editing task and propose an effective framework to solve this problem.

3. Method Description

3.1. Revisiting Cycle Consistency in Face Editing

Suppose G is a face editing model (*e.g.*, StarGAN [6] or RelGAN [33]), which typically requires two inputs: a source face image and the difference between target attributes and source attributes (StarGAN directly uses the target attributes). Given an image $\boldsymbol{x}$ with the attributes $\boldsymbol{a}_x$, we aim to change its attributes to $\boldsymbol{a}_y$. Namely, the generator G can take $\boldsymbol{x}$ and $\boldsymbol{\Delta} = \boldsymbol{a}_y - \boldsymbol{a}_x$ as input and generate the synthesis result $\hat{\boldsymbol{y}} = G(\boldsymbol{x}, \boldsymbol{\Delta})$. To guarantee that only the regions related to attribute changes in $\hat{\boldsymbol{y}}$ are different against



a) Input $\boldsymbol{x}$ b) $\hat{\boldsymbol{y}} = G(\boldsymbol{x}, \boldsymbol{\Delta})$ c) $\boldsymbol{y'} = G(\hat{\boldsymbol{y}}, \boldsymbol{\Delta'})$ d) $\hat{\boldsymbol{x}} = G(\hat{\boldsymbol{y}}, -\boldsymbol{\Delta})$ e) $\overline{\boldsymbol{y}} = G(\boldsymbol{x}, \boldsymbol{0})$ f) $\overline{\boldsymbol{x}} = G(\overline{\boldsymbol{y}}, \boldsymbol{0})$ g) $\boldsymbol{h} = \overline{\boldsymbol{y}} - \overline{\boldsymbol{x}}$

Figure 2: Previous methods (*e.g.*, StarGAN [6]) using cycle consistency tend to hide information in the output images.

$\boldsymbol{x}$, cycle consistency is usually applied between $G(\hat{\boldsymbol{y}}, -\boldsymbol{\Delta})$ and $\boldsymbol{x}$. However, we observe an intriguing phenomenon, that is, if we further input $\hat{\boldsymbol{y}}$ to the generator with any other attribute difference $\boldsymbol{\Delta'}$ ($\boldsymbol{\Delta'} \neq -\boldsymbol{\Delta}$) to get $\boldsymbol{y'} = G(\hat{\boldsymbol{y}}, \boldsymbol{\Delta'})$, the generated result $\boldsymbol{y'}$ will tend to be the same as input. This phenomenon is demonstrated in Figure 2(a,b,c), indicating that the generator learns to apply a tricky way to achieve the cycle consistency by "hiding" information in the output image. If we feed an image with hidden information to the generator, it tends to ignore the input attributes and only leverage the hidden information to reconstruct the original image.

Thereby, it is crucial to figure out this hidden information. One possible way is to calculate the difference between the edited image and its ground truth [7]. But, unfortunately, it is usually impossible to obtain the ground truth. To address this problem, we let the target attributes be identical to the original attributes by feeding the input image and $\boldsymbol{\Delta} = \boldsymbol{0}$ to the generator. Then we can get the synthesis result $\overline{\boldsymbol{y}} = G(\boldsymbol{x}, \boldsymbol{0})$, whose corresponding ground truth is the original input image. We further feed $\overline{\boldsymbol{y}}$ with $\boldsymbol{\Delta} = \boldsymbol{0}$ back to the generator to get the result $\overline{\boldsymbol{x}} = G(\overline{\boldsymbol{y}}, \boldsymbol{0})$. Examples of $\overline{\boldsymbol{y}}$ and its reconstructed image $\overline{\boldsymbol{x}}$ are shown in Figure 2(e, f). We find that there exist significant differences between $\overline{\boldsymbol{y}}$ and $\boldsymbol{x}$, especially the hair and wrinkles. However, the reconstructed image $\overline{\boldsymbol{x}}$ is almost perfect, which verifies the existence of hidden information in $\overline{\boldsymbol{y}}$. In this manner, we can get the hidden information $\boldsymbol{h} = \overline{\boldsymbol{y}} - \overline{\boldsymbol{x}}$ (see Figure 2(g)).

