FastSwap: A Lightweight One-Stage Framework for Real-Time Face Swapping

Sahng-Min Yoo[1,2], Tae-Min Choi[2], Jae-Woo Choi[2], Jong-Hwan Kim[2]
[1]KLleon AI Research
[2]RIT Lab., KAIST
sahngmin.yoo@klleon.io, {smyoo, tmchoi, jwchoi, johkim}@rit.kaist.ac.kr

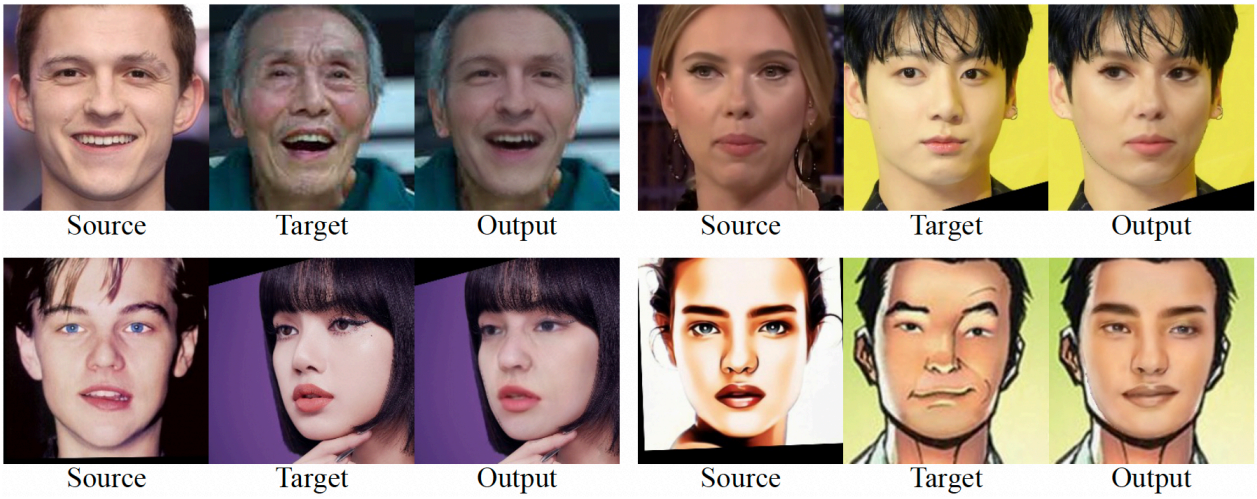

Figure 1: Face swapping results of FastSwap. The face in the target image is replaced with the face in the source image while preserving the pose and attributes. The source code is available in `https://github.com/sahngmin/fastswap`.

Abstract

Recent face swapping frameworks have achieved high-fidelity results. However, the previous works suffer from high computation costs due to the deep structure and the use of off-the-shelf networks. To overcome such problems and achieve real-time face swapping, we propose a lightweight one-stage framework, FastSwap. We design a shallow network trained in a self-supervised manner without any manual annotations. The core of our framework is a novel decoder block, called Triple Adaptive Normalization (TAN) block, which effectively integrates the identity and pose information. Besides, we propose a novel data augmentation and switch-test strategy to extract the attributes from the target image, which further enables controllable attribute editing. Extensive experiments on VoxCeleb2 and wild faces demonstrate that our framework generates high-fidelity face swapping results in 123.22 FPS and better preserves the identity, pose, and attributes than other state-of-the-art methods. Furthermore, we conduct an in-depth study to demonstrate the effectiveness of our proposal.

1. Introduction

In this work, we consider a face swapping task, which can replace the identity of a person in the image with another person while preserving the pose and attributes, e.g., skin tone, make-up, and lighting condition (see Figure 1). Given a *source image* and a *target image*, face swapping framework aims to generate a facial image with an identity of *source image*, and a pose and attributes of *target image*. We mainly focus on overcoming the computational limitation of the previous face swapping frameworks. Furthermore, we introduce a novel attribute editing manipulation within a lightweight one-stage framework.

The previously popular DeepFakes [6] is a subject-aware face swapping framework which has to be trained for each new input source and target pair. At least 500 face images of a source and target individual and 12 hours on GPU resource are required to train each network [6]. Despite ex-

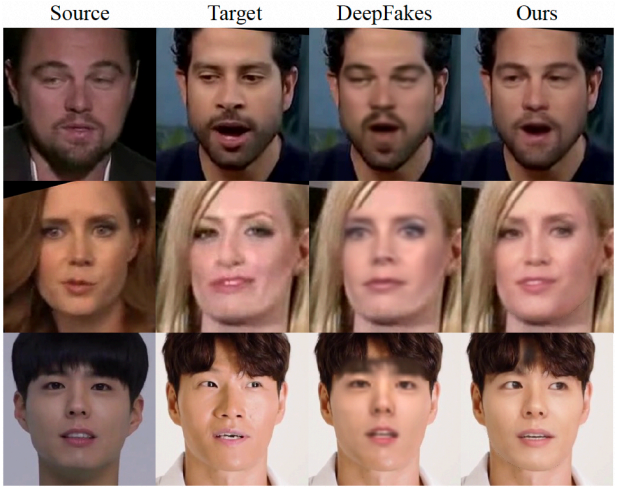

Figure 2: Failure cases of previous face swapping method, DeepFakes.

pensive data collection and time-consuming training procedures, the swapped faces are not perceptually appealing. DeepFakes fails to keep the identity of the source face and to mimic the pose and attributes of the target face (see Figure 2). However, recent studies have succeeded in generating realistic face swapping images of unseen individuals by designing deep generative networks and training the network with huge face datasets. Although the additional training is unnecessary, a high-end desktop GPU is still required to run the deep generative face swapping frameworks in real-time. To overcome the computational limitation, we design a lightweight face swapping framework which can be practically used in various application scenarios such as telepresence, gaming, AR/VR, etc.

