LOHO: Latent Optimization of Hairstyles via Orthogonalization

Rohit Saha[1,2] Brendan Duke[1,2] Florian Shkurti[1,4] Graham W. Taylor[3,4] Parham Aarabi[1,2]

[1]University of Toronto [2]Modiface, Inc. [3]University of Guelph [4]Vector Institute

Figure 1: **Hairstyle transfer samples synthesized using LOHO.** For given portrait images (a) and (d), LOHO is capable of manipulating hair attributes based on multiple input conditions. Inset images represent the target hair attributes in the order: appearance and style, structure, and shape. LOHO can transfer appearance and style (b), and perceptual structure (e) while keeping the background unchanged. Additionally, LOHO can change multiple hair attributes simultaneously and independently (c).

Abstract

Hairstyle transfer is challenging due to hair structure differences in the source and target hair. Therefore, we propose Latent Optimization of Hairstyles via Orthogonalization (LOHO), an optimization-based approach using GAN inversion to infill missing hair structure details in latent space during hairstyle transfer. Our approach decomposes hair into three attributes: perceptual structure, appearance, and style, and includes tailored losses to model each of these attributes independently. Furthermore, we propose two-stage optimization and gradient orthogonalization to enable disentangled latent space optimization of our hair attributes. Using LOHO for latent space manipulation, users can synthesize novel photorealistic images by manipulating hair attributes either individually or jointly, transferring the desired attributes from reference hairstyles. LOHO achieves a superior FID compared with the current state-of-the-art (SOTA) for hairstyle transfer. Additionally, LOHO preserves the subject's identity comparably well according to PSNR and SSIM when compared to SOTA image embedding pipelines. Code is available at https://github.com/dukebw/LOHO.

1. Introduction

We set out to enable users to make semantic and structural edits to their portrait image with fine-grained control. As a particular challenging and commercially appealing example, we study hairstyle transfer, wherein a user can transfer hair attributes from multiple independent source images to manipulate their own portrait image. Our solution, Latent Optimization of Hairstyles via Orthogonalization (LOHO), is a two-stage optimization process in the latent space of a generative model, such as a generative adversarial network (GAN) [12, 18]. Our key technical contribution is that we control attribute transfer by orthogonalizing the gradients of our transferred attributes so that the application of one attribute does not interfere with the others.

Our work is motivated by recent progress in GANs, enabling both conditional [15, 32] and unconditional [19] synthesis of photorealistic images. In parallel, recent works have achieved impressive latent space manipulation by learning disentangled feature representations [26], enabling photorealistic global and local image manipulation. However, achieving controlled manipulation of attributes of the synthesized images while maintaining photorealism remains an open challenge.

Previous work on hairstyle transfer [30] produced realistic transfer of hair appearance using a complex pipeline of GAN generators, each specialized for a specific task such as hair synthesis or background inpainting. However, the use of pretrained inpainting networks to fill holes left over by misaligned hair masks results in blurry artifacts. To produce more realistic synthesis from transferred hair shape, we can infill missing shape and structure details by invoking

the prior distribution of a single GAN pretrained to generate faces.

To achieve photorealistic hairstyle transfer even under said source-target hair misalignment we propose Latent Optimization of Hairstyles via Orthogonalization (LOHO). LOHO directly optimizes the extended latent space and the noise space of a pretrained StyleGANv2 [20]. Using carefully designed loss functions, our approach decomposes hair into three attributes: perceptual structure, appearance, and style. Each of our attributes is then modeled individually, thereby allowing better control over the synthesis process. Additionally, LOHO significantly improves the quality of synthesized images by employing two-stage optimization, where each stage optimizes a subset of losses in our objective function. Our key insight is that some of the losses, due to their similar design, can only be optimized sequentially and not jointly. Finally, LOHO uses gradient orthogonalization to explicitly disentangle hair attributes during the optimization process.

Our main contributions are:

- We propose a novel approach to perform hairstyle transfer by optimizing StyleGANv2's extended latent space and noise space.
- We propose an objective that includes multiple losses catered to model each key hairstyle attribute.
- We propose a two-stage optimization strategy that leads to significant improvements in the photorealism of synthesized images.
- We introduce gradient orthogonalization, a general method to jointly optimize attributes in latent space without interference. We demonstrate the effectiveness of gradient orthogonalization both qualitatively and quantitatively.
- We apply our novel approach to perform hairstyle transfer on in-the-wild portrait images and compute the Fréchet Inception Distance (FID) score. FID is used to evaluate generative models by calculating the distance between Inception [29] features for real and synthesized images in the same domain. The computed FID score shows that our approach outperforms the current state-of-the-art (SOTA) hairstyle transfer results.

2. Related Work

Generative Adversarial Networks. Generative models, in particular GANs, have been very successful across various computer vision applications such as image-to-image translation [15, 32, 40], video generation [34, 33, 9], and data augmentation for discriminative tasks such as object detection [24]. GANs [18, 3] transform a latent code to an image by learning the underlying distribution of training data. A more recent architecture, StyleGANv2 [20], has set the benchmark for generation of photorealistic human faces. However, training such networks requires significant amounts of data, significantly increasing the barrier to train SOTA GANs for specific use cases such as hairstyle transfer. Consequently, methods built using pretrained generators are becoming the de facto standard for executing various image manipulation tasks. In our work, we leverage [20] as an expressive pretrained face synthesis model, and outline our optimization approach for using pretrained generators for controlled attribute manipulation.

Latent Space Embedding. Understanding and manipulating the latent space of GANs via inversion has become an active field of research. GAN inversion involves embedding an image into the latent space of a GAN such that the synthesized image resulting from that latent embedding is the most accurate reconstruction of the original image. I2S [1] is a framework able to reconstruct an image by optimizing the extended latent space $\mathcal{W}^+$ of a pretrained StyleGAN [19]. Embeddings sampled from $\mathcal{W}^+$ are the concatenation of 18 different 512-dimensional w vectors, one for each layer of the StyleGAN architecture. I2S++ [2] further improved the image reconstruction quality by additionally optimizing the noise space $\mathcal{N}$. Furthermore, including semantic masks in the I2S++ framework allows users to perform tasks such as image inpainting and global editing. Recent methods [13, 27, 41] learn an encoder to map inputs from the image space directly to $\mathcal{W}^+$ latent space. Our work follows GAN inversion, in that our method optimizes the more recent StyleGANv2's $\mathcal{W}^+$ space and noise space $\mathcal{N}$ to perform semantic editing of hair on portrait images. We further propose a GAN inversion algorithm for simultaneous manipulation of spatially local attributes, such as hair structure, from multiple sources while preventing interference between the attributes' different competing objectives.

