Information Bottleneck Disentanglement for Identity Swapping

Gege Gao Huaibo Huang Chaoyou Fu Zhaoyang Li Ran He*
National Laboratory of Pattern Recognition, CASIA
Center for Excellence in Brain Science and Intelligence Technology, CAS
School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
{gege.gao,huaibo.huang}@cripac.ia.ac.cn {chaoyou.fu,rhe}@nlpr.ia.ac.cn zhaoyang0427@gmail.com

Appendix

1. Implementation Details

In order to improve the disentangled representation to generate identity-discriminative swapped faces, a novel learning-based information disentangling and swapping framework, InfoSwap, is proposed. In the main paper, we introduce the intuition and theoretical method of InfoSwap in detail. Three key points are discussed: the explicit supervision for disentanglement based on the information bottleneck principle, the improvement of information bottleneck objectives based on contrastive learning, and the metric for evaluating if the generated identities are discriminative based on the statistical features. In this section, we elaborate on the practical details of InfoSwap.

1.1. Method Details

Parametrization: To predict the information controller λ_i at i-th depth, as defined in Eq.(3), we collect m feature maps from different depths. As the spatial size of these features differs, we interpolate them bilinearly to match the size of the attached feature R_i. Then the information bottleneck IB_i uses the resized version of features to predict the controllers. Each bottleneck, denoted as $h_{IB_i}(*)$, consists of three $Conv + ReLu$ blocks. To simplify optimization, we parametrize the information controllers as $\lambda_i = \text{sigmoid}(r_i) \in [0, 1]$ where $r_i = h_{IB_i}(R_1, \cdots, R_m)$, to avoid any clipping during optimization.

Initialization: In the beginning, we retain all information in R_i by initializing $r_i = 5$ for all depths i and thus $\lambda_i \approx 0.993 \Rightarrow Z_i \approx R_i$, as formulated in Eq.(4). Then, λ_i deviates from this starting point to suppress unimportant areas by adding noise, as visualized in Fig. 1. As the face recognition model is trained already, adding noise should preserve the variance of the input to the following layers [11]. Therefore, we first use 10k images going through the pre-trained model to calculate the global mean and std. values of the feature R_i at each i-th depth. The noise is then sampled from the Gaussian distribution with the empirical mean and std. values.

Figure 1. Visualization of information variance in feature maps. The colormaps show the values of the information controller λ_i (in average). For representing the source identity, areas in deep blue have values $\lambda_i \approx 0$, *i.e.* most information is discarded for, and areas in hot red have $\lambda_i \approx 1$, *i.e.* most information is retained. While for representing the target perception, it is right the opposite as we use $1 - \lambda_i$ as formulated in Eq.(13).

Experimental Setup: During optimization, we set the objective of the information bottleneck (Eq.(2)) as $L_{IB} = \alpha L_{info} + \beta L_{task}$, with an additional parameter for easily control of the trade-off. Experimental results show that the uniformly uninformative features are obtained when $\alpha : \beta \geq 5$, namely all information gets discarded. When $\alpha : \beta \in (5, 1]$, more information passed and less noise is added. Finally the IIB is trained with the best performing value $\alpha : \beta = 1 : 5$. The hyper parameters for other loss terms in Eq.(21) are set to $\beta_1 = 1, \beta_2 = 5, \beta_3 = 1$. More detailed settings are shown in Tab. 1

Training Strategies: The training is performed on the training set of size 96000 collected from the FFHQ [5] and the CelebA-HQ [4] dataset. During training, we use Adam [6] optimizer for all modules with coefficients $(0.001, 0.999)$, weight decay 0.00002 and learning rate 0.00005. The entire model InfoSwap is trained end-to-end, the full training algorithm is summarized in Alg. 1.

*Corresponding Author

Parameter	Best Value	Search space
Pre-trained Backbone	ResNet-50	
Amount of internal features in use (m)	10	$\{8, 9, 10\}$
Optimizer	Adam ($b_1 = 0.001, b_2 = 0.999$)	
Learning Rate (lr_g, lr_d)	0.00005	$\{0.00002, 0.00005, 0.0001\}$
Epochs	15	
Batch Size (B)	4	$\{1, 2, 4\}$
Trade-off Factor ($\alpha : \beta$)	$1 : 5$	$\{10 : 1, 5 : 1, 1 : 1, 1 : 5, 1 : 10\}$

Table 1. Hyperparameters for InfoSwap. The pre-trained backbone is taken from the pre-trained face recognition model [2].