Face swapping frameworks based on deep generative networks attempt to handle an arbitrary source and target pairs without additional training. [24, 27, 25, 21] used an additional input such as landmark, action units (AU), and 3D Morphable Model (3DMM) [1] coefficients to provide distinct pose and attribute information by adopting off-the-shelf networks. However, such approaches are highly dependent on the pre-trained network, and additional computations are required to use the frameworks. Alternatively, [19, 20] presented the methods of decoupling an appearance and keypoint-based motion information. Since these methods used a relative face reenactment by tracing the most similar frame in the input video, they have difficulties in real-time conversion and suffer from video dependency since the video is used as a target input. To better preserve the source identity, [26, 3, 22, 9] used multiple source images and utilized an average feature as an identity representation. Consequently, it takes effort to collect various new images when creating a neural head of unseen source identity. In order to achieve real-time face swapping without additional networks or processes, we propose a lightweight one-stage framework with a novel decoder structure, data augmentation, and a switch-test strategy.

We introduce a novel face swapping framework, FastSwap, which addresses the computational limitation and generates a photorealistic face image in a self-supervised learning scheme. While designing a shallow and lightweight network to fulfill the real-time face swapping, we utilize adaptive normalization to overcome the low-fidelity problem appearing as a trade-off of the network reduction. The proposed Triple Adaptive Normalization (TAN) block integrates the identity and pose by applying three different adaptive normalizations in each dimensional space. Furthermore, we introduce a novel data augmentation and switch-test strategy noting the task gap between training and test steps and handling the inputs of pose and attributes independently. In the face swapping task, the output aims to follow the attributes of the target image when testing. The target image is recommended as an attribute provider, though it is a contradiction where the input becomes a ground truth. Hence, we use the source image as an attribute provider by matching the color augmentation with the ground truth image during training. Then, when testing, we switch the provider to the target image to generate a suitable output. Consequently, our strategy enables attribute manipulation while preserving identity and pose through independent attributes input, unlike other face swapping frameworks [22, 27, 25, 9, 24, 26, 3, 21].

In summary, the main contributions of our work are as follows: **(1)** We propose a lightweight one-stage face swapping framework. Our framework swaps the face in 123.22 FPS and shows high-fidelity face swapping results with quantitative and qualitative evaluations. **(2)** We design a TAN block to achieve an effective disentanglement and integration of identity and pose in an adaptive fashion. **(3)** We introduce a novel data augmentation and switch-test strategy which deals with an attribute input in a self-supervised manner. Our strategy enables controllable attribute editing with a one-stage framework. **(4)** We analyze the effect of each component of our proposed framework by conducting an ablation study.

2. Related Work

2.1. Neural Talking Head

Neural talking head synthesis frameworks focus on the source face to imitate the pose of the target face while maintaining the attributes of the source image. LPD [3] concatenated both identity and pose features to generate adaptive parameters and used AdaIN [11] in each decoding layer. On the other hand, [23] presented an appearance adaptive normalization mechanism inspired by SPADE [17] to optimize