Hairstyle Transfer. Hair is a challenging part of the human face to model and synthesize. Previous work on modeling hair includes capturing hair geometry [8, 7, 6, 35], and using this hair geometry downstream for interactive hair editing. However, these methods are unable to capture key visual factors, thereby compromising the result quality. Although recent work [16, 23, 21] showed progress on using GANs for hair generation, these methods do not allow for intuitive control over the synthesized hair. MichiGAN [30] proposed a conditional synthesis GAN that allows controlled manipulation of hair. MichiGAN disentangles hair into four attributes by designing deliberate mechanisms and representations and produces SOTA results for hair appearance change. Nonetheless, MichiGAN has difficulty handling hair transfer scenarios with arbitrary shape change. This is because MichiGAN implements shape change using a separately trained inpainting network to fill "holes"

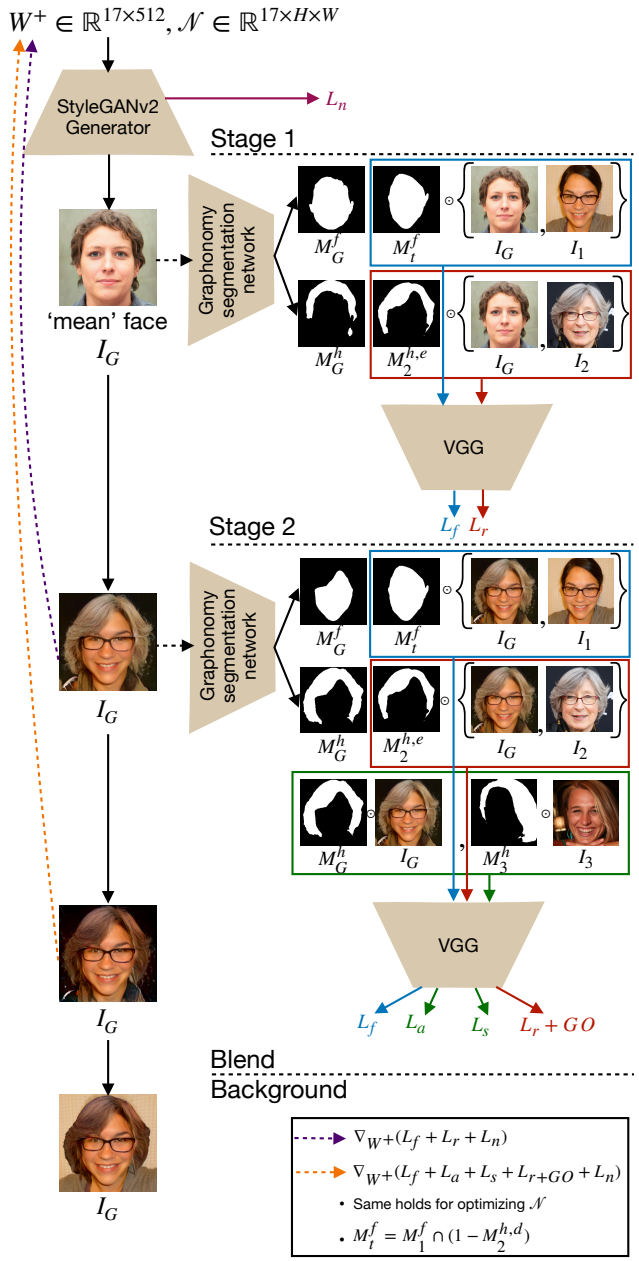

Figure 2: **LOHO**. Starting with the 'mean' face, LOHO reconstructs the target identity and the target perceptual structure of hair in Stage 1. In Stage 2, LOHO transfers the target hair style and appearance, while maintaining the perceptual structure via Gradient Orthogonalization (GO). Finally, I_G is blended with I_1's background. (Figure best viewed in colour)

created during the hair transfer process. In contrast, our method invokes the prior distribution of a pretrained GAN to "infill" in latent space rather than pixel space. As compared to MichiGAN, our method produces more realistic synthesized images in the challenging case where hair shape changes.

3. Methodology

3.1. Background

We begin by observing the objective function proposed in Image2StyleGAN++ (I2S++) [2]:

$$\begin{aligned} L = &\lambda_s L_{\text{style}}(M_s, G(w,n), y) \\ &+ \lambda_p L_{\text{percept}}(M_p, G(w,n), x) \\ &+ \frac{\lambda_{\text{mse1}}}{N}\|M_m \odot (G(w,n) - x)\|_2^2 \\ &+ \frac{\lambda_{\text{mse2}}}{N}\|(1 - M_m) \odot (G(w,n) - y)\|_2^2 \end{aligned} \tag{1}$$

where w is an embedding in the extended latent space $\mathcal{W}^+$ of StyleGAN, n is a noise vector embedding, M_s, M_m, and M_p are binary masks to specify image regions contributing to the respective losses, $\odot$ denotes the Hadamard product, G is the StyleGAN generator, x is the image that we want to reconstruct in mask M_m, and y is the image that we want to reconstruct outside M_m, i.e., in $(1 - M_m)$.

[2] use variations on the I2S++ objective function in Equation 1 to improve image reconstruction, image crossover, image inpainting, local style transfer, and other tasks. In our case for hairstyle transfer we want to do both image crossover and image inpainting. Transferring one hairstyle to another person requires crossover, and the left-over region where the original person's hair used to be requires inpainting.

3.2. Framework

For the hairstyle transfer problem, suppose we have three portrait images of people: I_1, I_2 and I_3. Consider transferring person 2's (I_2's) hair shape and structure, and person 3's (I_3's) hair appearance and style to person 1 (I_1). Let M_1^f be I_1's binary face mask, and M_1^h, M_2^h and M_3^h be I_1's, I_2's, and I_3's binary hair masks. Next, we separately dilate and erode M_2^h by roughly 20% to produce the dilated version, $M_2^{h,d}$, and the eroded version, $M_2^{h,e}$. Let $M_2^{h,\text{ir}} \equiv M_2^{h,d} - M_2^{h,e}$ be the ignore region that requires inpainting. We do not optimize $M_2^{h,\text{ir}}$, and rather invoke StyleGANv2 to inpaint relevant details in this region. This feature allows our method to perform hair shape transfer in situations where person 1 and person 2's hair shapes are misaligned.

In our method the background of I_1 is not optimized. Therefore, to recover the background, we soft-blend I_1's background with the synthesized image's foreground (hair and face). Specifically, we use GatedConv [36] to inpaint the masked out foreground region of I_1, following which we perform the blending (Figure 2).