1.2. Network Architectures

The network architectures of IIB (Informative identity Bottleneck) and AII (Adaptive Information Integration) for generating 512×512 images are shown in Fig 3. In each information bottleneck IB_i, we use the Gaussian Smooth with kernel size 1 and standard variance 0.25 to enforce the local smoothness in each information controller λ_i, as the pooling operations (in ResNet-50) and convolutional layers with stride greater than 1 are ignoring parts of the input. During training, we replace the first pre-trained feature with another of the same size $(32 \times 256 \times 256)$ for better reconstructing the background of the target images, which is obtained by inputting the target images into one convolutional layer $(Conv4 \times 4, 2, 1)$. The network architectures for generating 1024×1024 images are similar to this.

2. Experimental Results

In this section, we show the qualitative results of the ablation study and more swapped examples on the test set of the FFHQ [5] and the CelebA-HQ [4] datasets.

2.1. Ablation Studies

The qualitative results of the ablation study are shown in Fig. 2. It can be observed that in the first configuration (i) removing the IIB module (InfoSwap w/o IIB), the identities of the swapped results appear to be affected by the target identity and look less like the source. The second configuration (ii) replacing the ICL with conventional identity loss (InfoSwap w/o ICL) suffer the same problem. And in the third configuration (iii) discarding the information controller $\tilde{\lambda}_i^t$ (InfoSwap w/o $\tilde{\lambda}_i^t$) causes problems of information integration such as the inconsistent position of the eyes. While the full model has much better performance without such problems, indicating that each of the three designs is necessary for generating identity-discriminative results with high-fidelity.

2.2. More Quantitative Results

As shown in Fig 4, we provide more results of InfoSwap. The swapping is performed between images that have large gaps in genders (Fig 4(a)), ages (Fig 4(b)), skin colors (Fig 4(c)) and lighting conditions (Fig 4(d)), and in all situations our method has good performances. Besides, we show more high-fidelity swapping examples of the same gender and celebrities in Fig. 6 and Fig. 7 respectively. As shown in Fig. 5, our method performs well at higher resolution of 1024×1024. Moreover, in movie scenes with more complex conditions, our method can also produce good results as shown in Fig. 8. These demonstrate that our method is very robust and able to work well even under very difficult situations, which is mainly due to our efficient disentanglement method.

3. Forgery Detection

We use the model from FaceForensics++ [10] to examine the performance of the SOTA forgery detector on faces swapped by InfoSwap. For each of the 1000 videos in FF++, we evenly select 10 results from all the frames manipulated by our method, along with 10 corresponding target frames, making up a test set of 20k images. The experimental results are reported in Tab. 2. It can be observed that current detection algorithms have limited performance on our method (almost equals random classification). This indicates that our swapped results can be used as useful training data for better development of the data-driven forgery detectors.

To this end, we further train the forgery detection model from FF++ using 100k fake images (100 frames per video) generated by our method. We test the performance of the original detector and further train one on the manipulated dataset provided by [8]. The results are shown in Tab. 3. We will provide our manipulated videos on FF++ as soon as possible.

FF++ metrics / method	AUC	AP	F_1 measure	MSE
InfoSwap	0.5002	0.5001	0.0031	0.4993
FaceShifter [8]	<u>0.5222</u>	<u>0.5228</u>	0.0047	<u>0.4805</u>
FSGAN [9]	0.5321	0.5288	0.0049	0.4794
FaceSwap [7]	0.9930	0.9928	0.9929	0.0070
Deepfakes [1]	0.9941	0.9939	0.9940	0.0059

Table 2. Face Forgery Detection results of FF++ on fake faces produced by, reporting in AUC (Area under the ROC Curve), AP (Average Precision), F_1 measure, and MSE (Mean Square Error). Values underlined are from [8], others are computed following the same protocol.

Figure 2. Qualitative results of the ablation experiments.

FaceShifter / metrics / detector	AUC	AP	F_1	MSE
FF++ [10]	0.5222	0.5228	0.0047	0.4805
FF++ (further trained)	0.5609	0.5585	0.0041	0.3959

Table 3. Face Forgery Detection results on manipulated videos [8] with detectors the original FF++ model and the further trained FF++ using our swapped results. Values underlined are from [8], others are computed following the same protocol.