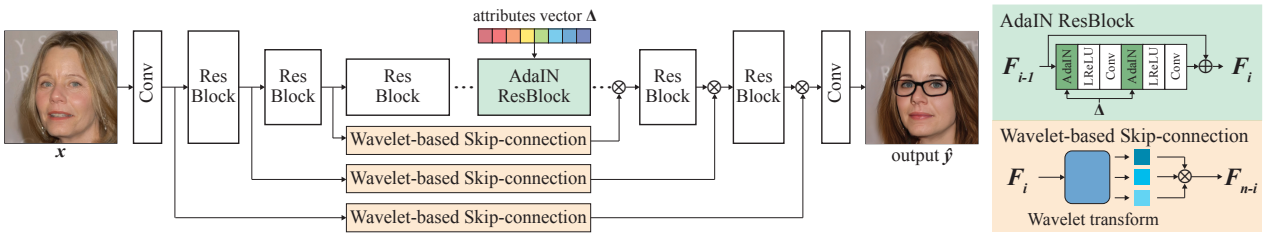
Motivated by the analysis mentioned above, we can prevent the generator from hiding information by simply restricting the hidden information $\boldsymbol{h}$ to be $\boldsymbol{0}$, which is equivalent to restrict $G(\boldsymbol{x}, \boldsymbol{0})$ to be equal to $\boldsymbol{x}$. We notice that the above strategy has been adopted by several existing methods [24, 8]. But this restriction is harmful to the face editing model and makes the edited results remain unchanged during the editing process. Another possible way to prevent the generator from hiding information is to corrupt the hidden information during training. We have tried to add image transformations such as Gaussian blur, flip and rotation on the synthesis results to encourage the generator to give up hiding information. However, we find that the generator still struggles to synthesize rich details and learns to satisfy cycle consistency via a trivial solution.

This paper proposes addressing this problem with a novel framework. The key idea is to prevent the generator from encoding hidden information and encourage it to synthesize perceptible information. By inspecting the hidden information, we find that it is highly related to the high-frequency signals of the input image. Therefore, we choose to utilize the widely used wavelet transformation to decompose the image into domains with different frequencies and take the high-frequency parts to represent rich details.

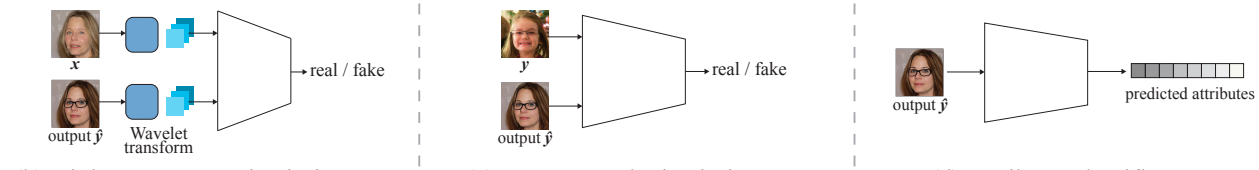
More specifically, we propose solving the problem of cycle consistency from two perspectives. On the one hand, we directly feed the high-frequency signals to the end of the generator to mitigate the generator's struggles for synthesizing rich details so that it gives up to encode the hidden signals. On the other hand, we also employ an additional discriminator to encourage the generator to generate rich details, thus further preventing the generator from finding a trivial solution for cycle consistency.

3.2. Wavelet Transformation

Wavelet transformation has achieved remarkable success in applications that require decomposing images into domains with different frequencies. Following this idea, we adopt a classic wavelet transformation method, the Haar wavelet, which consists of two operations: wavelet pooling and unpooling. Wavelet pooling contains four kernels, $\{LL^\top, LH^\top, HL^\top, HH^\top\}$, where the low (L) and high (H) pass filters are $L^\top = \frac{1}{\sqrt{2}}[1, 1]$ and $H^\top = \frac{1}{\sqrt{2}}[-1, 1]$, respectively. The low-pass filter concentrates on the smooth surface which is mostly related to low-frequency signals while the high-pass filter captures most high-frequency signals like vertical, horizontal and diagonal edges. Figure 4 shows the information of four frequency domains (*i.e.*, **LL**, **LH**, **HL** and **HH**) decomposed from two images by implementing the Haar wavelet transformation. **LL** mainly con-



