HairNet: Hairstyle Transfer with Pose Changes

Peihao Zhu[1], Rameen Abdal[1], John Femiani[2], and Peter Wonka[1]

[1] KAUST, Saudi Arabia
[2] Miami University, USA

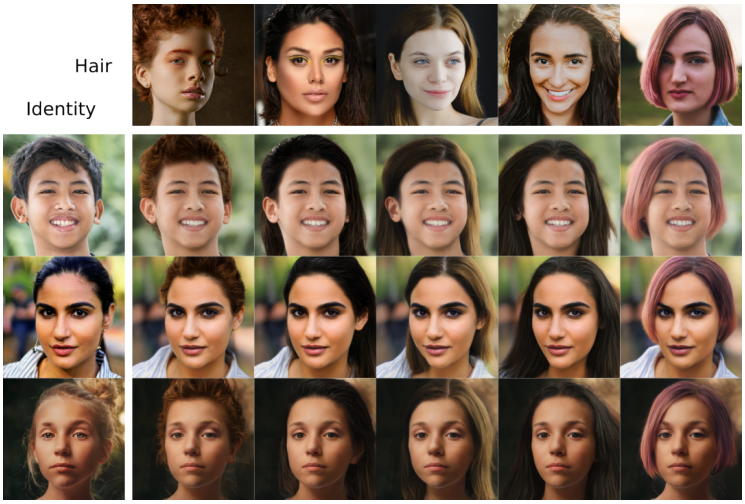

Fig. 1. Given an image of a hairstyle (top row) and a face (left column) which may be in a *different* pose than the hair, we seamlessly transfer the hair onto the face image. Unlike previous approaches, details are preserved even when hair and face images have different poses and head-shapes.

Abstract. We propose a novel algorithm for automatic hairstyle transfer, specifically targeting complicated inputs that do not match in pose. The input to our algorithm are two images, one for the hairstyle and one for the identity (face). We do not require any additional inputs such as segmentation masks. Our algorithm consists of multiple steps and we contribute three novel components. The first contribution is the idea to include baldification into hairstyle editing pipelines to simplify inpainting of background and face regions covered by hair. The second contribution is a novel embedding algorithm that can handle both pose changes and semantic image blending. The third contribution is the *hairnet* architecture that semantically blends the hairstyle and identity images, performing multiple tasks jointly, such as baldification of the identity image, transformation estimation between the two images, warping, and hairstyle copying. Our results show a clear improvement over current

state of the art methods in both quantitative and qualitative results. Code and data will be released.

Keywords: Hairstyle transfer, GANs, StyleGAN, deep learning, image editing

1 Introduction

Choosing a new hairstyle is an important decision, and for many applications that range from marketing to social media and entertainment, there is a need to 'try on' different hairstyles. For this problem we are given a reference image I_{hair} and an identity image I_{ident} which includes a face and background. The goal is to generate a new image I_{mix} that is as similar as possible to I_{ident} with the hair of I_{hair}, but to keep the generated image plausible as a realistic portrait image. The problem is challenging because hair is complex in its interaction with the environment – it is reflective and translucent, and the lighting of the hair including scattering and shadows must be consistent with the I_{ident}. It may pass in front of or behind the face, ears, and clothing. It may be backlit, may have sub-pixel strands, and it reflects light anisotropically.

The current state of the art is Barbershop [44], which uses StyleGAN to invert I_{hair} and I_{ident}, aligns both images to a target segmentation mask, and then copies the activations of an early style-block that correspond to the 'hair' into I_{ident}. However, Barbershop has two main limitations that we try to address. First, the pipeline is inherently limited to work for two input images I_{hair} and I_{ident} that have similar pose. Second, it requires a target segmentation mask to determine the shape of the hair region in relation to the face – but the mask is produced by a naïve heuristic approach that may result in impossible hair shapes, e.g. when complex inpainting of the mask would be needed to deal with translation, scale, occlusion, or disocclusion of the face in I_{ident}.

To overcome these limitations we propose three novel concepts in our work. First, we introduce baldification as a pipeline stage in hairstyle editing. Baldificaton can address disocclusion issues in I_{ident} and provides a dedicated stage to consistently inpaint occluded background and face regions.

Second, we propose a novel embedding algorithm for hairstyle transfer combining two recent embedding algorithms. StyleGAN has biases towards certain hairstyles and so GAN-inversion tends to lose many of the unique characteristics of a hairstyle. Barbershop [44] addresses this by optimizing the activations of an early style-block (called an F-code) of the StyleGAN generator. This embedding is compatible with SOTA semantic image blending algorithms, such as combining hair and face images, but it cannot work for pose changes. On the other hand, generator fine tuning proposed by PTI [31] is a high quality embedding method that is great for pose changes, but it doesn't work with SOTA image blending algorithms. The reason is that PTI generates a separate generator for each input image, but a single generator is required for image blending. Our solution is to combine F code embedding with generator fine tuning [31] to create

the first high quality embedding method that allows for both pose changes as well as image blending.

