Smooth-Swap: A Simple Enhancement for Face-Swapping with Smoothness

Jiseob Kim[1,2], Jihoon Lee[2], Byoung-Tak Zhang[1]
[1]Seoul National University, [2]Kakao Brain
jkim@bi.snu.ac.kr, jihoonlee.in@gmail.com, btzhang@bi.snu.ac.kr

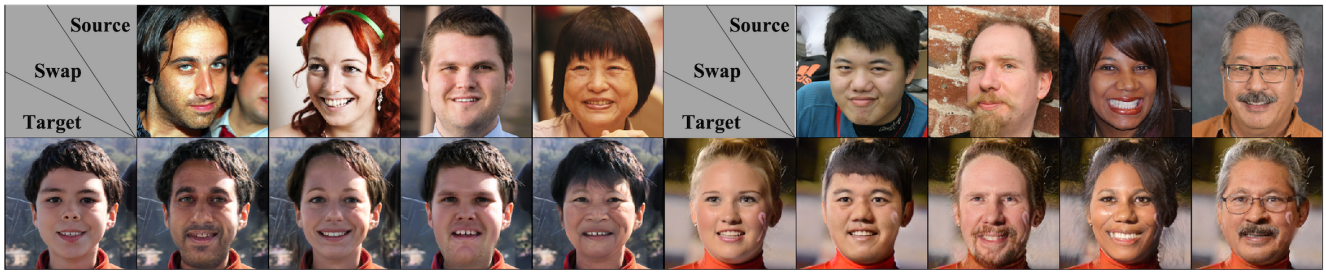

Figure 1. Face-swapped images generated by our *Smooth-Swap* model. In the swapped images, the identities of the target images are replaced with that of the source images. See the face-shape, hair, and mustache change in accordance with different sources.

Abstract

Face-swapping models have been drawing attention for their compelling generation quality, but their complex architectures and loss functions often require careful tuning for successful training. We propose a new face-swapping model called 'Smooth-Swap', which excludes complex handcrafted designs and allows fast and stable training. The main idea of Smooth-Swap is to build smooth identity embedding that can provide stable gradients for identity change. Unlike the one used in previous models trained for a purely discriminative task, the proposed embedding is trained with a supervised contrastive loss promoting a smoother space. With improved smoothness, Smooth-Swap suffices to be composed of a generic U-Net-based generator and three basic loss functions, a far simpler design compared with the previous models. Extensive experiments on face-swapping benchmarks (FFHQ, FaceForensics++) and face images in the wild show that our model is also quantitatively and qualitatively comparable or even superior to the existing methods.

1. Introduction

Face swapping is a task to switch the person-identity of a given face image with another, preserving other attributes like facial expressions, head poses, and backgrounds. The task has been highlighted for its wide use of real-world applications, such as anonymization in privacy protection and the creation of new characters in the entertainment industry. With progress made over years [3, 6, 16, 21, 22, 28, 31, 33], state-of-the-art face-swapping models can generate a swapped image of decent quality using a single shot of a new source identity.

Despite the performance improvement, however, existing models usually adopt complex model architectures and numerous loss functions to change *face shape*. Face shape is a crucial component of identity, but changing it is a nontrivial task; it incurs a dramatic change of pixels, but no guidance can be given due to the inherent absence of the ground-truth swapped images. Thus, previous studies have focused on using handcrafted components such as mask-based mixing [6] or 3D face-shape modeling [16, 31]. Although such components are effective for changing shape and improving the swapped-image quality, the models have added complexity of hyperparameters and loss functions that require careful tuning for successful training.

In this study, we postulate that the approaches based on handcrafted components are not the best way to resolve the difficulty of face-swapping. We propose instead a new identity embedding model having improved smoothness, which we assume to be related most to the gist of the problem. An identity embedding model, or an embedder, plays a key role during the training of the swapping model. It gives gradients for the generator, to which direction it has to tune to change the identity. It is thus important the embedder has a smooth space, since the gradients can be erroneous or noisy otherwise. In our proposed model, *Smooth-Swap*, we con-

sider a new embedder trained with supervised constrastive loss [14]. [30]. We find it has a smoother space than the ArcFace embedder [7], one used in the most of the existing models, and helps faster and stable training.

Through the smooth embedder, Smooth-Swap works without any handcrafted components. It adopts a simple U-Net [24]-based generator, and we train it using only three basic loss functions—identity change, target preserving, and adversarial (Fig. 2). While this set-up is simpler than the existing models, we find that our model can still achieve comparable or superior performance by taking a data-driven approach and minimizing inductive bias.

The advantages of Smooth-Swap can be summarized as follows. **1) Simple architecture**: Smooth-Swap uses a simple U-Net [24]-based generator, which does not involve any handcrafted components as the existing models. **2) Simple loss functions**: The Smooth-Swap generator can be trained using minimal loss functions for face-swapping—identity, pixel-level change, and adversarial loss. **3) Fast training**: The smooth identity embedder allows faster training of the generator by providing more stable gradient information.

