StyleGAN Salon: Multi-View Latent Optimization for Pose-Invariant Hairstyle Transfer

Sasikarn Khwanmuang[☛] Pakkapon Phongthawee[☛] Patsorn Sangkloy[✂] Supasorn Suwajanakorn[☛]
[☛]VISTEC, Thailand [✂]Phranakhon Rajabhat University, Thailand
{sasikarn.k_s18, pakkapon.p_s19, supasorn.s}@vistec.ac.th patsorn.s@pnru.ac.th

Figure 1. Our method can transfer the hairstyle from any reference hair image in the top row to Tom Holland [33], in the second row.

Abstract

Our paper seeks to transfer the hairstyle of a reference image to an input photo for virtual hair try-on. We target a variety of challenges scenarios, such as transforming a long hairstyle with bangs to a pixie cut, which requires removing the existing hair and inferring how the forehead would look, or transferring partially visible hair from a hat-wearing person in a different pose. Past solutions leverage StyleGAN for hallucinating any missing parts and producing a seamless face-hair composite through so-called GAN inversion or projection. However, there remains a challenge in controlling the hallucinations to accurately transfer hairstyle and preserve the face shape and identity of the input. To overcome this, we propose a multi-view optimization framework that uses two different views *of reference composites to semantically guide occluded or ambiguous regions. Our optimization shares information between two poses, which allows us to produce high fidelity and realistic results from incomplete references. Our framework produces high-quality results and outperforms prior work in a user study that consists of significantly more challenging hair transfer scenarios than previously studied. Project page:* https://stylegan-salon.github.io/.

1. Introduction

What makes Jennifer Aniston keep her same hairstyle for over three decades? Perhaps she likes the classic, or perhaps changing her hairstyle is a decision too high-stakes that she could later regret. Unlike garments or makeup, trying on a new hairstyle is not easy, and being able to imagine yourself in different hairstyles could be an indispensable tool.

Recent approaches for hairstyle transfer, StyleYourHair [20], Barbershop [46], LOHO [30], and MichiGAN [35], allow users to manipulate multiple hair attributes of an input image, such as appearance, shape, or color by providing *a reference image* for each different attribute. These methods [20, 30, 46] rely on a generative adversarial network [11], specifically StyleGAN2 [19], which can synthesize highly realistic face images. Their key idea, which also forms the basis of our method, is to leverage the realistic face distribution learned by StyleGAN and search for a hairstyle-transfer output whose latent code lies within the learned distribution using optimization (commonly known as GAN projection or inversion).

Our extensive study on these state-of-the-art techniques still reveal several unsolved challenges for *in-the-wild hairstyle transfer*. One of the main challenges is when the reference hair comes from a person with a very different head pose or facial shape. In this case, the transfer result often degrades significantly [30, 46]. HairFIT [8] and the recently proposed StyleYourHair [20] both attempt to solve this using an additional alignment step to align the pose of the target hair to the input face. In HairFIT [8], this is done explicitly via a flow-based warping module for hair segmentation, but this requires training on multi-view datasets [21, 26]. StyleYourHair [20] avoids this issue and also improves upon HairFIT's results by optimizing for an aligned pose within StyleGAN2's latent space using distances between the detected facial keypoints. While StyleYourHair can handle a certain degree of misalignment, it often struggles to preserve details of the reference hair texture, especially for intricate and non-straight hairstyles, *e.g.*, in Figure 3.

In general, we have observed that there is a trade-off between hallucinating new details, which is crucial for handling highly different poses, and preserving the original texture from the reference images. These two goals are often at odds with each others. This trade-off is also evident in the results from StyleYourHair, as reported in [20] and as shown in table 3, where BarberShop [46] can still produce better results, for example, when the pose difference is small.

We tackle this dilemma by performing a multi-stage optimization. This serves two purposes: first, to hallucinate new details necessary for aligning the poses, and second, to recover face-hair details from the original input images. We preserve these details in a form of two guide images in *both* viewpoints, which will be jointly optimized to allow new details to be filled while retaining face-hair texture in the original pose. Our pose alignment is done both explicitly via 3D projection [6] on *RGB* images, and implicitly via latent code(s) sharing during our multi-view optimization.

In summary, our contributions are as follows:

1. We propose StyleGAN Salon: a pose-invariant hairstyle transfer pipeline that is flexible enough to handle a variety of challenging scenarios including, but not limited to, bangs/hat removal and background inpainting.

2. Unlike previous works, our method operates entirely on RGB images, which are more flexible than segmentation masks. This allows us to first *draft* the output and then refine them via multi-stage optimization.

3. We introduce multi-view optimization for hairstyle transfer which incorporates 3D information to align the poses for both face and hair images. Our method leverages both views to help preserve details from the original images.

4. We thoroughly analyze the results in several experiments, including a user study with detailed breakdown into various challenging scenarios. Our method shows superior results over existing works in *all* scenarios.

2. Related Work

Generative Adversarial Networks. Beginning with [11, 18], StyleGANs [17–19] have shown great success in 2D image generation by learning realistic training data distribution corresponding to a fixed low-dimensional distribution, called latent code. The learned latent code can be applied to a downstream task such as image manipulation [13, 15, 32, 40]. Recent work on [6, 12, 44] also explore 3D aware architecture in StyleGAN, resulting in multi-view consistent images.

For hairstyle transfers, MichiGAN [35] uses a conditional hair generation network that can control hair shape, structure, and appearance. Recently, HairFIT [8], a pose-invariant hairstyle transfer network, aligns the reference hair to match the input face pose using a optical flow-based hair alignment module, but requires training on multi-view dataset [21, 26].

StyleGAN Latent Space and Projection Techniques. To generate an image, StyleGAN2 first maps a random latent code $z \sim N(\mathbf{0}, I)$ to an intermediate latent code $w \in \mathbb{R}^{512}$ in a new space called $\mathcal{W}$. This code w is then replicated (18x) and used as input to each layer of the generator that controls details at different scales through weight demodulation. Collectively, these replicated latent codes are referred to as $w^+ \in \mathbb{R}^{18\times512}$ in the extended latent space $\mathcal{W}^+$.

