FaceDancer: Pose- and Occlusion-Aware High Fidelity Face Swapping

Felix Rosberg[1,2] Eren Erdal Aksoy[2] Fernando Alonso-Fernandez[2] Cristofer Englund[2]
[1]Berge Consulting, Gothenburg, Sweden [2]Halmstad University, Halmstad, Sweden
felix.rosberg@berge.io, {eren.aksoy, fernando.alonso-fernandez, cristofer.englund}@hh.se

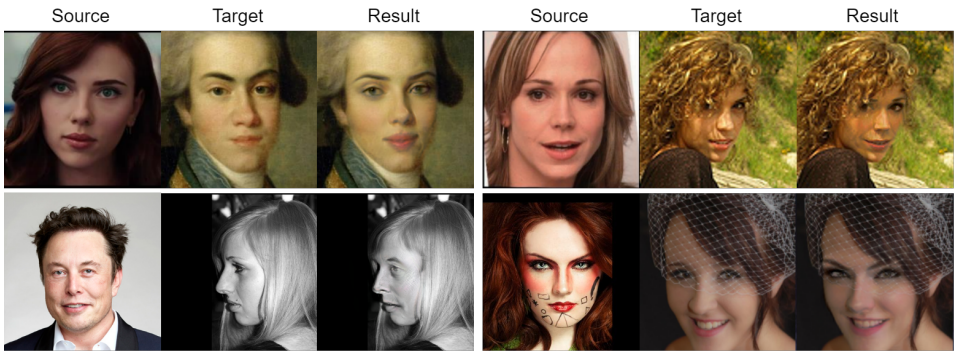

Figure 1: Face swapping results generated by FaceDancer.

Abstract

In this work, we present a new single-stage method for subject agnostic face swapping and identity transfer, named FaceDancer. We have two major contributions: Adaptive Feature Fusion Attention (AFFA) and Interpreted Feature Similarity Regularization (IFSR). The AFFA module is embedded in the decoder and adaptively learns to fuse attribute features and features conditioned on identity information without requiring any additional facial segmentation process. In IFSR, we leverage the intermediate features in an identity encoder to preserve important attributes such as head pose, facial expression, lighting, and occlusion in the target face, while still transferring the identity of the source face with high fidelity. We conduct extensive quantitative and qualitative experiments on various datasets and show that the proposed FaceDancer outperforms other state-of-the-art networks in terms of identity transfer, while having significantly better pose preservation than most of the previous methods. Code available at `https://github.com/felixrosberg/FaceDance`.

1. Introduction

Face swapping is a challenging task aiming at shifting the identity of a source face into a target face, while preserving the descriptive face attributes such as facial expression, head pose, and lighting of the target face. The idea of generating such non-existent face pairs has a vast range of applications in the film, game, and entertainment industry [2]. Therefore, face swapping has rapidly attracted increased research interest in computer vision and graphics. The challenge in swapping faces remains in achieving a high fidelity identity transfer from the source face with a set of attributes which need to be consistent with those in the target face.

There exist two mainstream approaches for face synthesis: *source-oriented* and *target-oriented* methods. The former approaches initially synthesize a source face with the attributes captured in the target face, which is then followed by blending the source face into the target counterpart [2], [30], [31]. These techniques still have difficulties in handling lighting, occlusion, and complexity. The latter approach directly convert the identity of the target face into one in the source face [8], [27], [45], [40], [39], [3]. These methods particularly rely on Generative Adversarial Networks (GAN) using a one-stage optimization setting. This helps preserve the target image attributes, such as pose and lighting, without requiring any additional processing step, *e.g.* by learning perceptual and deep features already in the training stage [8], [27], [32], [20], [42], [38].

In this work[1], we introduce a novel, *target-oriented*, and

[1]Work done within the Vinnova project MIDAS (2019-05873).r

single-stage method, named *FaceDancer*, to deal with challenges, e.g., lighting, occlusion, pose, and semantic structure (See Fig. 1). *FaceDancer* is simple, fast, and accurate.

Our core contribution is twofold: First, we introduce an Adaptive Feature Fusion Attention (AFFA) module, which adaptively learns during training to produce attention masks that can gate features. Inspired by the recent methods [27] and [39], the AFFA module is embedded in the decoder and learns attribute features without requiring any additional facial segmentation process. The incoming feature maps in AFFA are features that have been conditioned on the source identity information, but also the skip connection of the unconditioned target information in the encoder (See Fig. 2). The AFFA module, in a nutshell, allows *FaceDancer* to learn which conditioned features (*e.g*. identity information) to discard and which unconditioned features (*e.g*. background information) to keep in the target face. Our experiments show that gating from the AFFA module considerably improves the identity transfer.

Second, we present an Interpreted Feature Similarity Regularization (IFSR) method for boosting the attribute preservation. IFSR regularizes *FaceDancer* to enhance the preservation of facial expression, head pose, and lighting while still transferring the identity with high fidelity. More specifically, IFSR explores the similarity between intermediate features in the identity encoder by comparing the cosine distance distributions of these features in the target, source, and generated face triplets learned from a pretrained state-of-the-art identity encoder, ArcFace [12] (See Fig. 2).

