LOHO: Latent Optimization of Hairstyles via Orthogonalization [CVPR'21]

University of Toronto, Modiface, Inc.,
Rohit Saha et al.
Presenter : Taeu

Task

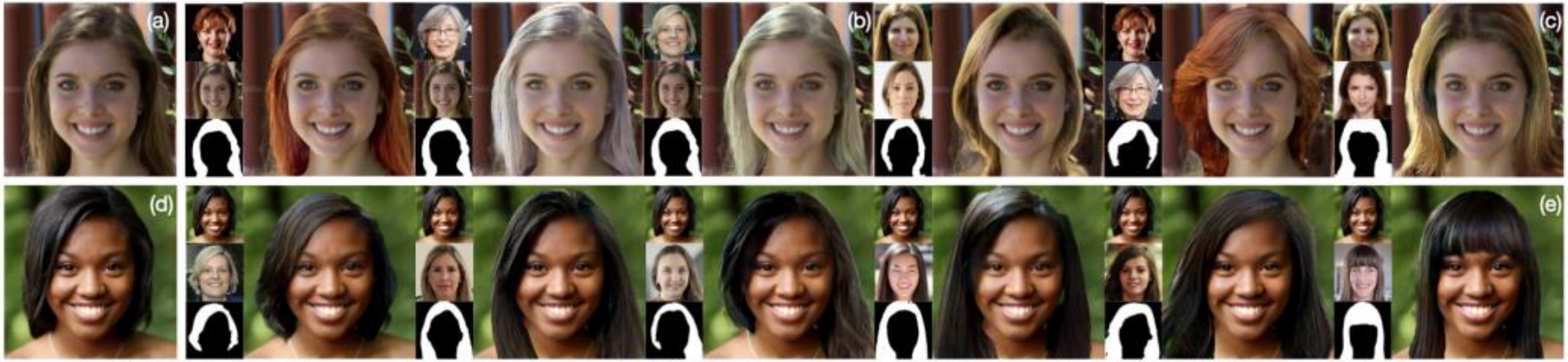


Figure 1: **Hairstyle transfer samples synthesized using LOHO.** For given portrait images (a) and (d), LOHO is capable of manipulating hair attributes based on multiple input conditions. Inset images represent the target hair attributes in the order: appearance and style, structure, and shape. LOHO 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).

Task

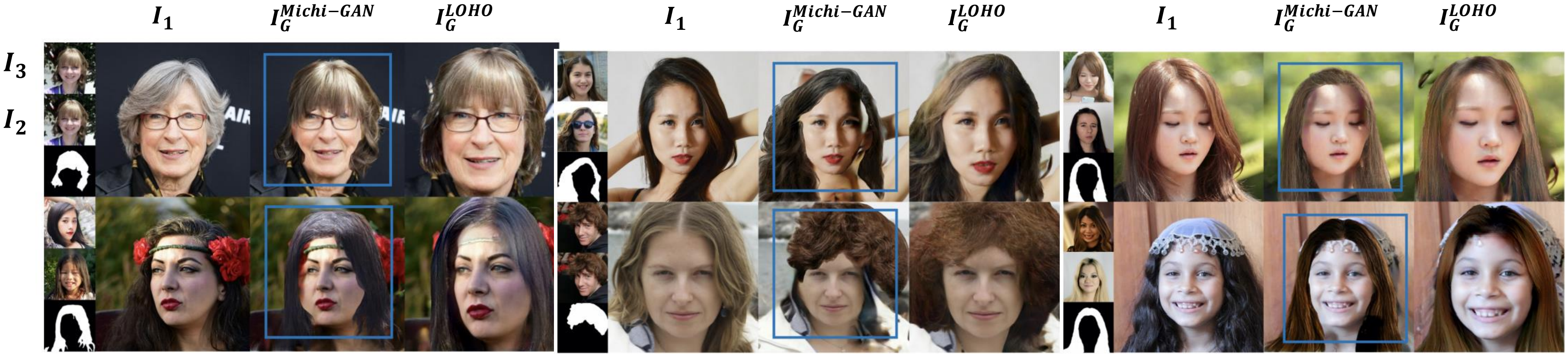
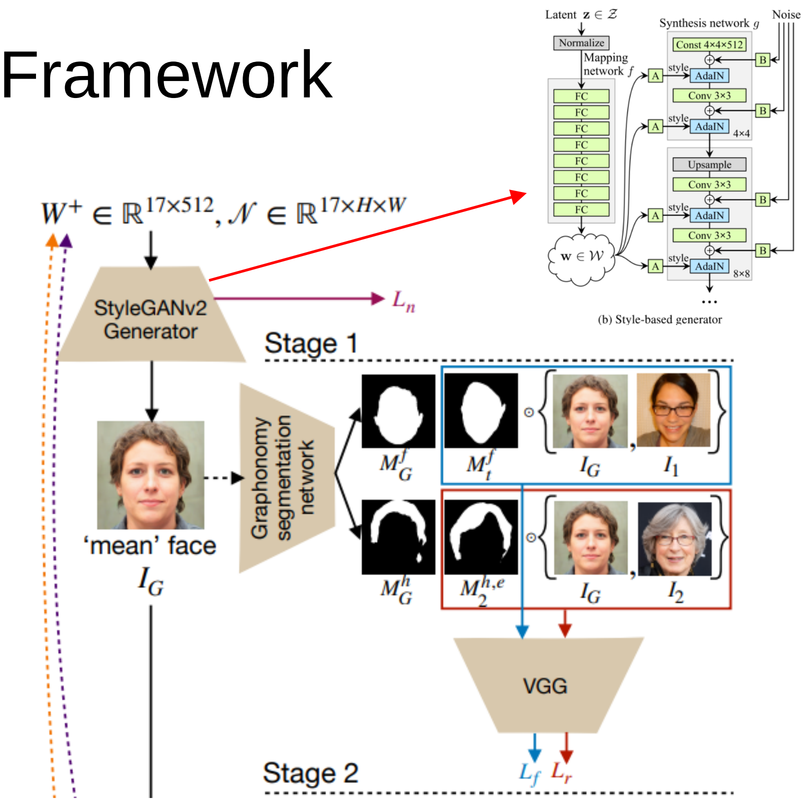


Figure 6: Qualitative comparison of MichiGAN and LOHO. Col 1 (narrow): Reference images. Col 2: Identity person Col 3: MichiGAN output. Col 4: LOHO output (zoomed in for better visual comparison). Rows 1-2: MichiGAN "copy-pastes" the target hair attributes while LOHO blends the attributes, thereby synthesizing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

Contribution

- To perform hairstyle transfer by **optimizing StyleGANv2's** extended latent space and noise space
- An objective that includes **multiple losses** catered to model each key hairstyle attribute.
- A **two-stage optimization strategy** that leads to significant improvements in the photorealism of synthesized images.
- A **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.
- The computed FID score shows that our approach outperforms the current **state-of-the-art (SOTA)** hairstyle transfer results