Pivotal Tuning Inversion (PTI) [29] further improves the projection quality by directly tuning generator weights around the optimized latent code w using their regularization technique. Other techniques [1, 2, 5] optimize the latent code w^+ in each layer separately to better match the input image.

PIE [37] introduces hierarchical optimization for semantic attributes editing that first optimizes the latent code in the $\mathcal{W}$ space, then transfers the code to the $\mathcal{W}^+$ space and continues the optimization in that space. We adopt similar hierarchical optimization that uses both $\mathcal{W}$ and $\mathcal{W}^+$. Part of our method is also inspired by PULSE [25], which reconstructs a high-resolution image from a small reference image (32x32) by searching for the closest image in the StyleGAN latent space that resembles the low-resolution reference.

Some methods [7, 41, 45] project multiple images into latent space simultaneously. However, all of their inputs are complete, whereas our method requires hair information from the reference hair image and additional information from the input face image.

StyleGAN-Based Hairstyle Transfer. LOHO [30] adopts loss functions from Image2StyleGAN++ [2] to combine face-hair inputs into a hairstyle transfer output with their orthogonalization optimization technique which reduces conflicts between multiple loss functions. Barbershop [46] first predicts semantic regions of both inputs and uses them to create a *"target segmentation mask"* of the output with rule- (in the original paper) or GAN-based inpainting (in their official code). Then, they optimize two separate latent codes in $\mathcal{W}^+$ space, one for matching the face and the other for hair, while conforming to the target segmentation mask. To preserve the original details, the latent optimization is done in their proposed $\mathcal{F}/\mathcal{S}$ space, which replaces the first seven blocks of $\mathcal{W}^+$ space with the corresponding activation maps of StyleGAN's convolution layers. To improve Barbershop's capacity to handle unaligned input poses, StyleYourHair [20] first aligns the reference hair to match the pose of the input face via the proposed local-style-matching loss. However, this alignment often leads to an unrealistic hair shape or inaccurate hair texture results. In contrast, our method simultaneously optimizes for two guides in face and hair views, resulting in a better hair texture.

Instead of estimating the final output from the segmentation mask [20, 46], our multi-view optimization uses face-hair composites in RGB space to overcome this problem, and produces results that better preserve the input facial structure and hairstyle across a wider, more challenging range of scenarios. Concurrent work, HairNet [47], arrives at a similar goal of removing the target segmentation mask via a two-step process that involves baldification using StyleFlow [3] followed by another network to transfer the hairstyle.

Nevertheless, handling pose differences is crucial for successful hairstyle transfer; our method incorporates 3D rotation to preserve the geometric consistency and systematically evaluate this aspect.

3. StyleGAN Salon

Given two input images I_h and I_f, the goal is to transfer the hair from the reference hair image I_h into the face in the image I_f, while preserving all remaining details in I_f, including identity, clothing, and background.

The key idea of our approach is to guide the optimization on the learned latent space of StyleGAN2 with two "guide" images, which represent rough composites of the final output based on simple cut-and-paste in the viewpoint of I_h and I_f. We leverage EG3D to construct these guided images in a geometrically consistent way, described in Section 3.1.

Optimization on StyleGAN2's latent space is commonly performed on either the original latent space $\mathcal{W}$ or the extended latent space $\mathcal{W}^+$. Optimizing on $\mathcal{W}$ space generally leads to more realistic results by staying within the original latent space, whereas optimizing on $\mathcal{W}^+$ allows a closer match to the reference [37]. In hairstyle transfer, it is important to stay faithful to the input images and preserve important details such as hair texture, face identity, and background. However, optimizing on $\mathcal{W}^+$ will lead to poor results because our guide images are rough and unrealistic composites. Thus, we propose to optimize on $\mathcal{W}$ space followed by $\mathcal{W}^+$ space, similar to a technique in PIE [37] used for editing semantic attributes of an image.

Our optimization incorporates both guide images from the two viewpoints, detailed in Section 3.2. Section 3.3 and Section 3.4 cover details of our optimization on $\mathcal{W}$ and $\mathcal{W}^+$, respectively. Finally, we also optimize StyleGAN2 weights while freezing the latent codes (Section 3.5) using PTI [29] to further improve detail fidelity. Figure 2 shows an overview of our complete pipeline.

3.1. Constructing the Guide Images

The purpose of our guide images is to provide an initial estimate of how the hair would look on I_f. We achieve this using a cut-and-paste composite of the face and background from I_f and the hair from I_h. To better handle a potentially large shift in viewpoint between I_f and I_h, we propose to leverage EG3D [6] to help generate geometrically consistent guide images. We argue that using multi-view guide images, each in the pose of I_f and I_h, helps preserve details that could otherwise be lost from using a single viewpoint alone.

A straightforward approach is to simply project I_h into the EG3D [6] latent space, and use their proposed neural rendering pipeline to render it in the view of I_f (and vice versa). However, while the projection can produce a geometrically consistent 3D shape, we found that the resulting texture is not very accurate. To address this, we replace any visible regions in the texture from I_h's viewpoint with the original pixels of I_h, while leaving the rest of the texture as the projected texture from EG3D's rendering.

We also additionally apply uniform scaling and translation to match the faces' widths and centers, which are computed based on detected facial keypoints [22]. These keypoints, along with semantic regions from [42], are also used to handle various corner cases, such as re-painting unwanted hair regions. We refer to Appendix A for more details of this operation.

The entire process is done for both I_f and I_h viewpoints, resulting in a pair of guide images $I_{\text{guide}} = [I^{\text{face}}_{\text{guide}}, I^{\text{hair}}_{\text{guide}}]$.We emphasize that by leveraging our multi-view optimization (Section 3.2), these guide images do not need to be precise or realistic to produce convincing final results, as demonstrated in the first row of Figure 4.