We conduct extensive quantitative and qualitative experiments on the FaceForensic++ [34] and AFLW2000-3D [44] datasets and show that the proposed *FaceDancer* significantly outperforms other state-of-the-art networks in terms of identity transfer, while maintaining significantly better pose preservation than most of the previous methods. To address the scalability of our network, we further apply *FaceDancer* to low resolution images with harsh distortions and qualitatively show that *FaceDancer* can still improve the pose preservation in contrast to other methods.

2. Related Work

There are two leading approaches for face swapping: source- and target-oriented methods. Although our proposed method falls into the later category, we here provide a brief review of the literature related to both approaches.

Source-oriented approaches first transform the source face to match the expression and posture of the target face, and then blend with the target frame. One of the earliest approaches is The Digital Emily project [2] which performs face swapping through expensive and time-consuming 3D scanning of a single actor. Getting one face ready with this method to insert in a scene can, however, take months. Banz et al. [4] presents an early approach for utilizing 3D Morphable Models (3DMM) [13] to generate source faces with matching target attributes. This approach, however, comes with the cost that for each image the subject hair must be carefully marked out. Nirkin et al. [31] also utilizes 3DMM to extract pose and expression coefficients from the target face. These coefficients are then employed for reconstructing the source face. The reconstructed image is finally combined with the output from a facial segmentation network in order to automate the entire face swapping process. This method, however, struggles with textures and lighting conditions. FSGAN [30] introduces a reenactment network particularly designed to reenact the source face based on the target landmarks. In this work, the blending process is performed in an additional step which combines outputs from a segmentation network together with outputs from an inpainting network. This method also struggles with lighting conditions. More importantly, due to relying on the target landmarks for reenactment, the reenacted source falls short in having an effective identity transfer.

Target-oriented approaches mostly rely on generative models to manipulate features of an encoded target face, together with a semi-supervised loss function or a regularization method to preserve attributes. Almost all of these methods, including ours, utilize facial recognition models in order to extract identity information to be later used for conditioning of the target features. FaceShifter [27] robustly transfers the identity while maintaining attributes by having an attribute encoder-decoder model trained in a semi-supervised fashion. This model is coupled with a generator that injects the source identity information and adaptively learns to gate features between the generator and the attribute model. FaceShifter also has a secondary stage to improve the occlusion awareness. This approach succeeds well with identity and occlusion, but struggles with hard poses, which is solved by our new IFSR loss function. SimSwap [8] has an encoder-decoder model that utilizes the identity information to manipulate the bottleneck features. To preserve attributes, SimSwap uses a modified version of the feature matching loss from pix2pixHD [38]. This approach achieves state-of-the-art performance for preserving the pose at an arguably large trade-off for the identity transferability. HifiFace [39] uses a combination of GANs and 3DMMs to achieve state-of-the-art identity performance. Although HifiFace produces high resolution photo-realistic face swaps, it, however, seems not to improve the pose considerably and performs worse than SimSwap. In addition, HifiFace relies on a 3DMM model, which particularly works well with high resolution images only [18], [13].

Our approach differs from these methods in that ours rely on the identity encoder for simplicity and can handle harsh image distortions such as artifacts that emerge in low resolution images. Our method also reaches state-of-the-art identity performance and improves the pose preservation in