4. Broader Impact

As mentioned in the main paper, Deepfakes may cause threats to the public. Therefore, we further discuss ways to prevent it from being misused according to our work.

Stay on top of the latest Deepfake technologies. To build good detection algorithms, it is crucial to first know the features of various manipulation algorithms. And developing uniform standards to record recent progress can facilitate the track of the latest technologies. Recent work [10] collates and summarises a wide range of the latest Deepfake developments. Such work is a great help for researchers to understand current Deepfake technologies and develop better strategies for forgery detection.

Collect fake data to develop better forgery detectors. One efficient way to prevent the misuse of Deepfakes is using forgery detectors to automatically identify fake data. As current forgery detection algorithms are mainly data-driven, it is shown in recent studies that using various Deepfake data from a different manipulated method as training data can improve the global performance and robustness of detection algorithms. Therefore, it is highly suggested that researchers expand their training set with the latest Deepfake data and increase the data variance.

Our contributions. Apart from proposing the efficient information disentangling and swapping method to improve the performance of face swapping, we are also striving to mitigate the potential harms of Deepfakes: (i) We are building a new Deepfake dataset synthesized by our method to help improve the data-driven forgery detectors. (ii) We commit to supporting the development of Deepfake detection algorithms in all possible ways, including but not limited to summarizing the latest Deepfake methods and developing novel forgery detectors.

References

[1] *Deepfakes*, Accessed: 2020-09-21. https://github.com/deepfakes/faceswap.

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[3] Kai-Ming He, Xiang-Yu Zhang, Shao-Qing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[4] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[6] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1703.00810*, 2017.

[7] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *ICCV*, 2017.

[8] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *CVPR*, 2020.

[9] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *ICCV*, 2019.

[10] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[11] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. In *ICLR*, 2020.