(a) Wavelet-Based Generator G



(b) High-Frequency Discriminator D_H (c) Image-Level Discriminator D_I (d) Attribute Classifier C

Figure 3: An overview of our proposed method, which contains four parts: a) the Wavelet-Based Generator G; b) the High-Frequency Discriminator D_H; c) the Image-Level Discriminator D_I and; d) the Attribute Classifier C. $\bigoplus$ denotes element-wise plus and $\bigotimes$ denotes channel-wise concatenation.

sists of information in the low-frequency domain, depicting the overall appearance of an image, while **LH**, **HL** and **HH** contain information representing rich details. We find that the combination of **LH**, **HL** and **HH** can be considered a good approximation of the hidden information $\boldsymbol{h}$ (see Figure 2(g)). Besides, wavelet unpooling is employed to exactly reconstruct the original image from the signal components decomposed via wavelet pooling as follows. We first apply a component-wise transposed-convolution on the signal of each component and then sum all resulted features up to precisely reconstruct the image.

Wavelet transformation is usually applied at the image level, but here we implement it at the feature level. Specifically, we first adopt the above-mentioned wavelet pooling to extract features in the domains of different frequencies from different layers of the encoder (see Figure 3). Then we ignore the information of **LL**, and apply wavelet unpooling to **LH**, **HL** and **HH** to reconstruct the information for high-frequency components of the original feature.

3.3. HifaFace

In this section, we introduce our proposed method called HifaFace. It requires two inputs: the input image $\boldsymbol{x}$ and the difference of attributes $\boldsymbol{\Delta}$, and outputs the result $\hat{\boldsymbol{y}}$ with the target attributes. Figure 3 gives an overview of our method which mainly contains the following four parts: 1) the Wavelet-based Generator, 2) the High-frequency Discriminator, 3) the Image-level Discriminator and 4) an attribute Classifier.

Wavelet-Based Generator. Our generator G mainly follows the "encoder-decoder" structure, which contains the encoding part, bottleneck part and decoding part. The input image $\boldsymbol{x}$ is directly fed to the front of the network. We adopt the widely used AdaIN [18] module in the bottleneck

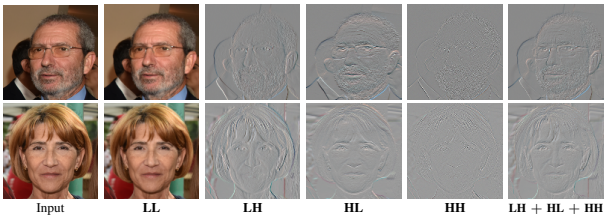


Figure 4: An illustration of wavelet transformation.

part to input the vector of condition attributes $\boldsymbol{\Delta}$. To alleviate the generator's pressure of synthesizing rich details, we propose to use a *wavelet-based skip-connection* to feed the high-frequency information directly to the decoding part of the generator. Specifically, for the i-th layer in the encoding part, we adopt wavelet pooling to extract frequency features $\boldsymbol{E}^i_{\boldsymbol{LL}}, \boldsymbol{E}^i_{\boldsymbol{LH}}, \boldsymbol{E}^i_{\boldsymbol{HL}}$ and $\boldsymbol{E}^i_{\boldsymbol{HH}}$. Then we ignore the low-frequency feature $\boldsymbol{E}^i_{\boldsymbol{LL}}$, and feed the remaining three high-frequency feature maps to the wavelet unpooling module which transforms them to the different frequency domains of the original feature. Finally, we use a skip-connection to feed them to the $(n-i)$-th layer in the decoding part, where n is the number of all layers. This branch aims to maintain the high-frequency details of the input image.

High-Frequency Discriminator. To ensure that synthesis results with rich details can be obtained by the generator, we also adopt a high-frequency discriminator D_H. For both real images and generated images, we first use wavelet pooling to extract their high-frequency features $\boldsymbol{E}_{\boldsymbol{LH}}, \boldsymbol{E}_{\boldsymbol{HL}}$ and $\boldsymbol{E}_{\boldsymbol{HH}}$. Then we feed them into the high-frequency discriminator, thus encouraging the generator to synthesize images with high-frequency information.