Third, we propose an image blending architecture, called Hairnet, that is a learned substitute for the heuristic segmentation map editing step in Barbershop. In particular, hairnet is capable to learn how to make difficult semantic decisions that are required when merging a hairstyle image with an identity image. For example, should long hair pass infront or behind the shoulders or ears. Hairnet is trained to jointly perform multiple steps required to blend the two input images, such as baldification, estimating translation and scale between two images, warping, and hairstyle copying. The most interesting aspect about hairnet is that it can be trained in an unsupervised fashion, since there are no ground truth images telling us how to transfer hair from one image to another.

In summary, we make the following contributions: 1) We extend the current SOTA method barbershop to handle input images with different pose. 2) We introduce the concept of baldification to tackle disocclusion problems of the background and face regions. 3) We combine two SOTA GAN embedding algorithms to enable pose changes and image blending. 4) We introduce an image blending architecture hairnet that can be trained in an unsupervised manner.

2 Prior Art

State-of-the-art GANs. Generative adversarial Networks (GANs) [15,29] have seen recent improvements to the loss functions, architecture as well as availability of high quality datasets [22]. Karras together with a team of changing co-authors developed the current state of the art GAN called StyleGAN [19,22,23,20,21]. These GANs are trained on quality high quality datasets like FFHQ [22], AFHQ [8] and LSUN objects [41]. Apart from photo-realistic image synthesis, StyleGAN learns a rich latent space which has been used to perform various downstream tasks such as image editing [33,1,27,4]. Moreover, the architecture of StyleGAN is now used to model other tasks like unsupervised dense correspondences [28] and 3D GANs which are able to generate high-resolution multi-view-consistent images along with approximate 3D geometry [7,13,26]. In the context of this work, we build upon a StyleGAN based hair editing framework, BarberShop [44], to improve the generalization capabilities and the speed of the framework. Based on the issues section in the official released repository of StyleGAN3 [21], there is no solid evidence as of now that it is better than StyleGAN2 in real image projection and editing quality, hence, we use StyleGAN2 [23] to build and compare our framework. This is also important for a comparison, as none of the competing methods uses StyleGAN3.

GAN latent space projection and GAN-based editing. In order to extract meaningful information from a GAN, there are two important components: projecting existing images into a GAN latent space and extracting latent directions to edit an images. First, to enable image editing, image embedding/projection is used as a technique to project real images into the GAN's latent space. In the

StyleGAN domain, Image2StyleGAN [1] uses the extended $W+$ latent-space to project a real image into the StyleGAN latent space using optimization. Other methods like II2S [46] and PIE [36] improve the reconstruction-editing quality trade-off. Other works like IDInvert [43], pSp [30], e4e [37], and Restyle [5] use encoders and identity preserving loss functions to maintain the semantic meaning of the embedding. PTI [31] and HyperStyle [6] modify the generator weights to better map out-of-distribution images. Secondly, image editing frameworks extract important semantic directions in a pretrained GAN. Related to StyleGAN, the editing frameworks [16,33,35,4] analyze the linear and non-linear nature of the underlying W and $W+$ spaces. We use StyleFlow to perform two editing operations critical to our HairNet framework i.e. "Bald" and "Pose" edits. StyleSpace [39] proposes to edit images in StyleSpace S. Another important area of StyleGAN based editing is CLIP based image editing [27,14,2] and domain transfer [45,10].

Hair editing using GANs. Using StyleGAN, there are broadly two types of hair editing frameworks. The first category uses hair segmentation information to edit the hairstyles, e.g. [34,32,44]. We call such methods as semantic region based methods. Recent works use unsupervised analysis on the feature maps [3,11] to identify semantic regions. Editing in Style [11] uses k-means clustering. Based on the Editing in Style method, two relevant works, StyleFusion [18] and Retrieve in Style [9] modify the hair styles using StyleSpace [39] editing and interpolation. Apart from the segmentation and semantic region based methods, some other methods like StyleCLIP [27] and HairCLIP [38] use the CLIP model to modify the hairstyle of a person based on the text prompts. Our method falls in the realm of segmentation based methods. In this work we quantitatively and qualitatively compare our method with the segmentation and semantic region based methods in section 4.3.

3 Method

3.1 Background

We build on the approach of Barbershop [44], and we briefly summarize the key points here but encourage the reader to consult the source for details. The main idea of Barbershop is to use a new latent space for images that allows for spatial control of image features. The new FS-code starts with the frequently used $W+$ latent representation of an image [1], but replaces the first $7 \cdot 512$ elements of the embedding vector by the activations of style-block 7 of the StyleGAN generator. The new latent code consists of a $32 \times 32 \times 512$ tensor of activations F and a vector S with the remaining elements of the $W+$ embedding. The new F code has more degrees of freedom than the $W+$ latent-code, and an optimization process is used to find values of F that are similar to the original $W+$ based activations, but that also improve the reconstruction of the input image. That

optimization process is described in [44] and we omit it here for brevity, as it is not critical for understanding our contribution.

The tensor F can be understood as a 32×32 coarse spatial representation of an image, and so copying and pasting regions of the F-code allows coarse details (called *structure*) of the image to be transferred. This allows the Barbershop to preserve the shape of medium-to-large features, such as the shape of a hair region or the structure of large curls or braids. The style code S used by barbershop is global, and so an optimization process was described to find a single S that best captures the appearance (color and texture) of the hair in I_{hair} in regions of hair while preserving the appearance of I_{ident} in other parts of the image.