Framework



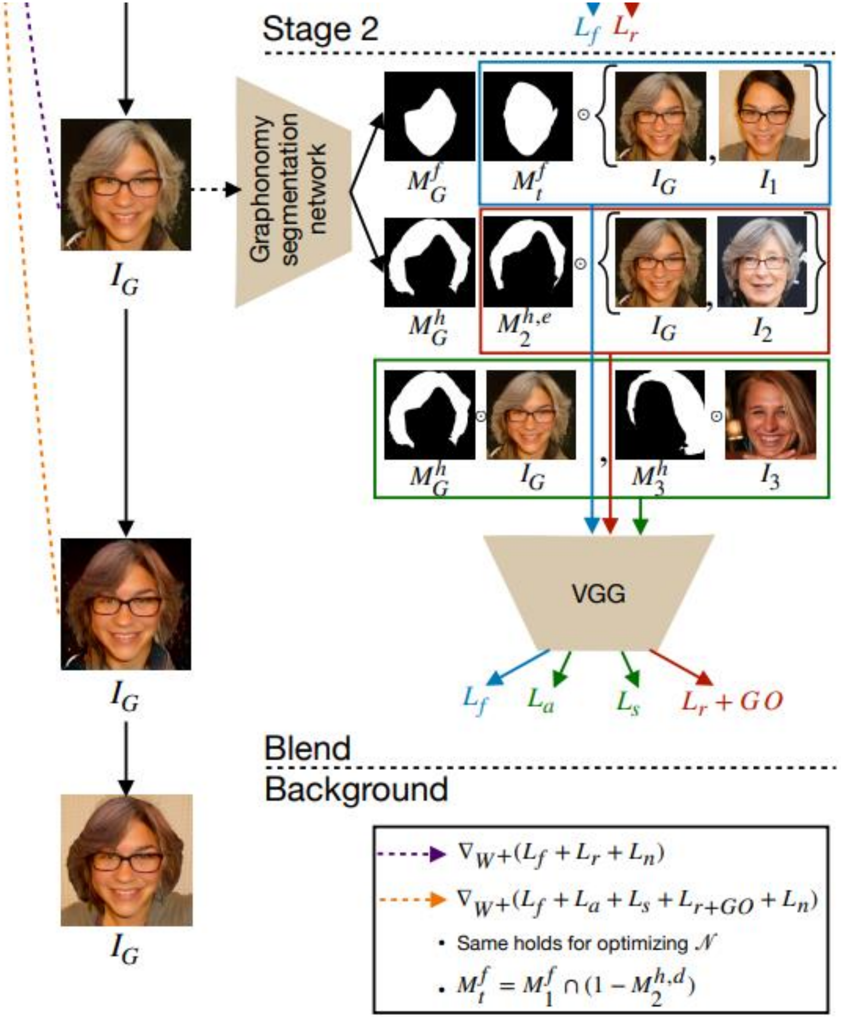
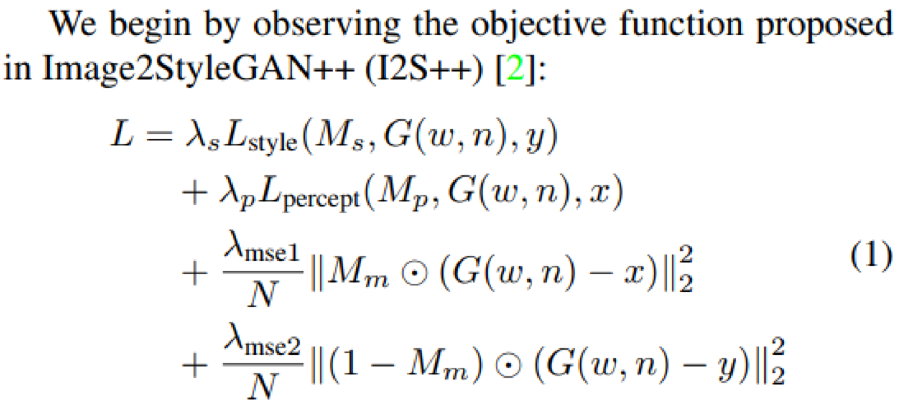

Stage 2
L_f
L_r
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_2
M_G^h
I_G
M_3^h
I_3
VGG
L_f
L_a
L_s
$L_r + GO$
Blend
Background
$\nabla_{W^+}(L_f + L_r + L_n)$
$\nabla_{W^+}(L_f + L_a + L_s + L_{r+GO} + L_n)$
Same holds for optimizing $\mathcal{N}$
$M_t^f = M_1^f \cap (1 - M_2^{h,d})$

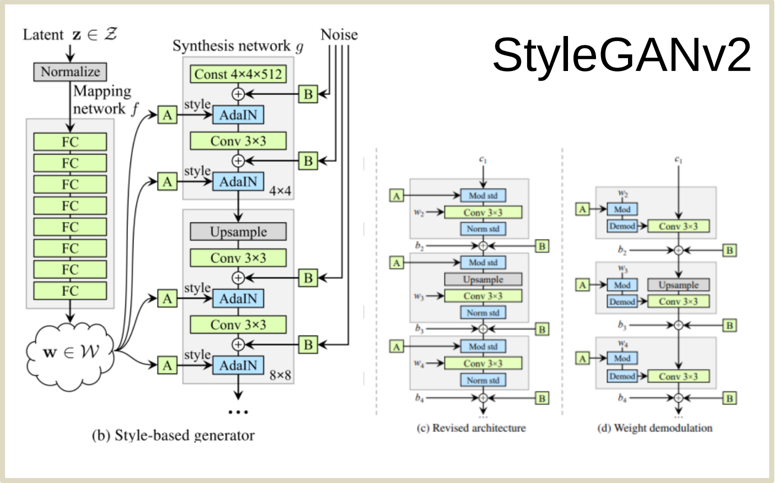
Figure 2: **LOHO**. Starting with the 'mean' face, LOHO reconstructs the target identity and the target perceptual structure of hair in Stage 1. In Stage 2, LOHO transfers the target hair style and appearance, while maintaining the perceptual structure via Gradient Orthogonalization (GO). Finally, I_G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

Background



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

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



StyleGANv2

(b) Style-based generator

(c) Revised architecture

(d) Weight demodulation

- Improve image reconstruction, image crossover, image inpainint, local style transfer, and other tasks.
- For this task, to do both image crossover and image inpainting.
- **Transferring one hairstyle** to another person requires **crossover**.
- **The leftover region** where the original person’s hair used to be requires **inpainting**.

Framework

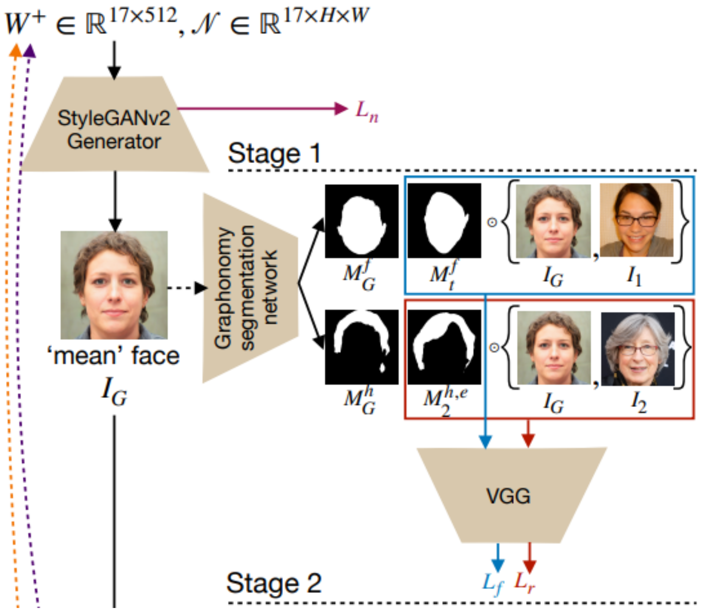
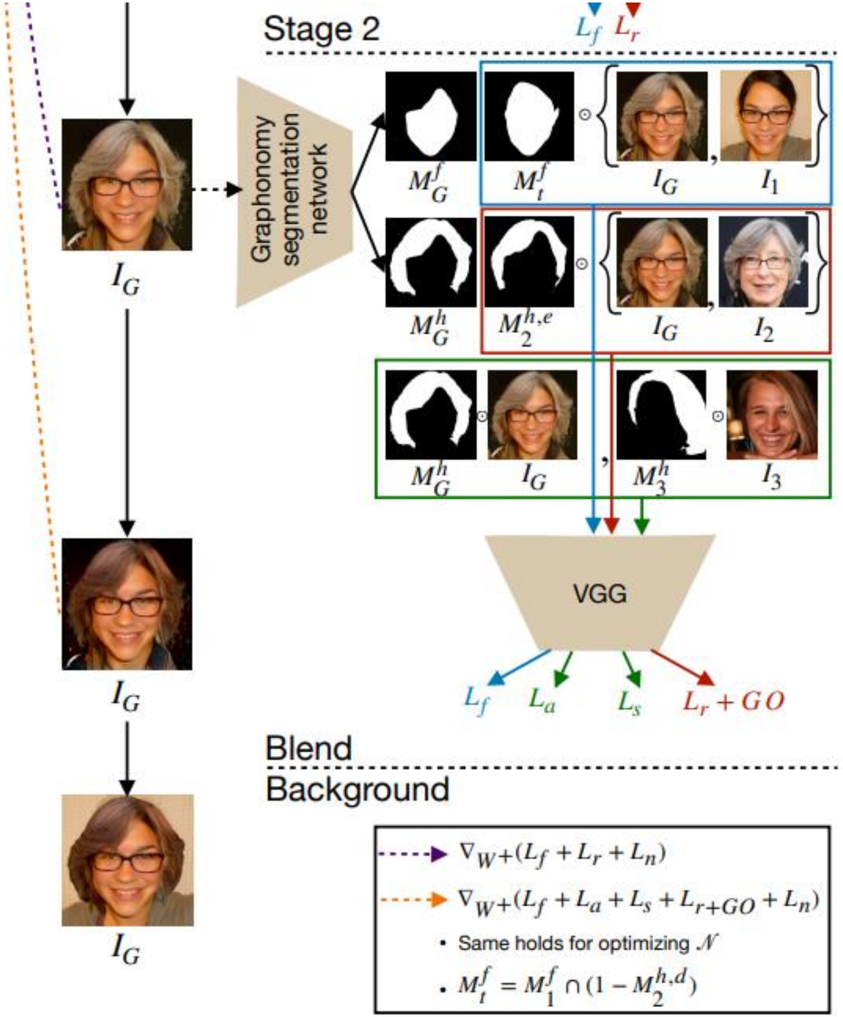
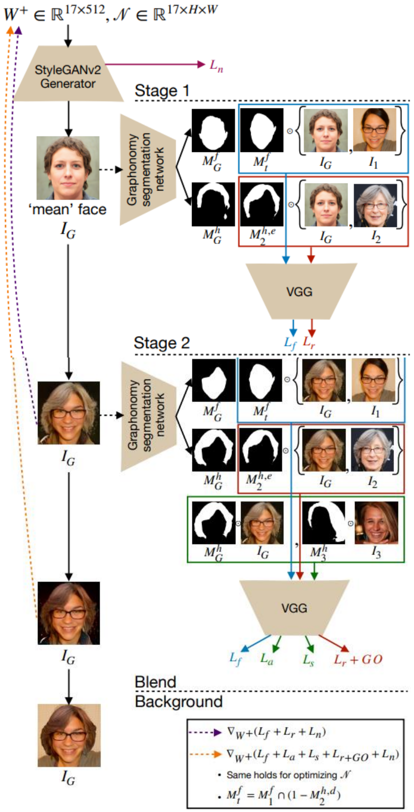