Image-Level Discriminator. To encourage the generated image to be realistic, we adopt an image-level discriminator D_I to distinguish between real images and generated images.

Attribute Classifier. To guarantee the consistency between synthesis results and their corresponding target attributes, we design an auxiliary attribute classifier C. Specifically, it consists of K binary classifiers on top of the feature extractor, where K denotes the number of attributes. To achieve a faster training procedure, we first train the classifier only on the labeled dataset. Then when training the whole HifaFace model, we apply the learned classifier to ensure that the generated image possesses the target attributes.

3.4. Objective Function

We train our model using the following losses.

Image-Level Adversarial Loss. We adopt the adversarial loss to encourage the generated image to be realistic. Let $\boldsymbol{y}$ denote the sampled real images, the image-level adversarial loss is defined as:

$$\mathcal{L}_{GAN}^{I} = \mathbb{E}[\log D_I(\boldsymbol{y}) + \log(1 - D_I(G(\boldsymbol{x}, \boldsymbol{\Delta})))]. \quad (1)$$

High-Frequency Domain Adversarial Loss. To encourage the generator to maintain rich details, we apply the adversarial loss in the high-frequency domain. Here, we choose the combination of three domains (*i.e.*, **LH**, **HL** and **HH**) as the high-frequency domain, and define the high-frequency domain adversarial loss as:

$$\mathcal{L}_{GAN}^{H} = \mathbb{E}[\log D_H(\boldsymbol{x}) + \log(1 - D_H(G(\boldsymbol{x}, \boldsymbol{\Delta})))]. \quad (2)$$

Cycle Reconstruction Loss. In order to guarantee the generated image properly preserving the characteristics of the input image $\boldsymbol{x}$ that are invariant to the target attributes, we employ the cycle reconstruction loss which is defined as:

$$\mathcal{L}_{cyc} = \mathbb{E}[\| \boldsymbol{x} - G(G(\boldsymbol{x}, \boldsymbol{\Delta}), -\boldsymbol{\Delta}) \|_1]. \quad (3)$$

Attribute Classification Loss. To ensure that the synthesis result $\hat{\boldsymbol{y}}$ possesses the desired attributes $\boldsymbol{a}_y$, where a_y^k is the k-th element of $\boldsymbol{a}_y$, we introduce the attribute classification loss to constrain the generator G. The attribute classification loss is only applied on the attributes that are changed. Specifically, we use Δ^k to determine whether the k-th attribute has been changed (*i.e.*, $|\Delta^k| = 1$). Suppose p^k is the probability value of the k-th attribute estimated by the classifier C, we have the attribute classification loss as:

$$\mathcal{L}_{ac} = -\mathbb{E}[\sum_{k=1}^{K} \mathbb{1}_{\{|\Delta^k|=1\}}(a_y^k \log p^k + (1 - a_y^k) \log(1 - p^k))], \quad (4)$$

where $\mathbb{1}$ denotes the indicator function, which is equal to 1 when the condition is satisfied. The attribute classification loss restricts the generator to synthesize high-quality images with the corresponding target attributes.

Attribute Regression Loss. Most existing works only consider discrete editing operations [6, 24, 8] or support a limited range of continuous editing [33], making them less practical in real-world scenarios. We expect to precisely control the attributes with a scale factor α, requiring that the generator is capable of synthesizing face images with different levels of attribute editing. Thus the output image can be denoted as $\boldsymbol{y}_\alpha = G(\boldsymbol{x}, \alpha \cdot \boldsymbol{\Delta})$, in which $\alpha \in [0, 2]$. For example, we may want the people in an image to smile. If $\alpha = 0.5$, we expect a smile, if $\alpha = 2$, what we expect is laughing. We calculate the attribute regression loss by:

$$\mathcal{L}_{ar} = \mathbb{E}[(d(\boldsymbol{f}_0, \boldsymbol{f}_\alpha) - d(\boldsymbol{f}_1, \boldsymbol{f}_0)) - (\alpha - 1)], \quad (5)$$

where $\boldsymbol{f}_\alpha$ means the ℓ_2 normalized feature vector extracted by the attribute classifier C: $\boldsymbol{f}_\alpha = C(G(\boldsymbol{x}, \alpha \cdot \boldsymbol{\Delta}))$, in which $d(.,.)$ denotes the ℓ_2 distance of two feature vectors.