A key challenge in transferring the hairstyle from one image to another is that the pose and head-geometry of the subject in I_{hair} may be different from I_{ident}. In order to address this issue, Barbershop aligned each image to a common target-segmentation mask. The target mask was constructed heuristically, and so occasionally implausible target masks were created. In addition, the alignment mainly works for smaller translations, but cannot compensate for larger pose differences.

3.2 Overview

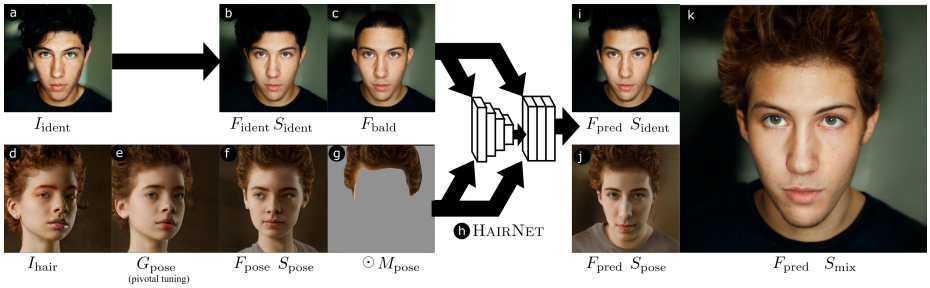

Fig. 2. Overview of hair blending pipeline. Input identity image (a) is embedded into FS-space (b) and then a latent-edit is used to baldify the image and get its F-code (c). Another *hair* input image (d) cannot be embedded the same way, so pivotal tuning is used to make a new generator (e) and then after a pose-edit the generator is used to get FS codes (f) for the hair. The hair is masked (g) and then (c) and (g) are inputs to a new HairNet network (h) that predicts a blended F code. The identity image's global style information (i) and the hair's style (j) are combined to produce final output (k).

We borrow the idea of FS-codes from Barbershop, but we propose to use an alternative approach to deal with changes in pose and head shape. Our approach eliminates the need to generate a target segmentation mask, resulting in a more robust approach to hairstyle transfer. Rather than a target mask, we first *remove hair* from I_{ident} to make a new image I_{bald} using a latent-edit, eliminating the

need to handle disocclusion after this step. The baldification method is described in section 3.3.

Furthermore, prior approaches to hairstyle transfer struggle when I_{hair} and I_{ident} are from very different viewpoints or head-poses. We use a latent-edit to change the pose of I_{hair} to match the pose of I_{ident}. We adapt the pivotal tuning approach [31] to fine-tune a StyleGAN generator in order to create a detailed pose-edited hair image, I_{pose}. Our pose-editing approach is described in section 3.4.

Next, we train a new network called HairNet to blend the F-codes. Let F_{pose} and S_{pose} denote the FS-code of the pose-edited hair image I_{pose}, and similarly let F_{bald} and S_{bald} denote the FS-code of I_{bald}. Then

$$F_{\text{mix}} = \text{HairNet}(F_{\text{pose}}, F_{\text{bald}}). \tag{1}$$

The purpose of HairNet is to use unsupervised machine learning rather than heuristics to learn how to copy hair from one image into another. A key element of our approach is a process for training HairNet in an unsupervised way, which we discuss in detail in section 3.5.

Proceeding with the overview of our method, once F_{mix} is determined, the corresponding style-code S_{mix} must also be determined by mixing elements of S_{ident} and S_{hair}. This is important to preserve the color and texture of the hair, as described in section 3.6.

Finally, the image I_{mix} is found by applying the StyleGAN generator with the activations style-block 7 set to F_{mix} and using S_{mix} as for the remaining style blocks.

3.3 Baldification

A key element of our approach to hair transfer is removal of the existing hair in I_{ident}, which we call 'baldification'. To do so, we first find a latent code W_{ident} in $W+$ space by applying GAN inversion to I_{ident}. In addition, we find an F-code F_{ident} for the identity image to capture a more detailed and spatially-aligned representation. Then we use the StyleFlow [4] method to generate a latent code W_{bald}. StyleFlow in $W+$ space may cause details other than head-hair to change – for example, facial hair may be removed, and the expression or facial features may slightly change. In addition, $W+$ space does not have the capacity to reconstruct all images as well as FS space does, so after the edit we use W_{bald} to find an initial F-code $F^{\text{init}}_{\text{bald}}$. Then we use an automatic segmentation method, BiseNET [40], to label pixels. A binary hair mask for the hair-region is formed and then down sampled bicubically to a shape of 32×32 pixels (so that each pixel is a real-valued number between zero and one) to form M_{hair}. Then

$$F_{\text{bald}} = (1 - M_{\text{hair}})F_{\text{ident}} + M_{\text{hair}}F^{\text{init}}_{\text{bald}}. \tag{2}$$

where the expression is evaluated for each pixel. Note that the the latent edit using StyleFlow only modifies the early layers of $W+$ as described in [4, section

6.2.3], and so the S-code of the baldified image is simply the latter elements of W_{bald}.