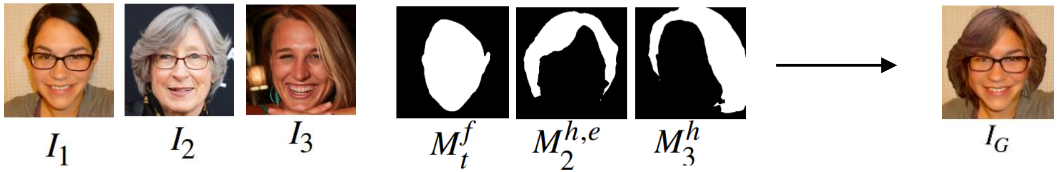
Figure 2: **LOHO**. Starting with the 'mean' face, LOHO reconstructs the target identity and the target perceptual structure of hair in Stage 1. In Stage 2, LOHO transfers the target hair style and appearance, while maintaining the perceptual structure via Gradient Orthogonalization (GO). Finally, I_G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


Stage 2
L_f
L_r
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
M_G^h
I_G
M_3^h
I_3
VGG
L_f
L_a
L_s
$L_r + GO$
I_G
Blend
Background
I_G
$\nabla_{W^+}(L_f + L_r + L_n)$
$\nabla_{W^+}(L_f + L_a + L_s + L_{r+GO} + L_n)$
Same holds for optimizing $\mathcal{N}$
$M_t^f = M_1^f \cap (1 - M_2^{h,d})$

Framework

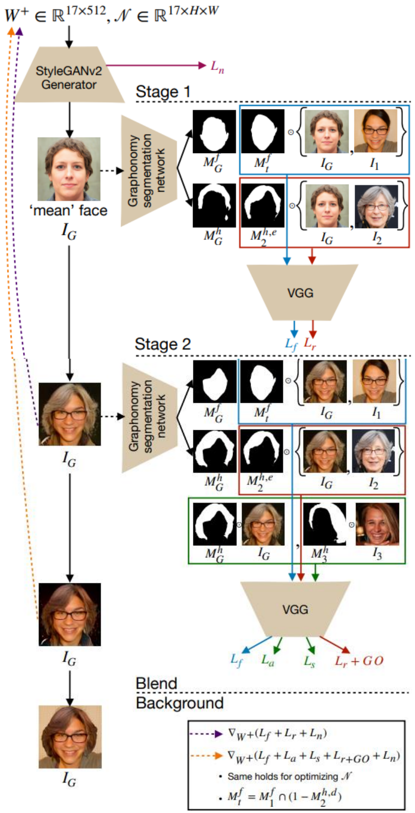


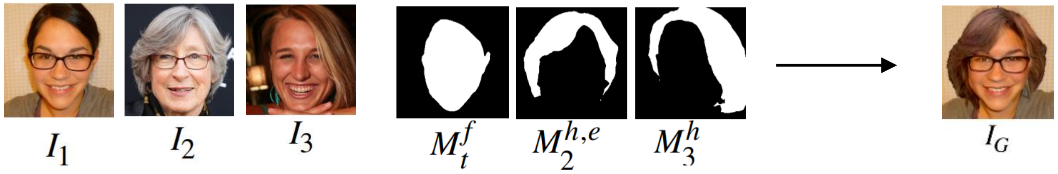
$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
'mean' face
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
VGG
L_f
L_r
Stage 2
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
M_G^h
I_G
M_3^h
I_3
VGG
L_f
L_a
L_s
$L_r + GO$
I_G
Blend
Background
I_G
$\nabla_{W^+}(L_f + L_r + L_n)$
$\nabla_{W^+}(L_f + L_a + L_s + L_{r+GO} + L_n)$
• Same holds for optimizing $\mathcal{N}$
• $M_t^f = M_1^f \cap (1 - M_2^{h,d})$



- **I$_1$** : Keep image except hair **I$_{2,}$** : hair shape and structures **I$_3$** : hair appearance and style
- $\boldsymbol{M_1^h}, \boldsymbol{M_2^h}, \boldsymbol{M_3^h}$: I$_1$, I$_{2,}$ I$_3$'s binary hair masks.
- $M_1^{h,\boldsymbol{d}}$ 중 d : 20% to produce the dilated version
- $M_1^{h,\boldsymbol{e}}$ 중 e : 20% to produce the eroded version
- $\boldsymbol{M_2^{h,ir}}$: $M_2^{h,d}$ - $M_2^{h,e}$ be the ignore region that requires inpainting. We do not optimize $M_2^{h,ir}$, 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.
- **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 I1, following which we perform the blending (Figure 2)

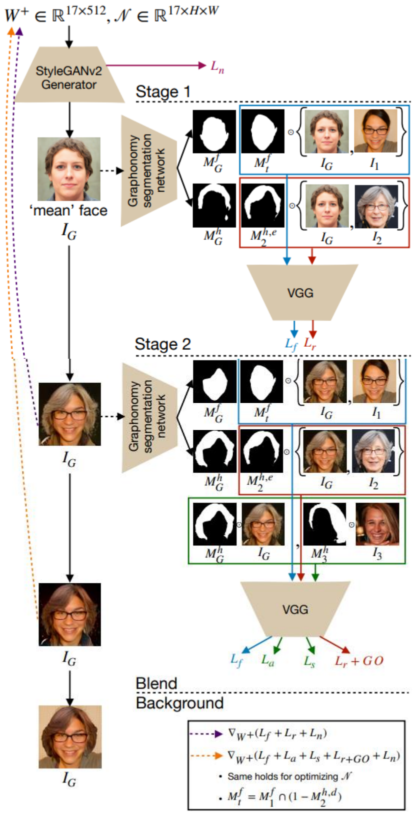
Objective


$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
'mean' face
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_2
VGG
L_f L_r
Stage 2
M_3^h
I_3
VGG
L_f L_a L_s L_r+GO
Blend
Background
$\nabla_{W+}(L_f+L_r+L_n)$
$\nabla_{W+}(L_f+L_a+L_s+L_{r+GO}+L_n)$
• Same holds for optimizing $\mathcal{N}$
• $M_t^f = M_1^f \cap (1-M_2^{h,d})$