Objective Function. The overall loss function of our model is:

$$\mathcal{L} = \lambda_{ar}\mathcal{L}_{ar} + \lambda_{ac}\mathcal{L}_{ac} + \lambda_{GAN}^{I}\mathcal{L}_{GAN}^{I} + \lambda_{GAN}^{H}\mathcal{L}_{GAN}^{H} + \lambda_{cyc}\mathcal{L}_{cyc}, \quad (6)$$

where λ_{ar}, λ_{ac}, λ_{GAN}^{I}, λ_{GAN}^{H} and λ_{cyc} denote the weights of corresponding loss terms, respectively.

3.5. Semi-Supervised Learning

Another important challenge for attribute editing is the limited size of the training dataset. Existing supervised learning based models [6, 24, 33, 8] are typically incapable of handling faces in the wild, especially for those with rich details, pose variance or complex background. However, manually annotating the attributes for a large number of face images is time-consuming, so we adopt a semi-supervised learning strategy to exploit large amounts of unlabeled data. We use our attribute classifier C to predict attributes for all images in the unlabeled dataset $\mathcal{D}_u$, and assign each image with the corresponding prediction result as the pseudo label. Then, both the labeled dataset $\mathcal{D}_l$ and the pseudo-labeled dataset $\mathcal{D}_u$ are employed for training.

4. Experiments

Datasets. We evaluate our model on the CelebA-HQ [20] and FFHQ [21] datasets. The classification model trained on CelebA-HQ is applied to get the pseudo labels for all images in FFHQ. The image resolution is chosen as 256×256 in our experiments.

Implementation Details. In the generator G, the wavelet-based skip-connection is employed between all the three encoding and decoding blocks. We apply the Spectral Normalization (SN) [25] for both D_H and D_I. For the attribute classifier C, we use the pre-trained ResNet-18 [14] as the feature extractor, and two non-linear classification layers are followed for each attribute. The classifier is fine-tuned on CelebA-HQ [20] with 92.8% accuracy on the test set. We use the Adam optimizer [22] with TTUR [16] for training.

Baselines. We compare our approach with two typical types of face editing models: 1) Latent space manipulation

Figure 5: Comparison of attribute-based face editing results obtained by the proposed method and other state-of-the-art approaches. All test images are wild images used in [2].

based models with pre-trained GANs: InterFaceGAN [30] and StyleFlow [2]; 2) Image-to-image translation-based methods: GANimation [28], STGAN [24], RelGAN [33], CAFE-GAN [10], CooGAN [5] and SSCGAN [8].

Evaluation Metrics. We apply several quantitative metrics to evaluate the performance of different face editing methods: 1) Frechét inception distance (FID) [16]; 2) Quality Score (QS) [12], which evaluates the quality of each sample; 3) Acc., which measures the classification accuracy of attributes for synthesized face images; 4) In addition to these commonly used metrics, we utilize the Self-Reconstruction Error (SRE) to quantify the models' capability of synthesizing rich details. Specifically, we compute the ℓ_1 distance between the original image and its projection for the latent space based methods. For the image-to-image translation-based methods, we calculate the ℓ_1 distance between the input image's self-reconstruction image $\overline{\boldsymbol{y}}$ and the reconstructed image $\overline{\boldsymbol{x}}$ of the self-reconstruction result.

4.1. Comparison with State of the Art Methods

In this section, we compare our face editing method with other existing approaches from two perspectives: attribute-based face editing and arbitrary face editing.

Methods	FID ↓	Acc. ↑	QS ↑	SRE ↓
GANimation	15.72	64.7	0.710	0.145
STGAN	14.78	83.2	0.543	0.041
RelGAN	10.13	83.6	0.729	0.024
CAFE-GAN	-	81.9	-	-
CooGAN	-	83.8	-	-
SSCGAN	4.69	94.2	-	-
InterFaceGAN	-	-	-	0.163
HifaFace	**4.04**	**97.5**	**0.803**	**0.021**

Table 1: Quantitative results of different methods.