3.4 Pose-Editing

A major failure-mode of hair-transfer is when the hair and face images are from different poses, however automatically changing the pose of the hair image while also preserving its structure is extremely challenging. Latent editing approaches, such as StyleFlow [4], are capable of changing the pose by modifying a $W+$ latent code, and therefore have a limited capacity to reconstruct details in the edited image. For baldification, equation (2) works only because F_{ident} and F_{bald} are spatially aligned, however this will not be the case if a latent-edit is used to change the hair pose.

We address this by using pivotal tuning (PTI) [31], which refines a generator G to create a specialized generator with slightly different weights, which does a better job at reconstructing details of a specific image. We adapt this approach to create a generator G_{pose} that is refined with frozen weights for all the StyleGAN blocks that *follow* the F-code, but the first $m = 7$ blocks are free to adapt to better-reconstruct I_{hair}. Because the FS code uses only the activations of style block m, the two generators produce identical images for the same FS-codes. We use StyleFlow to generate a latent-code W_{pose} from W_{hair} and then use the activations of layer m of $G_{\text{pose}}(W_{\text{pose}})$ as F_{pose}

3.5 Training HairNet

Overview In order to train hairNet we first describe an unsupervised way to generate inputs and the desired output of the hair-transfer network. We use images from the FFHQ [22] dataset as a source of training images for hair transfer. In order to do hair-transfer, we consider the FFHQ images to be an ideal result for I_{mix}, and we generate I_{bald} and I_{pose}[3] from it using latent edits and augmentation (described in §3.5). From these we train HairNet to minimize a reconstruction loss (§3.5), so that the image generated using the FS-code predicted by HairNet is perceptually similar to I_{mix}. Finally, we describe the network architecture of HairNet.

Hair Image Generation The process of generating I_{bald} was described in section 3.3. In order to train HairNet we need to do an inverse problem of hair transfer – we need to generate an image that preserves the same hair as I_{mix} but that is different elsewhere. We rely on a semantic segmentation of I_{hair} to determine a binary hair mask that is zero in regions that are not hair and one elsewhere, then downsample the mask bicubically to match the spatial dimensions (32×32) of the F-code and multiply it by the downsampled mask to produce an image with meaningful information only in the hair region.

[3] In this case the $I_{\text{pose}} = I_{\text{hair}}$ as the pose is perfectly aligned by construction.

Even after masking out the F-code while preserving the hair, the results are still different from the inputs one expects to provide HairNet for inference because the hair is still perfectly aligned to with the head-shape of I_{bald} (and the desired output in I_{mix}). In order to ensure that the network does not simply learn to copy the hair, we apply the following augmentations to I_{hair}: a) We apply a random translation to I_{hair}, drawn from a truncated normal distribution with $\sigma = 0.2$. b) We apply a random log-normally-distributed scale to I_{hair}, drawn from the same distribution. The augmentation applies a transform with parameters,

$$T_{\text{hair}} = (t_x, t_y, s_x, s_y) \tag{3}$$

where t_x and t_y are translations and s_x and s_y are the logarithms of the scale applied to the hair image. Note that we represent the transformation using the *log scale* so that all parameters are normally distributed with zero-mean.

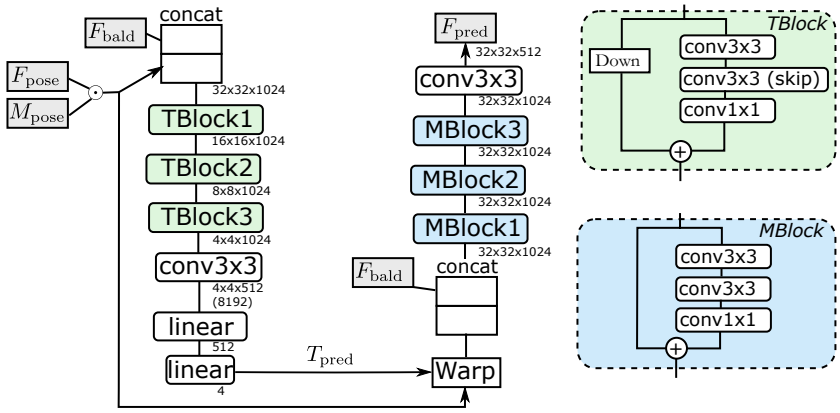

Fig. 3. The HairNet architecture. The input F_{pose} is multiplied by a hair-segmentation mask M_{pose} and concatenated channelwise with F_{bald}. The result is passed through the same set of residual blocks used by StyleGAN2 discriminator and a final convolution reduces the features to 512 channels, then two fully-connected layers predict T_{pred}. The F_{pose} tensor is warped and concatenated with F_{bald} before passing through a similar number of residual blocks. We call the StyleGAN2 blocks with strided convolution *TBlocks*, and the residual blocks without downsampling or strided convolution are called *MBlocks*.