2. Related Work

Approaches based on 3D Models and Segmentation Earlier face-swapping models rely on external modules such as 3D Morphable Models (3DMM) [4] and a face segmentation model. Face2Face [29] and [23] fit the source and the target images to 3DMM and transfers the expression (and the posture) parameters to synthesize the swapped image. RSGAN [21], FSNet [20], and FSGAN [22] use a segmentation model to separate the facial region from the background, generate the swapped image by switching and blending the regions. Despite the early success, these approaches do not produce high quality images since their performance depends on the non-trainable external modules.

Feature-based GAN models In contrast with the approaches above, recent models consider end-to-end training, generating a face-swapped image based on learned features. IPGAN [3] learns separate embedding vectors for the identity and the target attributes, switching and recombining them to generate a swapped image. FaceShifter [16] considers multi-level mixing using an encoder-decoder architecture, alleviating the information loss in the approach of IPGAN. SimSwap [6] proposes weak feature matching to focus more on preserving the facial expression of the source, whereas HifiFace [31] proposes a method integrating 3D shape model to focus more on active shape change. InfoSwap [8] uses information bottleneck for better disentangling the identity attributes from the rest. MegaFS [33] utilizes a pretrained StyleGAN2 [13] to generate high-resolution face-swapped images. [19] also tackles high resolution by training a separate generator for each identity. Although these models have continuously improved the quality of the generated images, they tend to show weak identity change or involve complexity due to handcrafted components.

3. Problem Formulation & Challenges

We first describe the problem formulation and main technical challenges of face-swapping. Then, we introduce how the smoothness of an identity embedder can alleviate them.

3.1. Problem Formulation

When a source x_{src} and a target x_{tgt} are given, a face-swapping model needs to generate the swap image, x_{swap}, which satisfies the following conditions:

C1. It has the identity of the source image.

C2. Other than the identity, it looks the same as the target image (having the same background, pose, etc.).

C3. It looks realistic (indistinguishable from real images).

To meet these requirements, most of face-swapping models [16, 31] consist of three components: an identity embedder f^*_{emb} for the source image, a generator f_{gen} for the swapped image, and a discriminator f_{dis} to improve the fidelity. Fig. 2 shows an overview of these face-swapping models including our approach. Note that the identity embedder is pre-trained and frozen during the training of other components, so the asterisk is included in the superscript.

3.2. Challenges for Changing Identity

The main difficulty for training a face-swapping model comes from the conflict between C1 and C2. Satisfying C1 makes x_{swap} move away from x_{tgt} to change the identity, whereas satisfying C2 enforces it to stay around. If we can accurately extract the *identity-irrelevant* change of x_{swap} from x_{tgt} and use it for the loss of C2, this conflict would have been relaxed. Unfortunately, designing such a loss is difficult, and a common fallback is to use an isotropic loss such as perceptual [32] or pixel-level L_p loss.

A major consequence of the conflict and an isotropic C2 loss is stagnant face-shape change. Shape-wise change such as round to sharp chin involves geometric transformations and entails dramatic variation in features and pixel values. It is thus a big fight against the C2 loss preventing any aspects of deviation from x_{tgt} and often compromised first. In this regard, previous work put much effort for changing face shape correctly, using a 3D face model, for example, to better capture the shape [31]. However, such a design introduces additional complication and requires a careful balancing between modules for successful training. In this work, we hypothesize that the conflict can be relaxed not by adding new modules but by introducing smoothness to an identity embedder. We will describe the details on this in the following section.

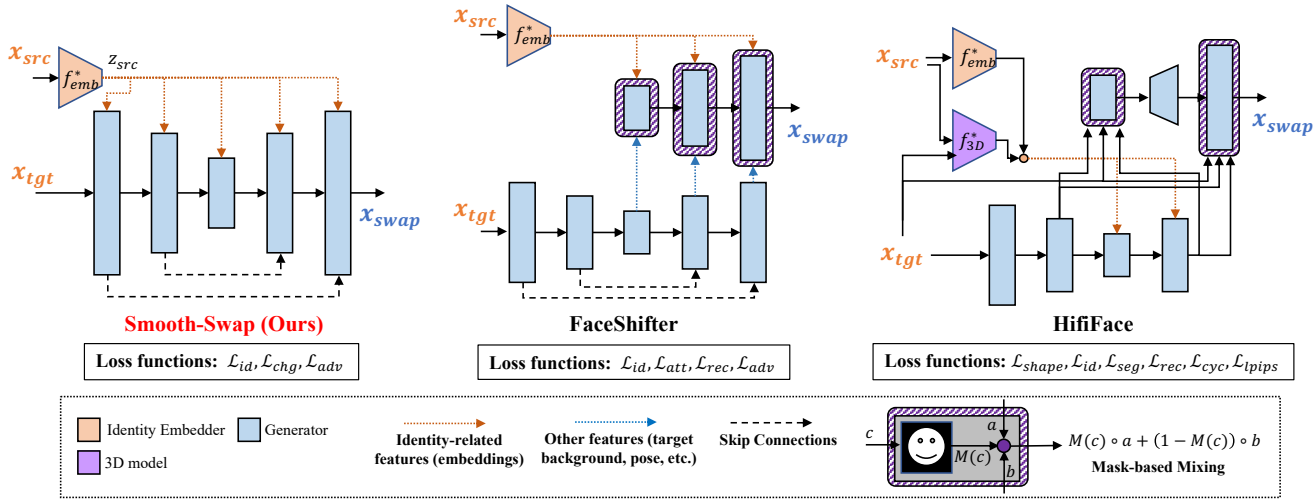

Figure 2. An illustrative comparison of the generator architectures and the loss functions of face-swapping models. Previous models (FaceShifter [16] and HifiFace [31]) have face-swapping-specific designs such as mask-based mixing (hatched in purple) or 3D face modeling (f^*_{3D}). Such designs induce complex architectures and various loss functions, which makes training difficult for balancing. On the contrary, our architecture is a simple U-Net extension excluding task-related heuristics, and trained by only three typical losses.