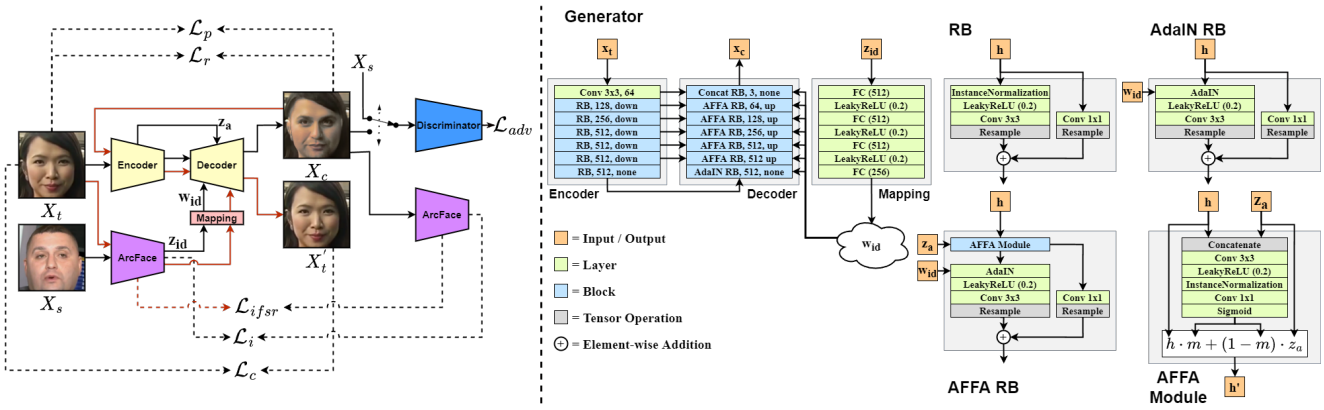

Figure 2: Overview of our proposed single-stage face swapping network *FaceDancer*. Left: The information flow in *FaceDancer* during training. Black lines indicate standard information flow, while red lines are the cycle consistency loss information flow and dashed lines represent inputs for losses (Section 3.4). Note that, the ArcFace model has two instances just to avoid having otherwise multiple intersecting arrows in the figure. Right: RB stands for ResBlock. X_s is the source face, X_t is the target face, X_c is the changed face, z_{id} is the identity vector extracted from ArcFace, w_{id} is the mapped identity vector, h is an incoming feature map, and z_a is a skip connection feature map. The layer Resample represents either an average pooling operation abbreviated as *'down'* or a bilinear upsampling indicated as *'up'* or an identity function shown as *'none'*. The layer Concat RB concatenates h and z_a without the AFFA module.

contrast to HifiFace.

3. Method

This section describes the *FaceDancer* network architecture shown in Fig. 2, together with the AFFA module, the IFSR method, and loss functions. Throughout this paper, we use the following notations: X_t refers to the *target face* which is the face image to be manipulated, X_s defines the *source face* which is the image of the face whose identity is transferred, and X_c is the *changed face* representing the manipulated target face with the identity of the source face.

3.1. Network Architecture

FaceDancer involves a generator and a discriminator forming a conditional GAN model coupled with a mapping network and ArcFace [12] as depicted in Fig. 2.

Generator: The generator G relies on an U-Net like encoder-decoder architecture combined with a mapping network M (See Fig. 2). The encoder consists of a set of residual blocks with gradually increasing number of filters. The decoder also involves a set of residual blocks, each of which employs either Adaptive Instance Normalization (AdaIN) [19], [23], [8] or an AFFA module or a concatenation layer for exploiting encoded skip connections. The main aim of G is to generate X_c from the encoded image X_t while conditioning the feature maps on the mapped identity vector w_{id} extracted from X_s as shown on the left in Fig. 2.

Discriminator: The discriminator D used for the adversarial loss is the same as the one in StarGan-v2 [9] and HifiFace [39], with the exception that we omit the multi-task discrimination, since we use the hinge loss.

Mapping network: *FaceDancer* has a mapping network M to boost the performance of G as already shown in [23], [24], [22], [9], [21]. The mapping network learns to transform the initial identity distribution to a new distribution in order to particularly inject the identity information. The M network consists of four fully-connected layers (FC) combined with leaky ReLU as non-linearity in all layers except the last (Fig. 2).

ArcFace: To extract and inject identity information from the source image X_s, *FaceDancer* employs a pretrained state-of-the-art identity encoder, ArcFace [12], coming with a ResNet50 backbone [16]. The resulting ArcFace output is an identity vector with the size of 512 that serves as an input to *FaceDancer*. The ArcFace model is also used for the computation of IFSR (Section 3.3) and the identity loss (Section 3.4).

3.2. The Adaptive Feature Fusion Attention (AFFA) Module

The AFFA module is inspired by previous works such as the Adaptive Attentional Denormalization layer in FaceShifters [27] and the Semantic Facial Fusion module in HifiFaces [39]. Unlike the former method where a separate attribute encoder-decoder model exists, we here keep everything condensed within the generator. In contrast to the latter method, which utilizes segmentation masks for supervision, we here avoid introducing any additional need to compute such segmentation masks for each training sample

by letting AFFA adaptively learn attention masks. In this regard, AFFA employs the information from skip connections in the generator encoder and forces the generator to learn whether it should rely on features (z_a) from skip connections or features (h) from the decoder conditioned on the source identity (Fig. 2). This way, AFFA can implicitly learn to extract relevant descriptive face features. Instead of naively concatenating or adding the two feature maps (h and z_a), AFFA first concatenates the feature maps and then passes them through a few learnable layers (Fig. 2). Finally, AFFA produces an attention mask m with the same filter number as in h and z_a. The following equation is used to gate and fuse h and z_a:

$$h' = h \cdot m + (1 - m) \cdot z_a \ , \tag{1}$$

where h' denotes the final fused feature map between h and z_a. We experimentally demonstrate the impact of the AFFA module by comparing with cases where either concatenation or addition is individually used to incorporate the information from skip connections in the generator encoder.

3.3. Interpreted Feature Similarity Regularization (IFSR)

Target-oriented face swapping methods rely particularly on semi-supervised or unsupervised techniques to make sure that the output image maintains the target attributes. To favor the preservation of attributes, we regularize the *FaceDancer* training by employing intermediate features captured by the ArcFace [12] identity encoder described in section 3.1. This idea of using pretrained identity encoders for exploring facial expressions is also supported by the recent work in [36].