3.2. Multi-View Latent Optimization

Our guide images provide complementary information about the target hair and face, albeit from different poses. However, each guide image is only fully accurate in regions with the original pixels seen in the original viewpoint and not warped by EG3D. Thus, our optimization goal is to combine information from both guide images to generate a final output that accurately captures the realistic hair from $I^{\text{hair}}_{\text{guide}}$, as well as other details from $I^{\text{face}}_{\text{guide}}$. We achieve this using multiple loss functions that attend to both viewpoints with different spatial emphasis. Specifically, we perform multi-view latent optimization on w/w^+ (and stochastic noise maps n) that fits both guide images:

$$\min_{\{w/w^+, n\}} \sum_{\text{i}}^{[\text{face, hair}]} \mathcal{L}_{\text{loss}}(O^{(\text{i})}, I^{(\text{i})}_{\text{guide}}), \tag{1}$$

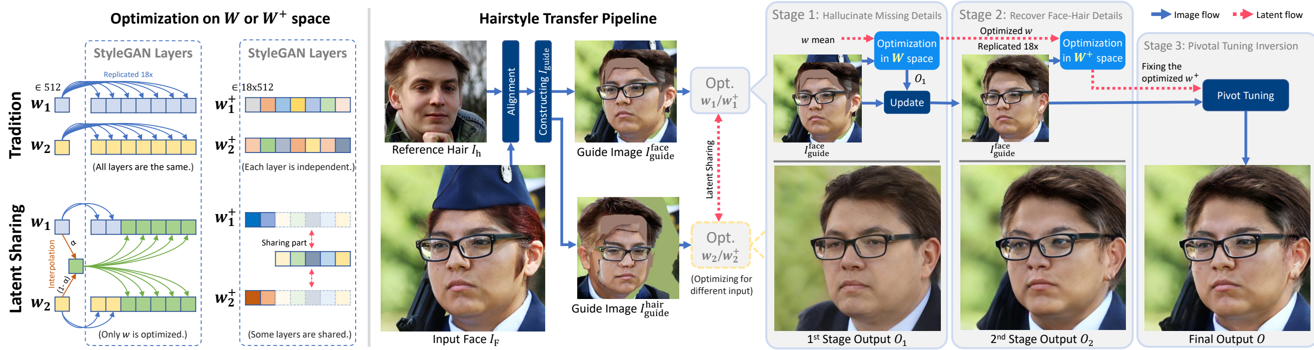

Figure 2. **Overview of StyleGAN Salon:** We first align the input face I_f and reference hair I_h and use them to construct guide images I_{guide}, in two different viewpoints, which specifies the target appearance for each output region. (see Section 3)

where $O^{(i)}$ denotes the output of the StyleGAN2's generator that takes a latent code $w^{(i)}$ or $w^{+(i)}$ and a stochastic noise input $n^{(i)}$. $\mathcal{L}_{\text{loss}}$ is a sum of all our loss functions:

$$\mathcal{L}_{\text{loss}} = \sum_{\text{j}}^{\text{[f, h, bg]}} \lambda_{\text{p}}^{(\text{j})} \mathcal{L}_{\text{per}}^{(\text{j})} \tag{2}$$

$$+ \lambda_{\text{g}} \mathcal{L}_{\text{global}} + \lambda_{\text{i}} \mathcal{L}_{\text{ini}} + \lambda_{\varepsilon} \mathcal{L}_{\varepsilon} + \lambda_{s} \mathcal{L}_{\text{sim}}, \tag{3}$$

where $\lambda_{(.)}$ are balancing weights, and [f, h, bg] refer to face, hair, and background. We next explain each loss function.

3.2.1 Loss Functions

The objective of our $\mathcal{W}$ and $\mathcal{W}^+$ optimization is to generate an output image that fits the corresponding I_{guide} without appearing unrealistic. The main challenge is that our guide images are unrealistic and can contain various unwanted artifacts. We design our loss functions to be tolerant of the imprecise nature of our guide images.

Masked Perceptual Loss: This loss is based on Learned Perceptual Image Patch Similarity (LPIPS) [43], which compares two input images in a *deep feature space*. Similar to [20, 30, 46], we apply a binary mask to indicate regions of interest in the deep feature space of the two input images. We use $L_{\text{LPIPS}}(I_1, I_2; M)$ to denote this LPIPS computation between two images (I_1, I_2) with mask M. However, we observe that applying this masking operation after the feature computation is insufficient to disregard unwanted regions. This is due to the fact that LPIPS is a *patch-based* similarity loss, and regions outside of the mask can still affect the loss computation. Based on this principle, we propose to apply additional **pre-masking** to mask out ambiguous regions in I_{guide} that should not be trusted. Specifically, our Masked Perceptual Loss is the following:

$$\mathcal{L}_{\text{per}}^{\text{i,roi}} = \Lambda_{\text{roi}}^{(\text{i})} L_{\text{LPIPS}}(O^{(\text{i})} \odot \neg M_{\text{roni}}^{(\text{i})}, I_{\text{guide}}^{(\text{i})} \odot \neg M_{\text{roni}}^{(\text{i})}; M_{\text{roi}}^{(\text{i})}), \tag{4}$$

where M_{roi} is the mask for region of interest (ROI), and M_{roni} is the mask for regions of *not* interest (RONI). $(.) \odot \neg M_{\text{roni}}$ is a simple element-wise multiplication, which effectively excludes these regions from the loss by setting them to 0 in both input images. And $\Lambda_{\text{roi}}^{(\text{i})}$ is a balancing weight in (i) viewpoint, which is set higher when the region of interest covers the original pixels of I_f (or I_h). For instance, the loss that attends to face in $I_{\text{guide}}^{\text{face}}$ has a higher weight than the face loss in $I_{\text{guide}}^{\text{hair}}$. We consider 3 regions of interest: $L_{\text{per}}^{\text{f}}$, $L_{\text{per}}^{\text{h}}$, and $L_{\text{per}}^{\text{bg}}$ for face, hair, and background regions. We refer to Section 3.3 and 3.4 for details on these mask generation for $\mathcal{W}$ and $\mathcal{W}^+$ optimization, respectively.