Attribute-Based Face Editing We compare our method with some recent works: GANimation [28] STGAN [24], RelGAN [33], InterFaceGAN [30] and StyleFlow [2], and show the qualitative results in Figure 5. GANimation utilizes a masking mechanism to force the generator to edit the regions that need to be edited. Unsatisfactory results indicate that the attention mask mechanism does not work well for the face editing task. STGAN is a typical method that adopts the reconstruction loss between the input image and its self-reconstruction instead of cycle consistency.

(a) Input (b) Interpolation of the "eyeglasses" attribute, in range of $[0.4, 2.0]$ with an interval of 0.2.

Figure 6: Interpolation results obtained by RelGAN [33], InterFaceGAN (IFGAN) [30] and our HifaFace.

Methods	Arched Eyebrows	Black Hair	Blond Hair	Brown Hair	Eye-glasses	Gray Hair	Heavy Makeup	Male	Mouth Open	Mustache	No Beard	Smiling	Young	Average
GANimation	69.2	74.0	52.8	54.1	87.2	77.1	75.9	65.9	57.2	54.8	44.6	67.7	57.5	64.7
STGAN	80.2	76.3	78.9	82.9	86.1	84.3	87.3	86.5	88.4	75.7	90.3	84.6	80.3	83.2
RelGAN	85.4	74.8	84.7	91.4	93.9	91.4	79.5	73.7	91.6	80.5	91.6	69.9	78.3	83.6
CAFE-GAN	-	-	88.1	-	-	-	-	95.2	97.2	40.1	-	-	88.6	81.9
CooGAN	89.1	80.1	84.2	64.1	99.8	-	-	85.0	94.9	59.4	96.3	-	85.2	83.8
SSCGAN	96.5	**99.3**	-	-	**99.9**	-	-	**99.1**	**99.9**	65.7	-	-	**99.0**	94.2
HifaFace	**98.4**	94.9	**98.4**	**92.7**	98.9	**98.3**	**96.0**	98.9	99.0	**98.2**	**97.7**	**98.3**	97.5	**97.5**

Table 2: The attribute editing accuracy of our method and other image-to-image translation-based face editing approaches.

Methods	FID ↓	Acc. ↑	QS ↑	SRE ↓
w/ VS in G	5.50	94.8	0.769	0.068
w/o WS in G	5.22	96.2	0.790	0.049
w/o D_H	5.34	96.4	0.762	0.057
HifaFace	**4.04**	**97.5**	**0.803**	**0.021**

Table 3: Quantitative results for ablation studies.

We observe that their results keep the rich details but don't have the desired attributes. On the contrary, RelGAN is designed based on cycle consistency. We observe that their results lose rich details and have a lot of artifacts. Meanwhile, for the latent space based methods InterFaceGAN [30] and StyleFlow [2], they lose too many details of the original input image. Compared to those existing approaches, our method obtains impressive results with higher quality, which not only preserve the rich details of the input face but also perfectly satisfy the desired attributes. Table 1 compares the quantitative results of different methods, from which we can see that our method clearly outperforms others considering the values of FID and QS, and generates face images containing more rich details as well as low self-reconstruction errors. Furthermore, the high Acc. value indicates that our method has better capability of editing facial attributes. In Table 2, we evaluate the attribute editing accuracy for each attribute. We can see that our method obtains the best results for most attributes.

Arbitrary Face Editing. In real-world scenarios, a fine-grained and wider-range control for each attribute is very useful. Previous works such as RelGAN [33] and InterFaceGAN [30] also provide continuous control for facial attribute editing. We perform attribute interpolation in the range of $\lambda \in [0.4, 2.0]$ with an interval of 0.2, where $\lambda = 0$ and $\lambda = 1$ denote that the desired attributes are equal to the source and target attributes, respectively. We show the visualization results in Figure 6, which shows that our model generates smoother and higher-quality interpolation results compared to InterFaceGAN [30]. When λ is larger than 1, interpolation results of RelGAN remain almost unchanged, while our method tends to strengthen the attribute of wearing eyeglasses. When λ is set to 2, the eyeglasses in the outputted face image of our method will be replaced by a pair of sunglasses. Both qualitative and quantitative results demonstrate that our method has a significantly stronger capability of implementing arbitrary face editing.