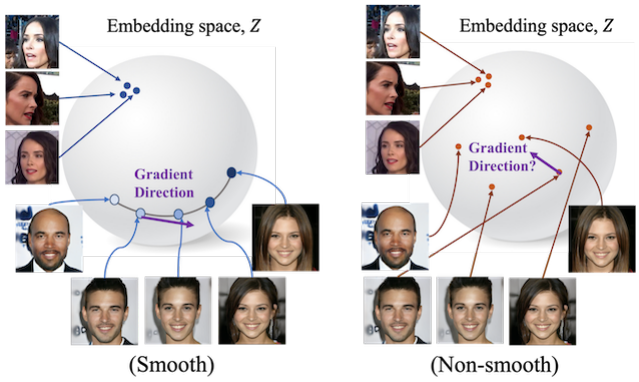

Figure 3. When identity is changed from one to another, the corresponding vector in a smooth embedding space would also change smoothly. In a non-smooth embedding space, however, the vector would make discrete jumps. The space can become non-smooth if the embedder is strongly trained on a discriminative task. In this case, the embedder cannot give a good gradient direction for the generator to change the identity correctly. See 3.3.

3.3. Importance of A Smooth Identity Embedder

Most of the previous face-swapping models use ArcFace [7] as an identity embedder (embedder for short) since it is one of the state-of-the-art face recognition models. Feeding images into the embedder and comparing features from the last layer (called embedding vectors), it provides a decent similarity metric for the person-identities of face images. Using ArcFace or any other face recognition models, we typically deal with a highly non-smooth embedding space, because these are trained only by a discriminative task.

The smoothness of the embedder, however, is crucial during the training of a face-swapping model. When a model generates x_{swap} with a wrong identity amid training, the embedder has to give a good gradient direction to correct it. This gradient has to be accurate and consistent; otherwise x_{swap} easily goes back to x_{tgt} by the loss for C2. If the embedding space is non-smooth, the gradient direction can be erroneous or noisy since gradients are only well-defined in a continuous space.

4. Method: Smooth-Swap

We explain our main model called *Smooth-Swap*. The model introduces a new identity embedder, trained using supervised contrastive learning [14] to improve the smoothness in the embedding space. It also introduces a simple U-Net style generator architecture, which is well suited to the new identity embedder.

Notations Our identity embedder takes images $x \in X$ and outputs the corresponding embedding vectors $z \in Z$ (e.g., $z_{src} := f^*_{emb}(x_{src})$). The generator takes a target image x_{tgt} and a source embedding z_{src}, and produces the swap image: $x_{swap} = f_{gen}(x_{tgt}, z_{src})$. f_{dis} takes x_{swap} and outputs a scalar ranging $[0, 1]$ (close to 0 for fake and 1 for real).

4.1. Smooth Identity Embedder

As discussed in Sec. 3.3, we desire a smooth embedder for stable and effective training. To train such an embedder,

we consult a supervised contrastive learning loss [14]:

$$\mathcal{L}(f_{emb}) = \mathop{\mathbb{E}}_{(x_i, x_p^i, x_n^i)} \left[-\log \frac{e^{(\langle z_i, z_p \rangle / \tau)}}{e^{(\langle z_i, z_p \rangle / \tau)} + \sum_n e^{(\langle z_i, z_n \rangle / \tau)}} \right]$$

where x_i denotes a sample from the training dataset; x_p^i and x_n^i denote positive (images having the same identity as x_i) and negative (having a different identity) samples, respectively.

An important property of contrastive learning is that it makes the embedding vectors keep the maximal information [30], and this is closely related to our need of a smooth embedder. If we have face images of the same identity but of a different age or of a different face shape (e.g., from a diet), discriminative embedders like ArcFace [7] remove this information aggressively to align the embedding vectors. While this is beneficial for classifying identities, it incurs a non-smooth embedding space. When changing the identity from elderly to young or from a round shape to sharp in this space, the embedding vectors cannot change smoothly as such information is removed. For our purpose, more desired are the embeddings with richer information—even if the alignment is compromised—as can be obtained from the contrastive learning. Then, changing from one identity to another is a smooth path and a good gradient direction can be obtained for training the swapping model (see Fig. 3).