To investigate which layers of ArcFace are responsible for facial expressions and, thus, contribute more to the attribute preservation, we perform a pre-study on a state-of-the-art face swapping model FaceShifter[27]. Note that, since the source code of FaceShifter[27], to the best of our knowledge, is not public, we here use our implementation of FaceShifter with minor modifications. For instance, in our implementation, the generator down samples to a resolution of 8×8, instead of 2×2. We also incorporate the weak feature matching loss from [8] together with the L1 reconstruction loss, instead of L2. Next, we use our baseline implementation of FaceShifter to perform random face swaps between identities in the VGGFace2 data set [7]. We then compare the cosine distances not only between the target and the generated face swaps, but also between the source and generated face pairs for exploring the intermediate features in each block in the ArcFace backbone. In addition, we compute the distance for intermediate features between negative pairs (imposters) of identities as qualitative reference. All these measured distributions of distances help us determine which layers, i.e., intermediate feature maps, are useful for preserving attribute information. For example, if there is a small distance between the target face and the generated face swap, it indicates that the intermediate features from that layer contain more attribute information. For this purpose, we also define a margin m_i for each ith layer based on the computed mean distances. The motivation for the margin is to regularize the generator to match the mean of the distribution, instead of completely minimizing the distance. The final regularization equation is as follows:

$$\mathcal{L}_{ifsr} = \sum_{i=k}^{n} min(1 - cos(I^{(i)}(X_t), I^{(i)}(X_c)) - m_i \cdot s, 0), \tag{2}$$

where $I^{(i)}$ denotes the ith intermediate feature map in the identity encoder ArcFace, m_i represents the aforementioned margin for the ith layer, s is a hyperparameter that scales the margin, $cos(.)$ represents the cosine similarity between two feature maps, k and n respectively denote the index of the first and final blocks, from which intermediate feature maps are extracted. Note that the feature maps are initially reshaped to a vector to have the appropriate dimensionality for the cosine similarity operation. In our experiments, k and n are set to 2 and 13, respectively. The main role of the margin scale s is to control the amount of feature similarity that can deviate from the margin. The lower the value of s, the stricter is the similarity.

3.4. Loss Functions

During training, *FaceDancer* employs various loss functions: Identity loss, reconstruction loss, perceptual loss, adversarial loss regularized with our IFSR method, and gradient penalty for the discriminator. See Fig. 2 for an overview of how these loss functions interact with inputs and outputs.

The identity loss is used to transfer the source identity as follows:

$$\mathcal{L}_i = 1 - cos(I(X_s), I(X_c)) \ , \tag{3}$$

where I is the identity encoder ArcFace and $cos(.)$ denotes the cosine similarity. The output of I is the identity embedding vector z_{id} (See Fig. 2).

The reconstruction loss is used to make sure that when the target X_t and source X_s are the same images, the final result X_c should be equal to the target image on a pixel-wise level. This reconstruction loss is defined as follows:

$$\mathcal{L}_r = \begin{cases} ||X_t - X_c||, & \text{if } X_t = X_s \\ 0, & \text{otherwise.} \end{cases} \ , \tag{4}$$

To further strengthen the above behavior and improve the semantic understanding of the image, a perceptual loss is deployed. The motivation is that deep features as a perceptual loss have shown to be robust in many reconstruction tasks [38], [20], [42]. The perceptual loss is defined as:

$$\mathcal{L}_p = \begin{cases} \sum_{i=0}^{n} ||P^{(i)}(X_t) - P^{(i)}(X_c)||, & \text{if } X_t = X_s \\ 0, & \text{otherwise.} \end{cases}, \tag{5}$$

where $P^{(i)}$ denotes the ith feature map output of the VGG16 model [20] pretrained on Imagenet [10] and n is the final index of outputs before the down sample step within the VGG16 model. In our experiments, n is 4.

Furthermore, we utilize the cycle consistence loss to motivate the model to keep important attributes and structures within the target image [43], [25], [39], [9]. The cycle consistence loss is formulated as follows:

$$\mathcal{L}_c = ||X_t - G(X_c, I(X_t))|| \,, \tag{6}$$

where I denotes the identity encoder ArcFace and G is the generator.

For adversarial loss $\mathcal{L}_{adv}$ we use the hinge loss [32], [29], [41], [5], [28]. The discriminator is regularized with a gradient penalty term $\mathcal{L}_{gp}$ [14]. The total loss function for the generator G is a weighted sum of above losses, formulated as:

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_i \mathcal{L}_i + \lambda_r \mathcal{L}_r + \lambda_p \mathcal{L}_p + \lambda_c \mathcal{L}_c + \lambda_{ifsr} \mathcal{L}_{ifsr} \,, \tag{7}$$

where $\lambda_i = 10$, $\lambda_r = 5$, $\lambda_p = 0.2$, $\lambda_c = 1$ and $\lambda_{ifsr} = 1$. The weighting for $\mathcal{L}_{gp}$ (λ_{gp}) is set to 10.