4.2. Ablation Studies

In this section, we conduct experiments to validate the effectiveness of each component of the proposed HifaFace. We mainly study the following three important issues: 1) the wavelet generator and discriminator; 2) the attribute regression loss; 3) the semi-supervised learning strategy.

Wavelet-Based Generator and Discriminator To verify the effectiveness of our proposed wavelet-based generator and discriminator, we evaluate the performance of several variants of our method. Specifically, we consider the following four variants: 1) our full model; 2) the proposed model without the wavelet-based skip-connection in the Generator (w/o WS in G); 3) the proposed model without the high-frequency Discriminator (w/o D_H); 4) the proposed model with the vanilla skip-connection (w/ VS in

(a) Input (b) Interpolation of the "old" attribute, in range of $[0.4, 2.0]$ with an interval of 0.2.

Figure 7: Comparison of interpolation results obtained using our models without and with the attribute regression loss.

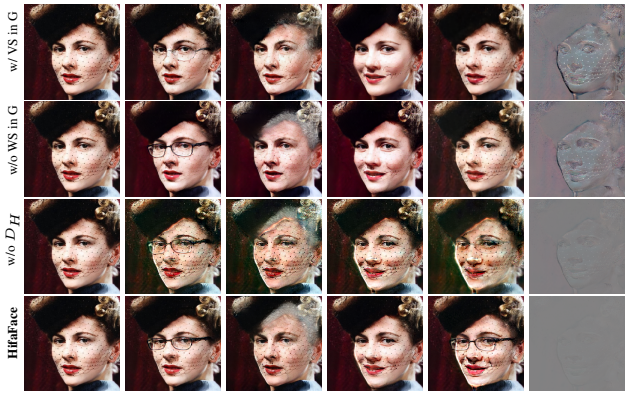


Figure 8: Ablation studies of our proposed model.

Figure 9: Comparison of synthesis results obtained by our models without and with semi-supervised learning.

G). Qualitative and quantitative results of these methods are shown in Figure 8 and Table 3, respectively, from which we can draw two conclusions as follows. 1) The wavelet-based generator and discriminator are extremely important for the overall framework, since it is impossible to obtain satisfactory face editing results without them. 2) Our generator does not encode hidden information in its output image. This is intuitively demonstrated in Figure 8, where we expect to add eyeglasses on the synthesized smile face images (the fifth column). We can see that only our full model is able to satisfactorily achieve this goal. The hidden information of each model is shown in the rightmost column of Figure 8.

Attribute Regression Loss. The attribute regression loss is introduced to achieve arbitrary face editing, which aims to obtain both high-quality interpolated results and extrapolated results, as shown in Figure 7. With our proposed attribute regression loss, we can get a smooth and wider-range control for each attribute.

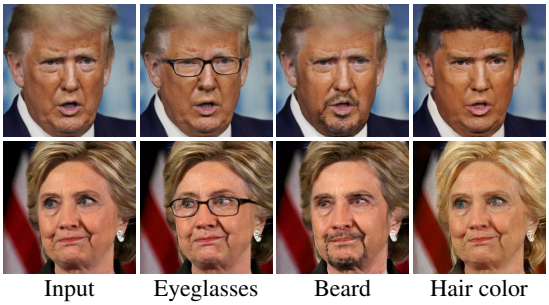


Figure 10: Examples of face editing results for wild images obtained by our HifaFace.

Semi-Supervised Learning. To validate the effectiveness of our semi-supervised learning strategy, we conduct an ablation study and show the results in Figure 9. It can be observed that benefiting from the utilization of large amounts of data, the quality of synthesis results obtained by the method with semi-supervised learning is significantly better, especially for some attributes such as "eyeglasses" and "old", where very few labeled data are available.

4.3. Editing Wild Faces

Finally, we demonstrate the strong capability of HifaFace by editing wild face images downloaded from the Internet. As shown in Figure 10, our method works well for face images under various poses and expressions.