4.2. Generator Architecture

Our generator architecture is an adaptation from the noise conditional score network (NCSN++), which is one of the state-of-the-art architectures in score-based generative modeling [27] (Fig. 2). While the original usage of NCSN++ is far different from face-swapping, we find its U-Net nature [24] and conditioning structure is useful for our task. We modify two parts from NCSN++; the time embedding is replaced with the identity embedding and a direct skip connection from the input to the output is added.

Details on Structure NCSN++ is basically a U-Net [24] with a conditioning structure and modern layer designs such as residual and attention blocks. Its original goal is to take a noisy image and output a score vector having the same dimensionality as the image. Since it has to output a vector conditioned on varying noise levels controlled by time, it also takes a time embedding vector that is added to each residual block after being broadcasted over the width and height dimensions. In our design, we replace this embedding vector with identity embedding, as illustrated in Fig. 2. Also, since the score vector is close to a difference between images rather than an image itself, we add the input image when making the final output image, instead of directly passing the output (i.e., an input-to-output skip connection).

Note our architecture does not include any task-specific design components such as a 3D face model or mask-based mixing from the previous work. It is universal and mostly compatible with score modeling by design.

Loss Functions To train this generator, we use three most basic loss functions, each corresponding to the conditions for x_{swap} described at the beginning of Sec. 3.

$$\begin{cases} \mathcal{L}_{id} &= 1 - \cos(z_{swap}, z_{src}) \\ \mathcal{L}_{chg} &= \|x_{swap} - x_{tgt}\|_2^2 / D \\ \mathcal{L}_{adv} &= -\log(f_{dis}(x_{swap})) \end{cases}$$

The total loss is computed by combining these functions and taking the expectation over (x_{tgt}, x_{src}) pairs:

$$\mathcal{L}(f_{gen}) = \mathbb{E}_{(x_{tgt}, x_{src})}[\lambda_{id}\mathcal{L}_{id} + \lambda_{chg}\mathcal{L}_{chg} + \lambda_{adv}\mathcal{L}_{adv}].$$

Note that $\cos(\cdot, \cdot)$ stands for cosine similarity and D stands for the number of dimensions of X; f_{dis} is trained with the original loss from [9] and R1 regularizer [17]. The loss functions are generally the same as [16], except we use a simpler pixel-level change loss instead of the feature-level loss (denoted as attribute loss in the paper). For each mini-batch, we include one (x_{tgt}, x_{tgt}) pair, whose change loss effectively acts as a reconstruction loss.

5. Experiments

5.1. Training Details

Datasets For training the generator, we use FFHQ dataset [12], which contains 70k aligned face images. We use the 10% of images for testing. For training the identity embedder, we use the VGGFace2 dataset [5], which contains 3.3M identity-labeled images of 9k subjects. We crop and align VGGFace2 images using the same procedure as FFHQ. All images including FFHQ are resized to 256×256 scale.

Architecture Details Our identity embedder is based on ResNet50 [10] architecture. The final, average-pooled feature vector is passed through two fully-connected layers and normalized to unit length. The generator architecture is mostly the same as NCSN++ [27], except we use half as many channels. The discriminator is set to the same as StyleGAN2 [13]. The detailed structure of the networks is included in the appendix.

Training We set $\lambda_{id} = 4$, $\lambda_{chg} = 1$, and $\lambda_{adv} = 1$ for training. The discriminator is trained with the non-saturating loss [9] along with the R1 regularizer [17] to prevent the overfitting. Adam optimizer [15] is used for training with learning rates 0.001 (generator) and 0.004 (discriminator). It is run for 800k steps with batch-size eight,

Model	VGG↓	VGG-R↓	Arc↑	Arc-R↑	Shp↓	Shp-R↓	Expr↓	Expr-R↓	Pose↓	Pose-R↓	PoseHN↓	Overall↓
Deepfakes	120.907	0.493	0.443	0.524	0.639	**0.464**	0.802	0.541	0.188	0.445	4.588	0.927
FaceShifter	110.875	0.482	†	†	0.658	0.492	**0.653**	0.456	**0.177**	**0.381**	**3.175**	-0.202
SimSwap	**99.736**	**0.435**	†	†	0.662	0.479	**0.644**	**0.449**	0.178	**0.385**	3.749	**-0.558**
HifiFace	106.655	0.469	**0.527**	**0.550**	**0.616**	0.465	0.702	0.484	**0.177**	0.387	**3.370**	-0.329
MegaFS	110.897	0.461	†	†	0.701	0.500	0.678	**0.436**	0.182	0.398	5.456	0.234
Smooth-Swap	**101.678**	**0.435**	**0.464**	**0.611**	**0.565**	**0.403**	0.722	0.477	0.186	0.395	4.498	**-0.617**
50% steps	101.905	0.430	-	-	0.578	0.404	0.726	0.476	0.186	0.399	5.979	-0.398
$\lambda_{id} = 1$	107.096	0.446	0.421	0.581	0.610	0.415	0.669	0.461	0.185	0.398	4.636	-0.419
(Arc) $\lambda_{id} = 1$	103.767	0.437	†	†	0.682	0.460	0.728	0.493	0.192	0.416	5.457	0.266
(Arc) $\lambda_{id} = 4$	98.115	0.421	†	†	0.684	0.441	0.914	0.543	0.207	0.430	5.655	0.699

Shp: shape, **Expr**: expression, **PoseHN**: pose metric with Hopenet [26], (Arc): trained using ArcFace, †: scores cannot be compared because the model uses ArcFace in training.