Global loss: This loss function attempts to match the overall appearance of the output to I_{guide}:

$$\mathcal{L}_{\text{global}} = L_{\text{MSE}}^{32}(O, I_{\text{guide}}). \tag{5}$$

L_{MSE}^{32} is the mean square error computed on 32x32 downsampled input images. The goal of the downsampling is to reduce the effect of matching visible seams from higher resolution I_{guide} because those seams become imperceptible once downsampled. This loss is inspired by a similar idea used for image super-resolution by Menon et al. [25].

Initial Value Loss: To prevent the output from deviating too far from the learned distribution and becoming unrealistic, we force the optimized latent code(s) w and w^+ to be close to its initial value w_0 through L2 loss, as used in [2]:

$$\mathcal{L}_{\text{ini}} = \|w_{1:18} - w_0\|_2^2. \tag{6}$$

We use the estimated mean of $\mathcal{W}$ latent space as initial values for optimizing w. This is computed by averaging many latent codes $w_i = f(z_i)$ drawn randomly through $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the StyleGAN's mapping network f. Then the optimized latent code w from the first stage becomes the initial value for the optimization in $\mathcal{W}^+$ extended latent space.

Noise Regularization Loss: We also use the noise regularization loss ($\mathcal{L}_{\varepsilon}$) proposed in StyleGAN2 [19] to ensure that the noise maps will be used for capturing the stochastic variations, and not for encoding the content meant to be captured by the latent code. We refer to StyleGAN2 for the details of this loss.

Latent Similarity Loss: To enable information sharing between the two guide images ($I^{\text{face}}_{\text{guide}}$ and $I^{\text{hair}}_{\text{guide}}$) from two viewpoints, we force both their latent codes to be close together by using the following L2 loss, similarly to [7].

$$\mathcal{L}_{\text{sim}} = \|w^{(\text{face})} - w^{(\text{hair})}\|_2^2, \tag{7}$$

where w can be in $\mathcal{W}$ space during $\mathcal{W}$ optimization or replaced with w^+ during $\mathcal{W}^+$ optimization.

3.2.2 Sharing Latent Code

In StyleGAN2, latent codes in early layers have been shown to correspond to high level concepts (*e.g.* human pose) whereas the remaining layers capture low-level information (*e.g.* colors) [15, 18]. This motivates the optimization in $\mathcal{W}^+$ extended space, as each layer can be optimized for different aspects of the image. Similarly, we optimize latent codes to match guide images of the same person and hairstyle but in different poses. Rather than optimizing two latent codes independently, we can enforce information sharing between the two poses by constraining the latent codes of the last few layers ($w_{l:18}$) to be the same. We set l to be 4 in all our experiments, loosely based on the head rotation experiments shown in GANSpace [15].

Sharing w Latent Code: Optimization in the space of StyleGAN2 latent code $\mathcal{W}$ is generally done as a single latent code ($\mathbb{R}^{512}$), which are duplicated (18x) and fed into all the layers. In our case, we also optimize for a single latent code w but only duplicate them for the first l layers ($w_{1:l}$). The remaining layers ($w_{l:18}$) are interpolated from the current latent codes of each guide image with a coefficient α.

$$w_{l:18} = \alpha w^{(\text{face})}_{1:l} + (1-\alpha) w^{(\text{hair})}_{1:l}. \tag{8}$$

We randomize this coefficient to make the latent codes stay within the space of $\mathcal{W}$. This random interpolation forces the latent code for each guide image to be partly similar while still producing realistic results from $\mathcal{W}$ latent space.

Sharing w^+ Latent Code: Optimization in $\mathcal{W}^+$ extended space is straightforward, as latent code for each layer can already be separately optimized. We simply share the last few layers ($w_{l:18}$) of both w^+ latent codes during optimization. We refer to the Appendix A.5 for more details on the optimization process.

3.3. $\mathcal{W}$ Optimization: Hallucinate Missing Details

This stage aims to hallucinate details not currently visible in the guide images. The output images of this stage are O^{face}_1 and O^{hair}_1, which are essentially the projection of the unrealistic $I^{\text{face}}_{\text{guide}}$ and $I^{\text{hair}}_{\text{guide}}$ into the real image distribution learned in StyleGAN2 [19]. This is done by optimizing on W latent space to fit each guide image with the objective function (Equation 2).

While our guide images ($I^{\text{face}}_{\text{guide}}$ and $I^{\text{hair}}_{\text{guide}}$) capture the overall appearance of the desired outputs, they still lack details in certain regions. These unknown regions may correspond to unseen facial features that were occluded by the hair in I_f or incomplete reference hair from I_h caused by, *e.g.*, a hat or image cropping. These details will need to be hallucinated and seamlessly blended with the rest of the image.

To accomplish this, we design our **pre-masking** in Masked Perceptual Loss ($\mathcal{L}^{\text{f}}_{\text{per}}$, $\mathcal{L}^{\text{h}}_{\text{per}}$, and $\mathcal{L}^{\text{bg}}_{\text{per}}$; Equation 4) to include all the unknown regions that need to be hallucinated. Specifically, $M^{\text{f}}_{\text{roni}}$ represents the face region *occluded* by the hair in I_f, *e.g.*, the forehead behind the bangs or the ears that should become visible in the final output. Analogously, $M^{\text{h}}_{\text{roni}}$ represents the hair region occluded by other objects (*e.g.*, a hat) or not visible in I_h due to image cropping. We set $M^{\text{f}}_{\text{roi}}$ and $M^{\text{h}}_{\text{roi}}$ to be segmentation masks for face and hair regions (M_{f} and M_{h}), respectively. Both $M^{\text{bg}}_{\text{roi}}$ and $\neg M^{\text{bg}}_{\text{roni}}$ is set to be the background regions in I_f not covered by the transferred hair (M_{bg}). These masks are constructed by composing different semantic regions (union, intersection, etc.) from I_f and I_h, detailed in Appendix A.3

We note that M_{roi} is applied on the deep feature maps, similar to [20, 30, 46], while $\neg M_{\text{roni}}$ is applied on raw RGB images. Using both masking techniques is crucial to our natural and seamless blending. The additional pre-masking allows the hair shape (or face shape) in the output to be different from I_{guide}'s and freely grow outward outside M_{roi} if this leads to a more natural result (see Figure 4).