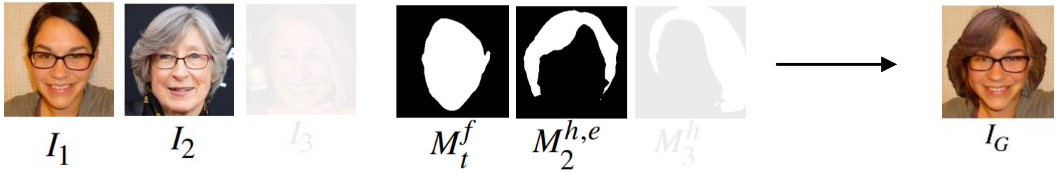


- $\mathbf{I_G}$: $\boldsymbol{G(W^+, N)}$: Synthesized image, $\boldsymbol{M_G^f, M_G^h}$ its corresponding face and hair regions.
- [1] Identity Reconstruction.
- [2] Hair Shape and Structure Reconstruction.
- [3] Hair Appearance Transfer.
- [4] Hair Style Transfer.
- [5] Noise Map Regularization.

Objective – [1] Identity Reconstruction



$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
'mean' face
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
VGG
L_f L_r
Stage 2
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
M_G^h
I_G
M_3^h
I_3
VGG
L_f L_a L_s L_r+GO
I_G
Blend
Background
I_G
$\nabla_{W+}(L_f+L_r+L_n)$
$\nabla_{W+}(L_f+L_a+L_s+L_{r+GO}+L_n)$
• Same holds for optimizing $\mathcal{N}$
• $M_t^f = M_1^f \cap (1-M_2^{h,d})$



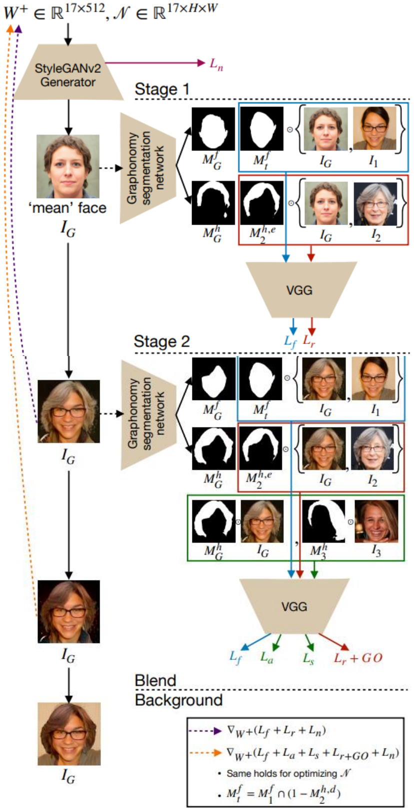
- **LPIPS** : Learned Perceptual Image Patch Similarity. LPIPS is a perceptual loss based on human similarity judgements and, therefore, is well suited for facial reconstruction.
- Pretrained VGG to extract high-level features for both $\mathbf{I_1}$ and $\mathbf{I_G}$

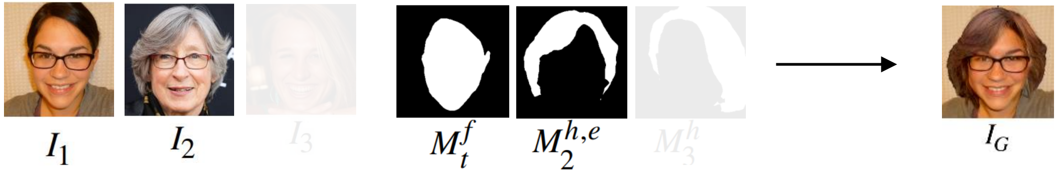
$$L_f = \frac{1}{5}\sum_{b=1}^{5} \text{LPIPS}\big[\text{VGG}^b(I_1 \odot (M_1^f \cap (1 - M_2^{h,d}))),$$

$$\text{VGG}^b(I_G \odot (M_1^f \cap (1 - M_2^{h,d})))\big] \quad (2)$$

- Where b denotes a VGG block, and $\boldsymbol{M_1^f} \cap \boldsymbol{M_2^{h,d}}$ represents the target mask
- This formulation places a soft constraint on the target mask.

Objective – [2] Hair Shape and Structure Reconstruction


$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
'mean' face
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
VGG
L_f L_r
Stage 2
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
M_G^h
I_G
M_3^h
I_3
VGG
L_f L_a L_s L_r+GO
I_G
Blend
Background
I_G
$\nabla_{W+}(L_f+L_r+L_n)$
$\nabla_{W+}(L_f+L_a+L_s+L_{r+GO}+L_n)$
• Same holds for optimizing $\mathcal{N}$
• $M_t^f = M_1^f \cap (1-M_2^{h,d})$

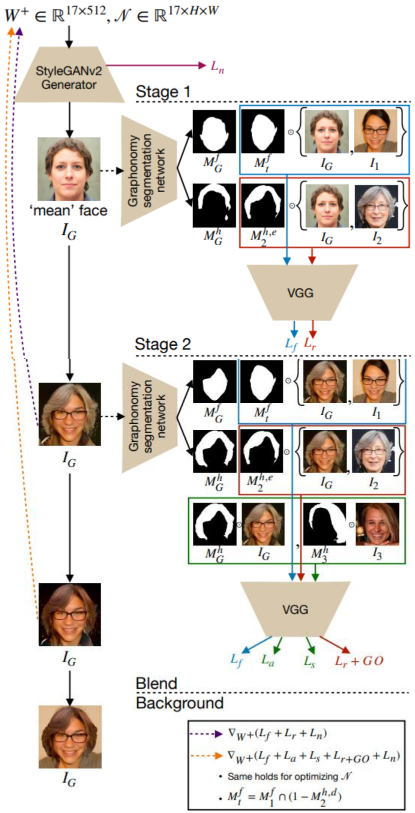


- **LPIPS loss** for hair shape and structure
- ***Naively use*** $\boldsymbol{M_2^h}$ as the target hair (GT) cause the generator to synthesize hair on undesirable regions of $\mathbf{I_G}$ of especially when the target face and hair regions do not align well.

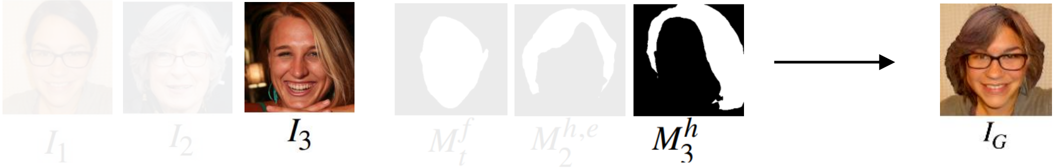
$$L_r = \frac{1}{2} \sum_{b\in\{4,5\}} \text{LPIPS}\big[\text{VGG}^b(I_2 \odot M_2^{h,e}), \quad \text{VGG}^b(I_G \odot M_2^{h,e})\big] \tag{3}$$

- Therefore, use $\boldsymbol{M_2^{h,e}}$, the eroded mask, as it places a soft constraint on the target placement of synthesized hair.

