HairMapper: Removing Hair from Portraits Using GANs

Yiqian Wu[1,2] Yong-Liang Yang[3] Xiaogang Jin[1,2] *
[1]State Key Lab of CAD&CG, Zhejiang University
[2]ZJU-Tencent Game and Intelligent Graphics Innovation Technology Joint Lab
[3]University of Bath
onethousand@zju.edu.cn, y.yang@cs.bath.ac.uk, jin@cad.zju.edu.cn



Figure 1. Given portrait images with their faces partially occluded by hair (a), our method is able to generate portraits without hair while preserving facial identity (b). After removing the effects of hair, the resulting portrait images can be well employed in hair design by simply blending the clean face with some hairstyle templates (c and d) without the interference from existing hair. Our results can also benefit 3D face reconstruction [12] by using the clean face textures (e, f) generated by our method (rows 2 and 4) in contrast to the results of the original images (rows 1 and 3).

Abstract

Removing hair from portrait images is challenging due to the complex occlusions between hair and face, as well as the lack of paired portrait data with/without hair. To this end, we present a dataset and a baseline method for removing hair from portrait images using generative adversarial networks (GANs). Our core idea is to train a fully connected network $HairMapper$ to find the direction of hair removal in the latent space of StyleGAN for the training stage. We develop a new separation boundary and diffuse method to generate paired training data for males, and a novel "female-male-bald" pipeline for paired data of females. Experiments show that our method can naturally deal with portrait images with variations on gender, age, etc. We validate the superior performance of our method by comparing it to state-of-the-art methods through extensive experiments and user studies. We also demonstrate its applications in hair design and 3D face reconstruction.

1. Introduction

Hair is not only an important component of the human body, but also a key element of personality and fashion. However, the presence of hair in a portrait image poses significant challenges for digital hair design and 3D face reconstruction. Regarding hair design, a direct overlay of the new hair can easily cause problems due to mixing up with the old hair, while replacing the old hair with the new hair requires error-prone matting and inpainting techniques [38]. For 3D face reconstruction, most existing methods cannot handle the hair in front of the face and it remains in the texture [10, 23], resulting in noticeable artifacts of the re-

*Corresponding author.

constructed face (see Fig. 1f). This motivates us to develop a hair manipulation method that can naturally remove hair from portraits to facilitate such real applications.

Although image inpainting methods [24, 25, 39] can help to generate or edit facial structure, they only allow manipulating face semantic attributes at the image level. Thanks to the development of StyleGAN [19], the exploration in its latent space [3, 29, 31, 36, 37] enables editing facial semantics at the manifold level. However, none of these methods is able to remove hair while preserving facial identity due to the following main challenges. First, a dataset of paired portraits with/without hair is not available. Moreover, it is not easy to prepare such a dataset, especially for females. Second, besides the lack of ground truth of "bald woman", synthetic portrait generation based on StyleGAN is infeasible as "bald woman" is an invalid semantic combination. Third, hair removal is not a simple inpainting task, since the newly generated contents for the original hair region should be compatible with the original face in terms of skin color, shadow effect, etc. However, extreme light conditions, shadows, and different hairstyles can easily cause imperfections given the variety of faces.

To address the above challenges, we present a novel method that can effectively remove hair from portraits while preserving original face semantics and portrait quality, even for female portraits with long and complex hair. In the StyleGAN latent space, hair removal is not a simple linear mapping problem. To find a specific hair manipulation path and avoid expensive data annotation, we design two different pipelines to generate paired latent codes with/without hair for males and females while keeping their facial identities. The paired data with/without hair are used to train a fully connected network $HairMapper$ to manipulate the latent code of a real portrait for hair removal during testing. The final result is obtained by exploiting Poisson editing to blend the mapped portrait with the original portrait. The experiments demonstrate that our method results in high-quality portraits of different ages and gender groups. The user studies further show that our work can generate satisfactory outputs that accord with human preferences.

The major contributions of our paper are: 1) We introduce an automatic method to remove hair from real portrait images. It can generate a new portrait without hair while preserving facial identity. 2) We develop a novel "female-male-bald" pipeline to generate bald female data that does not exist in the StyleGAN latent space, and use a fully connected network to find the hair removal path in the latent space. 3) We create the first dataset that contains 6,000 high-quality portrait images with hair removed.

2. Related Work

Portrait Image Synthesis Thanks to the seminal work of generative adversarial networks [13], researchers have been exploring different generative models for portrait generation and editing. Progressively growing generator and discriminator [17] can achieve high-resolution portrait images. Freeform user input like freehand sketches [8, 9] and masks [14, 22] can also be used as conditions to generate realistic portraits. StyleGAN [18–20] has a disentangled latent space and can generate high-quality portraits by manipulating different semantics. By projecting images into the latent space of StyleGAN [1, 2, 30, 35], the input image's semantic information can be embedded into a latent code, thus the face editing task is transformed into simple latent code manipulation.