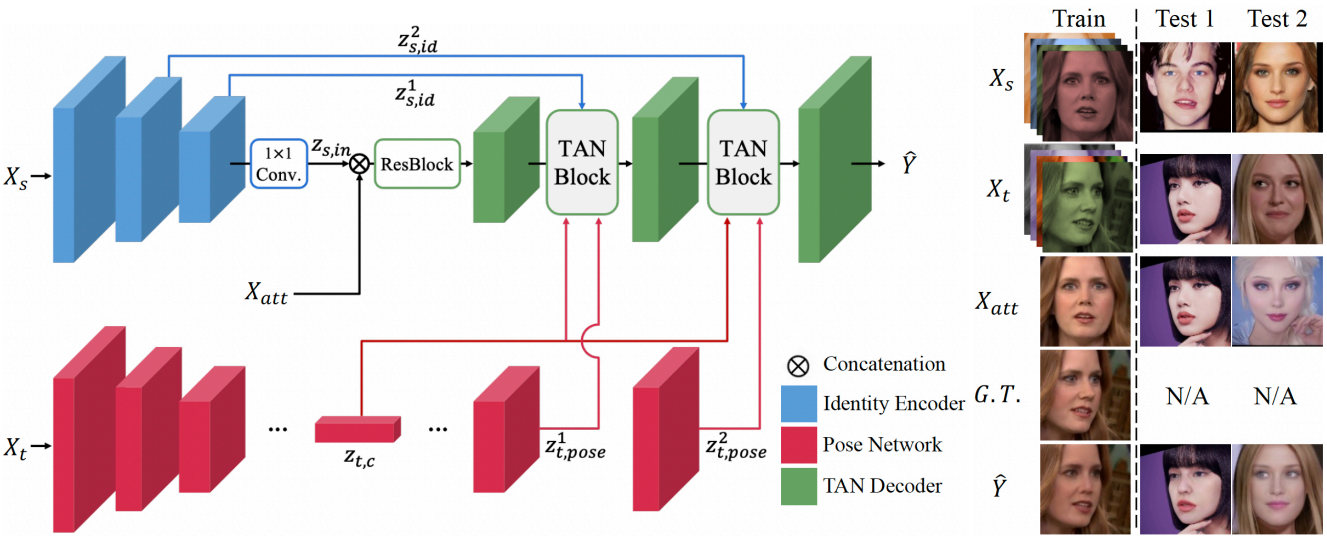

Figure 3: Overall architecture of FastSwap (left) and examples of inputs and outputs of training and test steps (right). X_s, X_t and X_{att} (a resized image) are used as inputs of our framework. The color-distorted data augmentation is applied to X_s and X_t in the training step to disentangle the identity, pose, and attributes from X_s, X_t, and X_{att}, respectively. Test 1 and Test 2 indicate a normal face swapping case and a controllable attribute editing case, respectively. Note that for Test 1, X_t is used as X_{att}, and for Test 2, the desired attribute image is used as X_{att}.

the layers to improve the identity appearance locally. In our framework, we design a differentiated decoder block that can combine the identity and pose information comprehensively by executing triple adaptive normalization: AdaIN, 1-channel SPADE, and multi-channel SPADE.

2.2. Face Swapping

Face swapping frameworks replace the face of the target image with the reenacted source face while preserving the attributes of the target face. FSGAN [16] suggested a cascaded face swapping framework consisting of reenactment, inpainting, and blending modules. FaceShifter [13] is a two-stage framework consisting of a face synthesis decoder utilizing adaptive attentional normalization and an anomaly recovering module. SimSwap [4] presented a one-stage framework with an ID injection module that transfers the identity information into the decoder using AdaIN. In our framework, we propose a switch-test strategy to apply the attributes of the target image into a reenacted image without any additional networks.

3. Methods

Given three input images, a source image $X_s \in \mathbb{R}^{3\times256\times256}$, a target image $X_t \in \mathbb{R}^{3\times256\times256}$, and an attribute input $X_{att} \in \mathbb{R}^{3\times H_1\times W_1}$ (resized image), our goal is to generate a swapped image $\hat{Y}$ preserving the identity of X_s, pose of X_t, and attributes of X_{att} with a lightweight framework. To achieve the goal, we propose a FastSwap network structure, train the network with novel data augmentation and use a switch-test strategy. Note that we handle the inputs of pose and attributes independently, whereas the previous face swapping frameworks deal with pose and attributes from the target image at once.

The proposed FastSwap network extracts identity features of X_s and pose features of X_t by using an identity encoder and pose network, respectively. Then FastSwap takes advantage of an adaptive normalization mechanism inspired by AdaIN [11] and SPADE [17] to integrate the features in a Triplet Adaptive Normalization (TAN) decoder. In addition, our data augmentation induces the network to extract attributes from X_{att} in the training step. Then, when testing, the desired attributes are applied to the output $\hat{Y}$ through the switch-test strategy.

3.1. FastSwap Architecture

We focus on disentangling the identity and pose with a shallow network and designing an effective integration method. Hence, as shown in Figure 3, our FastSwap consists of three modules: *1) Identity Encoder* which extracts the identity feature and provides the skip connections to the generator, *2) Pose Network* which extracts pose from target image and decodes the spatial pose feature, and *3) Decoder with TAN Block* which effectively integrates the features from *1)* and *2)* in an adaptive fashion. FastSwap network is trained in a self-supervised manner without any manual annotations or the off-the-shelf network.

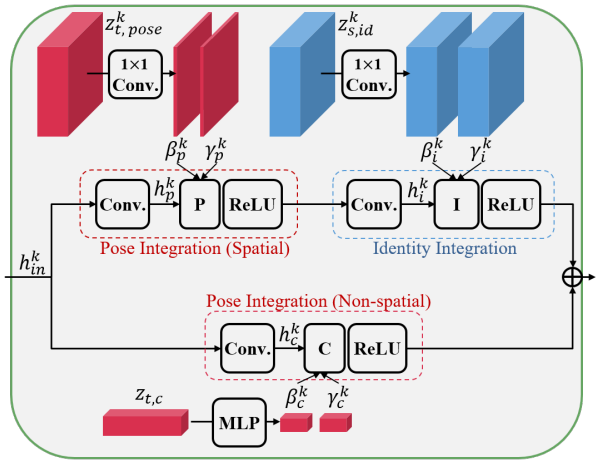

Figure 4: Detailed structure of the k-th TAN block with three separate adaptive normalizations. $\oplus$ denotes a sum operation.

3.1.1 Identity Encoder

The identity encoder extracts identity information from X_s. We only use two down-sampling blocks and extract the identity feature $z^1_{s,id}$ in a quarter size of each input height and width. $z^1_{s,id}$ passes through one 1×1 convolution layer, becomes $z_{s,in}$, then $z_{s,in}$ is used as an input of the generator. Moreover, the intermediate outputs of the identity encoder $\{z^k_{s,id}\}^N_{k=1}$, where $N = 2$ is a number of down-sampling blocks, are further used to generate the adaptive normalization parameters in each TAN block for identity integration.

3.1.2 Pose Network

In the pose network, X_t is encoded into $z_{t,c} \in \mathbb{R}^{C\times1\times1}$ to avoid any spatial identity information from X_t. While the identity encoder maintains the spatiality of the feature map in a quarter, a low-dimensional bottleneck target code $z_{t,c}$ is extracted to induce a self-disentanglement of the pose [3]. We then decode $z_{t,c}$ to train spatial pose features to reenact the pose of X_t. Target code $z_{t,c}$ and multi-level pose features $\{z^k_{t,pose}\}^N_{k=1}$ from the pose network are fed into TAN block for pose integration, where $N = 2$ is the number of TAN blocks in the TAN decoder.