3.4. $\mathcal{W}^+$ Optimization: Recover Face-Hair Details

The output of the previous stage, O^{face}_1 and O^{hair}_1, may still look different from the input person and not yet capture the hair details from the reference. This stage aims at refining these output images to better reproduce the hair details from I_h and the rest from I_f. The optimization is done in the extended $\mathcal{W}^+$ space with respect to $\{w^+, n\}$. In other words, we allow $w \in \mathbb{R}^{512}$ that was previously replicated to each layer to be optimized separately as $w^+ \in \mathbb{R}^{18\times512}$. We denote O^{face}_2 and O^{hair}_2 to be the output of this stage generated by StyleGAN2 from our optimized latent codes in $\mathcal{W}^+$ extended latent space.

We update the target images for the optimization from $I^{\text{face}}_{\text{guide}}$ and $I^{\text{hair}}_{\text{guide}}$ to the more complete versions based on O^{face}_1 and O^{hair}_1 from the first stage. However, because the first stage aims at hallucinating new details into the pre-masking regions, it may lose the original texture details. To address this, we replace the known regions in O^{face}_1 and O^{hair}_1 with the correct details from the original I_{guide} by setting

$$I_{\text{new\_guide}} \leftarrow I_{\text{guide}} \odot M_{\text{c}} + O_1 \odot \neg M_{\text{c}}, \tag{9}$$

where $M_c = M_{\text{f}} \cup M_{\text{h}} \cup M_{\text{bg}}$, which corresponds to the regions of interest in our Masked Perceptual Loss where we want to match with I_{guide}. In short, we simply copy the hallucinated parts from O_1^{face} and O_1^{hair}, and combine them with the part in I_{guide} that are known to be correct.

For our Masked Perceptual Loss, we use the same masking procedure as the previous stage (Section 3.3). However, we observe that the projection from EG3D, which produces I_{guide}, generally is inferior to the known details from the original images. For example, $I_{\text{guide}}^{\text{hair}}$ contains more details on the hair than $I_{\text{guide}}^{\text{face}}$ due to the latter being hallucinated from the 3D rotation by EG3D's rendering pipeline. To *de-prioritize* these inaccurate details, we blur both I_{guide} and O when computing the perceptual loss for the less accurate regions of interest. Concretely, we only blur images when computing $\mathcal{L}_{\text{per}}^{\text{f}}$ for $I_{\text{guide}}^{\text{hair}}$ optimization and when computing $\mathcal{L}_{\text{per}}^{\text{h}}$ for $I_{\text{guide}}^{\text{face}}$ optimization.

3.5. Pivotal Tuning Inversion

The purpose of this stage is to further refine the output images (O_2^{face} and O_2^{hair}) by allowing the optimization of the StyleGAN's weights θ (while fixing the optimized latent code $w_{\text{optimized}}$). We closely follow the proposed optimization in PTI [29] with the same objective function. However, we update the reconstruction loss to reflect the goal of our task, which is to match the original details from I_f and I_h.

$$\mathcal{L}_{\text{pti}} = \sum_{\text{i}}^{[\text{face, hair}]} (L_{\text{LPIPS}}(O_{\text{tune}}^{(\text{i})} \odot M_{\text{raw}}^{(\text{i})}, I_{\text{guide}}^{(\text{i})} \odot M_{\text{raw}}^{(\text{i})}; M_{\text{raw}}^{(\text{i})}) + L_{\text{MSE}}^{32}(O_{\text{tune}}^{(\text{i})}, I_{\text{guide}}^{(\text{i})}; \neg M_{\text{raw}}^{(\text{i})})), \tag{10}$$

where O_{tune} is the generated image using the tuned weights θ, the mask $M_{\text{raw}}^{\text{face}}$ is $(M_{\text{bg}} \cup M_{\text{f}})$, and $M_{\text{raw}}^{\text{hair}}$ is M_{h}.

4. Experiments

In this section, we compare our method to state-of-the-art StyleYourHair [20], Barbershop [46], and LOHO [30]. Our evaluation criteria are i) user preference via a user study on a wide variety of scenarios, ii) hairstyle transfer quality, iii) hair reconstruction quality, and iv) how well the input face shape is preserved. Section 4.5 presents ablation studies on our multi-view sharing latent optimization, optimization stage, and loss functions.

We use the official code of LOHO [30] and Barbershop [46] with the default configurations. For StyleYourHair [20], we use the configuration where the hair reference is never flipped.

4.1. Qualitative Comparison

We provide a qualitative comparison to StyleYourHair [20], Barbershop [46], and LOHO [30] in Figure 3. We observe that LOHO and Barbershop often struggle to fit the reference hair accurately when the poses are not well aligned (rows 3-5), resulting in various artifacts, such as remnants of the original hair or wrong placement of the target hair. Barbershop performs well in preserving hair texture when the poses are similar. However, it falls short when handling challenging cases, such as transitioning from a long to a short hairstyle or removing bangs.

Figure 3. Comparison to current state-of-the-art methods for transferring hair from 1st column-top, to the face of 1st column-bottom. Our method can accurately transfer hairstyle even when the input face and hair are misaligned (3rd, 4th row). It can also hallucinate missing details such as a forehead that was previously occluded (1st row), or shorten the hair (2nd row). In general, LOHO and Barbershop struggle with pose misalignment, while StyleYourHair struggles to preserves the input's face shape and hair details.