StyleGAN Latent Space Finding the general rule of latent code manipulation under the StyleGAN framework is an effective way for portrait editing. It can be regarded as seeking a specific path in the StyleGAN latent space. Establishing a separation boundary in the latent space [31] allows editing face semantics through a simple linear combination. However, the separation boundary may significantly change facial identity. To address this problem, Coarse-to-Fine [36] further proposes a pipeline to refine the separation boundary, thus better preserving facial identity. The paths can also be learned by Principal Component Analysis or self-supervised approaches without annotation [15, 16]. Instead of working in the StyleGAN latent space, the latent semantic path can be discovered by directly decomposing the pre-trained weights [32], or in a new space of channel-wise better-disentangled style parameters [37]. By adding or subtracting the standard deviation of a specific style channel, facial attributes like the amount of hair or hair greyness can be effectively controlled. StyleClip [29] further uses the Contrastive Language-Image Pre-training model to achieve text-based semantic image manipulations by dynamically finding the direction of a given text prompt in latent space. Besides local semantic attributes, the nonlinear path in the latent space for a continuous global aging process can be achieved by a regression task [4]. Another solution of attribute-conditioned sampling and editing is continuous normalizing flows in the latent space [3].

Hair Manipulation Hair manipulation provides an effective way for users to design and visualize different hairstyles. Layers that are extracted from human faces can be applied to beard removal [27]. 3D models offer more geometric information for hair generation and manipulation. Based on the modeling results, hair editing can be achieved by smoothly changing each strand [5, 6]. To circumvent the expensive 3D modeling and rendering progress, neural networks are exploited to directly synthesize realistic hair for 2D images. Simple and sparse guided strokes can provide high-level hair structure information for target hairstyle generation [7, 28, 38]. Painted masks or reference photos

give users more degrees of freedom for interactive hair manipulation [33, 34, 40]. Although the above methods have good performance in generating high-quality portraits with different hairstyles, the facial identity can be affected when synthesizing the new portrait as a whole from the input. On the other hand, if a new hairstyle is composed onto the input portrait directly, artifacts can easily arise such as redundant hair, hair-face inconsistency, etc.

3. Method

3.1. Overview

Fig. 2 illustrates the pipeline of our method. Given a real portrait image of either male or female with any hairstyle, our goal is to find a specific path in the StyleGAN latent space that can completely remove hair while preserving portrait structure and facial identity. The basic idea is to first generate paired latent codes of portraits with/without hair, then train a $HairMapper$ which can generalize well for plausible hair removal. The key challenge is how to generate high-quality paired data (latent codes) for training, especially for females. To this end, we design two pipelines for males and females as summarized below.

To manipulate hair and gender for paired data generation, we first randomly sample two latent code datasets and get their hair scores through a hair classifier (Subsection 3.2). Then, we leverage the datasets and hair scores to train a male hair separation boundary. We also use the gender transition results of StyleFlow to train a gender separation boundary (Subsection 3.3). For male portrait with hair, inspired by [36], we manipulate hair based on the male hair separation boundary to generate a new portrait without hair, then diffuse the new portrait with the original portrait to obtain quality paired latent codes (Subsection 3.4). The paired male latent codes are used to train a male $HairMapper$ (Subsection 3.5) to find the latent path of hair removal for males. For females, since StyleGAN [19] prevents the sampling of invalid combinations (*e.g.*, bald-females, long-haired males), it is impossible to generate bald female data with a single linear transformation. Therefore, we introduce a novel "female-male-bald" pipeline to solve this problem. We first leverage the aforementioned gender separation boundary to edit the latent code, and get the male portrait that has the same face postures and skin characteristics as the original female portrait. After that, we feed the latent code of the male portrait to the pre-trained male $HairMapper$ and achieve a bald male portrait, and then diffuse it with the original female portrait, resulting in paired female latent codes (Subsection 3.6). Finally, we train the final $HairMapper$ on paired data of both males and females. For a real input portrait image (male or female), we first embed it into the latent space using the $e4e$ StyleGAN encoder [35], then we feed the latent code to the final $HairMapper$. After that, we utilize image blending to achieve the final result (Subsection 3.7).

	description
D_0	codes in each layer are same and limited to W
D_{noise}	same with D_0, but add noise
$e4e$	codes in each layer close to the distribution of W

Table 1. Comparison between latent codes from D_0, D_{noise} and latent codes generated by $e4e$ encoder.

3.2. Data Preparation

The $e4e$ encoder [35] can map a real image into its latent code w^+, while balancing the distortion-editability and distortion-perception trade-offs. w^+ consists of a series of latent vectors w with low variance, each close to the distribution of the W latent space of StyleGAN. We employ $e4e$ to encode real images. In order to apply our method to latent codes of real images, we sample two w^+ latent code set D_0 and D_{noise} (see details in Tab. 1) that are close to the distribution of $e4e$ encoder results.

D_0. For each latent code $w_i^+ \in D_0$, we sample a latent code $w_i \in W$, and set codes in each layer of w_i^+ as w_i:

$$D_0 = \{w_i^+ \mid_{i=0}^{n}\}, w_i^+ = [w_i, \cdots, w_i], \tag{1}$$

where $[\cdots]$ denotes the stack of layers.

D_{noise}. For each latent code $w_i^+ \in D_{noise}$, we sample a latent code $w_i \in W$, and set codes in each layers of w_i^+ as w_i, then add noise to each layer:

$$D_{noise} = \{w_i^+ \mid_{i=0}^{n}\}, w_i^+ = [w_i + n_{i0}, \cdots, w_i + n_{i17}], \tag{2}$$

where $n_{i0}, \cdots, n_{i17}$ denote noises that are added to each layer (see details of noise sample in Section 4).