Table 1. Quantitative comparison between the models (see Sec. 5.2 and Sec. 5.3 for the details). The arrow ↓ (or ↑) denotes that the score is the lower (or the higher) the better; the best two are marked as bold. The vertical line in the middle divides the scores into two groups: ones related to the identity change (left) and ones related to keeping the target attributes (right). The overall score is the average of each score after standardization (Arc and Arc-R are excluded as some models are ineligible). The last four rows are ablation models (Sec. 5.4).

where the number matches with the total number of images shown to HifiFace. As described in Sec. 4.2, one pair in the batch is set to (x_{tgt}, x_{tgt}) for considering the self-reconstruction case. Adam is also used for training the embedder (prior to training the swapping model), where the learning rate is set to 0.001 and decreased by a factor of 10 at 60, 75, and 90% during the total 101K steps. The batch size is 128 (32 identities, four instances per each) and the temperature τ is 0.07 as suggested in [14].

5.2. Evaluation Details

Compared Models We compare our Smooth-Swap model with the latest feature-based face-swapping models: FaceShifter [16], MegaFS [33], HifiFace [31], SimSwap [6], and Neural Textures [28]. We also compare two of the earliest models: Deepfakes [1] and Faceswap [2].

Quantitative Evaluations Since the most of the compared models do not open their source code to the public, the current standard for evaluating the models is to compare their generated images[1] on the FaceForensics++ (FF++) datasets [25], and we follow accordingly.

We evaluate various metrics that can be grouped into the following: identity, shape, expression, and pose. We want x_{swap} to be close to x_{src} for the first two and close to x_{tgt} for the other two. To evaluate identity, we use VGGFace2 [5] and ArcFace [7] embedders and compute the embedding distance and cosine similarity, respectively, between x_{swap} and x_{src}. Compared with the retrieval accuracy used in [6, 16, 31], which classifies x_{swap} among fixed candidates, this metric allows more fine-grained comparison. To evaluate shape, expression, and pose, we follow the evaluation protocol of [31]; i.e., we use a 3D face model of [26] to get the parameters of each class and compute the L2 distances.

When applicable, we compute relative distances and similarities (denoted by '-R') as well. For example,

$$\text{dist-R} := \frac{\text{dist}(x_{swap}, x_{src})}{\text{dist}(x_{swap}, x_{src}) + \text{dist}(x_{swap}, x_{tgt})},$$

is computed for VGGFace2 embedding distance[2]. This is to reflect how humans perceive the changes; to our eyes, important is not only the identity of x_{swap} being close to x_{src} but also its being far from x_{tgt}.

5.3. Basic Face-Swapping Performance

We apply face-swapping on the FaceForensics++ dataset and compare the results with other models in Fig. 4. The figure shows that our Smooth-Swap model is more aggressive in changing identity, especially in face shape. For example, in the second and the fourth row, our swapped images show more round and grown chin shapes reflecting the characteristics of the source identity (more extreme cases can be found in Fig. 5); the images from the other models are mostly confined to textural change. Also, we can observe other identity-related attributes, such as skin tones or hair colors, are matched more to the source in our results, making the overall figure visually more close to the source. Fig. 5 and 6 show the swapping results on FFHQ and face images in the wild (see appendix for more samples and discussion on the failure cases).

The same trend can be seen from the quantitative results summarized in Table 1. In the table, Smooth-Swap shows good identity and shape scores (VGG, Arc, and Shp). While it is not as good in the other scores, it at least shows comparable numbers (not the worst at all times). Considering that a bypass model (not changing the identity at

[1] Available on https://github.com/ondyari/FaceForensics; some are on the project page of each model.

[2] For pose and expressions, numerator is changed to (x_{swap}, x_{tgt})