StyleYourHair can produce more realistic results in unaligned cases than LOHO and Barbershop. However, the hair details often look different from the reference hairstyle (row 5). We also observe that StyleYourHair may perform poorly when the pose difference becomes too large (row 3), as also shown in our user study for pose in Table 1. In contrast, our method can transfer hairstyles convincingly, regardless of the misalignment, while still preserving the original face shape and hair details. We refer to Appendix F for more results.

4.2. User Study

Qualitative results can be misleading and biased, so we conducted a user study using Amazon Mechanical Turk on hairstyle transfer results using randomly selected pairs grouped into various difficulty levels. We compare results of our method with the current state of the art: StyleYourHair, Barbershop, and LOHO. Each participant was shown an input face, marked as 'Face', and a reference hair, marked

	Test Datasets			FFHQ Scenario Breakdown												
				Easy	Medium				Difficult							
	All	FFHQ-P	FFHQ-S	-	✓	-	-	-	✓✓	✓	✓	✓	-	-	-	Pose Misalignment
				-	-	✓	-	-	-	✓	-	-	✓	✓	-	Needs Face Inpainting
				-	-	-	✓	-	-	-	✓	-	✓	-	✓	Needs BG Inpainting
				-	-	-	-	✓	-	-	-	✓	-	✓	✓	I_h wears a hat
LOHO	8.8	9.6	8.4	12.0	5.3	4.0	13.3	8.0	5.3	5.3	4.0	10.7	5.3	13.3	14.7	
Barbershop	16.4	<u>13.8</u>	17.8	17.3	<u>21.3</u>	<u>13.3</u>	21.3	18.7	<u>17.3</u>	17.3	20.0	12.0	18.7	<u>14.7</u>	<u>21.3</u>	
StyleYourHair	<u>18.5</u>	12.4	<u>21.6</u>	<u>25.3</u>	<u>21.3</u>	10.7	<u>22.7</u>	<u>33.3</u>	10.7	<u>24.0</u>	<u>28.0</u>	<u>14.7</u>	<u>32.0</u>	<u>14.7</u>	<u>21.3</u>	
Ours	**56.2**	**64.2**	**52.2**	**45.3**	**52.0**	**72.0**	**42.7**	**40.0**	**66.7**	**53.3**	**48.0**	**62.7**	**44.0**	**57.3**	**42.7**	

Table 1. User study results on hairstyle transfer (percentage of user preferring each method). Our method outperforms state-of-the-art hairstyle transfer methods on FFHQ datasets in all challenging scenarios. A total of 450 pairs are used in this study, 150 pairs in FFHQ-P and 300 in FFHQ-S. For each pair, we asked 3 unique participants to select the best result for hairstyle transfer.

as 'Hair', and asked to pick only one output that best accomplishes the task of transferring the hairstyle from image 'Hair' to the person in image 'Face'. The output row consists of four images from each method *in random order*. Each task was evaluated by 3 different participants. All images were in 256x256 resolution.

4.2.1 Datasets

For the user study, we construct two challenging benchmarks: FFHQ-P and FFHQ-S, from the test set of Flickr-Faces-HQ dataset (FFHQ) [18].

FFHQ-P contains 150 random input face-hair pairs from FFHQ, covering yaw differences in ranges of $[0-15), [15-30), \ldots, [75-90)$, with 25 pairs in each range.

FFHQ-S contains 300 pairs, categorized into 12 different configs with varying levels of difficulty (Table 1). Each config contains 25 random input pairs and is a combination of four possible scenarios (see details in Appendix B):

- Pose Misalignment: When the yaw difference is between [15-30) or [30-45), a single checkmark or double checkmarks are used, respectively in Table 1.
- Needs Face Inpainting: This includes scenarios that require hallucinating parts of the original face, *e.g.*, inpainting the forehead to remove bangs. This is challenging because the identity can easily change from the hallucination. We detect such scenarios based on the face/hair regions in I_f and I_h (Appendix B).
- Needs BG Inpainting: This includes scenarios where the hair shape becomes smaller and requires background inpainting. We detect such scenarios automatically based on the hair regions in I_f and I_h (Appendix B).
- I_h Contains Hat: This represents scenarios where the hair reference is not fully visible in I_h. We detect such scenarios based on the hat region in I_h.

	Pose Difference Range (FFHQ-P)					
	[0,15)	[15,30)	[30,45)	[45,60)	[60,75)	[75,90)
LOHO	10.7	5.3	10.7	12.0	8.0	10.7
Barbershop	<u>24.0</u>	13.3	<u>12.0</u>	<u>8.0</u>	10.7	<u>14.7</u>
StyleYourHair	17.3	<u>16.0</u>	<u>12.0</u>	6.7	<u>14.7</u>	8.0
Ours	**48.0**	**65.3**	**65.3**	**73.3**	**66.7**	**66.7**

Table 2. User study on pose-invariant hairstyle transfer. Our method outperforms others on all pose difference ranges.

	Hair Reconstruction Metrics				Face Shape
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	RMSE ↓
LOHO	25.76	0.86	<u>0.07</u>	**10.40**	15.84
Barbershop	**29.18**	**0.89**	**0.05**	<u>10.46</u>	17.25
StyleYourHair	26.89	0.87	0.09	10.93	<u>14.37</u>
Ours	<u>27.84</u>	<u>0.88</u>	<u>0.07</u>	10.85	**10.89**

Table 3. Hair reconstruction results in the self-transfer experiment (Section 4.3), and RMSEs between facial landmarks detected on the input and output images.(Section 4.4).

4.2.2 Results

Using all test pairs, the participants preferred our results 56.2% of the time, whereas StyleYourHair's, Barbershop's, and LOHO's results were selected for 18.5%, 16.4% and 8.8%, respectively. Our method perform the best in both FFHQ-P (64.2%) and FFHQ-S (52.2%), and in all configurations in Table 1-2.