3.3. Objective

To perform hairstyle transfer, we define the losses that we use to supervise relevant regions of the synthesized im-

age. To keep our notation simple, let $I_G \equiv G(\mathcal{W}^+, \mathcal{N})$ be the synthesized image, and M_G^f and M_G^h be its corresponding face and hair regions.

Identity Reconstruction. In order to reconstruct person 1's identity we use the Learned Perceptual Image Patch Similarity (LPIPS) [39] loss. LPIPS is a perceptual loss based on human similarity judgements and, therefore, is well suited for facial reconstruction. To compute the loss, we use pretrained VGG [28] to extract high-level features [17] for both I_1 and I_G. We extract and sum features from all 5 blocks of VGG to form our face reconstruction objective

$$L_f = \frac{1}{5}\sum_{b=1}^{5} \text{LPIPS}\big[\text{VGG}^b(I_1 \odot (M_1^f \cap (1 - M_2^{h,d}))), \text{VGG}^b(I_G \odot (M_1^f \cap (1 - M_2^{h,d})))\big] \quad (2)$$

where b denotes a VGG block, and $M_1^f \cap (1 - M_2^{h,d})$ represents the target mask, calculated as the overlap between M_1^f, and the foreground region of the dilated mask $M_2^{h,d}$. This formulation places a soft constraint on the target mask.

Hair Shape and Structure Reconstruction. To recover person 2's hair information, we enforce supervision via the LPIPS loss. However, naïvely using M_2^h as the target hair mask can cause the generator to synthesize hair on undesirable regions of I_G. This is especially true when the target face and hair regions do not align well. To fix this problem, we use the eroded mask, $M_2^{h,e}$, as it places a soft constraint on the target placement of synthesized hair. $M_2^{h,e}$, combined with $M_2^{h,\text{ir}}$, allow the generator to handle misaligned pairs by inpainting relevant information in non-overlapping regions. To compute the loss, we extract features from blocks 4 and 5 of VGG corresponding to hair regions of I_2, I_G to form our hair perceptual structure objective

$$L_r = \frac{1}{2}\sum_{b\in\{4,5\}} \text{LPIPS}\big[\text{VGG}^b(I_2 \odot M_2^{h,e}), \text{VGG}^b(I_G \odot M_2^{h,e})\big] \quad (3)$$

Hair Appearance Transfer. Hair appearance refers to the globally consistent colour of hair that is independent of hair shape and structure. As a result, it can be transferred from samples of different hair shapes. To transfer the target appearance, we extract 64 feature maps from the shallowest layer of VGG ($relu1\_1$) as it best accounts for colour information. Then, we perform average-pooling within the hair region of each feature map to discard spatial information and capture global appearance. We obtain an estimate of the mean appearance A in $\mathbb{R}^{64\times 1}$ as $A(x,y) = \sum \frac{\phi(x)\odot y}{|y|}$, where $\phi(x)$ represents the 64 VGG feature maps of image x, and y indicates the relevant hair mask. Finally, we compute the squared L_2 distance to give our hair appearance objective

$$L_a = \|A(I_3, M_3^h) - A(I_G, M_G^h)\|_2^2 \quad (4)$$

Hair Style Transfer. In addition to the overall colour, hair also contains finer details such as wisp styles, and shading variations between hair strands. Such details cannot be captured solely by the appearance loss that estimates the overall mean. Better approximations are thus required to compute the varying styles between hair strands. The Gram matrix [10] captures finer hair details by calculating the second-order associations between high-level feature maps. We compute the Gram matrix after extracting features from layers: $\{relu1\_2, relu2\_2, relu3\_3, relu4\_3\}$ of VGG

$$\mathcal{G}^l(\gamma^l) = \gamma^{l\intercal}\gamma^l \quad (5)$$

where, γ^l represents feature maps in $\mathbb{R}^{HW\times C}$ that are extracted from layer l, and $\mathcal{G}^l$ is the Gram matrix at layer l. Here, C represents the number of channels, and H and W are the height and width. Finally, we compute the squared L_2 distance as

$$L_s = \frac{1}{4}\sum_{l=1}^{4} \|\mathcal{G}^l(\text{VGG}^l(I_3 \odot M_3^h)) - \mathcal{G}^l(\text{VGG}^l(I_G \odot M_G^h))\|_2^2 \quad (6)$$

Noise Map Regularization. Explicitly optimizing the noise maps $n \in \mathcal{N}$ can cause the optimization to inject actual signal into them. To prevent this, we introduce regularization terms of noise maps [20]. For each noise map greater than 8×8, we use a pyramid down network to reduce the resolution to 8×8. The pyramid network averages 2×2 pixel neighbourhoods at each step. Additionally, we normalize the noise maps to be zero mean and unit variance, producing our noise objective

$$L_n = \sum_{i,j}\left[\frac{1}{r_{i,j}^2}\cdot\sum_{x,y} n_{i,j}(x,y)\cdot n_{i,j}(x-1,y)\right]^2 + \sum_{i,j}\left[\frac{1}{r_{i,j}^2}\cdot\sum_{x,y} n_{i,j}(x,y)\cdot n_{i,j}(x,y-1)\right]^2 \quad (7)$$

where $n_{i,0}$ represents the original noise map and $n_{i,j>0}$ represents the downsampled versions. Similarly, $r_{i,j}$ represents the resolution of the original or downsampled noise map.

Combining all the losses the overall optimization objective is

$$L = \arg\min_{\{\mathcal{W}^+,\mathcal{N}\}} \big[\lambda_f L_f + \lambda_r L_r + \lambda_a L_a + \lambda_s L_s + \lambda_n L_n\big] \quad (8)$$

Figure 3: **Effect of two-stage optimization. Col 1 (narrow)**: Reference images. **Col 2**: Identity person. **Col 3**: Synthesized image when losses arre optimized jointly. **Col 4**: Image synthesized via two-stage optimization + gradient orthogonalization.

3.4. Optimization Strategy

Two-Stage Optimization. Given the similar nature of the losses L_r, L_a, and L_s, we posit that jointly optimizing all losses from the start will cause person 2's hair information to compete with person 3's hair information, leading to undesirable synthesis. To mitigate this issue, we optimize our overall objective in two stages. In stage 1, we reconstruct only the target identity and hair perceptual structure, i.e., we set λ_a and λ_s in Equation 8 to zero. In stage 2, we optimize all the losses except L_r; stage 1 will provide a better initialization for stage 2, thereby leading the model to convergence.

However, this technique in itself has a drawback. There is no supervision to maintain the reconstructed hair perceptual structure after stage 1. This lack of supervision allows StyleGANv2 to invoke its prior distribution to inpaint or remove hair pixels, thereby undoing the perceptual structure initialization found in stage 1. Hence, it is necessary to include L_r in stage 2 of optimization.