4. Results

Implementation Details: *FaceDancer* is trained on the datasets VGGFace2 [7] and LS3D-W [6]. All faces are aligned with five point landmarks extracted with RetinaFace [11]. The alignment is performed to match the input into ArcFace [12]. We keep all images in the data sets. ArcFace is pretrained on MS1M [15] with a ResNet50 backbone. We used the Adam [26] optimizer with $\beta_1 = 0$, $\beta_2 = 0.99$, a learning rate of 0.0001, and exponential learning rate decay of 0.97 every 100K steps. The target (X_t) and source (X_s) images are randomly augmented with brightness, contrast, and saturation. Each configuration is trained for 300K steps for the ablation study (Table 2 and Table 3). We further train all the best performing configurations in the ablation studies (B, C, D) up to 500K steps to compare with the recent works using a batch size of 10. Image resolution for all of our models are 256×256. There is a 20% chance that an image pair is the same, with at least one pair in the batch being the same. Margin scale s in Eq. 2 is set to 1.2.

4.1. Quantitative Results

We perform quantitative evaluation of *FaceDancer* using the FaceForensics++ [34] dataset and compare it to the other state-of-the-art face swapping networks, such as SimSwap [8], FaceShifter [27], HifiFace [39], and FaceController [40]. The metrics evaluated are identity retrieval (ID), pose error, expression error, and Frechét Inception Distance (FID) [17]. For the identity retrieval, we initially perform random swaps for each image in the test set and then retrieve the correct identity with a secondary identity encoder, CosFace [37]. To compare pose, we use the pose estimator in [35] and report the average L2 error. The expression metric is often omitted for comparison due to poor accessibility of models. However, we here use an implementation of an expression embedder [33] and report the average L2 error. FID is calculated between the swapped version of the test set and the unaltered test set and helps demonstrate when a model has problems with lighting, occlusion, visual quality, and posture.

Table 1: Quantitative experiments on FaceForensics++ [34]. See Table 2 for the definition of each *FaceDancer* configuration (Config B to D). These models has been trained for 500k iterations.

Method	ID↑	Pose↓	Exp↓	FID↓
FaceSwap [1]	54.19	2.51	N/A	N/A
FaceShifter [27]	97.38	2.96	N/A	N/A
MegaFS [45]	90.83	2.64	N/A	N/A
FaceController [40]	98.27	2.65	N/A	N/A
HifiFace [39]	98.48	2.63	N/A	N/A
SimSwap [8]	92.83	**1.53**	8.04	**11.76**
FaceDancer (Config B)	98.54	2.24	8.52	25.11
FaceDancer (Config C)	**98.84**	2.04	7.97	16.30
FaceDancer (Config D)	98.19	2.15	**5.70**	19.10

Similar to the previous works [27], [8], [39], we sample 10 frames from each video in FaceForensic++ which yields a test data set of 10K. As shown in Table 1 our method *FaceDancer* outperforms all the previous works by leading to the highest identity retrieval performance. Regarding the pose metric, we have comparable results, i.e., *FaceDancer* achieves the second-lowest pose error (2.04) after SimSwap [8].

4.2. Qualitative Results

For the qualitative evaluation, we compare the performance of our model *FaceDancer* with the recent state-of-the-art works SimSwap [8], FaceShifter [27], HifiFace [39], and FaceController [40] as shown in Fig. 3. We here note that SimSwap [8] is the only work coming with a public and easy to access model. Due to this fact, we have more in depth comparison with SimSwap, whereas for the other baseline models we show qualitative results for sample images only reported in these works.

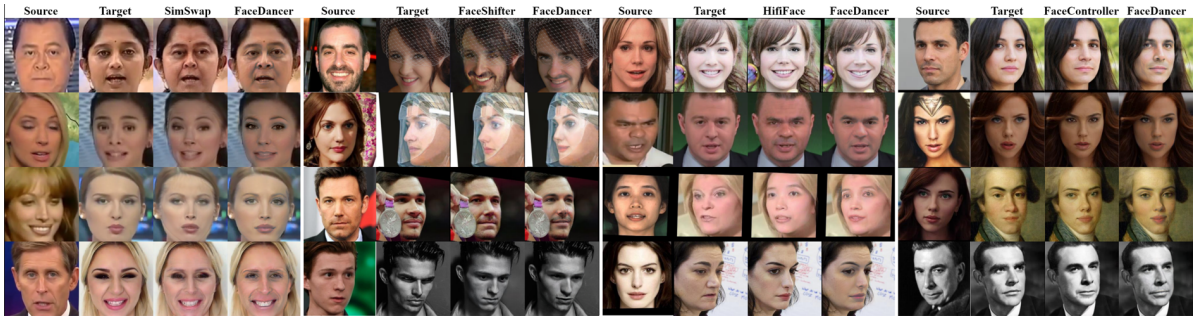

Figure 3: Comparing our model FaceDancer with SimSwap [8], FaceShifter [27], HifiFace [39], and FaceController [40].