4.3. Quality of Hair Reconstruction

Following MichiGAN and LOHO, we perform a self-transfer experiment where we set both the input face and input reference hair to be the same and evaluate the reconstruction accuracy using various metrics: PSNR, SSIM, IPIPS, FID [14]. For every method, the hair region of the output will be blended back to the input image to ensure that the difference in score only comes from the hair.

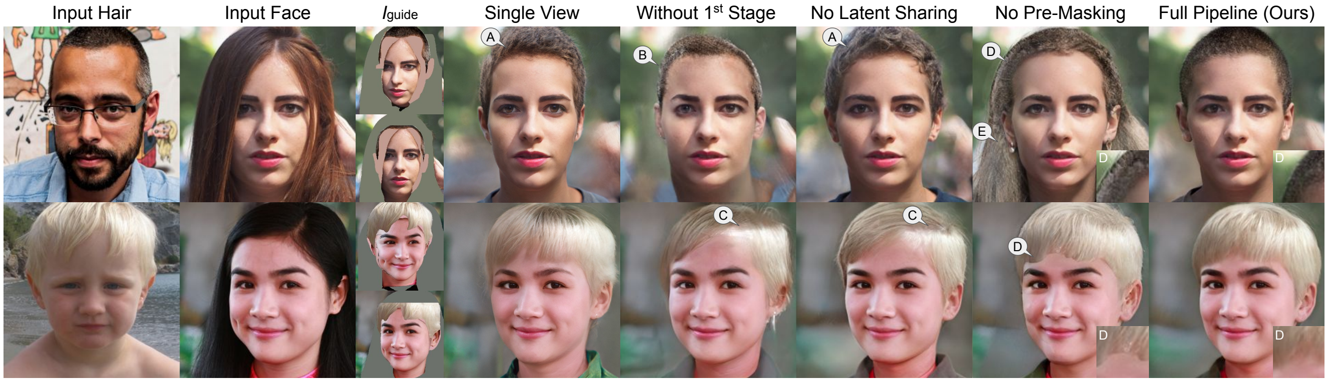

Figure 4. Ablation study of optimization stages and pre-masking. These ablated versions of our pipeline may produce results that (A) look realistic, but fail to match the I_h's hairstyle; (B, C, D) have unlikely hair shapes or structures; (E) have sharp boundaries; and (F) contain an incomplete background.

The scores of our method, LOHO, Barbershop, and StyleYourHair are reported in Table 3. Unsurprisingly, when there is no misalignment between input face and hair, Barbershop generally achieves excellent hair reconstruction quality. Like ours, StyleYourHair is specifically designed to be pose-invariant, but their hair reconstruction quality seems to suffer greatly, as shown in our comparison (row 5 of Figure 3). Our method achieves better performance at hair reconstruction than StyleYourHair, but also suffers from similar drawbacks.

4.4. How Well Is Face Shape Preserved?

We also propose a novel evaluation metric for hairstyle transfer that focuses on the ability to preserve facial shape of the original person. We accomplish this by comparing detected keypoints of the input face I_f and those on the output using a simple Root-Mean-Square Error (RMSE). These keypoints were detected using an off-the-shelf library Dlib [22], and we only used keypoints on the facial contour (ID 0-16) for this evaluation.

For this evaluation, we randomize additional 1,550 FFHQ test pairs into our FFHQ-P and FFHQ-S datasets (2,000 images in total). Table 3 shows that our method outperforms others with an RMSE of 10.89, which is 24.2% lower than StyleYourHair's (the second best), 36.9% lower than Barbershop's and 31.3% lower than LOHO's.

4.5. Ablation Studies

Here we assess the importance of each component in our pipeline. The results are shown in Figure 4. We test 4 ablation configurations: **i)** by using only $I^{\text{face}}_{\text{guide}}$ from a single view, **ii)** without using the hallucination stage (Section 3.3), **iii)** without using the latent sharing structure (Section 3.2.2), and **iv)** without using pre-masking in $\mathcal{L}_{\text{per}}$ (Section 3.2.1).

Optimizing with a single view guide $I^{\text{face}}_{\text{guide}}$ (Config i) yields inaccurate hairstyles (Figure 4-A). Without optimization in W (Config ii), the method produces unrealistic hair results with various artifacts, *e.g.*, sharp edges, unnatural hair shapes (Figure 4-B) or structures (Figure 4-C). This is

Figure 5. Failure cases; (A) the semantic regions are incorrect; (B) the facial keypoint is incorrect; (C) the EG3D projection is incorrect; (D) the reference hair color is too similar to the background color; (E) lighting looks unnatural; (F) eccentric hairstyle.

because the second stage, which has higher image fitting capability, tries to fit the initial rough estimation I_{guide}. Without the latent sharing structure, the hair detail cannot be shared from the reference hair view ($\mathcal{L}_{\text{sim}}$ alone is not strong enough to force consistency) (Figure 4-A), resulting in inaccurate hair colors or structures (Figure 4-C).

Without pre-masking in $\mathcal{L}_{\text{per}}$, the boundary of the face and hair regions is forced to be the same as in I_{guide}, leading to visible and sharp seams between the face and hair or between different facial features. Here the optimizer fails to refine the face-hair boundary to make the results look natural (Figure 4-D), and the original details, such as the background from I_f cannot be seamlessly blended (Figure 4-E).

5. Conclusion and Limitations

We have presented a flexible and effective hairstyle transfer system that can handle a variety of challenging in-the-wild scenarios and produce perceptually convincing hairstyle transfer results. Our user study shows that human evaluators prefer our results over previous methods across all tested scenarios. Nevertheless, our method can still fail in certain scenarios, for example, when the face and hair are too eccentric (Figure 5). We refer to Appendix G for more details.