Objective – [3] Hair Appearance Transfer



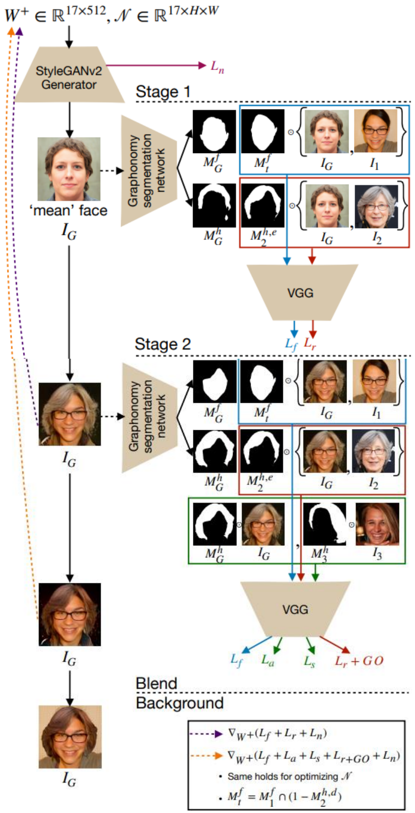
$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
'mean' face
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_2
VGG
L_f L_r
Stage 2
M_3^h
I_3
L_f L_a L_s $L_r + GO$
Blend
Background
$\nabla_{W^+}(L_f + L_r + L_n)$
$\nabla_{W^+}(L_f + L_a + L_s + L_{r+GO} + L_n)$
Same holds for optimizing $\mathcal{N}$
$M_t^f = M_1^f \cap (1 - M_2^{h,d})$



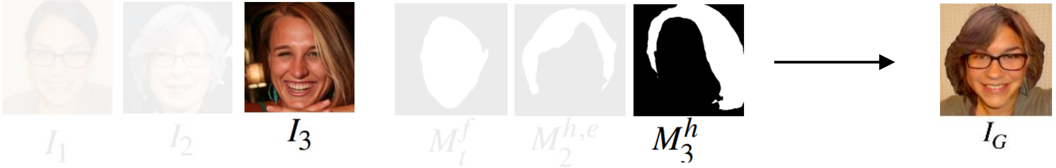
- **Hair appearance** refers to the globally consistent colour of hair that is independent of hair shape and structure.
- extract 64 feature maps from the shallowest layer of VGG (relu1 1) as it best accounts for colour information. Then, we perform average-pooling within the hair region of each feature map to discard spatial information and capture global appearance.
- the mean appearance A in R 64×1 as A(x, y) = $$\sum \frac{\phi(x)\odot y}{|y|},$$
- where φ(x) represents the 64 VGG feature maps of image x, and y indicates the relevant hair mask.
- Finally, we compute the squared L2 distance to give our hair appearance objective

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

Objective – [4] Hair Style Transfer



$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
'mean' face
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
VGG
L_f L_r
Stage 2
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
M_G^h
I_G
M_3^h
I_3
VGG
L_f L_a L_s $L_r + GO$
I_G
Blend Background
I_G
$\nabla_{W^+}(L_f + L_r + L_n)$
$\nabla_{W^+}(L_f + L_a + L_s + L_{r+GO} + L_n)$
• Same holds for optimizing $\mathcal{N}$
• $M_t^f = M_1^f \cap (1 - M_2^{h,d})$



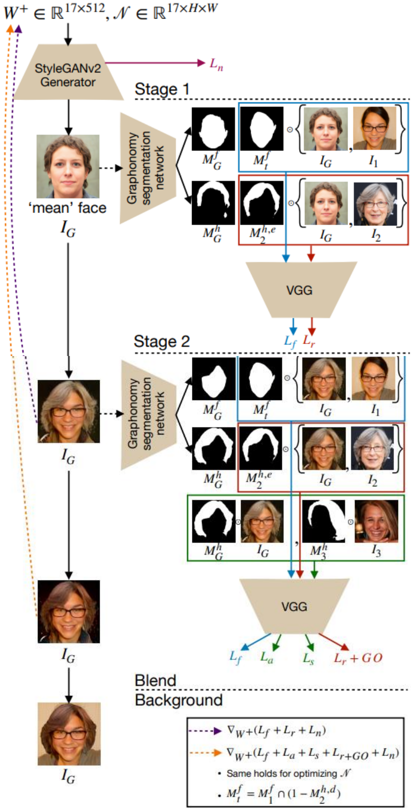
- In addition to the overall colour, hair also contains finer details such as **wisp** styles, and **shading** variations between hair strands. Such details cannot be captured solely by the appearance loss that estimates the overall mean.
- The Gram matrix captures finer hair details by calculating the second-order associations between high-level feature maps. Extracting features from layers: {relu1 2, relu2 2, relu3 3, relu4 3} of VGG where, γ l represents feature maps in R HW×C that are extracted from layer l, and G l is the Gram matrix at layer l.

$$\mathcal{G}^l(\gamma^l) = {\gamma^l}^{\mathsf{T}} \gamma^l$$

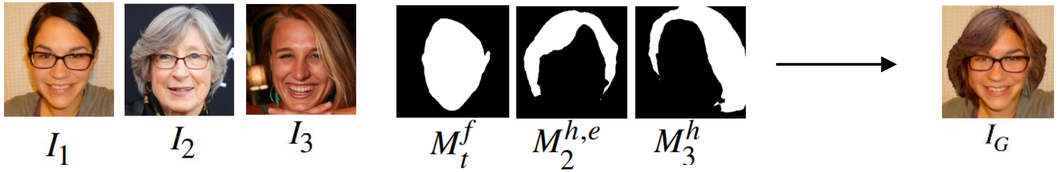
- Finally, we compute the squared L2 distance as

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

Objective – [5] Noise Map Regularization



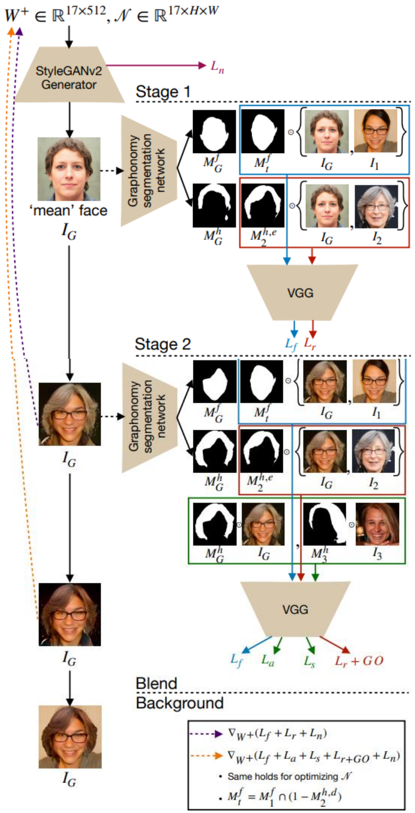
$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
'mean' face
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
VGG
L_f
L_r
Stage 2
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
M_G^h
I_G
M_3^h
I_3
VGG
L_f
L_a
L_s
$L_r + GO$
I_G
Blend
Background
I_G
$\nabla_{W^+}(L_f + L_r + L_n)$
$\nabla_{W^+}(L_f + L_a + L_s + L_{r+GO} + L_n)$
• Same holds for optimizing $\mathcal{N}$
• $M_t^f = M_1^f \cap (1 - M_2^{h,d})$



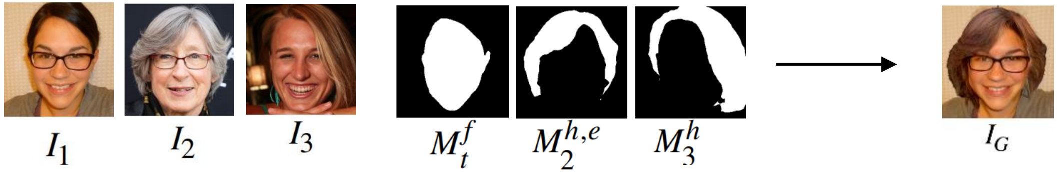
- Explicitly optimizing the noise maps n ∈ N can cause the optimization to inject actual signal into them. To prevent this, we introduce regularization terms of noise maps [20]. For each noise map greater than 8 × 8, we use a pyramid down network to reduce the resolution to 8×8. The pyramid network averages 2 × 2 pixel neighbourhoods at each step. Additionally, we normalize the noise maps to be zero mean and unit variance, where ni,0 represents the original noise map and ni,j>0 represents the downsampled versions. Similarly, ri,j represents the resolution of the original or downsampled noise map. producing our noise objective