To get the hair score of each portrait, we propose to train a hair classifier and leverage the CelebAHQ-mask [22] which includes attribute *Bald* as the training dataset. Since CelebAHQ-mask only contains 602 bald images, we add 184 bald images collected from the Internet to the dataset. Then we leverage ResNet50 to train a hair classifier C_{hair}. We input the latent codes in D_0 and D_{noise} to StyleGAN, then feed the output images into C_{hair}, the bald portrait's latent code will be scored as $s_{hair} = 0$ by C_{hair}, other portrait's latent code will be scored as $s_{hair} = 1$.

3.3. Separation Boundary Training

In this section, we train two separation boundaries in the latent space of StyleGAN, which provide general directions for male hair removal and gender transformation.

We use the separation boundary training algorithm in InterFaceGAN [31] to train a male hair separation boundary.

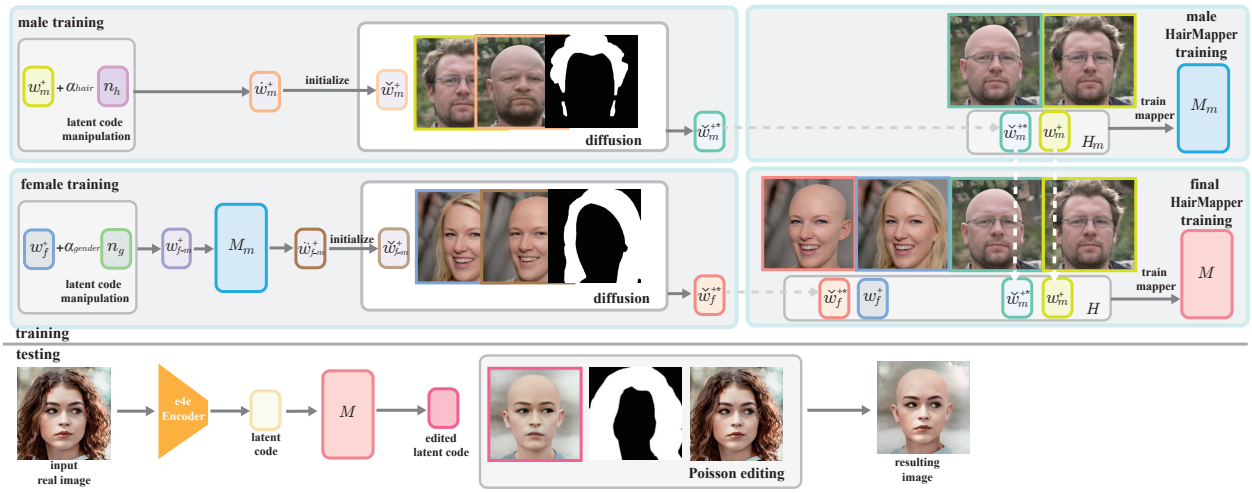


Figure 2. The pipeline of our $HairMapper$. We use the same color to mark the latent code and its corresponding portrait generated by StyleGAN. In the training stage, for males (top row), given a male latent code w_m^+ with hair, we first edit it by leveraging the normal vector n_h of the male hair separation boundary to achieve $\dot{w}_m^+$ and its corresponding bald portrait. To better preserve facial identity, we further leverage $\dot{w}_m^+$ to initialize the optimization of $\check{w}_m^+$ to obtain $\check{w}_m^{+*}$ by diffusing the bald portrait with the original portrait. We construct dataset H_m by composing pairs of male latent codes $\check{w}_m^{+*}$ and w_m^+, and train a $HairMapper$ M_m for males. For females (middle row), we first edit the input latent code w_f^+ by leveraging the normal vector n_g of the gender separation boundary to get a male portrait with latent code $w_{f\to m}^+$. Then we directly use the pre-trained M_m to edit $w_{f\to m}^+$ to get $\ddot{w}_{f\to m}^+$ and its corresponding male portrait without hair. After that, we perform the same diffusion process to obtain $\check{w}_f^{+*}$ as the paired latent code of w_f^+. We construct our complete dataset H with paired latent codes of both males and females, and train the final $HairMapper$ M. In the testing stage (bottom row), we first encode an input real image by $e4e$ encoder, then feed its latent code into M to remove hair. After blending with the input image, we achieve our final result.

We use D_0 as the training set, an SVM is trained on the male latent codes and their corresponding hair scores (hair-1 and bald-0), then obtain the male hair separation boundary's normal vector n_h.

We use StyleFlow [3] to edit the *Gender* attribute for 1,000 latent codes, resulting in 1,000 pairs of male-female latent codes. Then an SVM is trained on the latent code pairs and their corresponding scores (male-1 and female-0) to obtain the gender separation boundary's normal vector n_g. This gender separation boundary has better performance in preserving skin characteristics and face posture than that which we simply train on unpaired latent codes and gender scores (see comparisons in Section 5).

3.4. Male Hair Removal

In the following, we describe the pipeline to generate paired latent codes with/without hair for males. As the bald man data is valid in latent space, we directly use hair separation boundary to edit the original latent code of the male portrait and get the baldness information as:

$$\dot{w}_m^+ = w_m^+ + \alpha_{hair} \times n_h, \tag{3}$$

where α_{hair} is a hyper-parameter that controls the distance that w_m^+ moves along n_h, w_m^+ denotes the original latent code, and $\dot{w}_m^+$ denotes the edited latent code.