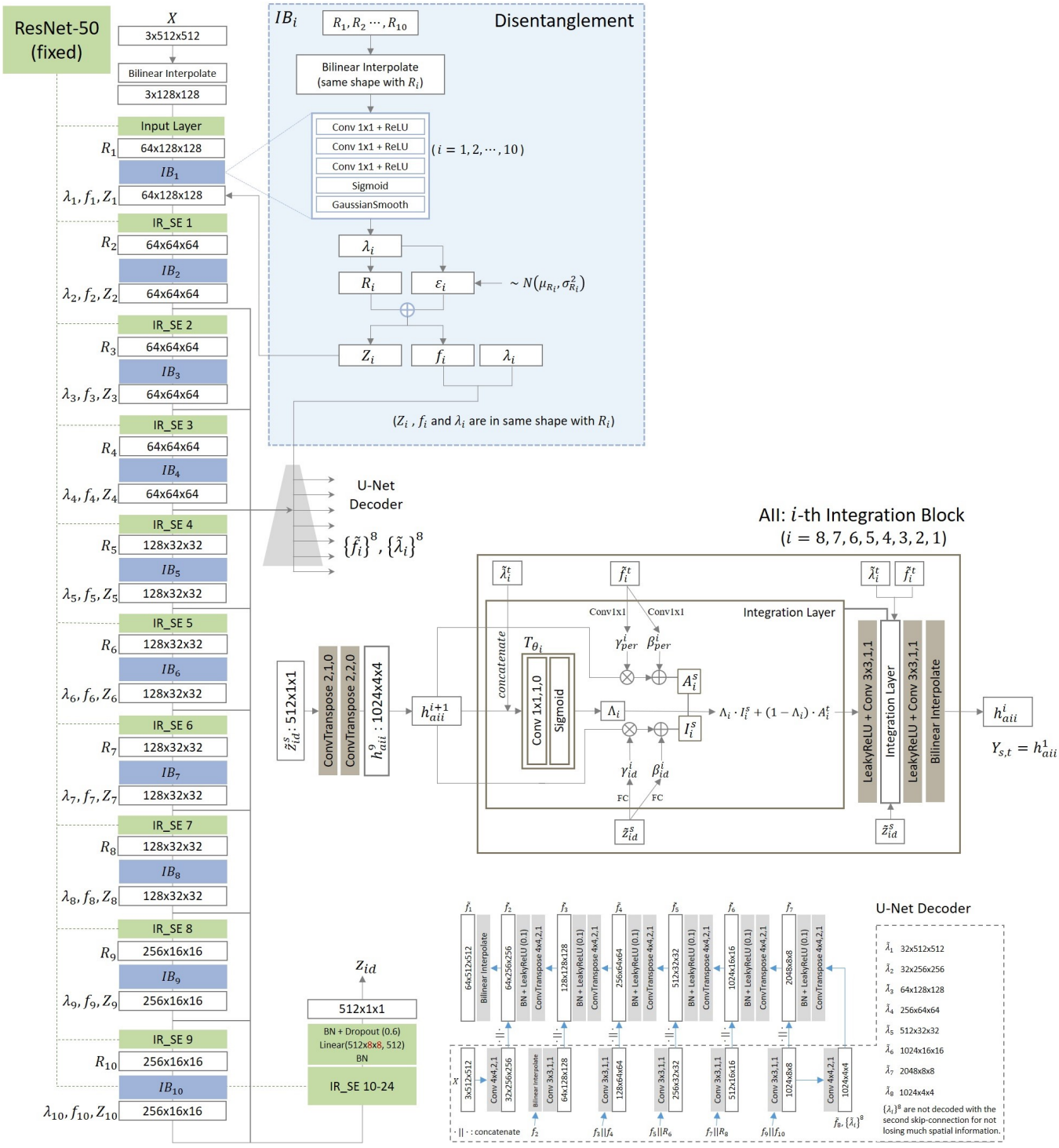


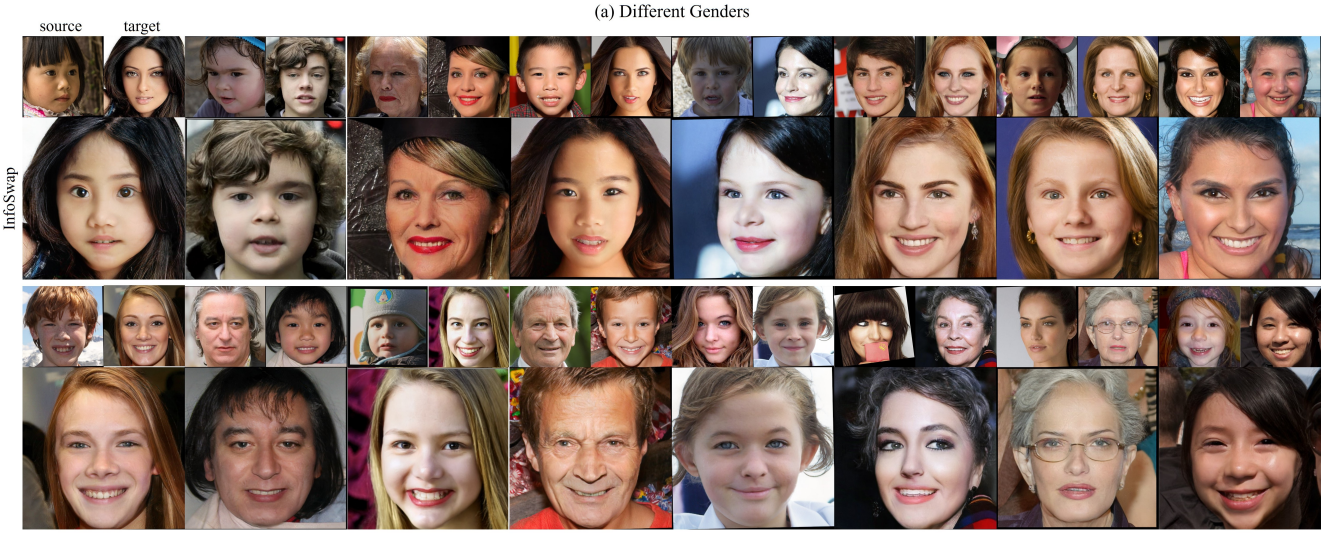
Figure 3. The network architecture of InfoSwap.

Algorithm 1 Training algorithm for InfoSwap.