3.1.3 Decoder with TAN Block

We incorporate $z_{s,in}$, $\{z^k_{s,id}\}^N_{k=1}$ from the identity encoder and $z_{t,c}$, $\{z^k_{t,pose}\}^N_{k=1}$ from the pose network to generate a swapped face image $\hat{Y}$. We propose a novel *Triple Adaptive Normalization* (TAN) block inspired by AdaIN and SPADE. TAN block guides the fusion of identity and pose with three adaptive normalizations considering each feature dimension. The TAN decoder is constructed with multiple TAN blocks to generate the output.

In the k-th TAN block, we design two parallel branches which combine spatial adaptive parameters from $z^k_{s,id}$ and $z^k_{t,pose}$, and non-spatial adaptive parameters from $z_{t,c}$ as shown in Figure 4. We arrange spatial and non-spatial pose integration in the two branches, respectively, and identity integration is placed in the rear since the identity feature $z_{s,in}$ is used as a decoder input. In other words, two adaptations are applied in a sequence of pose integration and identity integration in a spatial-adaptive branch, and a non-spatial pose integration is held in the other.

We perform three different adaptive normalizations of the activation map with their corresponding parameters generated from each input: 1) spatial pose integration with $z^k_{t,pose}$, 2) identity integration with $z^k_{s,id}$, 3) non-spatial pose integration with target code $z_{t,c}$ (see Figure 4). Let $h^k_p, h^k_i, h^k_c \in \mathbb{R}^{C_k\times H_k\times W_k}$ denote the activation map that is fed into each adaptive normalization of the k-th TAN block as an input, where C_k is the number of channels and $H_k \times W_k$ is the spatial dimensions.

For spatial pose integration, pose activation function P denormalizes the normalized $\bar{h}^k_p$ with 2D adaptive parameters generated from $z^k_{t,pose}$:

$$\begin{aligned} \bar{h}^k_p &= \frac{h^k_p - \mu^k_p}{\sigma^k_p} \\ P(h^k_p) &= \gamma^k_p \odot \bar{h}^k_p + \beta^k_p \end{aligned} \tag{1}$$

where μ^k_p, $\sigma^k_p \in \mathbb{R}^{1\times H_k\times W_k}$ are the mean and standard deviation of h^k_p over HW-wise activations, and $\beta^k_p, \gamma^k_p \in \mathbb{R}^{1\times H_k\times W_k}$ are modulation parameters convolved from $z^k_{t,pose}$ and $\odot$ is an element-wise multiplication.

For identity integration, we define the identity activation function I as denormalizing the normalized $\bar{h}^k_i$ according to the $z^k_{s,id}$:

$$\begin{aligned} \bar{h}^k_i &= \frac{h^k_i - \mu^k_i}{\sigma^k_i} \\ I(h^k_i) &= \gamma^k_i \odot \bar{h}^k_i + \beta^k_i \end{aligned} \tag{2}$$

where μ^k_i, $\sigma^k_i \in \mathbb{R}^{C_k\times H_k\times W_k}$ are the mean and standard deviation of h^k_i over CHW-wise activations, and $\beta^k_i, \gamma^k_i \in \mathbb{R}^{C_k\times H_k\times W_k}$ are modulation parameters generated from $z^k_{s,id}$.

For non-spatial pose integration, code activation function C denormalizes the normalized $\bar{h}^k_c$ according to the target code $z_{t,c}$:

$$\begin{aligned} \bar{h}^k_c &= \frac{h^k_c - \mu^k_c}{\sigma^k_c} \\ C(h^k_c) &= \gamma^k_c \odot \bar{h}^k_c + \beta^k_c \end{aligned} \tag{3}$$