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

Objective – ALL



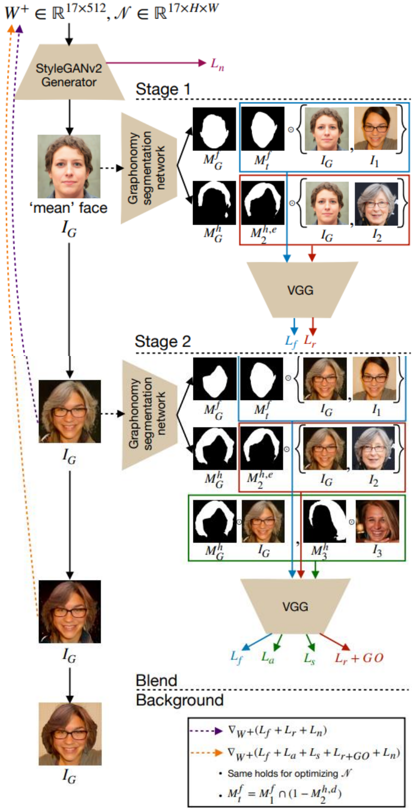
$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
'mean' face
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
VGG
L_f L_r
Stage 2
I_G
M_3^h
I_3
VGG
L_f L_a L_s $L_r + GO$
I_G
Blend Background
I_G
$\nabla_{W^+}(L_f + L_r + L_n)$
$\nabla_{W^+}(L_f + L_a + L_s + L_{r+GO} + L_n)$
• Same holds for optimizing $\mathcal{N}$
• $M_t^f = M_1^f \cap (1 - M_2^{h,d})$

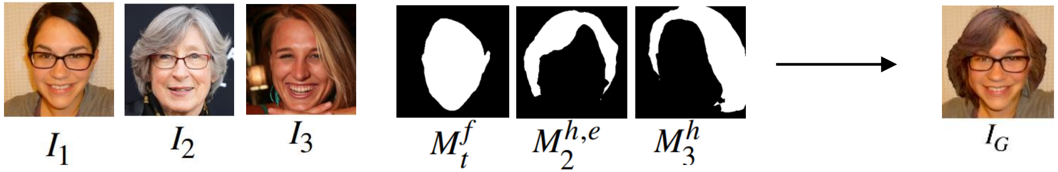


- $\mathbf{I_G}$: $G(W^+, N)$: Synthesized image, M_G^f, M_G^h its corresponding face and hair regions.
- [1] Identity Reconstruction. L_f
- [2] Hair Shape and Structure Reconstruction. L_r
- [3] Hair Appearance Transfer. L_a
- [4] Hair Style Transfer. L_s
- [5] Noise Map Regularization. L_n

$$L = \underset{\{\mathcal{W}^+,\mathcal{N}\}}{\arg\min}\big[\lambda_f L_f + \lambda_r L_r + \lambda_a L_a + \lambda_s L_s + \lambda_n L_n\big] \tag{8}$$

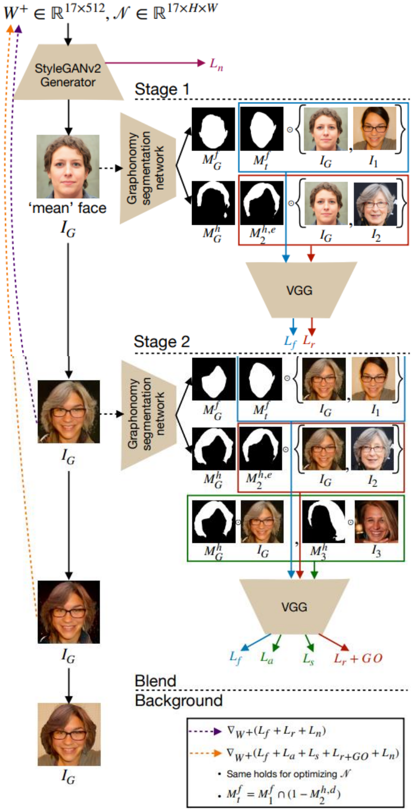
Optimization Strategy


$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
'mean' face
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
VGG
L_f L_r
Stage 2
I_G
Graphonomy segmentation network
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_G
I_2
M_G^h
I_G
M_3^h
I_3
VGG
L_f L_a L_s L_r+GO
I_G
Blend
Background
I_G
$\nabla_{W+}(L_f + L_r + L_n)$
$\nabla_{W+}(L_f + L_a + L_s + L_{r+GO} + L_n)$
• Same holds for optimizing $\mathcal{N}$
• $M_t^f = M_1^f \cap (1 - M_2^{h,d})$

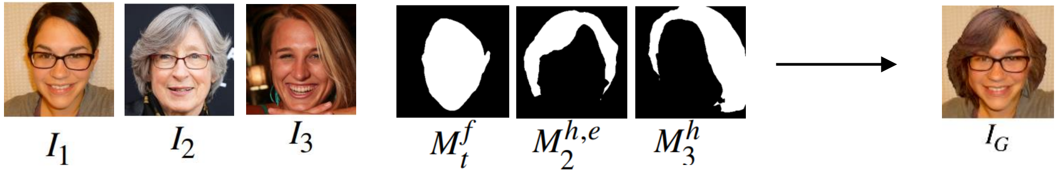


- **Two-stage optimization.**
- Optimize our overall objective in two stages. **In stage 1**, we reconstruct only the target identity and hair perceptual structure, i.e., we set λa and λs in Equation 8 to zero.
- **In stage 2**, optimize all the losses except $\boldsymbol{L_r}$; stage 1 will provide a better initialization for stage 2, thereby leading the model to convergence.
- 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 $\boldsymbol{L_r}$ in stage 2 of optimization**.**

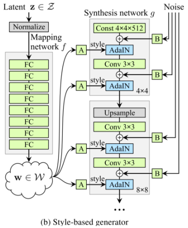
Optimization Strategy



$W^+ \in \mathbb{R}^{17\times512}, \mathcal{N} \in \mathbb{R}^{17\times H\times W}$
StyleGANv2 Generator
L_n
Stage 1
Graphonomy segmentation network
'mean' face
I_G
M_G^f
M_t^f
I_G
I_1
M_G^h
$M_2^{h,e}$
I_2
VGG
L_f L_r
Stage 2
M_3^h
I_3
L_f L_a L_s $L_r + GO$
Blend Background
$\nabla_{W+}(L_f + L_r + L_n)$
$\nabla_{W+}(L_f + L_a + L_s + L_{r+GO} + L_n)$
Same holds for optimizing $\mathcal{N}$
$M_t^f = M_1^f \cap (1 - M_2^{h,d})$



- **Gradient Orthogonalization.**
- L_r, by design, captures all hair attributes of person 2: perceptual structure, appear ance, 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.
- 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 W+ latent space, the gradients computed are



(b) Style-based generator

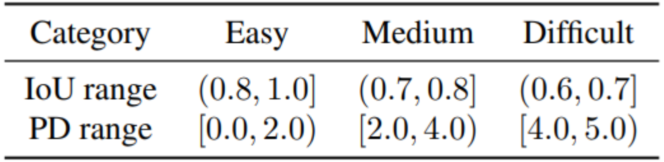
$$g_{R_2} = \nabla_{W+} L_r, g_{A_2} = \nabla_{W+} L_a, g_{S_2} = \nabla_{W+} L_s, \quad (9)$$

- where, L_r, L_a, and L_s s are the LPIPS, appearance, and style losses computed between I_2 and I_G. To enforce orthogonality, we would like to minimize gR_2^T (gA_2 + gS_2). We achieve this by projecting away the component of gR_2 parallel to (gA_2 + gS_2). , using the structure-appearance gradient orthogonalization after every iteration in stage 2 of optimization

$$g_{R_2} = g_{R_2} - \frac{{g_{R_2}}^{\mathsf{T}}(g_{A_2} + g_{S_2})}{\|g_{A_2} + g_{S_2}\|_2^2}(g_{A_2} + g_{S_2}) \qquad (10)$$

4. Experiments and Results

- **4.1. Datasets : FFHQ**
- Flickr-Faces-HQ dataset (FFHQ) 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.
- select tuples of images (I1, I2, I3) based on the following con straints: (a) each image in the tuple should have at least 18% of pixels contain hair, and (b) I1 and I2's face regions must align to a certain degree.
- To enforce these constraints we extract hair and face masks using the Graphonomy segmentation network and estimate 68 2D facial landmarks using 2D-FAN.
- For every I1 and I2, 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.**