The image generated from $\dot{w}_m^+$ is a male portrait without hair but its facial identity is changed (see Fig. 3). In order to get the latent code whose hair is removed while preserving facial identity, inspired by Coarse-to-Fine [36], we utilize a method to diffuse the baldness semantic information with the original portrait. We first compose the original portrait and the bald male portrait to obtain a synthetic image:

$$\begin{aligned} \check{X}_m &= G(w_m^+) \odot (1 - m_h) + G(\dot{w}_m^+) \odot m_h, \\ m_h &= FaceParsing(G(w_m^+)), \end{aligned} \tag{4}$$

where $\odot$ is the element-level multiplication, G is the StyleGAN generator, $G(w_m^+)$ denotes the original portrait, $G(\dot{w}_m^+)$ denotes the bald male portrait. m_h denotes the hair mask, $FaceParsing$ is a network that can extract the face mask excluding hair (see details in Section 4).

The synthetic image is used as the prior information to



Figure 3. From left to right: original image $G(w_m^+)$, edited image $G(\dot{w}_m^+)$, synthetic image $\check{X}_m$, and diffusion result $G(\check{w}_m^{+*})$.

achieve a bald portrait that preserves the facial identity of $G(w_m^+)$. We diffuse $G(w_m^+)$ with $G(\ddot{w}_m^+)$ by optimizing a latent code (denoted as $\check{w}_m^+$) to obtain latent code $\check{w}_m^{+*}$ whose corresponding portrait has the same face identity as $G(w_m^+)$ and has the same baldness information as $G(\ddot{w}_m^+)$.

In the optimization, Coarse-to-Fine [36] uses the original latent code (w_m^+ in our case) to initialize the latent code $\check{w}_m^+$ that needs to be optimized. But we find that directly eliminating hair by optimization is difficult due to the complexity of the hair structure (see details in Section 5). Fortunately, face semantics can be easily manipulated by optimization. Thus we leverage $\ddot{w}_m^+$ to initialize $\check{w}_m^+$, then the optimization mainly focuses on transforming the face region of $G(\ddot{w}_m^+)$ to that of $G(w_m^+)$, which is much easier than transforming from hair to skin. The optimization and its loss $\mathcal{L}_{diffuse}$ are defined as:

$$\begin{aligned}
\check{w}_m^{+*} &= \arg\min_{\check{w}_m^+} \mathcal{L}_{diffuse}, \\
\mathcal{L}_{diffuse} &= \lambda_{rec}\mathcal{L}_{rec} + \lambda_{per}\mathcal{L}_{per}, \\
\mathcal{L}_{rec} &= \|\check{X}_m \odot (1 - m_h) - G(\check{w}_m^+) \odot (1 - m_h)\|_2, \\
\mathcal{L}_{per} &= \|\phi(\check{X}_m) - \phi(G(\check{w}_m^+)))\|_2,
\end{aligned} \tag{5}$$

where λ_{rec} and λ_{per} are the weights to control face reconstruction loss $\mathcal{L}_{rec}$ and overall perceptual loss $\mathcal{L}_{per}$, ϕ denotes the VGG16 model, and $\mathcal{L}_{per}$ is the L_2 distance between activation maps of ϕ.

3.5. Male HairMapper Training

We use the procedure described in Subsection 3.4 to process 10,774 male latent codes in D_0 and 5,328 male latent codes in D_{noise}, yielding the male latent code dataset H_m:

$$H_m = \{(w_{mi}^+, \check{w}_{mi}^{+*}) \mid_{i=0}^{n_m}\}, \tag{6}$$

where $n_m = 16,102$ is the size of H_m.

We do not use H_m to train a "fine separation boundary" as in [36] for hair removal, since the hair structure is complicated and cannot be removed by a single linear combination of latent code and separation boundary (see details in supplementary material). Inspired by the Latent Mapper of StyleClip [29], we propose to use a fully connected network to find a specific path:

$$\ddot{w}_m^+ = w_m^+ + \beta \times M_m(w_m^+), \tag{7}$$

where M_m is the fully connected network, and β is the hyper-parameter.

The objective function to train M_m is :

$$\mathcal{L} = \lambda_l\mathcal{L}_{latent} + \lambda_h\mathcal{L}_{hair} + \lambda_f\mathcal{L}_{face} + \lambda_i\mathcal{L}_{id}, \tag{8}$$

where $\mathcal{L}_{latent}$ penalizes the difference between $\ddot{w}_m^+$ and $\check{w}_m^{+*}$:

$$\mathcal{L}_{latent} = \|\ddot{w}_m^+ - \check{w}_m^{+*}\|_2, \tag{9}$$

$\mathcal{L}_{hair}$ penalizes the difference of the hair region between the two images generated from $\ddot{w}_m^+$ and $\check{w}_m^{+*}$:

$$\mathcal{L}_{hair} = \|G(\ddot{w}_m^+) \odot m_h - G(\check{w}_m^{+*}) \odot m_h\|_2, \tag{10}$$

and $\mathcal{L}_{face}$ penalizes the difference of the face region between the image generated from $\ddot{w}_m^+$ and the original image w_m^+:

$$\mathcal{L}_{face} = \|G(\ddot{w}_m^+)\odot(1-m_h) - G(w_m^+)\odot(1-m_h)\|_2. \tag{11}$$

We also employ the identity loss [30] to penalize the difference of the identity between the image generated from $\ddot{w}_m^+$ and the original image:

$$\mathcal{L}_{id} = 1 - \langle R(G(\ddot{w}_m^+)), R(G(w_m^+))\rangle, \tag{12}$$

where R is the ArcFace network [11], and $\langle\cdot,\cdot\rangle$ computes the cosine similarity.