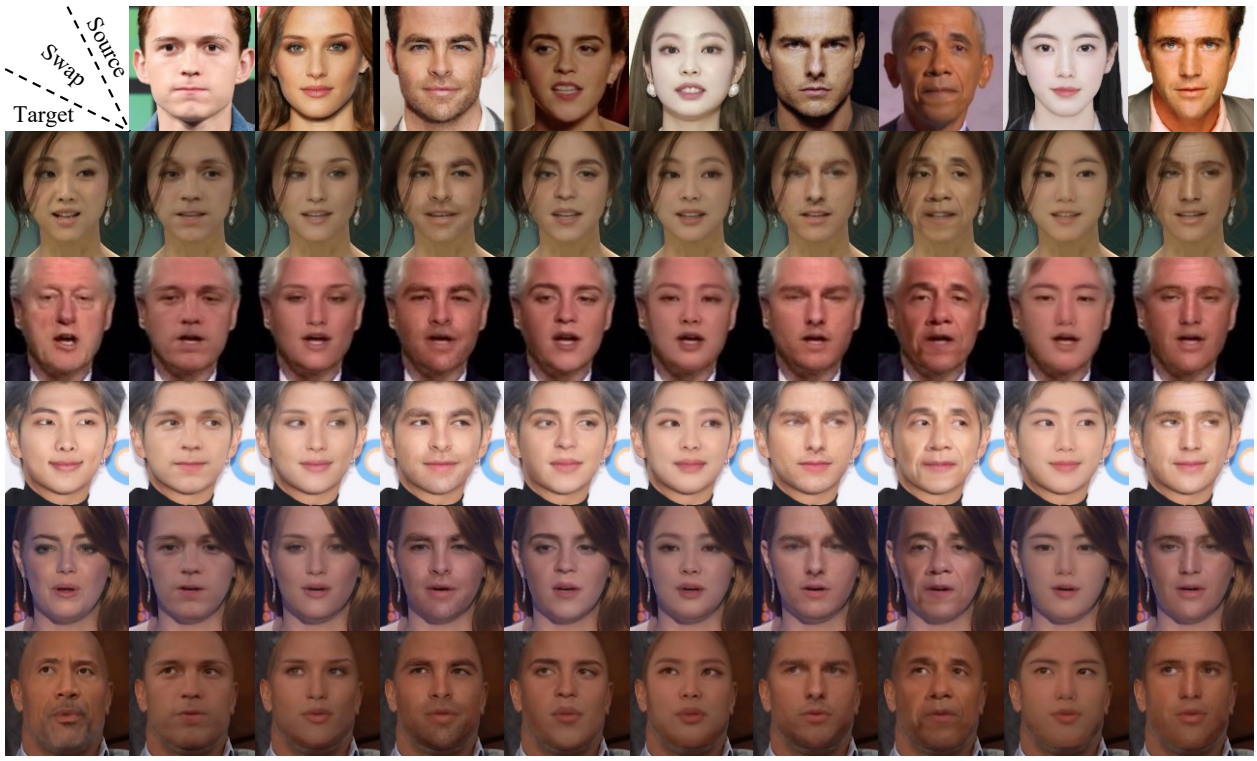

Figure 5: Results of FastSwap with various source and target pairs.

where $\mu_c^k, \sigma_c^k \in \mathbb{R}^{C_k \times 1 \times 1}$ are the mean and standard deviation of h_c^k over C-wise activations, and $\beta_c^k, \gamma_c^k \in \mathbb{R}^{C_k \times 1 \times 1}$ are modulation parameters learned from MLP with flattenned $z_{t,c}$ input.

The total activation of the k-th TAN block is formulated as

$$\begin{aligned} &TAN^k(h_{in}^k, z_{s,id}^k, z_{t,c}, z_{t,pose}^k) \\ &= I\big(Conv\big(P(Conv(h_{in}^k))\big)\big) + C\big(Conv(h_{in}^k)\big) \end{aligned} \tag{4}$$

where h_{in}^k is the input of the k-th TAN block, $Conv$s are 1×1 convolution layers, and ReLU activation is omitted for readability.

3.2. Data Augmentation and Switch-Test Strategy

Our data augmentation facilitate the FastSwap network to extract identity information from X_s, pose information from X_t, and attribute information from X_{att}. We exploit the characteristic of color distortion for data augmentation. As shown in the Train case in Figure 3, we operate different color distortion augmentation on X_s and X_t, respectively, since identity and pose information in the images are not compromised by color distortion. On the other hand, attribute information is sensitive to color changes. Consequently, X_{att} and the ground truth (G.T.) maintain their original color so that the attributes are extracted from X_{att}. Note that we use a resized source image as X_{att} in accordance with our switch-test strategy.

We introduce a switch-test strategy that considers the task gap between the training and test steps of the face swapping task. X_s and X_t have the same attributes in the training stage, but different attributes are used in the test stage. Considering the test stage, it is recommended to use X_t as X_{att}, but this is a situation where the G.T. is provided as input. Therefore, we use self-supervised learning by taking advantage of the fact that the attributes of X_s and X_t are the same in the training stage. We set X_{att} with a resized X_s maintaining the original color at the training stage. Then, at the test stage, we switch X_{att} with a resized X_t to reconstruct the attributes of X_t, as shown in the Test 1 case in Figure 3. Furthermore, we can generate various outputs with the desired attributes by adjusting the independent input X_{att}, as in the Test 2 case in Figure 3. The effects of using various X_{att} are reported in Section 4.4.

3.3. Training Objectives

We combine five losses to train the FastSwap framework. First, we define a $\mathcal{L}$-2 reconstruction loss L_{rec} and a VGG-19-based perceptual loss [12] L_{per} between the output $\hat{Y}$ and the ground truth (G.T.).

Next, we take advantage of adversarial training with the

discriminator to improve image quality. The discriminator is trained via its adversarial loss L_{adv}^{D}, while FastSwap is trained with the adversarial loss L_{adv}^{G}. Multi-scale discriminator [17] is used and each original binary cross entropy loss is substituted with hinge loss [15].

To preserve identity and pose of X_s and X_t, we utilize identity preservation loss L_{id} and pose reconstruction loss L_{pose}. L_{id} is calculated with cosine similarity of the identity feature from Arcface [7] between $\hat{Y}$ and X_s. L_{pose} is the $\mathcal{L}$-2 distance between $z_{t,c}$ and $\hat{z}_c$, where $\hat{z}_c$ is the target code reconstructed by output $\hat{Y}$ fed to the pose network encoder. $\hat{z}_c$ is expected to be close to $z_{t,c}$ since $\hat{Y}$ is intended to have the same pose with X_t.

FastSwap is finally trained to minimize a weighted sum of the above losses formulated as

$$\begin{aligned} &L_{rec}(\hat{Y}, G.T.) + \lambda_{per} L_{per}(\hat{Y}, G.T.) \\ &+ \lambda_{adv} L_{adv}^{G}(\hat{Y}, G.T.) + \lambda_{id} L_{id}(\hat{Y}, X_s) \\ &+ \lambda_{pose} L_{pose}(z_{t,c}, \hat{z}_c) \end{aligned} \tag{5}$$

with $\lambda_{per} = \lambda_{adv} = 1$, $\lambda_{id} = 0.1$, and $\lambda_{pose} = 10$.