Network Architecture. The architecture of HairNet is inspired by the StyleGAN2 discriminator network. The inputs to the network are the F-codes F_{bald} and F_{pose} along with a segmentation mask, M_{pose}, for the hair in I_{bald} computed using BiseNet [40]. The network first predicts a spatial transformation, T_{pred}, which is then used to warp the masked F_{pose} tensor so that the features

corresponding to hair are positioned properly relative to F_{bald}. The spatial transformer network portion of the architecture uses residual blocks that are identical to the ones used in the StyleGAN2 discriminator, indicated in Fig. 3 as TBlocks. After three residual TBlocks, a final convolution is used to reduce channels to 512, before fully-connected layers predict the transformation (T_{pred}). Then a spatial transformation is applied to the masked activations of F_{pose}, warping it so that the features corresponding to hair are in new spatial locations (ideally aligned to F_{bald}).

Rather than simply copy/pasting the F-code values as was done in Barbershop [44], we use another residual 'blending' network to predict F_{pred}. The aim is to allow the network to learn when hair should cover the face or be occluded by it. The residual blocks of this network do not downsample their inputs, and they are labeled as MBlocks in Fig 3. A final convolution applied to the output of the last MBlock reduces the channels from 1024 down to 512, producing F_{pred} as output.

Loss Function. The goal of the HairNet network is to generate an F-code,

$$F_{\text{pred}} = \text{HairNet}(F_{\text{pose}}, F_{\text{bald}}), \tag{4}$$

so that the generated image

$$I_{\text{pred}} = G(F_{\text{pred}}, S_{\text{ident}}), \tag{5}$$

where the function $G(\cdot)$ is the StyleGAN image generator, is perceptually as similar as possible to I_{mix}. Our training process augments the hair image by applying a translation scale transformation, T_{hair}, to the hair image, and architecture predicts the same spatial transformation, T_{pred} in order to align hair. We use the L_2 loss between the two transformation parameters in order to encourage the network to learn the correct transformation. We also use L_{PIPS} [42] as a perceptual similarity metric. A secondary goal is to minimize the L_2-error between I_{pred} and I_{mix}, and finally, we also want to keep the L_2 error between the latent code F_{pred} and F_{mix} small. We minimize the following loss function:

$$\begin{aligned} L_{\text{rec}} = {} & L_2(T_{\text{hair}}, T_{\text{pred}}) + \lambda_1 L_{\text{PIPS}}(I_{\text{mix}}, I_{\text{pred}}) \\ & + \lambda_2 L_2(F_{\text{mix}}, F_{\text{pred}}) + \lambda_3 L_2(I_{\text{mix}}, I_{\text{pred}}) \end{aligned} \tag{6}$$

The contribution of different loss terms are evaluated empirically in supplemental materials.

Preventing Overfitting. The F-codes contain significantly more information than a $W+$ code for an image, and it is possible for a system such as the one we described to learn how to infer the missing information about an image's original hair from a 'baldified' input image. This means that our training process

if capable of ignoring F_{hair} completely in order to generate F_{mix} using baldified images if I_{mix} is always the same as I_{ident}. We address this by randomly using I_{bald} as I_{mix}. With probability p we replace F_{hair} with zeros, and we use I_{bald} and F_{bald} as I_{mix} and F_{mix} when evaluating the losses. We find $p = 0.5$ produces reasonable results.

3.6 Styling

RetreiveInStyle (RiS) [9] and EncodeInStyle (EiS) [30] use a fast method to interpolate latent codes by first selecting a set of elements of a latent code in style-space. Style-space latent-codes modulate the channels of each stylegan block. RiS and EiS threshold total activatiosn *within* a masked region to determine which latent code elements are relevant to that region. In order to edit the region-of-interest, the latent code elements that are not relevant are frozen and the others are free to change.

One important caveat is that the layer relevance approach uses elements in *style space* rather then $W+$ space. We build on this approach and we also use style-space. This slightly changes our interpretation of the S-code in FS-space as the style space elements are the result of an affine transform applied to the $W+$ vector.

The methods of RiS and EiS use an effective heuristic to change the relevant elements of a latent code - however we find that their approach, while fast, would cause unexpected changes to the color of the hair. Instead, wo use the same optimization criteria as Barbershop (the masked LPIPS function, L_{mask}) to solve for the relevant portions of the code. This loss function is the same function used by Barbershop [44, section 3.5] for style mixing, with one modification. Let that R be a mask so that $R_i = 1$ if the corresponding element S_i of the style code is relevant, and $R_i = 0$ otherwise. We use the mask to change only the relevant parts of the stylecode using optimization. Assume that, for some vector C_{mix}, we have $S_{\text{mix}} = (1 - R)S + RC_{\text{mix}}$, and

$$C_{\text{mix}} = \arg\min_{C} L_{\text{mask}}((1 - R)S + RC), \tag{7}$$

where L_{mask} is calculated using I_{pose} and I_{bald} using M_{pose} as the segmentation mask.

4 Evaluation

4.1 Metrics

Quantitative evaluation of the success of an image editing task has always been a challenging task. A successful edit does two things; it correctly preserves some attributes while changing others (e.g. the face vs the hair), and it also produces a high quality image as output (e.g. free of artifacts). In our case, a successful edit reconstructs the *face* from one image (I_{ident}) and it correctly constructs the hair from another image (I_{hair}) after a pose change.