Gradient Orthogonalization. L_r, by design, captures all hair attributes of person 2: perceptual structure, appearance, and style. As a result, L_r's gradient competes with the gradients corresponding to the appearance and style of person 3. We fix this problem by manipulating L_r's gradient such that its appearance and style information are removed. More specifically, we project L_r's perceptual structure gradients onto the vector subspace orthogonal to its appearance and style gradients. This allows person 3's hair appearance and style to be transferred while preserving person 2's hair structure and shape.

Assuming we are optimizing the $\mathcal{W}^+$ latent space, the gradients computed are

$$g_{R_2} = \nabla_{\mathcal{W}^+} L_r, g_{A_2} = \nabla_{\mathcal{W}^+} L_a, g_{S_2} = \nabla_{\mathcal{W}^+} L_s, \quad (9)$$

where, L_r, L_a, and L_s are the LPIPS, appearance, and style losses computed between I_2 and I_G. To enforce orthogonality, we would like to minimize ${g_{R_2}}^\intercal (g_{A_2} + g_{S_2})$. We achieve this by projecting away the component of g_{R_2} parallel to $(g_{A_2} + g_{S_2})$, using the structure-appearance gradient orthogonalization

$$g_{R_2} = g_{R_2} - \frac{{g_{R_2}}^\intercal (g_{A_2} + g_{S_2})}{\|g_{A_2} + g_{S_2}\|_2^2} (g_{A_2} + g_{S_2}) \quad (10)$$

after every iteration in stage 2 of optimization.

4. Experiments and Results

4.1. Implementation Details

Datasets. We use the Flickr-Faces-HQ dataset (FFHQ) [19] that contains 70 000 high-quality images of human faces. Flickr-Faces-HQ has significant variation in terms of ethnicity, age, and hair style patterns. We select tuples of images (I_1, I_2, I_3) based on the following constraints: (a) each image in the tuple should have at least 18% of pixels contain hair, and (b) I_1 and I_2's face regions must align to a certain degree. To enforce these constraints we extract hair and face masks using the Graphonomy segmentation network [11] and estimate 68 2D facial landmarks using 2D-FAN [4]. For every I_1 and I_2, we compute the intersection over union (IoU) and pose distance (PD) using the corresponding face masks, and facial landmarks. Finally, we distribute selected tuples into three categories, *easy*, *medium*, and *difficult*, such that the following IoU and PD constraints are both met

Category	Easy	Medium	Difficult
IoU range	$(0.8, 1.0]$	$(0.7, 0.8]$	$(0.6, 0.7]$
PD range	$[0.0, 2.0)$	$[2.0, 4.0)$	$[4.0, 5.0)$

Table 1: **Criteria used to define the alignment of head pose between sample tuples.**

Training Parameters. We used the Adam optimizer [22] with an initial learning rate of 0.1 and annealed it using a cosine schedule [20]. The optimization occurs in two stages, where each stage consists of 1000 iterations. Based on ablation studies, we selected an appearance loss weight λ_a of 40, style loss weight λ_s of 1.5×10^4, and noise regularization weight λ_n of 1×10^5. We set the remaining loss weights to 1.

Figure 4: **Effect of Gradient Orthogonalization (GO). Row 1**: Reference images (from left to right): Identity, target hair appearance and style, target hair structure and shape. **Row 2**: Pairs (a) and (b), and (c) and (d) are synthesized images and their corresponding hair masks for no-GO and GO methods, respectively.

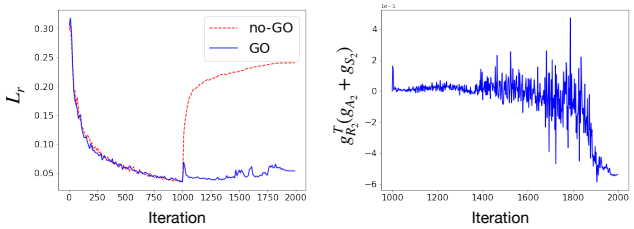

Figure 5: **Effect of Gradient Orthogonalization (GO). Left**: LPIPS hair reconstruction loss (GO vs no-GO) vs iterations. **Right**: Trend of ${g_{R_2}}^{\intercal}(g_{A_2} + g_{S_2})$ ($\times$1e-5) in stage 2 of optimization.

4.2. Effect of Two-Stage Optimization

Optimizing all losses in our objective function causes the framework to diverge. While the identity is reconstructed, the hair transfer fails (Figure 3). The structure and shape of the synthesized hair is not preserved, causing undesirable results. On the other hand, performing optimization in two stages clearly improves the synthesis process leading to generation of photorealistic images that are consistent with the provided references. Not only is the identity reconstructed, the hair attributes are transferred as per our requirements.

4.3. Effect of Gradient Orthogonalization

We compare two variations of our framework: no-GO and GO. GO involves manipulating L_r's gradients via gradient orthogonalization, whereas no-GO keeps L_r untouched. No-GO is unable to retain the target hair shape, causing L_r to increase in stage 2 of optimization i.e., after iteration 1000 (Figures 4 & 5). The appearance and style losses, being position invariant, do not contribute to the shape. GO, on the other hand, uses the reconstruction loss in stage 2 and retains the target hair shape. As a result, the IoU computed between M_2^h and M_G^h increases from 0.857 (for no-GO) to 0.932 (GO).

In terms of gradient disentanglement, the similarity between g_{R_2} and $(g_{A_2} + g_{S_2})$ decreases with time, indicating that our framework is able to disentangle person 2's hair shape from its appearance and style (Figure 5). This disentanglement allows a seamless transfer of person 3's hair appearance and style to the synthesized image without causing model divergence. Here on, we will use the GO version of our framework for comparisons and analysis.

4.4. Comparison with SOTA

Hair Style Transfer. We compare our approach with the SOTA model MichiGAN. MichiGAN contains separate modules to estimate: (1) hair appearance, (2) hair shape and structure, and (3) background. The appearance module bootstraps the generator with its output feature map, replacing the randomly sampled latent code in traditional GANs [12]. The shape and structure module outputs hair masks and orientation masks, denormalizing each SPADE ResBlk [25] in the backbone generation network. Finally, the background module progressively blends the generator outputs with background information. In terms of training, MichiGAN follows the pseudo-supervised regime. Specifically, the features, that are estimated by the modules, from the same image are fed into MichiGAN to reconstruct the original image. At test time, FID is calculated for 5000 images at 512 px resolution uniform randomly sampled from FFHQ's test split.