3.6. Female Hair Removal

StyleGAN [19] prevents the sampling of invalid combinations, such as *bald-female*. Also, simply using male hair separation boundary cannot completely remove hair and significantly changes facial identity (see detailed explanation in supplementary material). Hence we propose a **female-male-bald** pipeline to process female portraits as shown in Fig. 4. Our approach is to generate a bald-man portrait first, and diffuse it with the original female portrait.

First, we leverage the gender separation boundary to edit the latent code w_f^+ of the original female portrait. The edited latent code $w_{f\to m}^+$ can generate a male portrait which has similar skin characteristics and face posture with w_f^+:

$$w_{f\to m}^+ = w_f^+ + \alpha_{gender} \times n_g. \tag{13}$$

Second, we feed $w_{f\to m}^+$ to the pre-trained male $HairMapper$ M_m to get a bald male latent code $\ddot{w}_{f\to m}^+$:

$$\ddot{w}_{f\to m}^+ = w_{f\to m}^+ + \beta \times M_m(w_{f\to m}^+). \tag{14}$$

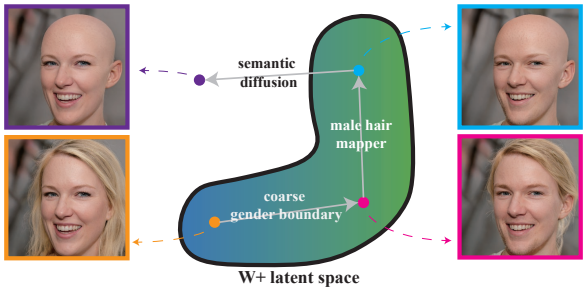


Figure 4. Example **female-male-bald** pipeline. The orange box highlights the original female image ($G(w_f^+)$), the pink box highlights the male image ($G(w_{f\to m}^+)$) edited by n_g, the blue box shows the bald male image ($G(\ddot{w}_{f\to m}^+)$) edited by M_m, and the purple box highlights the diffusion bald female result ($G(\check{w}_f^{+*})$).

We then use the hair mask to obtain a synthetic image $\check{X}_f$ as the prior information as we did for male portraits:

$$\check{X}_f = G(w_f^+) \odot (1 - m_h) + G(\ddot{w}_{f\to m}^+) \odot m_h. \quad (15)$$

Finally, we apply the diffusion method described in Subsection 3.4 to generate the result $\check{w}_f^{+*}$, and we also use $\ddot{w}_{f\to m}^+$ to initialize the optimization.

After processing 8,034 latent codes in D_0 and 5,878 latent codes in D_{noise}, we build a dataset H including female latent code pairs and those male latent code pairs in H_m:

$$H = \{(w_{mi}^+, \check{w}_{mi}^{+*}) \mid_{i=0}^{n_m}\} \bigcup \{(w_{fi}^+, \check{w}_{fi}^{+*}) \mid_{i=0}^{n_f}\}, \quad (16)$$

where $n_m = 16,102$ and $n_f = 13,912$. We train the final $HairMapper$ M on H by following the same procedure in Subsection 3.5.

3.7. Image Blending

For real image editing, we first use the $e4e$ encoder to encode the real image and get its latent code. Then we feed the latent code to M and obtain the resulting latent code and its corresponding image. Finally, we apply Poisson editing to blend the resulting image with the original face.

However, the hair mask cannot cover the shadow under the hair, and the Poisson editing forces the color to be consistent on the edge of the hair mask and maintain the gradient of the synthetic image (keep the light difference between shadow and skin unchanged). As a result, the resulting image cannot perfectly integrate with the original face as the face region is wrongly "lightened" (see the top row in Fig. 5). To resolve this, we propose to dilate and blur the hair mask, which can effectively cover the shadow (see the bottom row in Fig. 5) and achieve a smooth transition.



Figure 5. Image blending using the original hair mask (top) vs. the dilated and blurred hair mask (bottom). Image from left to right: hair mask visualization, synthetic image, and blended image.