Fig. 3 shows that our model *FaceDancer* behaves similar to SimSwap, but one can easily notice the substantially improved identity transfer in our results. FaceShifter performs good identity transfer and is able to transfer relevant attributes such as facial hair while preserving occlusion and the identity face shape. FaceShifter, however, struggles with lighting and gaze direction as it heavily relies on the second stage model. FaceController exhibits good identity transferability and decent pose error, however, still fails noticeably often with the gaze direction. Our approach *FaceDancer* deals with all these problems better. Finally, HifiFace demonstrates promising results regarding all these metrics, particularly when it comes to the facial shape. For instance, HifiFace exhibits better face shape preservation of the identity than our model. Otherwise, it is not feasible to compare qualitatively with HifiFace since our model *FaceDancer* quantitatively performs better (See Table 1).

Furthermore, to address the scalability of our model, we qualitatively analyze the performance of *FaceDancer* compared to SimSwap on low resolution face images. Fig. 4 shows that *FaceDancer* has enough capacity to capture the semantic structure of the face images even under low resolution cases. *FaceDancer* is able to maintain the pixelation artifacts, while SimSwap either produces a smooth face or completely fails, as depicted in the first row of Fig. 4. *FaceDancer* also works well on videos without any temporal information. We refer to supplementary materials for video results. In supplementary materials we also include further results of images in higher resolution, further comparisons, occlusion, difficult poses, extreme cases and finally failure cases. Failure usually occurs when the face poses away from the camera or the face pose is an uncommon angle represented in the data.

4.3. Ablation Study

We here ablate different *FaceDancer* components (such as the AFFA module and the IFSR method) and compare to two baselines as shown in Table 2. The ablations shown in Table 2 are evaluated on FaceForensic++ [34] and the ablations shown in Table 3 are evaluated on AFLW2000-

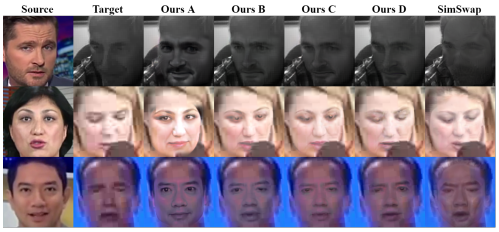

Figure 4: Qualitative comparison on low resolution images. See Table 2 for the definition of each *FaceDancer* configuration (Config B to D).

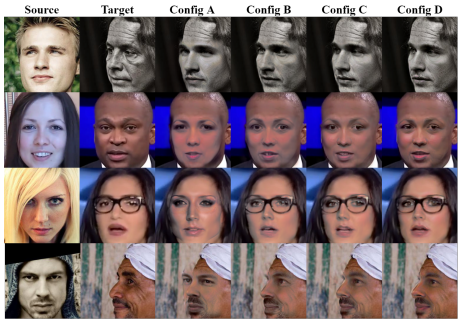

Figure 5: Illustration of the impact of IFSR. Config A given in the 3rd column here shows results once IFSR is omitted during training as described in Table 2.

3D [44]. Baseline 1 and 2 respectively employ concatenation and addition in order to fuse feature maps from the decoder and skip connection. For baseline 1, baseline 2, configuration A and configuration B, the feature fusion is performed at top three resolutions (256, 128, 64). For configuration C, we use concatenation at resolution 256 and AFFA at resolution 128, 64, and 32 are processed. Configuration D is the same as C, but with two additional AFFA modules and skips at resolution 16 and 8 (Fig. 2). Configuration E is the same as D with the only difference that the mapping network (M) in the generator is omitted (Fig. 2). We refer to the supplementary materials for detailed figures of the baselines and configurations. All ablation

Table 2: Ablative analysis together with the runtime performance. Inference time is given in millisecond and memory usage in GB. All models in this table were trained for 300k iterations.

Config	IFSR	AFFA	Concat final skip*	6 skips	Mapping	ID↑	Pose↓	Exp↓	FID↓	Inference	Memory
Baseline 1	✓	-	-	-	✓	97.66	1.97	8.20	16.72	74.9	1.25
Baseline 2	✓	-	-	-	✓	92.61	**1.87**	7.97	13.51	70.2	1.25
A	-	✓	-	-	✓	98.14	3.61	9.82	31.63	75.8	**1.18**
B	✓	✓	-	-	✓	96.96	2.48	8.25	23.11	75.8	**1.18**
C	✓	✓	✓	-	✓	**98.57**	2.27	7.98	14.59	78.3	1.26
D	✓	✓	✓	✓	✓	97.53	2.04	7.76	**13.50**	78.2	1.27
E	✓	✓	✓	✓	-	97.38	2.07	**5.73**	14.68	**64.6**	1.21

* Concatenation instead of AFFA at resolution 256 + one extra AFFA modules at resolution 32. See supplementary materials for detailed figures for each configuration.

configurations are trained for 300K steps. As reported in Table 2, baselines 1 and 2 achieve the lowest pose errors, however, with the cost of having either high FID score or poor identity performance. Configuration A improves identity performance but does not use IFSR which leads to poor pose error, expression error, and FID. Since Configuration B employs IFSR, it improves the expression and pose problem, however, still struggles with the FID. Configuration C overcomes these problems and achieves state-of-the-art performance on identity. Adding two more AFFA modules in lower resolutions in the decoder slightly disrupts the identity performance, but improves the other metrics further. This is mainly because Configuration D fuses more features from the target face. The last row in this table shows that the mapping information employed by the *FaceDancer* generator improves identity transfer and FID with expression error as trade off.