To ensure that our results are comparable, we follow the above procedure and compute FID scores [14] for LOHO. In addition to computing FID on the entire image, we calculate the score solely relying on the synthesized hair and facial regions with the background masked out. Achieving a low FID score on masked images would mean that our model is indeed capable of synthesizing realistic hair and face regions. We call this LOHO-HF. As MichiGAN's background inpainter module is not publicly available, we use GatedConv [36] to inpaint relevant features in the masked out hair regions.

Quantitatively, LOHO outperforms MichiGAN. Our method achieves an FID score of 8.419, while MichiGAN achieves 10.697 (Table 2). This improvement indicates that our optimization framework is able to synthesize high quality images. LOHO -HF achieves an even lower score of 4.847, attesting to the superior quality of the synthesized hair and face regions.

Qualitatively, our method is able to synthesize better results for challenging examples. LOHO naturally blends the target hair attributes with the target face (Figure 15). MichiGAN naïvely copies the target hair on the target face, causing lighting inconsistencies between the two regions. LOHO handles pairs with varying degrees of misalignment whereas MichiGAN is unable to do so due to its reliance on blending background and foreground information in pixel space rather than latent space. Lastly, LOHO transfers rele-

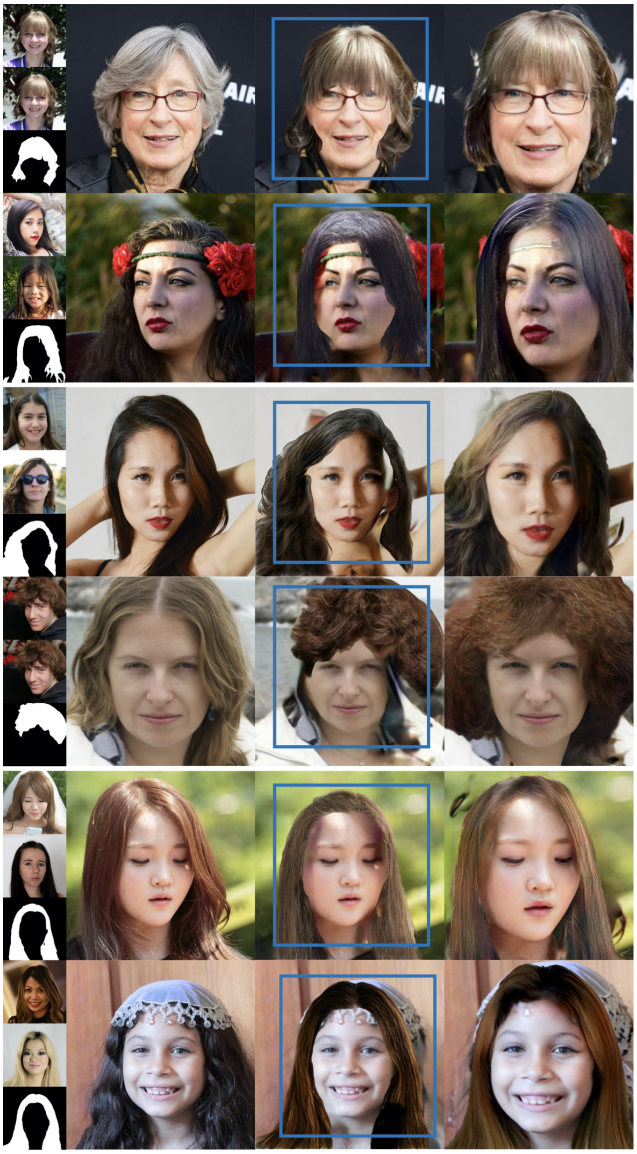

Figure 6: Qualitative comparison of MichiGAN and LOHO. **Col 1 (narrow)**: Reference images. **Col 2**: Identity person **Col 3**: MichiGAN output. **Col 4**: LOHO output (zoomed in for better visual comparison). **Rows 1-2**: MichiGAN "copy-pastes" the target hair attributes while LOHO blends the attributes, thereby synthesizing more realistic images. **Rows 3-4**: LOHO handles misaligned examples better than MichiGAN. **Rows 5-6**: LOHO transfers the right style information.

vant style information, on par with MichiGAN. In fact, due to our addition of the style objective to optimize second-order statistics by matching Gram matrices, LOHO synthesizes hair with varying colour even when the hair shape source person has uniform hair colour, as in the bottom two rows of Figure 15.

Identity Reconstruction Quality. We also compare LOHO with two recent image embedding methods: I2S [1] and I2S++ [2]. introduces the framework that is able to

Method	MichiGAN	LOHO-HF	LOHO
FID (↓)	10.697	4.847	8.419

Table 2: **Frechet Inception Distance (FID) for different methods**. We use 5000 images uniform-randomly sampled from the testing set of FFHQ. ↓ indicates that lower is better.

Method	I2S	I2S++	LOHO
PSNR (dB) (↑)	-	22.48	32.2 ± 2.8
SSIM (↑)	-	0.91	0.93 ± 0.02
$\|w^* - \hat{\mathbf{w}}\|$	$[30.6, 40.5]$	-	37.9 ± 3.0

Table 3: **PSNR, SSIM and range of acceptable latent distances $\|w^* - \hat{\mathbf{w}}\|$ for different methods**. We use randomly sampled 5000 images from the testing set of FFHQ. - indicates N/A. ↑ indicates that higher is better.

reconstruct images of high quality by optimizing the $\mathcal{W}^+$ latent space. I2S also shows how the latent distance, calculated between the optimized style latent code w^* and $\hat{\mathbf{w}}$ of the average face, is related to the quality of synthesized images. I2S++, additionally to I2S, optimizes the noise space $\mathcal{N}$ in order to reconstruct images with high PSNR and SSIM values. Therefore, to assess LOHO's ability to reconstruct the target identity with high quality, we compute similar metrics on the facial region of synthesized images. Since inpainting in latent space is an integral part of LOHO we compare our results with I2S++'s performance on image inpainting at 512 px resolution.

Our model, despite performing the difficult task of hair style transfer, is able to achieve comparable results (Table 3). I2S shows that the acceptable latent distance for a valid human face is in $[30.6, 40.5]$ and LOHO lies within that range. Additionally, our PSNR and SSIM scores are better than I2S++, proving that LOHO reconstructs identities that satisfy local structure information.

4.5. Editing Attributes

Our method is capable of editing attributes of in-the-wild portrait images. In this setting, an image is selected and then an attribute is edited individually by providing reference images. For example, the hair structure and shape can be changed while keeping the hair appearance and background unedited. Our framework computes the non-overlapping hair regions and infills the space with relevant background details. Following the optimization process, the synthesized image is blended with the inpainted background image. The same holds for changing the hair appearance and style. LOHO disentangles hair attributes and allows editing them individually and jointly, thereby leading to desirable results (Figures 7 & 8) .