4. Experiments and Results

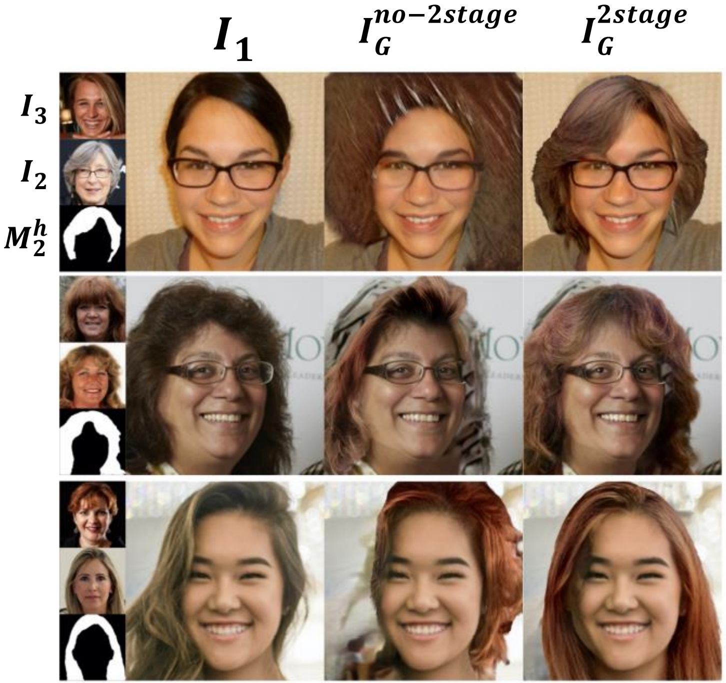


Figure 3: **Effect of two-stage optimization. Col 1 (narrow)**: Reference images. **Col 2**: Identity person. **Col 3**: Synthesized image when losses arre optimized jointly. **Col 4**: Image synthesized via two-stage optimization + gradient orthogonalization.

- **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. Experiments and Results

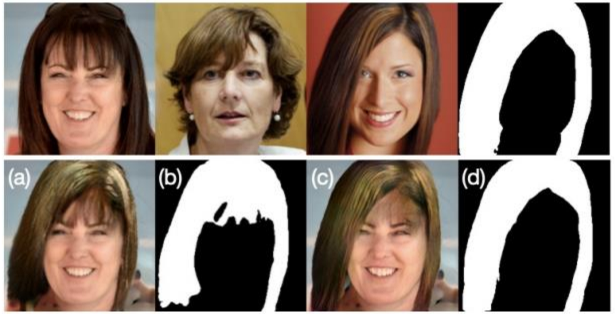


Figure 4: **Effect of Gradient Orthogonalization (GO). Row 1**: Reference images (from left to right): Identity, target hair appearance and style, target hair structure and shape. **Row 2**: Pairs (a) and (b), and (c) and (d) are synthesized images and their corresponding hair masks for no-GO and GO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

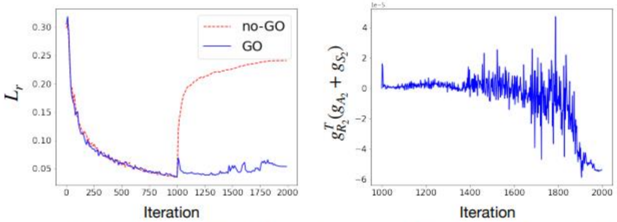


Figure 5: **Effect of Gradient Orthogonalization (GO). Left**: LPIPS hair reconstruction loss (GO vs no-GO) vs iterations. **Right**: Trend of ${g_{R_2}}^{\top}(g_{A_2}+g_{S_2})$ ($\times$1e-5) in stage 2 of optimization.

- **4.3. Effect of Gradient orthogonalization(GO)**
- We compare two variations of our framework: no-GO and GO. GO involves manipulating $\boldsymbol{L_r}$'s gradients via gradient orthogonalization, whereas no-GO keeps $\boldsymbol{L_r}$ untouched. No-GO is unable to retain the target hair shape, causing $\boldsymbol{L_r}$ 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 $\boldsymbol{M_2^h}$ and $\boldsymbol{M_G^h}$ increases from 0.857 (for no-GO) to 0.932 (GO).
- In terms of gradient disentanglement, the similarity beween $\boldsymbol{gR_2}$ and ($\boldsymbol{gA_2}$ + $\boldsymbol{gS_2}$) 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. Experiments and Results

- **4.4. Comparison with SOTA**

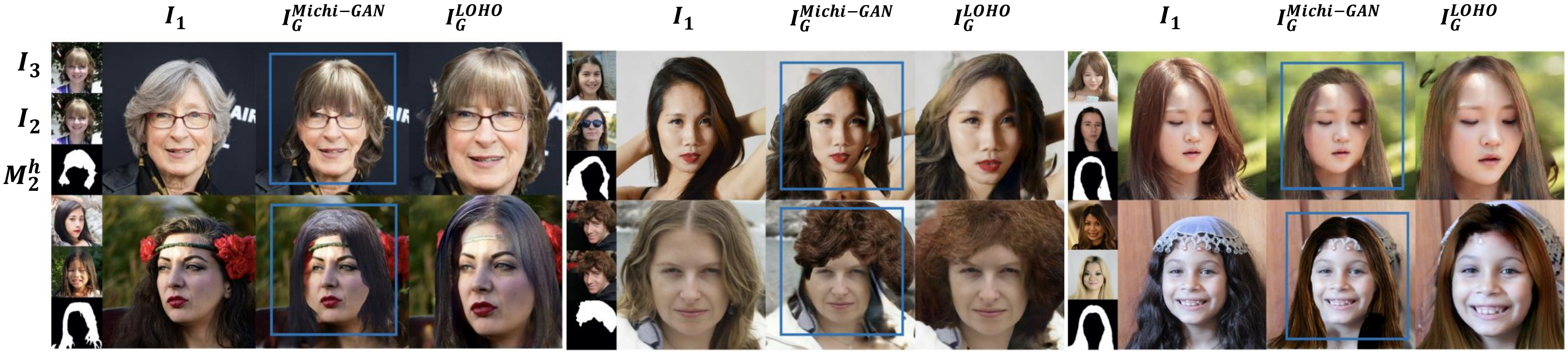
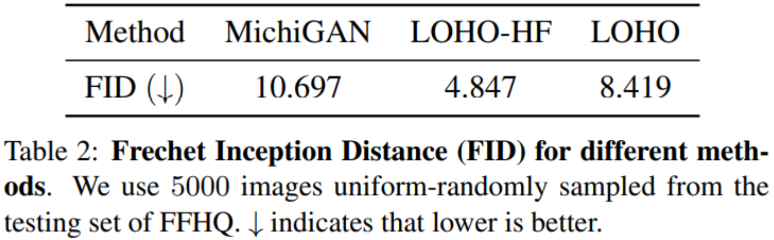
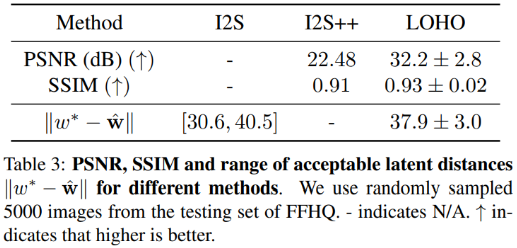


Figure 6: Qualitative comparison of MichiGAN and LOHO. Col 1 (narrow): Reference images. Col 2: Identity person Col 3: MichiGAN output. Col 4: LOHO output (zoomed in for better visual comparison). Rows 1-2: MichiGAN "copy-pastes" the target hair attributes while LOHO blends the attributes, thereby synthesizing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



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.