In Table 2, we also provide the total runtime performance for each *FaceDancer* configuration. Inference and memory consumption profiling are done on a single Nvidia RTX 3090 with a batch size of 32. Profiling includes inference for ArcFace.

The contribution of IFSR and AFFA becomes clearer when we ablate on the pose challenging dataset AFLW2000-3D [44] (Table 3). We here use AFLW2000-3D as the target data set and FaceForsenic++ as the source data set. In this case, after randomly swapping all faces in AFLW2000-3D with faces from FaceForensic++, we try to retrieve the original identity in FaceForensic++.

Our findings in Table 3 depict that baseline 1 still performs the best for pose, but falls short on the other metrics. Configurations A through D perform significantly better for ID, however, they have comparable pose error and similar or better expression error. Configuration E demonstrates the impact of not having the mapping network M used in *FaceDancer*. Configuration E falls short for identity performance and pose error (Table 3).

Table 3: Ablative analysis using AFLW2000-3D [44] as target and FaceForensics++ [34] as source. See Table 2 for configuration details.

Config	ID↑	Pose↓	Exp↓	FID↓
Baseline 1	89.10	**5.63**	5.34	19.26
Baseline 2	94.95	6.23	5.60	21.30
A	**98.50**	14.97	7.07	40.34
B	97.95	5.86	5.74	21.50
C	97.65	5.82	**4.13**	18.50
D	97.10	5.75	4.15	20.41
E	95.45	6.16	4.19	**18.13**

4.4. Analysis of the AFFA Module

In this section, we provide a comprehensive study showing the role of the Adaptive Feature Fusion Attention (AFFA) module. For this purpose, we first trained *FaceDancer* using AFFA in the three upper resolutions of the decoder (256, 128, 64). However, this leads to noticeable color defects in the swapped face images (Fig. 5). Experimental findings reported in Fig. 7 show that this is due to the usage of AFFA in the end of *FaceDancer* generator. The attention maps generated at the resolution of 256 are mostly gray, as depicted in Fig. 7. We hypothesize that in the highest layer of the generator, the attention maps are far away from fusing the feature maps as expected. As shown in Table 2, baselines 1 and 2 do not have any problems with the color defect as demonstrated by the substantially lower FID scores compared to configurations A and B which rely on AFFA at resolution 256. To remedy this problem, we replace the final AFFA module with a simple concatenation operation while adding an AFFA module to resolution 32. In Fig. 7, we show examples of attentions maps for each configuration in Table 2 at each resolution of the decoder where the *FaceDancer* generator uses an AFFA module.

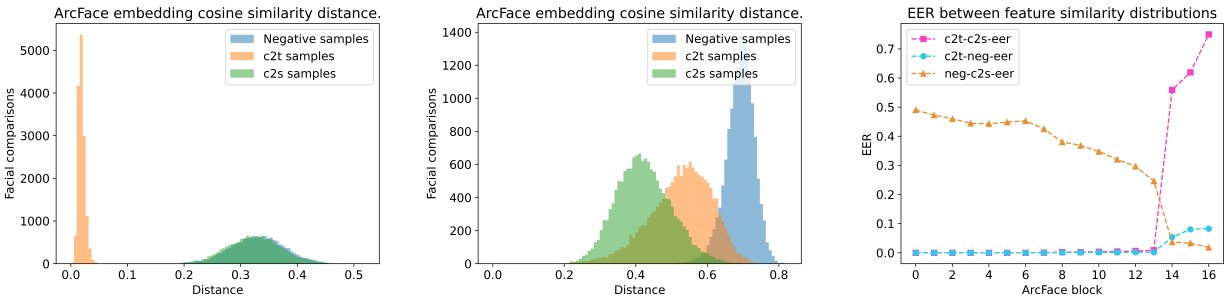

Figure 6: Cosine similarity between intermediate features between changed and target faces (c2t), changed and source faces (c2s), and different identities (Negative Samples). (a) Distances between features from first block of ArcFace. (b) Distances between features from final block of ArcFace. (c) Equal error rates (EER) between the distance distributions for intermediate features in every block.

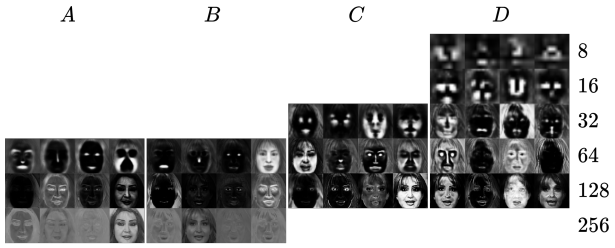

Figure 7: Comparison between example attention maps at different resolutions for different configurations in Table 2.

4.5. Analysis of IFSR

We now provide a comprehensive experimental evaluation on the role of the Interpreted Feature Similarity Regularization (IFSR) method.