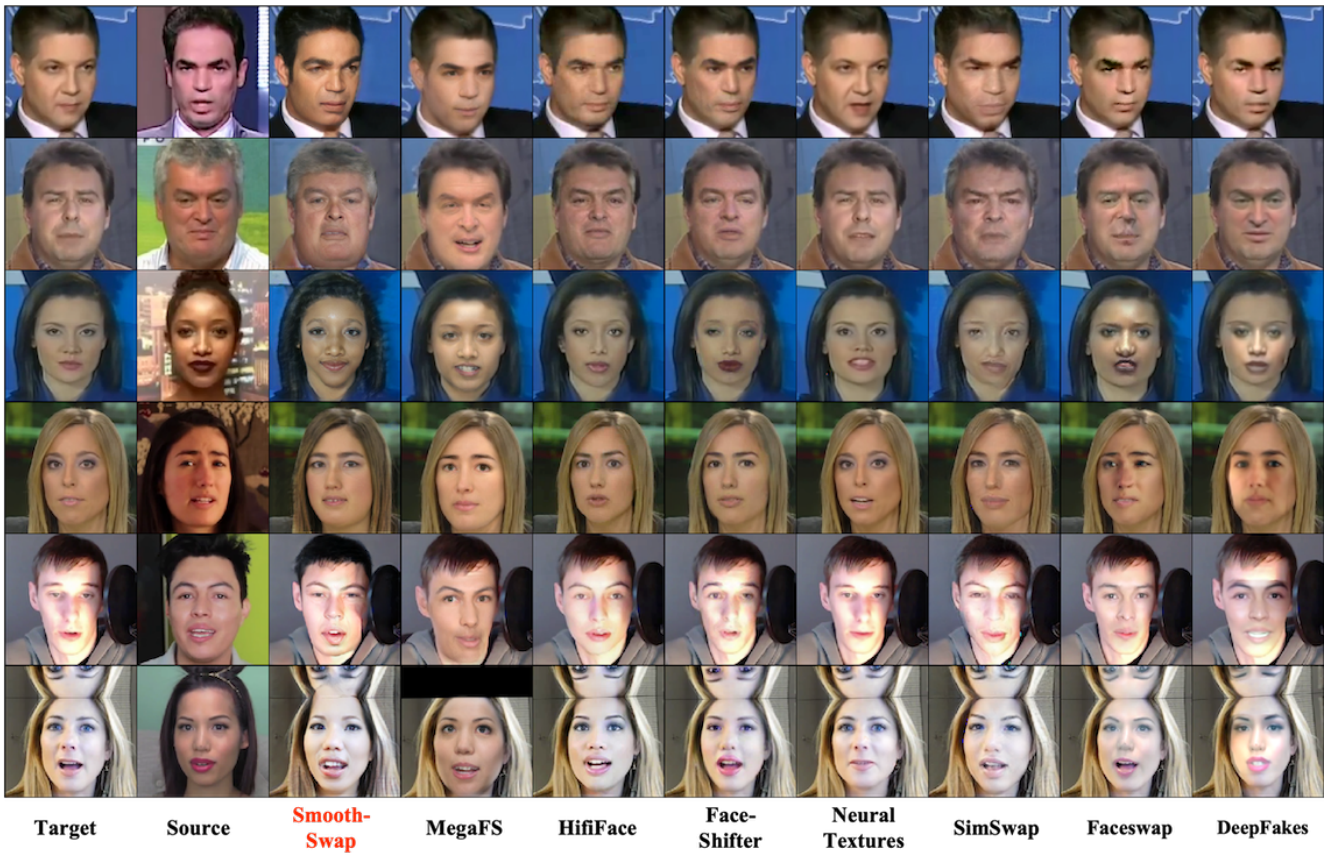

Figure 4. Comparison of the face-swapping results of various models on the FaceForensics++ dataset [25]. The results from our models show the most active identity and shape change, reflecting the characteristics of the source identities. Note there are minor frame differences among the results as the images are extracted from videos.

all) would achieve the best score in expression and pose scores, we emphasize that the overall competence is important here. Thus, we report an overall score in the last column—average of the metrics after standardizing each—where our model marks the best.

5.4. Ablation Study on the Identity Embedder

To see how our identity embedder makes a difference, we train our generator using ArcFace [7] as well. As seen from the lower part of Table 1, the models using ArcFace perform worse in most of the metrics.

More importantly, we observe that our embedder enables faster and stable training. In Fig. 7, the left graph shows that the identity loss of our model converges faster compared with the one using ArcFace. Note this is not due to the scales or the choice of λ_{id}, since Arc16, which has a similar rate of identity-loss drop, shows a significantly worse curve for the change loss.

The same trend can be seen in Fig. 8. When paired with ArcFace embedder, the models show slow training, rarely changing identity until 400k training steps. In contrast, the models with our embedder begins to change the identity as early as 100k steps, and the overall score at 400k steps (50% training) is already better than HifiFace (Table 1).

5.5. Identity Embedding Performance

The advantage expected from our identity embedder is the smoothness; in particular, smooth change of identities along the interpolation curve as shown in Fig. 3. To quantitatively evaluate this, we devised a smoothness score and compared with other baseline embedders.

$$d_{smooth} := E_{x_A, x_B \sim p(x)} \left[\frac{\|Slerp(z_A, z_B; r) - z_C\|}{\|z_A - z_B\|} \right].$$

The score measures the (normalized) gap between the average point of the two identity embedding vectors, $Slerp(z_A, z_B; r)$, and the closest valid embedding to it, z_C (here, r is an averaging ratio). If the embedding space is smooth, this gap has to be small.

The notion of valid embedding is subject to the settings. When measured using samples, x_A and x_B are samples

Figure 5. The results of Smooth-Swap on the FFHQ test split (uncurated). An active change of identity (e.g., row-1, column-2) is observed, but some artifacts can be also found when the source identity has a complicated hair pattern (column-1).

Figure 6. Face swapping results of Smooth-Swap on wild images. More samples are included in appendix.

from the FFHQ dataset D_{test}, and $z_C = f_{emb}(x_C)$ where $x_C = \arg\min_{x \in D_{train}} \|Slerp - f_{emb}(x)\|$. When measured using GAN, $x_A = g(y_A)$ and $x_B = g(y_B)$ are samples generated from a pretrained StyleGAN2 [13], where $z_C = f_{emb}(g(Lerp(y_A, y_B; r)))$ (g is the generator, and y's are the latent codes).