4. Experiments and Results

- **4.5. Editing Attributes**

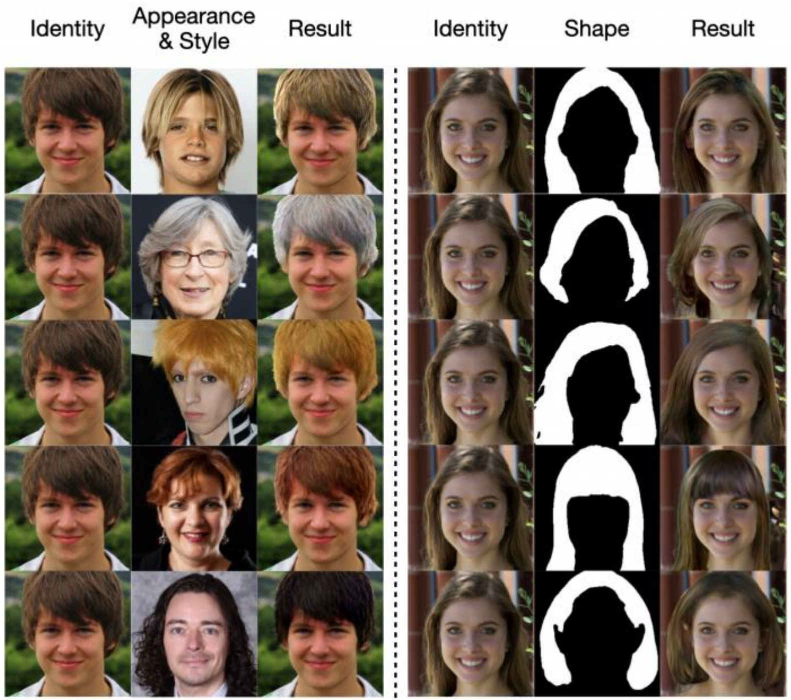


Figure 7: **Individual attribute editing**. The results show that our model is able to edit individual hair attributes (**left**: appearance & style left, **right**: shape) without them interfering with each other.

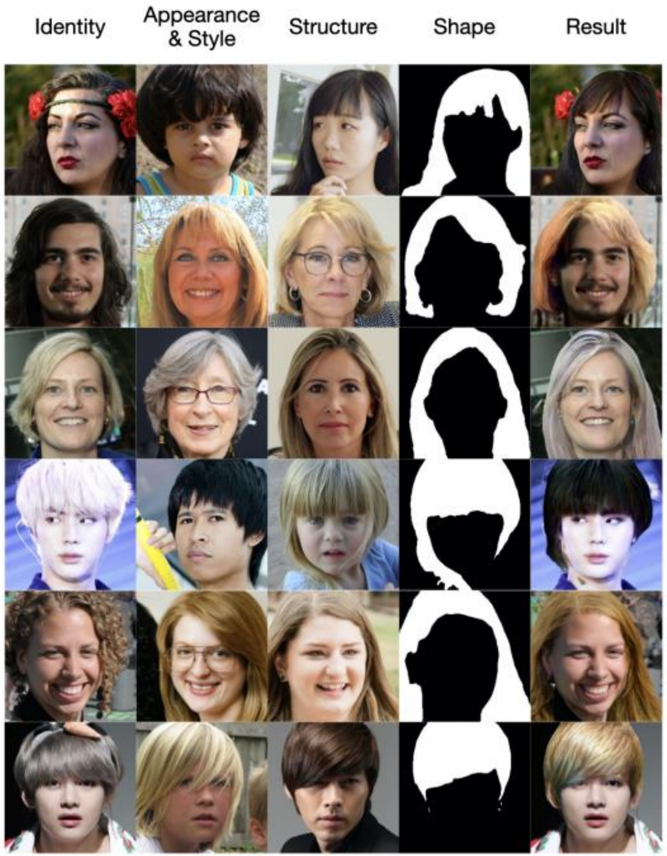


Figure 8: **Multiple attributes editing**. The results show that our model is able to edit hair attributes jointly without the interference of each other.

5. Limitations

- Our approach is susceptible to extreme cases of mis alignment (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 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's imperfect segmentation of hair. More sophisticated segmentation networks can be used to mitigate this issue.

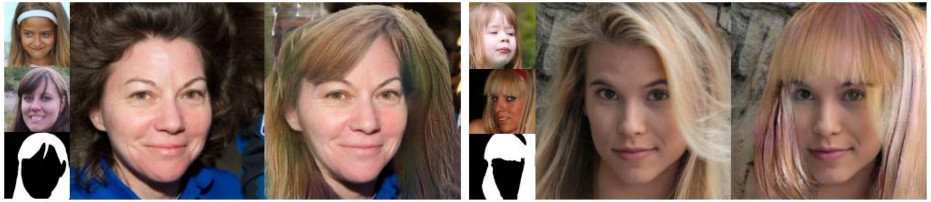


Figure 9: **Misalignment examples**. **Col 1 (narrow)**: Reference images. **Col 2**: Identity image. **Col 3**: Synthesized image. Extreme cases of misalignment can result in misplaced hair.

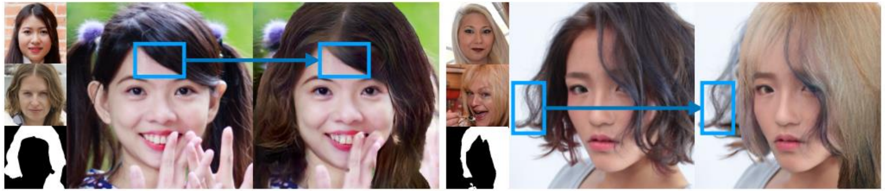
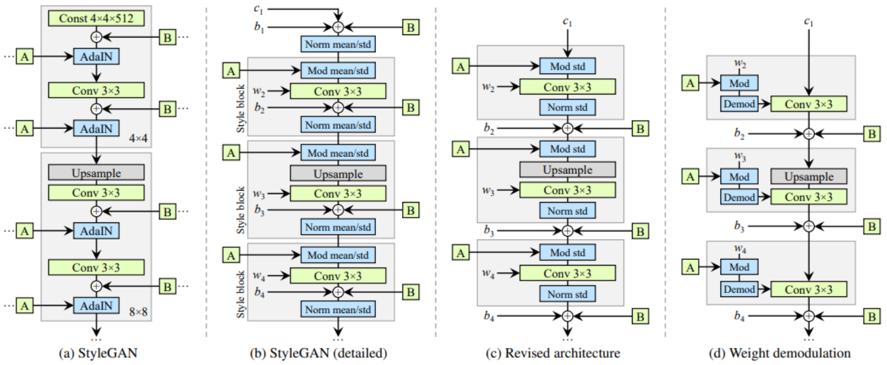


Figure 10: **Hair trail**. **Col 1 (narrow)**: Reference images. **Col 2**: Identity image. **Col 3**: Synthesized image. Cases where there are remnants of hair information from the identity person. The regions marked inside the blue box carries over to the synthesized image.

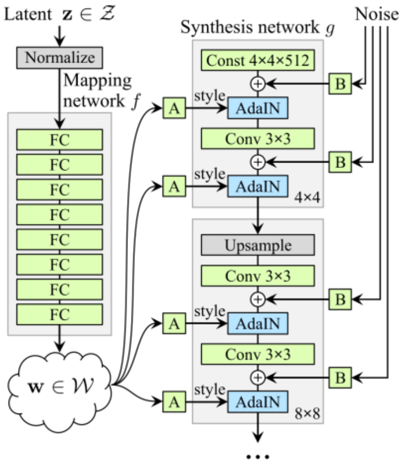
Reference

- https://arxiv.org/abs/2103.03891 (LOHO)
- https://arxiv.org/pdf/1912.04958.pdf (StyleGANv2)
- https://arxiv.org/pdf/1904.04536.pdf (Graphonomy network)

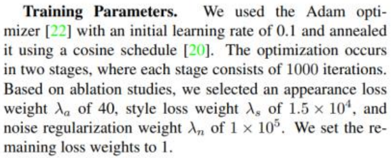


(a) StyleGAN (b) StyleGAN (detailed) (c) Revised architecture (d) Weight demodulation

Figure 2. We redesign the architecture of the StyleGAN synthesis network. (a) The original StyleGAN, where $\boxed{\text{A}}$ denotes a learned affine transform from $\mathcal{W}$ that produces a style and $\boxed{\text{B}}$ is a noise broadcast operation. (b) The same diagram with full detail. Here we have broken the AdaIN to explicit normalization followed by modulation, both operating on the mean and standard deviation per feature map. We have also annotated the learned weights (w), biases (b), and constant input (c), and redrawn the gray boxes so that one style is active per box. The activation function (leaky ReLU) is always applied right after adding the bias. (c) We make several changes to the original architecture that are justified in the main text. We remove some redundant operations at the beginning, move the addition of b and $\boxed{\text{B}}$ to be outside active area of a style, and adjust only the standard deviation per feature map. (d) The revised architecture enables us to replace instance normalization with a "demodulation" operation, which we apply to the weights associated with each convolution layer.



(b) Style-based generator



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.