4. Implementation Details

Experimental Environment: We use StyleGAN2-ada [18] PyTorch model in our experiments. The experimental platform is a desktop PC with i7-9700 3.0GHz CPU, 16 GB memory, and GeForce RTX 2060 GPU of 13.9 GB memory, all images in the training and testing stage are at 1024^2 resolution. **Data Generation**: We set truncation-psi as 0.8 for latent code samples. For sampling noises of D_{noise}, we first compute the standard deviation w_{std} and the noise scale $scale_{noise} = (0.5w_{std})^2$, then we use the noise scale to control the weight of randomly sampled noise and get the latent code noise in Equ. 2. **Diffusion**: We use the Adam solver [21] as the optimizer with a learning rate of 0.01, and train each latent code for 150 iterations. **HairMapper**: We use 8 fully connected layers to build M_m and M. For each mapper, we use the Adam solver with a learning rate of 0.005 to train M_m for 52,000 steps and M for 26,000 steps. **Style Preserving**: The first 8 layers of the latent code control the coarse feature of the face (posture, shape), keeping the rest of the layers unchanged can preserve fine features (skin appearance). Thus, we only edit the first 8 layers of the latent codes (using mapper or separation boundary). **HairMask**: We use the $FaceParsing$ network * proposed by MaskGAN [22] to generate hair mask for each portrait. For blending, we set dilate kernel size as 50 and blur kernel size as 30 for mask dilating and blurring. **Hyper parameter**: For diffusion, we set $\lambda_{rec} = 1.0$ and $\lambda_{per} = 5e^{-5}$. For mapper, we set $\beta = 0.4$ in the training stage, and set $\beta = 0.4 \times 1.2 = 0.48$ in the testing stage. In Equ. 8, we set $\lambda_l = 0.1$, $\lambda_h = 0.4$, $\lambda_f = 0.4$, and $\lambda_i = 0.4$. For separation boundary, we propose to use two algorithms to compute the hyper-parameter α (see details in supplementary material).

5. Evaluation

5.1. Results

Hair Removal. We test our method on various images collected from the Internet † (with the permission of non-commercial purposed download) across different ages, ethnicity, gender groups, and lighting conditions. Our method takes 0.7181 seconds to process a real image. Fig. 1 shows two typical examples with plausible hair removal. We also construct experiments on "pseudo ground-truth". We manually add hair to some bald portrait images collected from the internet, then apply our method to the hair-added portraits. More results can be found in the supplementary material.

Dataset. We also apply our method on FFHQ images [19] (CC BY-NC-SA 4.0) and present a non-hair-FFHQ dataset that contains 6,000 non-hair portraits ‡ (see samples in Fig. 6) to inspire and facilitate more works in the future.

*https://github.com/switchablenorms/CelebAMask-HQ
†https://unsplash.com/
‡https://github.com/oneThousand1000/non-hair-FFHQ



Figure 6. Samples from the non-hair-FFHQ dataset.



Figure 7. From left to right: The original image, the image edited by n_g, and the image edited by n_g^c.

Application. Fig. 1 demonstrates two applications of our method: digital hair design and single-view face reconstruction. First, hair design can be made easy by successful hair removal, as new hair templates can be directly added to our resulting image. Here we simply perform Poisson editing to blend hair templates with the image with no hair. Also, for face reconstruction, our hair removal results can provide clean texture for methods that suffer from hair at the forehead [10, 23]. As shown in Fig. 1, we apply GANFIT [12] to portraits after hair removal, resulting in a well-reconstructed face with little geometry change (compared with results from the original portraits) and no hair artifacts.

5.2. Ablation Study

Gender Transformation. In Subsection 3.3, we propose to use the results of Styleflow to train the gender separation boundary and get n_g, since it has better performance than n_g^c of coarse gender separation that trained on randomly sampled latent codes and gender scores. We show in Fig. 7 that n_g^c causes much bigger change of face shape and pose compared with n_g.

Optimization Initialization. In Subsection 3.4, we initialize the latent code using $\dot{w}_m^+$ instead of w_m^+ for optimizing hair removal. The benefit is shown in Fig. 8. There exist artifacts (the hair is not fully removed but still visible) if initialized by w_m^+. We attribute it to the big difference between hair structure and its surroundings including skin and background, which cannot be effectively minimized by optimization. Due to the preferable linearity of faces in the latent space, using $\dot{w}_m^+$ for initialization only requires the change of the face region and achieves much better results.

5.3. Comparisons

We compare our method with other image manipulation works based on path finding in the StyleGAN latent space, including InterFaceGAN [31], StyleSpace [37], StyleClip [29], and StyleFlow [3].



Figure 8. From left to right: the original image, and the results by initializing the optimized code using w_m^+ and $\dot{w}_m^+$, respectively.

	ours	IFG	SS	SC	SF
$\mathcal{L}_{id}$	<u>0.3451</u>	0.5933	**0.3382**	0.3986	0.4257

Table 2. Average identity loss ($\mathcal{L}_{id}$) of InterFaceGAN (IFG), StyleSpace (SS), StyleClip (SC), StyleFlow (SF), and ours.

As shown in Fig. 9, we sample input latent codes and get the original images (column 1). For InterFaceGAN, we train a separation boundary on D_0 and hair scores, then use the separation boundary to edit the input latent code. For StyleSpace, we map the input latent code to StyleSpace, get s, then add standard deviation to s at channel 364 of layer 6, which is claimed for hair control. For StyleClip, we find the global direction from "face with hair" to "face without hair", then move s along this direction. For StyleFlow, we edit the attribute "baldness" for input latent codes.

To compare the ability to preserve face identity, we randomly sampled 500 latent codes, then edit them using the above four methods and also ours. For our method, we do not perform blending (column 6 in Fig. 9) to blend the original face region to the resulting image, as our goal is to make fair comparisons on path-finding performances only. For quantitative measurement, we leverage ArcFace to compute the facial identity difference ($\mathcal{L}_{id}$) between the original face and the edited face (see Tab. 2). StyleSpace is a SOTA method to locally manipulate semantic attributes while preserving facial identity. Our method has quite a similar performance, indicating that the path identified by our method has little influence on the face.