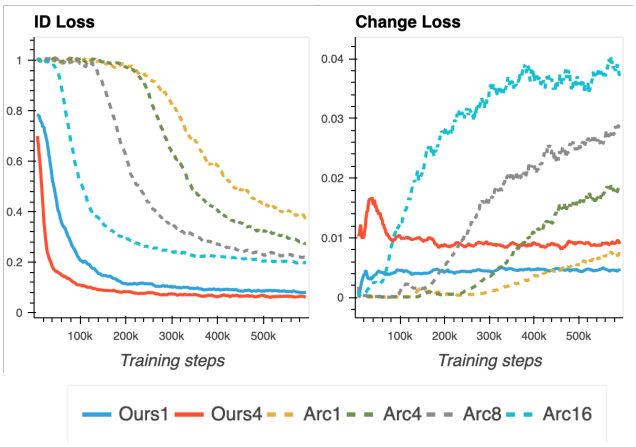

Figure 7. Ablation study of identity embedding model—Ours (solid) versus ArcFace (dashed) [7]. The number next to the model name indicates the identity-loss weight, λ_{id}, used for training. It can be seen that the model learns to change identity much faster with our embedder while being stable in the change loss. See Sec. 5.4 for the discussion.

	d_{smooth} w/smp	w/GAN	Verification AUC		
	r=0.25	r=0.5	r=0.5	VCHQ	VGG2	LFW
CE-Lin	0.333	0.354	0.797	0.939	0.994	1.000
CE-Arc	0.404	0.430	0.914	0.925	0.997	0.998
ArcFace	0.360	0.380	0.802	-	-	0.995
Ours	**0.116**	**0.135**	**0.671**	0.956	0.994	0.999

Table 2. Scores of the embedder models. Our model shows far better smoothness scores, maintaining comparable verification scores. CE-Lin and CE-Arc are reproduced versions of VGGFace2 [5] and ArcFace [7], trained from FFHQ-aligned VGGFace2 dataset. ArcFace is the original model provided in [7], trained from a larger dataset with a different alignment.

As seen from Table 2, our model shows substantially better smoothness while maintaining comparable verification performance with ArcFace and VGGFace2. Note LFW [11] is one of the standard benchmark dataset for verification; VCHQ is a dataset we derived from VoxCeleb [18] (see appendix for the details).

The same trend is also qualitatively confirmed in Fig. 9. The figure shows the retrieved x_C images for each of the interpolating points ($r \in [0.1, \cdots, 0.9]$). Our embedder tends to change smoothly while moving along the interpolation curve; others tend to stick with the same identities repeatedly. To quantify this, we compute the number of unique images for each interpolation (the lower, the more repetition, and the worse). Summarizing the results from 64 sample pairs, the numbers were 5.13±1.18 (Ours), 4.42±1.41 (VGGFace2), and 4.25±1.33 (ArcFace).

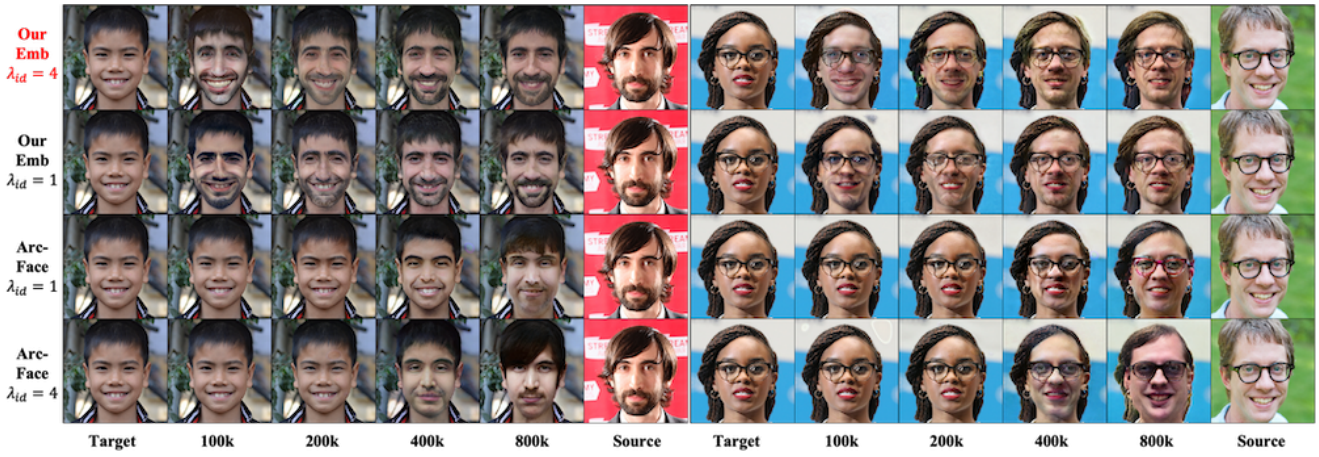

Figure 8. The progression of model training with different identity embedding models and loss weighting (λ_{id}); the generator architecture is fixed to ours. The models with ArcFace embedder [7] shows slow training, making little identity change until being trained for 400k steps. On the other hand, the models with our embedder show identity change at as early as 100k steps. See Sec. 5.4.