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, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 2, 4

[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. 3

[4] Ironhammer america. File:elizabeth-olsen-1632123202.jpg @ONLINE. `https://commons.wikimedia.org/wiki/File:Elizabeth-olsen-1632123202.jpg/`, 2022. 17

[5] Peter Baylies. Stylegan-encoder @ONLINE. `https://github.com/pbaylies/stylegan-encoder/`, 2019. 2

[6] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *arXiv*, 2021. 2, 3, 11

[7] Prashanth Chandran, Sebastian Winberg, Gaspard Zoss, Jérémy Riviere, Markus Gross, Paulo Gotardo, and Derek Bradley. Rendering with style: combining traditional and neural approaches for high-quality face rendering. *ACM Transactions on Graphics (ToG)*, 40(6):1–14, 2021. 2, 5

[8] Chaeyeon Chung, Taewoo Kim, Hyelin Nam, Seunghwan Choi, Gyojung Gu, Sunghyun Park, and Jaegul Choo. Hairfit: Pose-invariant hairstyle transfer via flow-based hair alignment and semantic-region-aware inpainting. *arXiv preprint arXiv:2206.08585*, 2022. 2

[9] Gordon Correll. File:julianne moore (15011443428) (2).jpg @ONLINE. `https://commons.wikimedia.org/wiki/File:Julianne_Moore_(15011443428)_(2).jpg/`, 2014. 17

[10] Chengdong Dong, Ajay Kumar, and Eryun Liu. Think twice before detecting gan-generated fake images from their spectral domain imprints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7865–7874, 2022. 16

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2

[12] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 2

[13] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[15] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Proc. NeurIPS*, 2020. 2, 5

[16] jerryhouuu. Face-yaw-roll-pitch-from-pose-estimation-using-opencv @ONLINE. `https://github.com/jerryhouuu/Face-Yaw-Roll-Pitch-from-Pose-Estimation-using-OpenCV/`, 2018. 13

[17] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 2

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2, 5, 7

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 2, 4, 5, 12, 13

[20] Taewoo Kim, Chaeyeon Chung, Yoonseo Kim, Sunghyun Park, Kangyeol Kim, and Jaegul Choo. Style your hair: Latent optimization for pose-invariant hairstyle transfer via local-style-aware hair alignment. *arXiv preprint arXiv:2208.07765*, 2022. 1, 2, 3, 4, 5, 6, 14, 19, 20, 21

[21] Taewoo Kim, Chaeyeon Chung, Sunghyun Park, Gyojung Gu, Keonmin Nam, Wonzo Choe, Jaesung Lee, and Jaegul Choo. K-hairstyle: A large-scale korean hairstyle dataset for virtual hair editing and hairstyle classification. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1299–1303. IEEE, 2021. 2

[22] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 3, 8, 11

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 13

[24] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 11

[25] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2445, 2020. 2, 4

[26] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017. 2

[27] nicolas genin. File:julianne moore 2009 venice film festival.jpg @ONLINE. `https://commons.wikimedia.org/wiki/File:Julianne_Moore_2009_Venice_Film_Festival.jpg/`, 2009. 17

[28] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. 13

[29] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 2, 3, 6, 11, 12, 13

[30] Rohit Saha, Brendan Duke, Florian Shkurti, Graham Taylor, and Parham Aarabi. Loho: Latent optimization of hairstyles via orthogonalization. In *CVPR*, 2021. 1, 2, 4, 5, 6, 19, 20, 21

[31] David Shankbone. File:dwayne johnson at the 2009 tribeca film festival.jpg @ONLINE. https://commons.wikimedia.org/wiki/File:Dwayne_Johnson_at_the_2009_Tribeca_Film_Festival.jpg/, 2009. 17

[32] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *TPAMI*, 2020. 2

[33] Gage Skidmore. Tom holland speaking at the 2016 san diego comic-con international in san diego, california. @ONLINE. https://commons.wikimedia.org/wiki/File:Tom_Holland_by_Gage_Skidmore.jpg/, 2016. 1

[34] Gage Skidmore. File:elizabeth olsen by gage skidmore 2.jpg @ONLINE. https://commons.wikimedia.org/wiki/File:Elizabeth_Olsen_by_Gage_Skidmore_2.jpg/, 2019. 17

[35] 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. *arXiv preprint arXiv:2010.16417*, 2020. 1, 2

[36] tenasia10. File:blackpink lisa gmp 240622.png trends @ONLINE. https://commons.wikimedia.org/wiki/File:Blackpink_Lisa_GMP_240622.png/, 2022. 17

[37] Ayush Tewari, Mohamed Elgharib, 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):1–14, 2020. 2, 3

[38] GABI THORNE. Jennifer aniston reacts to tiktok's best (and most absurd) trends @ONLINE. https://www.allure.com/story/jennifer-aniston-reacts-to-tiktok-trends/, 2022. 17

[39] Albert Truuväärt. File:re kaja kallas (cropped).jpg @ONLINE. https://commons.wikimedia.org/wiki/File:RE_Kaja_Kallas_(cropped).jpg/, 2011. 17

[40] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y Chang, and Shih-Wei Liao. Relgan: Multi-domain image-to-image translation via relative attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5914–5922, 2019. 2

[41] Yangyang Xu, Yong Du, Wenpeng Xiao, Xuemiao Xu, and Shengfeng He. From continuity to editability: Inverting gans with consecutive images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13910–13918, 2021. 2

[42] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 3, 11

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[44] Xiaoming Zhao, Fangchang Ma, David Güera, Zhile Ren, Alexander G. Schwing, and Alex Colburn. Generative multiplane images: Making a 2d gan 3d-aware. In *Proc. ECCV*, 2022. 2

[45] Tao Zhou, Changqing Zhang, Chen Gong, Harish Bhaskar, and Jie Yang. Multiview latent space learning with feature redundancy minimization. *IEEE transactions on cybernetics*, 50(4):1655–1668, 2018. 2

[46] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks. *arXiv preprint arXiv:2106.01505*, 2021. 1, 2, 3, 4, 5, 6, 14, 19, 20, 21

[47] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Hairnet: Hairstyle transfer with pose changes. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*, pages 651–667. Springer, 2022. 3

[48] zllrunnings. face-parsing.pytorch @ONLINE. https://github.com/zllrunning/face-parsing.PyTorch/, 2019. 11