We use the FID [17] of the generated images and the FFHQ dataset as a quantitative measure of the quality of generated images. The FID is a standard metric for evaluating GANs and if the generated I_{mix} images are not similarly distributed to the FFHQ dataset then they will have high FID scores, and that may indicate low quality results. However, the FID is a poor approximation to human perception of the quality of the input. Several other attempts have been made to quantitatively evaluate the quality of a generated image. The Naturalness Image Quality Estimator (NIQE) [25] measures overall image quality (not specific to face). Precision, Recall, and Realism were introduced in [24]. Precision and recall check whether edited images are on the same manifold as a ground-truth set of image, Realism is simply the distance of an image from the manifold of training images. These methods check the overall quality of an image, but do not capture whether the hair and face are preserved. ArcFace [12] measures the edit's ability to preserve the face of the identity image after the edits. However, we are unaware of any automatic quantitative way to evaluate whether the hairstyle was correctly transferred with a pose that matches the face image. For this problem, we must rely on a user-study and human perception.

4.2 Ablation Study

Many parts of our method are necessary to get any meaningful result, for example we cannot evaluate the effect of using pivotal tuning in isolation because without it a pose edit that produces an F-code that is different from the $W+$ embedding is not even possible. The main contributions we can ablate are the HairNet and our optimization method for mixing the S-codes, which is presented in Fig. 4.

Qualitative results of ablating HairNet are shown in Fig. 4(top), which highlights the importance of human perception in evaluating images, as we expect most readers would agree that the 'w/o HairNet' row of Fig. 4 is significantly worse than the last row, which uses HairNet. However, the quantitative metrics are nearly identical; FID without hairnet is 56.4 (vs 55.6 with HairNet). The NIQE is 11.82 (vs 11.85), the Precision is 95.5% (vs 97%), Recall is 57% (vs 60%) and Realism is 1.21 (vs 1.26%). Quantitatively, we show that HairNet only slightly changes each metric, even though the visual results are significant. We evaluate ablating the style mixing step qualitatively in Fig. 4(bottom), which shows that style mixing with L_{mask} better preserves the colors of the hair and face.

4.3 Comparison

We compare our results against several recent state-of-the-art hair editing methods; Barbershop [44], Retrieve in Style [9], Style Fusion [18], and MichiGAN [34]. Quantitative results are shown in Table 2, however as mentioned previously these metrics do not capture human perception of whether the edit was successful. For example methods that simply copy the face achieve high ArcFace scores but produce images with very undesirably cut&paste artifacts. The same is true for other

Fig. 4. Qualitative ablation studies: (top) results with, and without using the novel HairNet component proposed in our process, demonstrating both inpainting and handling of occlusions; (bottom) the effect of style-mixing using L_{mask} to produce face and hair colors that are more similar to the corresponding source images.

quantitative metrics, however, our method is within the top-2 for most quality metrics.

For a more reliable picture of our performance, we conducted a user study with Amazon Mechanical Turk. For each competing method, 764 image pairs (theirs vs ours) or (ours vs theirs) were shown to human subjects and they were asked which image better reconstructed the hair, the face, and which had the highest overall quality. These quantitative results are shown in Table 1 and it is clear that for the editing tasks of preserving hair and face our method dominates. With respect to overall image quality, we are nearly a tie with StyleFusion, however StyleFusion generates images restricted to a latent space that only has high quality images at the cost of reconstructing hair and face accurately. Many additional qualitative results are included in the supplemental materials.

	Face Rec.		Hair Rec.		Overall Qual.	
Method	Theirs	Ours	Theirs	Ours	Theirs	Ours
Barbershop	32%	**68%**	30%	**70%**	24%	**76%**
RetrieveInStyle	43%	**57**%	43%	**57**%	42%	**58**%
MichiGAN	8%	**92**%	4%	**96**%	2%	**98**%
StyleFusion	49%	**51%**	46%	**54%**	**51%**	49%

Table 1. User-study results comparing to Barbershop [44], Retreive In Style [9], MichiGAN [34], and StyleFusion [18]. Our method outperforms all others for reconstruction tasks. We only lose to StyleFusion (by less than 1%) for the overall quality question, however this is expected because images in the restricted StyleGAN latent space can be more generic and attractive than images with good reconstruction, which many users will perceive as high quality. However, we do significantly better at hair reconstruction than StyleFusion.

Method	NIQE↓	ArcFace↑	FID↓	Precision↑	Recall↑	Realism↑
Barbershop	12.52	<u>0.78</u>	**47.34**	0.93	**0.83**	**1.30**
RetrieveInStyle	12.18	0.58	60.74	0.96	0.31	1.17
MichiGAN	**11.65**	**0.88**	84.66	0.58	<u>0.72</u>	1.09
StyleFusion	12.12	0.56	68.46	**0.98**	0.19	1.19
Ours	<u>11.85</u>	0.66	<u>55.60</u>	<u>0.97</u>	0.60	<u>1.26</u>

Table 2. Quantitative evaluation of different methods using the following metrics: NIQE [25], ArcFace [12], FID [17], precision [24], recall [24], and realism [24]. The best result is bold and the second best is underlined.