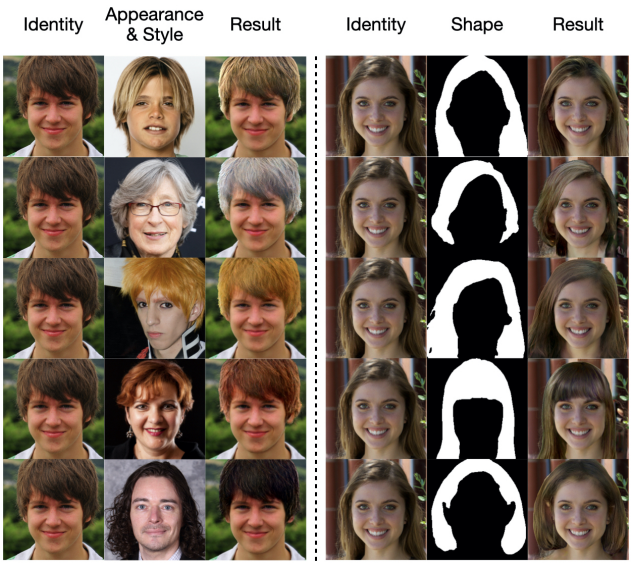

Figure 7: **Individual attribute editing**. The results show that our model is able to edit individual hair attributes (**left**: appearance & style left, **right**: shape) without them interfering with each other.

Figure 8: **Multiple attributes editing**. The results show that our model is able to edit hair attributes jointly without the interference of each other.

Figure 9: **Misalignment examples. Col 1 (narrow)**: Reference images. **Col 2**: Identity image. **Col 3**: Synthesized image. Extreme cases of misalignment can result in misplaced hair.

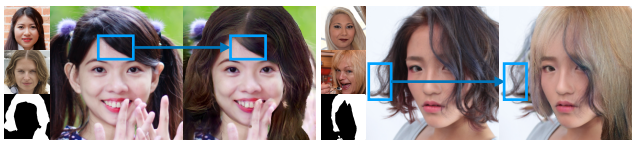

Figure 10: **Hair trail**. **Col 1 (narrow)**: Reference images. **Col 2**: Identity image. **Col 3**: Synthesized image. Cases where there are remnants of hair information from the identity person. The regions marked inside the blue box carries over to the synthesized image.

5. Limitations

Our approach is susceptible to extreme cases of misalignment (Figure 9). In our study, we categorize such cases as *difficult*. They can cause our framework to synthesize unnatural hair shape and structure. GAN based alignment networks [38, 5] may be used to transfer pose, or alignment of hair across *difficult* samples.

In some examples, our approach can carry over hair details from the identity person (Figure 10). This can be due to Graphonomy [11]'s imperfect segmentation of hair. More sophisticated segmentation networks [37, 31] can be used to mitigate this issue.

6. Conclusion

Our introduction of LOHO, an optimization framework that performs hairstyle transfer on portrait images, takes a step in the direction of spatially-dependent attribute manipulation with pretrained GANs. We show that developing algorithms that approach specific synthesis tasks, such as hairstyle transfer, by manipulating the latent space of expressive models trained on more general tasks, such as face synthesis, is effective for completing many downstream tasks without collecting large training datasets. GAN inversion approach is able to solve problems such as realistic hole-filling more effectively, even than feedforward GAN pipelines that have access to large training datasets. There are many possible improvements to our approach for hair synthesis, such as introducing a deformation objective to enforce alignment over a wide range of head poses and hair shapes, and improving convergence by predicting an initialization point for the optimization process.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 7

[2] R. Abdal, Y. Qin, and P. Wonka. Image2stylegan++: How to edit the embedded images? In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8293–8302, 2020. 2, 3, 7

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 2

[4] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017. 5, 11

[5] Egor Burkov, Igor Pasechnik, Artur Grigorev, and Victor Lempitsky. Neural head reenactment with latent pose descriptors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 8

[6] Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. High-quality hair modeling from a single portrait photo. *ACM Transactions on Graphics*, 34:1–10, 10 2015. 2

[7] Menglei Chai, Lvdi Wang, Yanlin Weng, Xiaogang Jin, and Kun Zhou. Dynamic hair manipulation in images and videos. *ACM Transactions on Graphics (TOG)*, 32, 07 2013. 2

[8] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. Single-view hair modeling for portrait manipulation. *ACM Transactions on Graphics*, 31, 07 2012. 2

[9] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 4

[11] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *CVPR*, 2019. 5, 8, 11

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, 2014. 1, 6

[13] Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. Collaborative learning for faster stylegan embedding, 2020. 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. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6626–6637. Curran Associates, Inc., 2017. 6

[15] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 1, 2

[16] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 4

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2017. 1, 2

[19] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 2, 4, 5

[21] Vladimir Kim, Ersin Yumer, and Hao Li. Real-time hair rendering using sequential adversarial networks. In *European Conference on Computer Vision*, 2018. 2

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

[23] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[24] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan. Perceptual generative adversarial networks for small object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1951–1959, 2017. 2

[25] T. Park, M. Liu, T. Wang, and J. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341, 2019. 6

[26] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [to appear]. 1

[27] 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. *arXiv preprint arXiv:2008.00951*, 2020. 2

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 4

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision.

In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. 2

[30] Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. Michigan: Multi-input-conditioned hair image generation for portrait editing. *ACM Transactions on Graphics (TOG)*, 39(4):1–13, 2020. 1, 2, 12

[31] A. Tao, K. Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *ArXiv*, abs/2005.10821, 2020. 8

[32] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 1, 2

[33] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[34] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018. 2

[35] Yanlin Weng, Lvdi Wang, Xiao Li, Menglei Chai, and Kun Zhou. Hair interpolation for portrait morphing. *Computer Graphics Forum*, 32, 10 2013. 2

[36] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang. Free-form image inpainting with gated convolution. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4470–4479, 2019. 3, 6

[37] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Computer Vision – ECCV 2020*, pages 173–190, 2020. 8

[38] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9458–9467, 2019. 8

[39] 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

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

[41] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 2

A. Images and Masks

For each selected tuple (I_1, I_2, I_3), we extract hair and face masks using Graphonomy [11]. We separately dilate and erode M_2^h, the hair mask of I_2, to produce the dilated version, $M_2^{h,d}$, and the eroded version, $M_2^{h,e}$. Using $M_2^{h,d}$ and $M_2^{h,e}$, we compute the ignore region $M_2^{h,ir}$. We exclude the ignore region from the background and let StyleGANv2 inpaint relevant features. We want to optimize for reconstruction of I_1's face, reconstruction of I_2's hair shape and structure, transfer of I_3's hair appearance and style, and inpainting of the ignore region. Given a tuple, Figure 11 shows the images and relevant masks used during optimization.