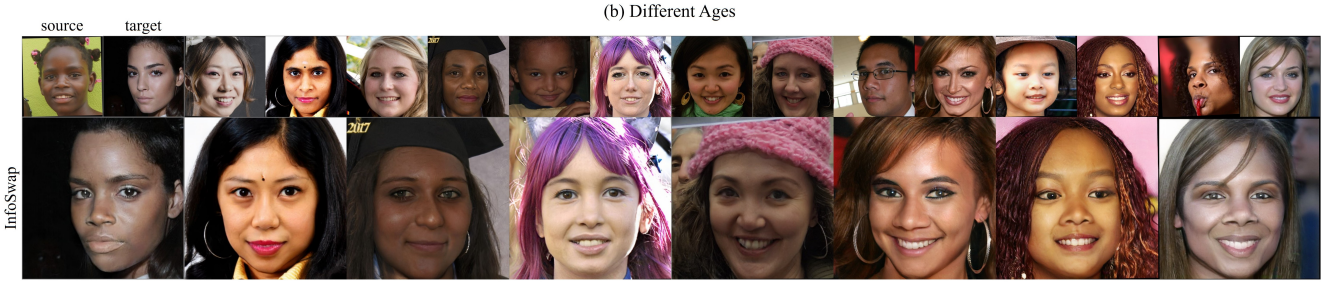
Input: $\{X_s\}^N$, $\{X_t\}^N$, N is the size of training set.
Require: Initialize IIB (with $h_{IB_i}(\cdot)$, $i = 1, \cdots, 10$), AII (with $T_{\theta_i}(\cdot)$, $i = 1, \cdots, 8$), Decoder (denoted as $dec(\cdot)$) and Discriminator (denoted as $D(\cdot)$) with values sampled from normal distribution [3];
Require: Load pre-trained face recognition model (denoted as $f(\cdot)$);
Require: Pre-calculated mean μ_{R_i} and std. σ_{R_i} values of R_i, $i = 1, \cdots, 10$.