Qualitative Results In addition to the quantitative results and the user study, several examples of our results compared with competing methods are shown in Fig. 5. We observe that the results visually agree with the user study, and both StyleFusion and our approach produce high quality results. However, our approach does a better job at preserving salient qualities of the hair and face.

5 Conclusion

We propose a novel algorithm for automatic hairstyle transfer, specifically targeting complicated inputs that do not match in pose. We introduced three main technical contributions to tackle this challenge. First, we introduce the concept of baldification to hairstyle editing pipelines. Second, we propose a novel embedding algorithm that combines the advantages of two recent state-of-the-art methods. Third, we propose the hairnet architecture that automatically combines two images at inference time and can be trained in an unsupervised fashion. In future work, we would like to extend our framework to recent 3D GANs such as EG3D or GRAM.

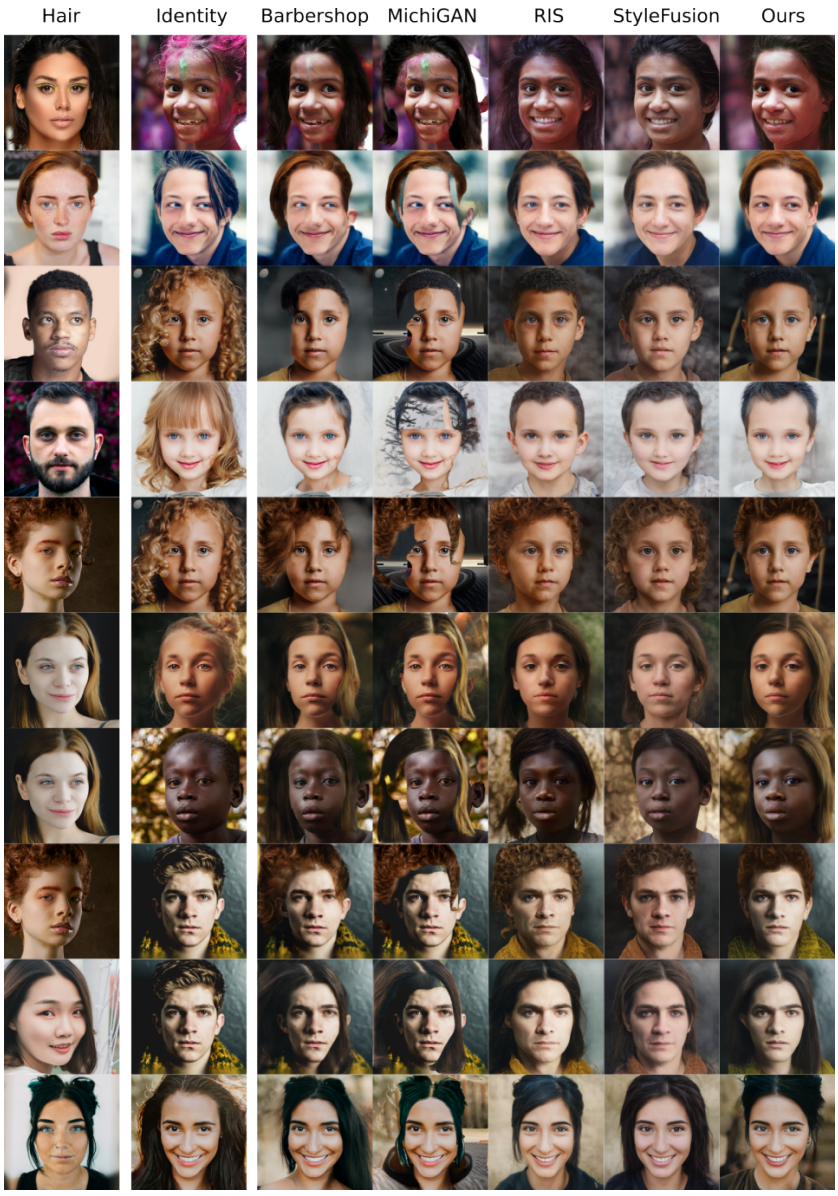

Fig. 5. Main comparison between competing methods. Notice that our method and StyleFusion both produce very realistic results, but our method preserves the appearance of the hair with high fidelity to the source image.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4432–4441. IEEE, Seoul, Korea (2019)
2. Abdal, R., Zhu, P., Femiani, J., Mitra, N.J., Wonka, P.: Clip2stylegan: Unsupervised extraction of stylegan edit directions. CoRR **abs/2112.05219** (2021), https://arxiv.org/abs/2112.05219
3. Abdal, R., Zhu, P., Mitra, N.J., Wonka, P.: Labels4free: Unsupervised segmentation using stylegan. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13970–13979 (October 2021)
4. Abdal, R., Zhu, P., Mitra, N.J., Wonka, P.: Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Trans. Graph. **40**(3) (may 2021). https://doi.org/10.1145/3447648, https://doi.org/10.1145/3447648
5. Alaluf, Y., Patashnik, O., Cohen-Or, D.: Restyle: A residual-based stylegan encoder via iterative refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2021)
6. Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.H.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing. CoRR **abs/2111.15666** (2021), https://arxiv.org/abs/2111.15666
7. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3d generative adversarial networks (2021)
8. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
9. Chong, M.J., Chu, W.S., Kumar, A., Forsyth, D.: Retrieve in style: Unsupervised facial feature transfer and retrieval (2021)
10. Chong, M.J., Forsyth, D.A.: Jojogan: One shot face stylization. CoRR **abs/2112.11641** (2021), https://arxiv.org/abs/2112.11641
11. Collins, E., Bala, R., Price, B., Süsstrunk, S.: Editing in style: Uncovering the local semantics of gans (2020)
12. Deng, J., Guo, J., Niannan, X., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019)
13. Deng, Y., Yang, J., Xiang, J., Tong, X.: Gram: Generative radiance manifolds for 3d-aware image generation (2021)
14. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators. arXiv preprint arXiv:2108.00946 (2021)
15. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2014)
16. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable gan controls. arXiv preprint arXiv:2004.02546 (2020)
17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6629–6640. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
18. Kafri, O., Patashnik, O., Alaluf, Y., Cohen-Or, D.: Stylefusion: A generative model for disentangling spatial segments (2021)

19. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation (2017)
20. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: Proc. NeurIPS (2020)
21. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks (2021)
22. Karras, T., Laine, S., Aila, T.: A Style-Based generator architecture for generative adversarial networks. IEEE transactions on pattern analysis and machine intelligence **43**(12), 4217–4228 (Dec 2021). https://doi.org/10.1109/TPAMI.2020.2970919, http://dx.doi.org/10.1109/TPAMI.2020.2970919
23. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proc. CVPR (2020)
24. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. CoRR **abs/1904.06991** (2019)
25. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a "completely blind" image quality analyzer. IEEE Signal Processing Letters **20**(3), 209–212 (2013). https://doi.org/10.1109/LSP.2012.2227726
26. Or-El, R., Luo, X., Shan, M., Shechtman, E., Park, J.J., Kemelmacher-Shlizerman, I.: Stylesdf: High-resolution 3d-consistent image and geometry generation (2021)
27. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery (2021)
28. Peebles, W., Zhu, J.Y., Zhang, R., Torralba, A., Efros, A., Shechtman, E.: Gan-supervised dense visual alignment (2021)
29. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks (2015)
30. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. arXiv preprint arXiv:2008.00951 (2020)
31. Roich, D., Mokady, R., Bermano, A.H., Cohen-Or, D.: Pivotal tuning for latent-based editing of real images. arXiv preprint arXiv:2106.05744 (2021)
32. Saha, R., Duke, B., Shkurti, F., Taylor, G.W., Aarabi, P.: Loho: Latent optimization of hairstyles via orthogonalization (2021)
33. Shen, Y., Yang, C., Tang, X., Zhou, B.: Interfacegan: Interpreting the disentangled face representation learned by gans. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
34. Tan, Z., Chai, M., Chen, D., Liao, J., Chu, Q., Yuan, L., Tulyakov, S., Yu, N.: Michigan. ACM Transactions on Graphics **39**(4) (Jul 2020). https://doi.org/10.1145/3386569.3392488, http://dx.doi.org/10.1145/3386569.3392488
35. Tewari, A., Elgharib, M., Bharaj, G., Bernard, F., Seidel, H.P., Pérez, P., Zollhofer, M., Theobalt, C.: Stylerig: Rigging stylegan for 3d control over portrait images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6142–6151 (2020)
36. Tewari, A., Elgharib, M., BR, M., Bernard, F., Seidel, H.P., Pérez, P., Zöllhofer, M., Theobalt, C.: Pie: Portrait image embedding for semantic control. vol. 39 (December 2020). https://doi.org/10.1145/3414685.3417803
37. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. arXiv preprint arXiv:2102.02766 (2021)
38. Wei, T., Chen, D., Zhou, W., Liao, J., Tan, Z., Yuan, L., Zhang, W., Yu, N.: Hairclip: Design your hair by text and reference image (2021)

39. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for stylegan image generation. arXiv preprint arXiv:2011.12799 (2020)
40. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. Lecture Notes in Computer Science **11217**, 334–349 (2018). https://doi.org/10.1007/978-3-030-01261-8_20, http://dx.doi.org/10.1007/978-3-030-01261-8_20
41. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
42. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
43. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain gan inversion for real image editing. In: European Conference on Computer Vision. pp. 592–608. Springer (2020)
44. Zhu, P., Abdal, R., Femiani, J., Wonka, P.: Barbershop: Gan-based image compositing using segmentation masks. ACM Trans. Graph. **40**(6) (dec 2021). https://doi.org/10.1145/3478513.3480537, https://doi.org/10.1145/3478513.3480537
45. Zhu, P., Abdal, R., Femiani, J., Wonka, P.: Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=vqGi8Kp0wM
46. Zhu, P., Abdal, R., Qin, Y., Femiani, J., Wonka, P.: Improved stylegan embedding: Where are the good latents? (2020)