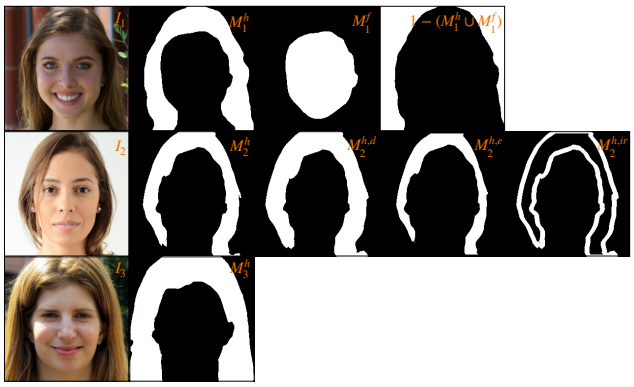

Figure 11: Tuple (I_1, I_2, I_3) and relevant masks used in LOHO.

B. Alignment Metrics

To categorize each selected tuple (I_1, I_2, I_3), we calculate the Intersection over Union (IoU) and pose distance (PD) between face masks, and 68 2D facial landmarks. We extract the masks using Graphonomy [11], and estimate landmarks using 2D-FAN [4].

IoU and PD quantify to what degree two faces align. Given the two binary face masks, M_1^f and M_2^f, we compute IoU as

$$\text{IoU} = \frac{M_1^f \cap M_2^f}{M_1^f \cup M_2^f}. \tag{11}$$

The pose distance (PD), on the other hand, is defined in terms of facial landmarks. Given the two 68 2D facial landmarks, $K_1^f \in \mathbb{R}^{68\times 2}$ and $K_2^f \in \mathbb{R}^{68\times 2}$, corresponding to I_1 and I_2, PD is calculated by averaging the L_2 distances computed between each landmark

$$\text{PD} = \frac{1}{68}\sum_{k=1}^{68} \|K_{1,k}^f - K_{2,k}^f\|_2 \tag{12}$$

where k indexes the 2D landmarks. Therefore, a tuple where I_1 and I_2 are the same person (Figure 12) would have an IoU of 1.0 and PD of 0.0.

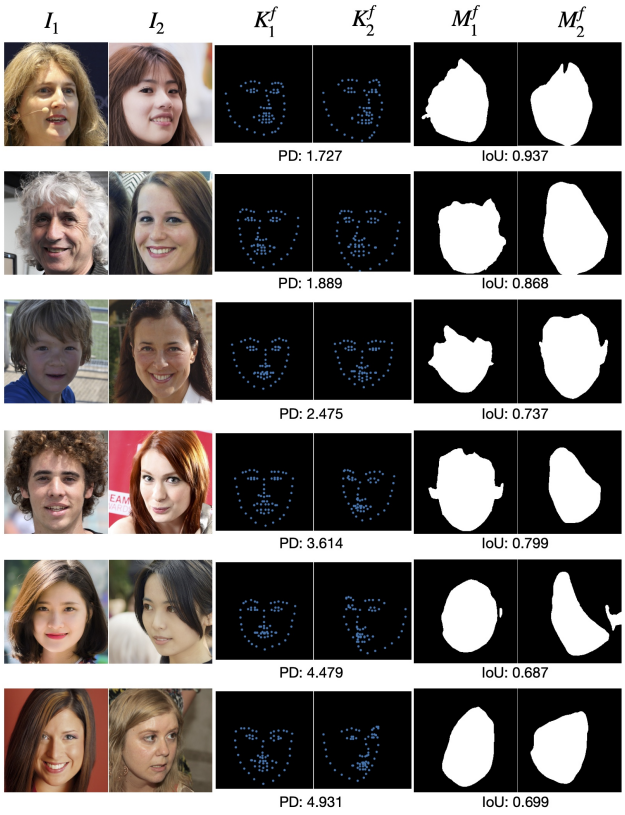

Figure 12: IoU and PD for tuples in each category. **Rows 1-2**: *Easy* tuples. **Rows 3-4**: *Medium* tuples. **Rows 5-6**: *Difficult* tuples.

C. StyleGANv2 Architecture

StyleGANv2 [**?**] can synthesize novel photorealistic images at different resolutions including 128^2, 256^2, 512^2 and 1024^2. The number of layers in the architecture therefore depends on the resolution of images being synthesized. Additionally, the size of the extended latent space $\mathcal{W}^+$ and the noise space $\mathcal{N}$ also depend on on the resolution. Embeddings sampled from $\mathcal{W}^+$ are concatenations of 512-dimensional vectors w, where $w \in \mathcal{W}^+$. As our experiments synthesize images of resolution 512^2, the latent space is a vector subspace of $\mathbb{R}^{15\times 512}$, i.e., $\mathcal{W}^+ \subset \mathbb{R}^{15\times 512}$. Additionally, noise maps sampled from $\mathcal{N}$ are tensors of dimension $\mathbb{R}^{1\times 1\times h\times w}$, where h and w match the spatial resolution of feature maps at every layer of the StyleGANv2 generator.

D. Effect of Regularizing Noise Maps

To understand the effect of noise map regularization, we visualize noise maps at different resolutions post optimization. When the regularization term is set to zero, we nor-

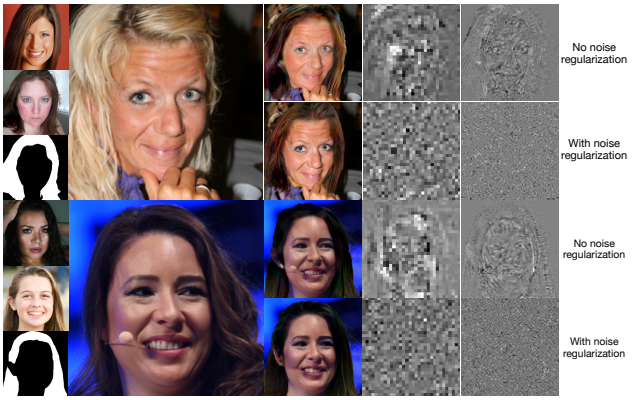

Figure 13: **Effect of regularizing noise maps**. **Col 1 (narrow)**: Reference images. **Col 2**: Identity person. **Col 3**: Synthesized images. **Cols 4&5**: Noise maps at different resolutions.

Figure 14: **Effect of Gradient Orthogonalization (GO). Rows 1&3**: Reference images (from left to right): Identity, target hair appearance and style, target hair structure and shape. **Rows 2&4**: Pairs (a) and (b), and (c) and (d) are synthesized images and their corresponding hair masks for no-GO and GO methods, respectively. The same holds for pairs (e) and (f), and (g) and (h).

malize the noise maps to be zero mean and unit variance. This causes the optimization to inject actual signal into the noise maps, thereby causing overfitting. Figure 13 shows that the noise maps encode structural information of the facial region, which is not desirable, and cause the synthesized images to have artifacts in the face and hair regions. Enabling noise regularization prevents this.