Moreover, Fig. 9 shows that only our method achieves plausible hair removal. It can completely remove hair from male and female portraits while keeping facial identity unchanged. In contrast, InterFaceGAN changes facial identity. StyleSpace only removes hair on the forehead and has obvious artifacts. StyleClip can remove most of the hair, but still causes artifacts in the case of "long hair female". StyleFlow always leads to a smaller head and changes facial identity.

We also show comparisons with 3D head reconstruction work [26] in the supplementary material.

5.4. User Study

We conducted two user studies to evaluate the effectiveness of our method. We evaluate (1) the authenticity of our

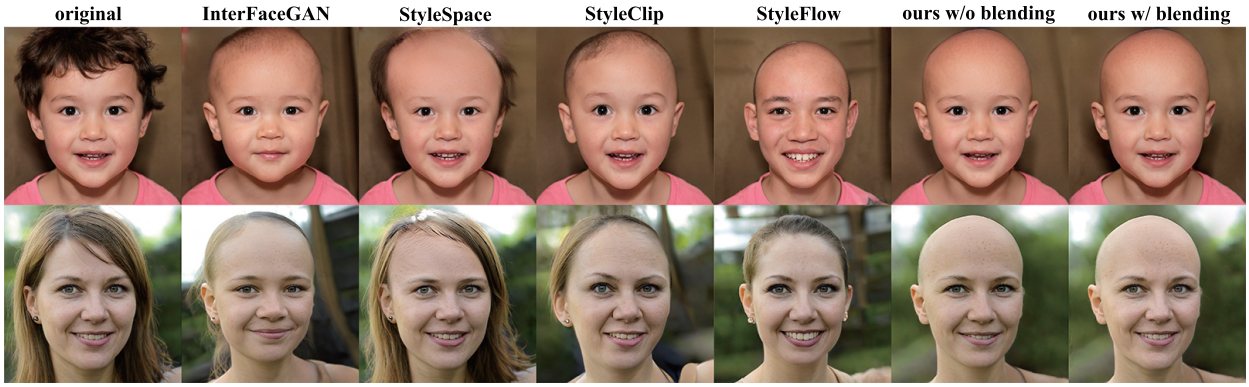


Figure 9. Qualitative comparison results for state-of-the-art methods. From left to right: the original image, InterfaceGAN, StyleSpace, StyleClip, StyleFlow, ours w/o blending, and ours w/ blending.

resulting image *w.r.t.* the real bald image, and (2) user preferences on our results over SOTA results.

For the first user study, we collected 40 real bald portrait images, and 40 randomly sampled results with hair removed from real portrait images. We randomly chose 20 images from the above, and asked the participant to judge whether each image is untouched or not. Results over 40 participants show that 48.37% of our results were labeled as untouched, while 59.74% of the real bald images were labeled as untouched. This indicates that our results are close to real bald data in terms of quality.

For the second user study, we applied the five methods (including ours) for comparison in Subsection 5.3 to 50 latent codes and obtained 50 groups of images. We randomly chose 20 image groups with unordered images for 35 participants to choose the best one from each group. Statistical results show that 66.86% of the participants chose our method as the best, while 8.43%, 6.14%, 9.86%, and 8.71% of the participants chose InterFaceGAN, StyleSpace, StyleClip, and StyleFlow, respectively. This indicates that our method is clearly better than others for hair removal.

6. Conclusion and Discussion

In this paper, we present a new method that can remove hair from portraits naturally while preserving facial identity and portrait quality. Our key idea is to train a fully connected network $HairMapper$ that can identify a path in the latent space of StyleGAN for plausible hair removal. To enable training, we first construct paired latent codes with/without hair for males and females respectively. In particular, we develop a novel "female-male-bald" method for creating paired data of females. During testing, we feed the input portrait to the mapper, then blend the output image that has no hair with the input. Our method can robustly deal with portrait images with variations on gender, age, race, hairstyle, expression, pose, occlusion, shadow, and lighting. The high-quality resulting portraits without hair can preserve facial identity, benefiting real applications such as digital hair design and single-view face reconstruction. We consider our work as an interesting step towards hair manipulation in the latent space. In addition, our high-quality portrait image dataset with hair removed can inspire more potential applications in the future.

Our method has limitations. First, our method relies on the $e4e$ encoder to embed real images. Some portraits that cannot be precisely encoded by $e4e$ may cause artifacts, such as having the overall color, the face or ear shape changed (see rows 1-3 of limitation samples in the supplementary). This can be overcome by improving the StyleGAN encoder or applying an additional diffusion method (details in the supplementary) to the resulting image. Second, hair not covered by hair mask can be noticeable in the result (row 4 of limitation samples). A more precise hair mask by a better extractor or by hand can help to resolve this problem. Third, although dilating and blurring the hair mask can prevent the artifact caused by the shadow under hair, it is still not good enough in extreme lighting conditions (row 5 of limitation samples). Shadow removal could be applied to the face region beforehand for improvement.

Although 3D faces reconstructed from single images or video usually do not contain hair, the resulting images generated by our work may still raise ethical concerns. Our work may benefit hair design and 3D face generation, but we still encourage such an editing operation on a real portrait image should have consent from its owner.