5. Conclusion

In this paper, we first revisit *cycle consistency* in current face editing frameworks and observed an interesting phenomenon called steganography. We then propose a novel model named HifaFace to address this problem. The key idea of our method is to adopt a wavelet-based generator and a high-frequency discriminator. Moreover, we also designed a novel attribute regression loss to achieve arbitrary face editing on a single attribute. Extensive experiments demonstrate the superiority of our framework for high-fidelity and arbitrary face editing. Hopefully, this paper will be able to inspire researchers for solving the similar problems of cycle consistency in many other tasks.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE international conference on computer vision*, pages 4432–4441, 2019. 2

[2] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ArXiv*, abs/2008.02401, 2020. 2, 6, 7

[3] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6713–6722, 2018. 2

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2

[5] Xuanhong Chen, B. Ni, Naiyuan Liu, Ziang Liu, Yiliu Jiang, Loc Truong, and Q. Tian. Coogan: A memory-efficient framework for high-resolution facial attribute editing. *ArXiv*, abs/2011.01563, 2020. 6

[6] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 1, 2, 3, 5

[7] Casey Chu, A. Zhmoginov, and Mark Sandler. Cyclegan, a master of steganography. *ArXiv*, abs/1712.02950, 2017. 1, 2, 3

[8] Wenqing Chu, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Sscgan: Facial attribute editing via style skip connections. *ECCV*, 2020. 1, 2, 3, 5, 6

[9] Yue Gao, Yuan Guo, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. Artistic glyph image synthesis via one-stage few-shot learning. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019. 2

[10] Jeong gi Kwak, David K Han, and Hanseok Ko. Cafe-gan: Arbitrary face attribute editing with complementary attention feature. *ECCV*, 2020. 6

[11] Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 2

[12] Shuyang Gu, Jianmin Bao, D. Chen, and Fang Wen. Giqa: Generated image quality assessment. *ArXiv*, abs/2003.08932, 2020. 6

[13] Shuyang Gu, Jianmin Bao, Hao Yang, D. Chen, Fang Wen, and L. Yuan. Mask-guided portrait editing with conditional gans. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2019. 2

[14] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[15] Zhenliang 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*, 28:5464–5478, 2019. 1

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. 5, 6

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018. 2

[18] X. Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, 2017. 4

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 1

[20] Tero Karras, Timo Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2018. 2, 5

[21] Tero Karras, S. Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019. 2, 5

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 5

[23] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2

[24] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, E. Ding, W. Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3677, 2019. 1, 2, 3, 5, 6

[25] Takeru Miyato, T. Kataoka, Masanori Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *ArXiv*, abs/1802.05957, 2018. 5

[26] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016. 2

[27] Horia Porav, Valentina Musat, and Paul Newman. Reducing steganography in cycle-consistency gans. In *CVPR Workshops*, pages 78–82, 2019. 2

[28] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the ECCV*, 2018. 6

[29] E. Sánchez and M. Valstar. A recurrent cycle consistency loss for progressive face-to-face synthesis. *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 53–60, 2020. 1, 2

[30] Yujun Shen, Ceyuan Yang, X. Tang, and B. Zhou. Interfacegan: Interpreting the disentangled face representation

learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2020. 2, 6, 7

[31] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019. 2

[32] Yizhi Wang, Yue Gao, and Zhouhui Lian. Attribute2font: creating fonts you want from attributes. *ACM Transactions on Graphics (TOG)*, 39(4):69–1, 2020. 2

[33] P. Wu, Yu-Jing Lin, Che-Han Chang, E. Chang, and S. Liao. Relgan: Multi-domain image-to-image translation via relative attributes. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5913–5921, 2019. 1, 2, 5, 6, 7

[34] Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K Kalra, Yi Zhang, Ling Sun, and Ge Wang. Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE transactions on medical imaging*, 37(6):1348–1357, 2018. 2

[35] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. 2

[36] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019. 2

[37] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 2

[38] Tinghui Zhou, Yong Jae Lee, Stella X Yu, and Alyosha A Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1191–1200, 2015. 2

[39] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 117–126, 2016. 2

[40] Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4032–4040, 2015. 2

[41] Jun-Yan Zhu, T. Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. 2