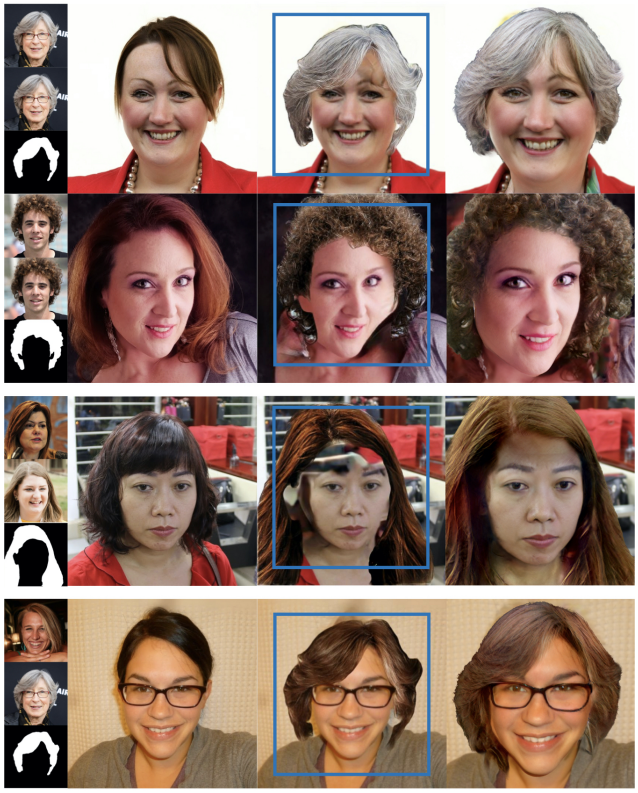

Figure 15: Qualitative comparison of MichiGAN and LOHO. **Col 1 (narrow)**: Reference images. **Col 2**: Identity person **Col 3**: MichiGAN output. **Col 4**: LOHO output (zoomed in for better visual comparison). **Rows 1-2**: MichiGAN "copy-pastes" the target hair attributes while LOHO blends the attributes, thereby synthesizing more realistic images. **Row 3**: LOHO handles misaligned examples better than MichiGAN. **Row 4**: LOHO transfers the right style information.

E. Additional Examples of Gradient Orthogonalization

Gradient Orthogonalization (GO) allows LOHO to retain the target hair shape and structure during stage 2 of optimization. Figure 14 shows that no-GO fails to maintain the perceptual structure. On the other hand, GO is able to maintain the target perceptual structure while transferring the target hair appearance and style. As a result, the IoU calculated between M_2^h and M_G^h increases from 0.547 (no-GO, Figure 14 (b)) to 0.603 (GO, Figure 14 (d)). In the same way, the IoU increases from 0.834 (no-GO, Figure 14 (f)) to 0.857 (GO, Figure 14 (h)).

F. Additional comparisons with MichiGAN

We provide additional evidence to show that LOHO addresses blending and misalignment better than MichiGAN [30]. The ignore region $M_2^{h,ir}$ (Figure 11), in addition

to StyleGANv2's powerful learned representations, is able to inpaint relevant hair and face pixels. This infilling causes the synthesized image to look more photorealistic as compared with MichiGAN. In terms of style transfer, LOHO achieves similar performance as MichiGAN (Figure 15).

G. Additional Results of LOHO

We present results to show that LOHO is able to edit individual hair attributes, such as appearance and style (Figure 16), and shape (Figure 17), while keeping other attributes unchanged. LOHO is also able to manipulate multiple hair attributes jointly (Figure 18,19,20).

Figure 16: **Transfer of appearance and style**. Given an identity image, and reference image, LOHO transfers the target hair appearance and style while preserving the hair structure and shape. **Row 1**: Identity images. **Rows 2-6**: Hair appearance and style references (Cols: $1, 3, 5$), and synthesized images (Cols: $2, 4, 6$).

Figure 17: **Transfer of shape**. Given an identity image, and reference image, LOHO transfers the target hair shape while preserving the hair appearance and style. **Row 1**: Identity images. **Rows 2-6**: Hair shape references (Cols: $1, 3, 5$), and synthesized images (Cols: $2, 4, 6$).

Figure 18: **Multiple attributes editing**. Given an identity image, and reference images, LOHO transfers the target hair attributes.

Figure 19: **Multiple attributes editing**. Given an identity image, and reference images, LOHO transfers the target hair attributes.

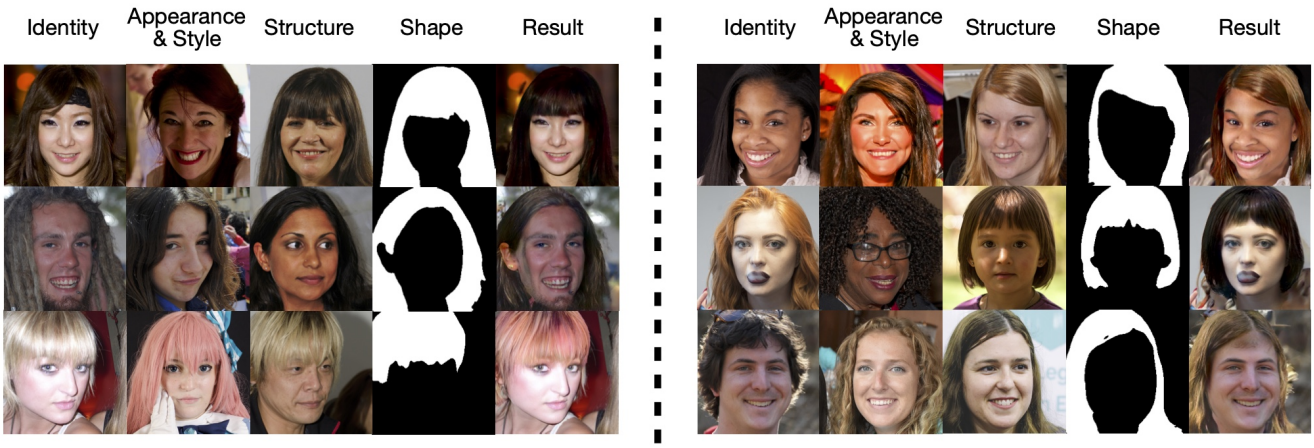

Figure 20: **Multiple attributes editing**. Given an identity image, and reference images, LOHO transfers the target hair attributes.