4. Experiments

4.1. Implementation Details

FastSwap is trained with the large face dataset VoxCeleb2 [5]. We align and crop the face in a size of 256×256 using [18]. The number of layers in identity encoder and TAN block, N, is set to 2 while the pose network downsamples the feature 8 times, which results in $z_{s,in} \in \mathbb{R}^{128\times64\times64}$, $z_{t,c} \in \mathbb{R}^{128\times1\times1}$, respectively.

4.2. Quantitative Comparison

4.2.1 Evaluation Metrics

We use various evaluation metrics to compare the efficiency of the swapping process and the plausibility of the results. Specifically, we use 1) Frames per second (*FPS*) representing the swapping speed, which are measured under a common environment with one RTX2080Ti GPU, 2) Multiply-accumulate operations (*MACs*) measuring the computational complexity and 3) Number of parameters (*Param.*) of each framework, 4) Identity similarity (*ID*), the cosine similarity between the embedding vectors of Arcface [7] from the output and the source image evaluating the identity match, 5) Pose error (*Pose*), the normalized mean error of the head pose by using 68 landmarks [2] of the synthesized image and the target image, and 6) Frechet-inception distance (*FID*) [10] measuring perceptual realism computed with target image as a ground truth.

Method	*FPS*↑	*MACs*↓	*Param.*↓	*ID*↑	*Pose*↓	*FID*↓
FOMM	41.64	56.24G	73.98M	0.65	<u>0.88</u>	<u>138.29</u>
LPD	<u>57.81</u>	<u>30.81G</u>	<u>40.07M</u>	<u>0.68</u>	0.96	138.45
OSFV	10.97	384.65G	195.08M	0.66	1.01	143.57
Ours-M	**123.22**	**14.34G**	**26.50M**	**0.70**	**0.71**	**90.63**
FSGAN	6.62	846.84G	226.36M	0.38	**0.57**	88.52
SimSwap	<u>24.48</u>	<u>55.79G</u>	<u>107.24M</u>	<u>0.48</u>	0.66	77.46
FaceShifter	17.36	81.58G	418.75M	0.44	0.70	**42.40**
Ours	**123.22**	**14.34G**	**26.50M**	**0.54**	<u>0.61</u>	<u>60.08</u>

Table 1: Quantitative comparison results with evaluation metrics. ↑ indicates that the higher the value, the better performance, and the ↓ indicates the opposite. The best performance is presented in bold, and the second-best performance is underlined.

4.2.2 Experimental Results

For the quantitative comparison, we sample 118 videos from the VoxCeleb2 test set (one video for each individual) and swap ten source faces on wild evenly distributed by gender and race. Table 1 shows the comparison results with the previous neural talking head frameworks and the face swapping frameworks in two sections, respectively.

FastSwap swaps the face at the fastest speed with the fewest parameters and computational cost when seeing the *FPS*, *MACs*, and *Param.*. Even though the *MACs* and *Param.* of LPD are relatively on par with Ours-M, LPD requires a few-shot fine-tuning process inevitably. Since FaceShifter focuses on preserving the unexpected attributes, FaceShifter has the lowest *FID* score calculated with the target image. FSGAN has the lowest *Pose* because FSGAN tends to maintain the shape and size of the eyes, nose, and mouth of the target image at the expense of missing the identity of the source image. However, it can be said that FastSwap preserves the identity of the source and the pose of the target in high quality when judging by the overall *ID*, *Pose*, and *FID* values. Finally, our study may exceptionally show performance on par or less fidelity against the comparison models, but it is clear that our framework has an evident strength in terms of swapping speed, which is 7 times faster than the FaceShifter.

4.3. Qualitative Comparison

We compare FastSwap with the state-of-the-art neural talking head frameworks, FOMM [19], LPD [3], and OSFV [20], as shown in Figure 6. The neural talking head methods follow the background and attributes of the source image while our framework follows the background and attributes of the target image. Since the frameworks follow different backgrounds, we mask out the background of the results of each framework with Graphonomy [8]. We denote our results without background as Ours-M. Here, it is challenging to compare identities at a glance since the skin tone varies depending on the attributes. However, our framework bet-

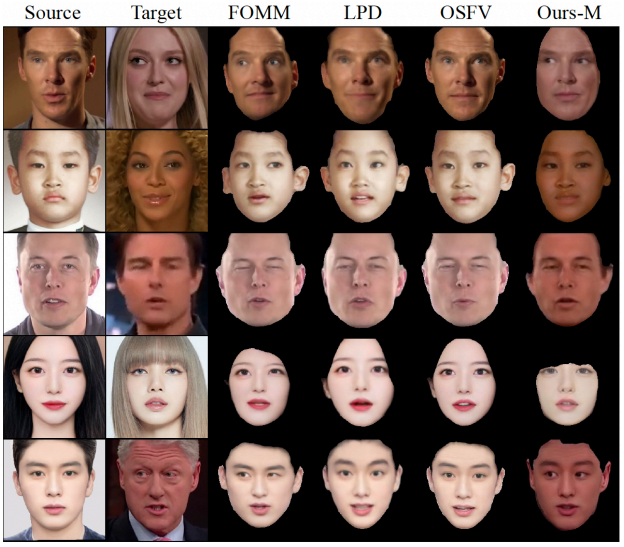

Figure 6: Comparison with state-of-the-art neural talking head methods. Ours-M denotes our results with the background masked out for ease of comparison.