while not converged **do**
  sample mini-batch $\{x_s\}^B$, $\{x_t\}^B$
  # I. Feed Forward:
  $\{x\}^B \leftarrow \{x_s\}^B || \{x_t\}^B$ ▷ $\cdot||\cdot$ denotes concatenate operation.
  $z_{id}, \{R_1, \cdots, R_m\} \leftarrow f(\{x\}^B)$ ▷ # (i) get original identity embedding and m internal features in one go.
  $z^s_{id} = z_{id}[:B]$, $z^t_{id} = z_{id}[B:]$
  $R^s_i = R_i$, $R^t_i = R_i[B:]$ ▷ calculate $2 \times B$ for source and target together.
  $L_{info} \leftarrow 0, L_{recog} \leftarrow 0$
  $\tilde{z}_{id} \leftarrow 0$
  for $i = 1, 2, \cdots, 10$ **do**
    $\lambda_i \leftarrow h_{IB_i}(R_1, \cdots, R_{10})$
    $\varepsilon_i \leftarrow \mathcal{N}(\mu_{R_i}, \sigma^2_{R_i})$
    $Z_i \leftarrow \lambda_i R_i + (1 - \lambda_i)\varepsilon_i$ ▷ # (ii) compress features.
    $\tilde{z}_{id} \leftarrow \tilde{z}_{id} + f_{IB_i}(\{x\}^B)$ ▷ identity compressed by IB_i alone: replace R_i by Z_i.
    $L_{info} \leftarrow L_{info} + I[Z_i, R_i]$ ▷ # calculate information bottleneck losses.
  end for
  $L_{info} \leftarrow L_{info}/10$
  $\tilde{z}_{id} \leftarrow \tilde{z}_{id}/10$
  $L_{recog} \leftarrow 1 - \cos\langle \tilde{z}_{id}, z_{id} \rangle$
  $\tilde{z}^s_{id} \leftarrow \tilde{z}_{id}[:B]$, $\tilde{z}^t_{id} \leftarrow \tilde{z}_{id}[B:]$
  $\lambda^s_i \leftarrow \lambda_i[:B]$, $\lambda^t_i \leftarrow \lambda_i[B:]$
  $f^t_i \leftarrow \lambda^t_i \varepsilon^s_i + (1 - \lambda^t_i) R^t_i$, $\varepsilon^s_i \sim \mathcal{N}(\mu_{R^s_i}, \sigma^2_{R^s_i})$ ▷ # (iii) get disentangled perceptual features
  $\tilde{f}^t_i \leftarrow dec(f^t_i)$, $\tilde{\lambda}^t_i \leftarrow dec(\lambda^t_i)$, $i = 1, 2, \cdots, 8$
  Output: $Y_{s,t} \leftarrow \text{AII}(\tilde{z}^s_{id}, \{\tilde{f}^t_i\}^8, \{\tilde{\lambda}^t_i\}^8)$ ▷ # (iv) generate swapped face
  $\tilde{z}_{id}(\{Y_{s,t}\}^B) \leftarrow \sum_{i=1}^{10} f_{IB_i}(\{Y_{s,t}\}^B)/10$
  $L_{pos} \leftarrow -\cos\langle \tilde{z}_{id}(\{Y_{s,t}\}^B), \tilde{z}^s_{id} \rangle$
  $L_{neg} \leftarrow [\cos\langle \tilde{z}_{id}(\{Y_{s,t}\}^B), \tilde{z}^s_{id} \rangle - \cos\langle \tilde{z}^s_{id}, \tilde{z}^t_{id} \rangle]^2$
  $L_{icl} \leftarrow L_{pos} + L_{neg}$ ▷ # (v) calculate ICL
  $L_{IB} \leftarrow \alpha L_{info} + \beta(L_{recog} + L_{icl})$
  $L_{per} \leftarrow 0$
  $\{R^{Y_{s,t}}_1, \cdots, R^{Y_{s,t}}_m\} \leftarrow f(\{Y_{s,t}\}^B)$
  for $i = 1, 2, \cdots, 10$ **do**
    $f_i(Y_{s,t}) \leftarrow \lambda^t_i \varepsilon^s_i + (1 - \lambda^t_i) R^{Y_{s,t}}_i$
  end for
  $\{\tilde{f}_i(Y_{s,t})\}^8 \leftarrow dec(\{f_i(Y_{s,t})\}^{10}, \{\lambda^t_i\}^{10})$
  $L_{per} \leftarrow \sum_{i=1}^{8} [\tilde{f}_i(Y_{s,t}) - \tilde{f}^t_i]^2/8$ ▷ # (vi) calculate perceptual loss
  With no grad: $\{\tilde{f}^s_i\}^8 \leftarrow dec(\{\lambda^s_i \varepsilon^t_i + (1 - \lambda^s_i) R^s_i\}^{10})$, $\varepsilon^t_i \sim \mathcal{N}(\mu_{R^t_i}, \sigma^2_{R^t_i})$, $\{\tilde{\lambda}^s_i\}^8 \leftarrow dec(\{\lambda^s_i\}^{10})$
  RequiresGrad(θ) $\leftarrow$ False for θ in AII.parameters()
  $\hat{x}_s \leftarrow \text{AII}(\tilde{z}^{Y_{s,t}}_{id}, \{\tilde{f}^s_i\}^8, \{\tilde{\lambda}^s_i\}^8)$
  RequiresGrad(θ) $\leftarrow$ True for θ in AII.parameters() ▷ to make all BP gradient calculated only after passing $Y_{s,t}$
  $L_{cyc} \leftarrow \|x_s - \hat{x}_s\|_1$ ▷ # (vii) calculate cycle-consistency loss
  $L^G_{adv} \leftarrow -\mathbb{E}_{(X_s, Y_{s,t})}[\log(\text{sigmoid}(D(Y_{s,t}) - D(X_s)))]$ ▷ # (viii) calculate adversarial loss
  # II. Update parameters of IIB, AII and Decoder (Θ_g) with learning rate lr_g:
  $L^G_{obj.} \leftarrow L_{IB} + \beta_1 L_{adv} + \beta_2 L_{per} + \beta_3 L_{cyc}$ ▷ Total Loss for generation
  $\nabla\Theta_g \leftarrow \nabla L^G_{obj.}$
  $\Theta_g \leftarrow \Theta_g + lr_g \cdot \text{Adam}(\nabla\Theta_g, 0.001, 0.999)$
  # III. Update parameters of Discriminator (Θ_d) with learning rate lr_d:
  $L^D_{adv} \leftarrow -\mathbb{E}_{(X_s, Y_{s,t})}[\log(\text{sigmoid}(D(X_s) - D(Y_{s,t})))]$
  $\nabla\Theta_d \leftarrow \nabla L^D_{adv}$
  $\Theta_d \leftarrow \Theta_d + lr_d \cdot \text{Adam}(\nabla\Theta_d, 0.001, 0.999)$
end while

(a) Different Genders



(b) Different Ages



(c) Different Skin Color

(d) Different Lighting Conditions

Figure 4. More qualitative results on the test sets of FFHQ and CelebA-HQ datasets. Swapped across large gaps between different: (a) genders, (b) ages, (c) skin colors, (d) lighting conditions.

Figure 5. Swapped results in 1024×1024 resolution.

Figure 6. Swapped results of the same gender.

Figure 7. Swapped results of celebrities.

Figure 8. Swapped results of stills from movie scenes.