ACKNOWLEDGMENTS Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant Nos. 61972344, 62036010), and the Ningbo Major Special Projects of the "Science and Technology Innovation 2025" (Grant No. 2020Z007). Yong-Liang Yang was supported by RCUK grant CAMERA (EP/M023281/1, EP/T022523/1), and a gift from Adobe. We thank Baris Gecer, Huiwen Luo and Liwen Hu for assisting the face reconstruction, and all Unsplash users for sharing portraits with non-commercial purposed usage.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 4431–4440, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8293–8302, 2020. 2

[3] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (TOG)*, 40(3):1–21, 2021. 2, 4, 7

[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a matter of style: age transformation using a style-based regression model. *ACM Transactions on Graphics (TOG)*, 40(4):45:1–45:12, 2021. 2

[5] Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. Dynamic hair manipulation in images and videos. *ACM Transactions on Graphics (TOG)*, 32(4):75:1–75:8, 2013. 2

[6] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. Single-view hair modeling for portrait manipulation. *ACM Transactions on Graphics (TOG)*, 31(4):116:1–116:8, 2012. 2

[7] Hong Chen and Song Chun Zhu. A generative sketch model for human hair analysis and synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1025–1040, 2006. 2

[8] Shu-Yu Chen, Feng-Lin Liu, Yu-Kun Lai, Paul L. Rosin, Chunpeng Li, Hongbo Fu, and Lin Gao. Deepfaceediting: deep face generation and editing with disentangled geometry and appearance control. *ACM Transactions on Graphics (TOG)*, 40(4):90:1–90:15, 2021. 2

[9] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deepfacedrawing: deep generation of face images from sketches. *ACM Transactions on Graphics (TOG)*, 39(4):72, 2020. 2

[10] Daniel E. Crispell and Maxim Bazik. Pix2face: Direct 3d face model estimation. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 2512–2518, 2017. 1, 7

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4690–4699, 2019. 5

[12] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. GANFIT: generative adversarial network fitting for high fidelity 3d face reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1155–1164, 2019. 1, 7

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[14] Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. Mask-guided portrait editing with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3436–3445, 2019. 2

[15] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable GAN controls. In *Advances in Neural Information Processing Systems*, 2020. 2

[16] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *8th International Conference on Learning Representations, ICLR*, 2020. 2

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR*, 2018. 2

[18] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, 2020. 2, 6

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4401–4410, 2019. 2, 3, 5, 6

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8107–8116, 2020. 2

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015. 6

[22] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5548–5557, 2020. 2, 3, 6

[23] Gun-Hee Lee and Seong-Whan Lee. Uncertainty-aware mesh decoder for high fidelity 3d face reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 6099–6108, 2020. 1, 7

[24] Haofu Liao, Gareth Funka-Lea, Yefeng Zheng, Jiebo Luo, and Shaohua Kevin Zhou. Face completion with semantic knowledge and collaborative adversarial learning. In *Asian Conference on Computer Vision*, volume 11361, pages 382–397, 2018. 2

[25] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. PD-GAN: probabilistic diverse GAN for image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 9371–9381, 2021. 2

[26] Huiwen Luo, Koki Nagano, Han-Wei Kung, Qingguo Xu, Zejian Wang, Lingyu Wei, Liwen Hu, and Hao Li. Normalized avatar synthesis using stylegan and perceptual refinement. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 11662–11672, 2021. 7

[27] Minh Hoai Nguyen, Jean-François Lalonde, Alexei A. Efros, and Fernando De la Torre. Image-based shaving. *Comput. Graph. Forum*, 27(2):627–635, 2008. 2

[28] Kyle Olszewski, Duygu Ceylan, Jun Xing, Jose Echevarria, Zhili Chen, Weikai Chen, and Hao Li. Intuitive, interactive

beard and hair synthesis with generative models. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 7444–7454, 2020. 2

[29] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 2065–2074, 2021. 2, 5, 7

[30] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: A stylegan encoder for image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2287–2296, 2021. 2, 5

[31] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 9240–9249, 2020. 2, 3, 7

[32] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1532–1540, 2021. 2

[33] Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. Michigan: multi-input-conditioned hair image generation for portrait editing. *ACM Transactions on Graphics (TOG)*, 39(4):95, 2020. 3

[34] Ayush Tewari, Mohamed Elgharib, Mallikarjun B. R., Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. PIE: portrait image embedding for semantic control. *ACM Transactions on Graphics (TOG)*, 39(6):223:1–223:14, 2020. 3

[35] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):133:1–133:14, 2021. 2, 3

[36] Yiqian Wu, Yong-Liang Yang, Qinjie Xiao, and Xiaogang Jin. Coarse-to-fine: facial structure editing of portrait images via latent space classifications. *ACM Transactions on Graphics (TOG)*, 40(4):46:1–46:13, 2021. 2, 3, 4, 5

[37] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 12863–12872, 2021. 2, 7

[38] Chufeng Xiao, Deng Yu, Xiaoguang Han, Youyi Zheng, and Hongbo Fu. Sketchhairsalon: Deep sketch-based hair image synthesis. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2021)*, 40(6), 2021. 1, 2

[39] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5505–5514, 2018. 2

[40] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks. *ACM Transactions on Graphics (TOG)*, 40(6):215:1–215:13, 2021. 3