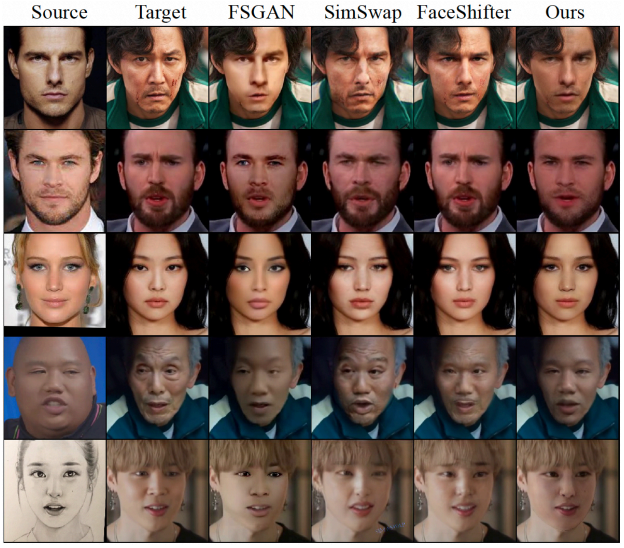

Figure 7: Comparison with state-of-the-art face swapping methods.

ter preserves the identity of the source face when looking at facial components separately. In addition, our framework best reconstructs the pose of the target when looking at the movement of the pupil or the shape of the mouth. Figure 6 row 3 shows that FastSwap determines the target pose even for low-fidelity input.

Figure 7 shows the comparison results with the state-of-the-art methods in face swapping, which are FSGAN [16], SimSwap [4], and FaceShifter [14]. Our framework best reenacts the pose of the target image, judging from the eyes, pupils, and lips movement of the results. In addition, FastSwap not only replaces the source face without loss of identity but also generates photo-realistic results by applying plausible attributes to the reenacted face than the other works. While unexpected attributes such as scars are better applied in SimSwap and FaceShifter (row 1), FastSwap focuses on preserving the source identities, including beard (row 2), wrinkles (row 4), and mole (row 5). Figure 1 right-lower result and Figure 7 row 5 show that FastSwap can extract the identity and pose of the input images even if the image is a cartoon or a drawing.

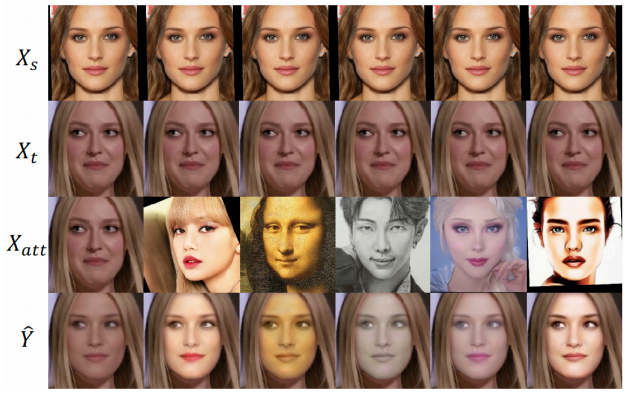

Figure 8: Results of FastSwap when using various X_{att}. The results follow the attributes of X_{att}, especially lip make-up and skin tone, while maintaining the same identity and pose.

4.4. Controllable Attribute Editing

In the previous experiments, we focused on the face swapping task using only two image inputs by putting the target image in X_{att}. However, our framework can edit the attributes of the result separately by using an extra image that has desired attributes. We visualize the results in Figure 8 by replacing X_{att} with several different images while maintaining input X_s and X_t the same. Figure 8 shows that the results are created according to the attributes of X_{att}, such as skin tone and make-up, while maintaining the same pose and identity. Unlike previous works, our framework can freely generate results representing the desired attributes by changing only X_{att}.

4.5. Analysis of FastSwap

4.5.1 TAN Block

To verify the necessity of each adaptive normalization in TAN block, we compare the results with a model without identity and pose activation functions (I, P, and C), respec-

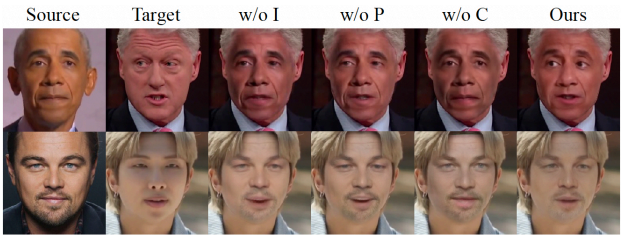

Figure 9: Comparison results of FastSwap with the ablation models detaching adaptive normalizations (I, P, and C) of TAN block. 'Ours' is omitted for readability from the name of ablation models.

tively. Figure 9 shows the results of the TAN block ablation study. As shown in Figure 9, I of TAN blocks improves the resolution of the output and integrates the detailed identity of the source image. P of TAN blocks affects mainly the detailed pose such as eyes and lips reenactment. C of TAN block reconstructs the general pose of the target image. The results denote that I, P, and C integrate identity and pose information as described in Section 3.1.3.

4.5.2 Data Augmentation

To examine the effect of our data augmentation (D.A.), we compare the results with a model trained without D.A. in Figure 10. Despite inputting the target image as X_{att} using the switch-test strategy, the model w/o D.A. follows the attributes of the source image since it is trained with source image attributes. X_{att} becomes meaningless since the model trained without D.A. extracts identity and attributes from X_s and pose from X_t. The results show that our proposed D.A. guides FastSwap to extract identity, pose, and attribute information from X_s, X_t, and X_{att}, respectively, during the training process.