Figure 9. Inspection of the smoothness of embedders via interpolation. For two randomly picked images from the FFHQ test split (the leftmost and the rightmost), we compute the interpolations in the embedding space. For each of the nine interpolating points, we retrieve the closest images (compared in the embedding space) from the train split. Our embedder tends to show continuously changing identities, whereas others show repeating identities, implying non-smoothness of the space. The graph on the right shows our embedder distributes the identities more uniformly. The distances are normalized by the average of 4k random pairs for each embedder. See Sec. 5.5.

6. Conclusion

We introduced Smooth-Swap, a new face-swapping model generating high-quality swap images with active change of face shape. While many existing models use handcrafted components to tackle the difficulty, our model stays with the simplest architecture and considers smooth identity embedding instead. By taking this data-driven approach with minimal inductive bias, we observed that Smooth-Swap can achieve the best overall scores with fast convergence.

We believe this study can open up opportunity for tackling more challenging face-swapping problems by reducing the complexity considerably. With reduced effort for balancing the components and reduced memory usage, one could consider an expanded problem scope, such as modeling face-swapping on videos in an end-to-end manner. A downside of our current model in that regard is some performance drop in preserving the pose and expression. However, we suppose a simple fine-tuning or different hyperparameter choice would be sufficient to meet the goal.

Potential Negative Societal Impact Face-swapping models, known as Deepfake to the public, have been maliciously used in making serious negative impacts (e.g., spread of fake news). Nonetheless, we believe studying on these models is important and necessary because deep understanding on them could set a good starting point for developing high-quality Deepfake detection algorithms [25]. We remark that they also have positive applications, including anonymization for privacy protection and creating new characters without heavy CGI techniques.

Acknowledgement This work was partly supported by IITP (2015-0-00310/20%, 2018-0-00622/15%, 2019-0-01371/20%, 2021-0-02068/15%, 2021-0-01343/15%) grants, and CARAI (UD190031RD/15%) grants.

References

[1] DeepFakes (https://github.com/deepfakes/faceswap), Nov. 2021. 5

[2] FaceSwap (https://github.com/MarekKowalski/FaceSwap), Nov. 2021. 5

[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, Salt Lake City, UT, USA, June 2018. IEEE. 1, 2

[4] 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*, SIGGRAPH '99, pages 187–194, USA, July 1999. ACM Press/Addison-Wesley Publishing Co. 2

[5] 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. 4, 5, 7

[6] Renwang Chen, Xuanhong Chen, B. Ni, and Yanhao Ge. SimSwap: An Efficient Framework For High Fidelity Face Swapping. *ACM Multimedia*, 2020. 1, 2, 5

[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*, Feb. 2019. 2, 3, 4, 5, 6, 7, 8

[8] Gege Gao, Huaibo Huang, Chaoyou Fu, Zhaoyang Li, and Ran He. Information Bottleneck Disentanglement for Identity Swapping. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3403–3412, Nashville, TN, USA, June 2021. IEEE. 2

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, 27, 2014. 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. 4

[11] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, Oct. 2007. 7

[12] 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 (CVPR)*, June 2019. 4

[13] 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. StyleGAN2. 2, 4, 7

[14] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 2, 3, 4, 5

[15] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*, 2015. 4

[16] 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. 1, 2, 3, 4, 5

[17] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which Training Methods for GANs do actually Converge? In *International Conference on Machine Learning*, pages 3481–3490. PMLR, 2018. 4

[18] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*, 2019. 7

[19] Jacek Naruniec, Leonhard Helminger, Christopher Schroers, and Romann M Weber. High-Resolution Neural Face Swapping for Visual Effects. In *Computer Graphics Forum*, volume 39, pages 173–184. Wiley Online Library, 2020. 2

[20] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. FSNet: An Identity-Aware Generative Model for Image-based Face Swapping. In *Asian Conference on Computer Vision*, pages 117–132. Springer, 2018. 2

[21] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. RSGAN: Face Swapping and Editing using Face and Hair Representation in Latent Spaces. In *ACM SIGGRAPH 2018 Posters*, SIGGRAPH '18, New York, NY, USA, 2018. Association for Computing Machinery. 1, 2

[22] 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. 1, 2

[23] 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, Xi'an, May 2018. IEEE. 2

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 2, 4

[25] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. FaceForensics++: Learning to Detect Manipulated Facial Images. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–11, Seoul, Korea (South), Oct. 2019. IEEE. 5, 6, 8

[26] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Learning to Regress 3D Face Shape and Expression from an Image without 3D Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7763–7772, 2019. 5

[27] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021. 4

[28] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred Neural Rendering: Image Synthesis using Neural Textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. Comment: Video: https://youtu.be/z-pVip6WeyY SIGGRAPH 2019. 1, 5

[29] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016. 2

[30] Tongzhou Wang and Phillip Isola. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020. 2, 4

[31] 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, Aug. 2021. 1, 2, 3, 5

[32] 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. 2

[33] 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*, pages 4834–4844, 2021. 1, 2, 5