We start investigating intermediate features within the ArcFace ResNet50 backbone by comparing the cosine distance between feature maps computed for the target face, the source face, the changed face, and negative pairs using the VGGFace2 [7] dataset. This process is repeated for each residual block output in ArcFace. The changed face is obtained by deploying a pretrained implementation of FaceShifter [27], briefly detailed in Section 3.3. As shown in Fig. 6(a), the changed and target faces ($c2t$) share significantly more similar features than those observed between the changed and source faces ($c2s$) in the early ArcFace layers. This behavior is not observed in the final residual block, as depicted in Fig. 6(b). This strongly suggests that the identity encoder contains important information such as pose, expression and occlusions in the earlier layers while the final blocks store identity information. To demonstrate the separability of the $c2t$ and $c2s$ distributions in Fig. 6, we calculate the equal error rate (EER) between these distributions. As shown by the EER plots in Fig. 6(c), the $c2t$ and $c2s$ distributions are completely separable until block 14. Afterwards, the EER jumps to more than 50%, which means that the $c2t$ distribution moves to the right of $c2s$, i.e., X_c shares more identity attributes with X_s in contrast to X_t in the same layer. This confirms that X_c successfully captures the identity of X_s. The qualitative impact of our proposed IFSR method is shown in Fig. 5. Without IFSR, the pasted face effect and lack of expression preservation become more apparent. Note that the layers and information from IFSR come from a frozen identity encoder. Therefore, any pretrained face swap framework could here be employed to calculate the IFSR margins. IFSR itself does not contain any learnable parameter. The process is just needed to gain an interpretable insight of what kind of information the layers contain (expression, pose, color, lightning, identity, etc.), and how to define the margins for IFSR.

5. Conclusion

In this work, we introduce *FaceDancer* as a new single-stage face swapping model that quantitatively reaches state-of-the-art. *FaceDancer* has a novel regularization component IFSR which utilizes intermediate features to preserve attributes such as pose, facial expression, and occlusion. Furthermore, the AFFA module in *FaceDancer* drastically improves identity transfer without a significant trade off for visual quality and attribute preservation when coupled with IFSR. *FaceDancer* is limited in two main aspects, transferring face shape and the need of calculating IFSR margins from a pretrained face swap model. Future directions for the latter is to figure out how to calculate the margins adaptively online. IFSR can potentially be used to compress complex face swap (or even image translation) models. Trying to combine IFSR with 3DMM to gain strong pose, occlusion and face shape preservation would be interesting future work.

References

[1] Faceswap. Accessed 2022-02-18.

[2] Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. Creating a photoreal digital actor: The digital emily project. In *2009 Conference for Visual Media Production*, pages 176–187, 2009.

[3] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6713–6722, 2018.

[4] Volker Blanz, Kristina Scherbaum, Thomas Vetter, and Hans-Peter Seidel. Exchanging faces in images. In *Computer Graphics Forum*, volume 23, pages 669–676. Wiley Online Library, 2004.

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[6] 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.

[7] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[8] Renwang Chen, Xuanhong Chen, Bingbing Ni, and Yanhao Ge. *SimSwap: An Efficient Framework For High Fidelity Face Swapping*, page 2003–2011. Association for Computing Machinery, New York, NY, USA, 2020.

[9] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[11] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5203–5212, 2020.

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2019.

[13] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38, 2020.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[15] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] 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.

[18] Guosheng Hu, Chi Ho Chan, Josef Kittler, and Bill Christmas. Resolution-aware 3d morphable model. In *BMVC*, pages 1–10. University of Surrey, 2012.

[19] 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.

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 694–711, Cham, 2016. Springer International Publishing.

[21] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

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

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

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[25] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International conference on machine learning*, pages 1857–1865. PMLR, 2017.

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

[27] 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, pages 5074–5083, 2020.

[28] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10551–10560, 2019.

[29] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[30] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7183–7192, 2019.

[31] Yuval Nirkin, Iacopo Masi, Anh Tran Tuan, Tal Hassner, and Gerard Medioni. On face segmentation, face swapping, and face perception. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 98–105, 2018.

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

[33] Felix Rosberg and Cristofer Englund. Comparing facial expressions for face swapping evaluation with supervised contrastive representation learning. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 01–05, 2021.

[34] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2019.

[35] Nataniel Ruiz, Eunji Chong, and James M Rehg. Fine-grained head pose estimation without keypoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2074–2083, 2018.

[36] Raviteja Vemulapalli and Aseem Agarwala. A compact embedding for facial expression similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5683–5692, 2019.

[37] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[38] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[39] 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. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1136–1142. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[40] Zhiliang Xu, Xiyu Yu, Zhibin Hong, Zhen Zhu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Facecontroller: Controllable attribute editing for face in the wild. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3083–3091, May 2021.

[41] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[42] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[43] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[44] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 146–155, 2016.

[45] Yuhao Zhu, Qi Li, Jian Wang, Cheng-Zhong Xu, and Zhenan Sun. One shot face swapping on megapixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4834–4844, June 2021.