4.5.3 Depth Design

To analyze whether our depth design is plausible, we compare the results with a deep identity encoder model *1*1 ID* ($z_{s,in} \in \mathbb{R}^{128\times1\times1}$) and a shallow pose network model *64*64 Pose* ($z_{t,c} \in \mathbb{R}^{128\times64\times64}$) in Figure 11. *1*1 ID* shows extreme pose and attributes loss leading to low-fidelity swapping results, and *64*64 Pose* fully reconstructs the target face. The results show that reducing the target code $z_{t,c}$ until 1×1 spatial resolution helps FastSwap to extract the pose, not the identity of the target images. The original shallow identity encoder improves the detail of identity from the source image by minimizing the loss of spatial feature size. The original deep pose network induces the activation P and C to focus on pose integration by preventing identity leakage from the target image. In short, our depth design

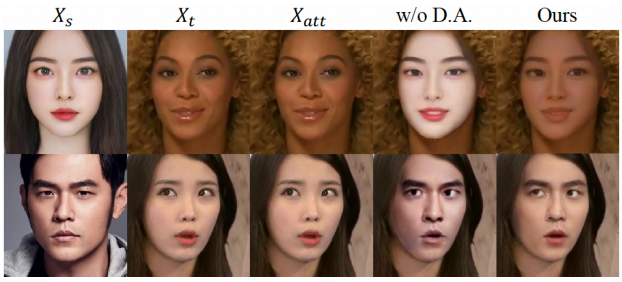

Figure 10: Comparison results of FastSwap between with and without the proposed data augmentation (D.A.).

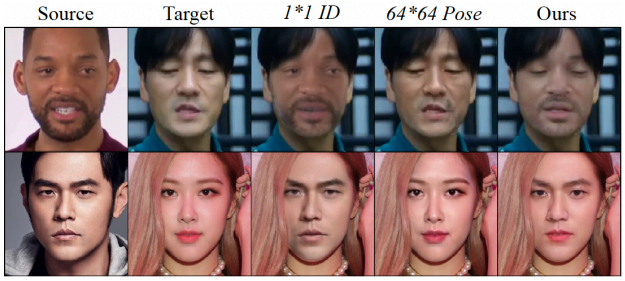

Figure 11: Results of FastSwap when changing the depth design. *1*1 ID* and *64*64 Pose* refer to the deep identity encoder model ($N = 8$) and the model in which the pose network downsamples X_t only twice, respectively.

supported TAN-block to prevent identity leakage of target images and improved the detail of identity from source images.

5. Conclusion

We have presented and evaluated our novel face swapping framework, FastSwap, which achieves the real-time swapping and preservation of identity, pose, and attributes of the given inputs. The main contribution of our paper is the TAN block that integrates identity and pose within a lightweight network. Our secondary finding is that the switch-test strategy with data augmentation guided an attributes extraction from the target image even though we used the source image during the training procedure. Our strategy facilitates controllable attribute editing, previously done through additional procedures, with a lightweight one-stage framework. Future work shall be on improvements to manipulate unexpected attributes.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by MSIT (South Korea) (No.2020-0-00440).

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[2] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017.

[3] Egor Burkov, Igor Pasechnik, Artur Grigorev, and Victor Lempitsky. Neural head reenactment with latent pose descriptors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13786–13795, 2020.

[4] Renwang Chen, Xuanhong Chen, Bingbing Ni, and Yanhao Ge. Simswap: An efficient framework for high fidelity face swapping. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2003–2011, 2020.

[5] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

[6] DeepFakes. Deepfakes github repository. `https://github.com/deepfakes/faceswap`, 2019. Accessed: 2021-11-28.

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[8] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7450–7459, 2019.

[9] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. Marionette: Few-shot face reenactment preserving identity of unseen targets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10893–10900, 2020.

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

[11] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[13] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457*, 2019.

[14] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[16] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7184–7193, 2019.

[17] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[18] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, et al. Deepfacelab: A simple, flexible and extensible face swapping framework. *arXiv preprint arXiv:2005.05535*, 2020.

[19] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in Neural Information Processing Systems*, 32:7137–7147, 2019.

[20] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10039–10049, 2021.

[21] Yuhan Wang, Xu Chen, Junwei Zhu, Wenqing Chu, Ying Tai, Chengjie Wang, Jilin Li, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Hififace: 3d shape and semantic prior guided high fidelity face swapping. *arXiv preprint arXiv:2106.09965*, 2021.

[22] Olivia Wiles, A Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–686, 2018.

[23] Guangming Yao, Yi Yuan, Tianjia Shao, Shuang Li, Shanqi Liu, Yong Liu, Mengmeng Wang, and Kun Zhou. One-shot face reenactment using appearance adaptive normalization. *arXiv preprint arXiv:2102.03984*, 2021.

[24] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In *European Conference on Computer Vision*, pages 524–540. Springer, 2020.

[25] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9459–9468, 2019.

[26] Xianfang Zeng, Yusu Pan, Mengmeng Wang, Jiangning Zhang, and Yong Liu. Realistic face reenactment via self-supervised disentangling of identity and pose. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12757–12764, 2020.

[27] Yunxuan Zhang, Siwei Zhang, Yue He, Cheng Li, Chen Change Loy, and Ziwei Liu. One-shot face reenactment. *arXiv preprint arXiv:1908.03251*, 2019.