Paste You Into Game: Towards Expression and Identity Consistency Face Swapping

Hao Zeng, Wei Zhang, Keyu Chen, Zhimeng Zhang, Lincheng Li and Yu Ding
Virtual Human Group, Netease Fuxi AI Lab, Hangzhou, China
{zenghao03, zhangwei05, chenkeyu02, zhangzhimeng, lilincheng, dingyu01}@corp.netease.com

***Abstract*—Customizing game characters for individual players has been a long-standing attractive feature in the game industry. However, traditional solutions like manual editing within a game engine are always time-consuming and unsatisfying. Our work proposes a novel automatic face swapping method for arbitrary users and game characters, addressing three challenges including style gap between human and game faces, identity preservation, and expression consistency. A game face dataset is collected to handle the cross-style gap; an identity compound embedding is proposed to ease the bias existing in the commonly-used ID identifiers and it provides a more robust identity representation; a novel expression embedding loss is proposed to enforce the expression consistency between the swapped and target faces and it achieves better expression consistency than the previous methods, especially when the expression is very subtle. The visualized results, as well as the qualitative and quantitative comparisons, reveal the significance and effectiveness of our proposed solutions.**

***Index Terms*—Game CG, Face Swapping, Identity, Expression, Image Synthesizing**

I. Introduction

Game CG videos or cutscenes are animations or pictures related to scene characters or plots in the game produced with the support of computer graphics (CG) technology. Game CG videos are very important for game promotion. To achieve personalized promotion, we propose to use face swapping to generate customized identity-specific CG videos. Given a template video (Target), we can replace the face in the video with a human face (Source) to obtain a personalized video (Result) and the resulting face is similar in appearance (identity) to the source face but maintains the attributes (e.g. face color, lighting) and facial expression of the target face. However, there are still several challenges that make it difficult to directly apply existing face swapping methods to game characters. First, the game character face swapping needs to be identity-agnostic, while several works [1], [2] are identity-specific. On the other hand, existing identity-agnostic methods [3], [4], [5] only focus on human faces, they cannot address the human-to-character cross-style problem. Second, previous works rely on a supervisor of identity embedding to perform the identity swapping. The identity embedding is provided with a single pre-trained human-based face identifier, but an identifier cannot provide sufficient identity constraint because of the inherent feature bias existing in an identifier, especially, when applied to game character face. Additionally, the existing methods without fine-grained expression constraints will cause inconsistent expressions in generated videos.

Considering the above problems, we propose a new face swapping method and make efforts in three aspects to better generalize the existing methods to the game character faces: solving the cross-domain problem, preserving the identity consistency (with the source face), and the expression consistency (with the target face).

To resolve the cross-style problem, we first train our face swapping model on human data and then fine-tune the model on the game face dataset we collected. Since there exist few game face datasets as large as real human face datasets, training directly on the game dataset will greatly reduce the robustness and generalization of the model, and fine-tuning can make full use of the knowledge learned by the model on human dataset and does not require a large amount of game data.

In the aspect of identity, a trivial solution to ensure the consistency between source and swapped images is to adopt face recognition models [6], [7], [8]. Nevertheless, the single extracted identity embedding lacks robustness and the face identity embedding is easily affected by the variances of expressions and other facial attributes, which is more serious in the game face domain. Therefore, we propose to use an identity compound embedding which is a fusion of several different embeddings extracted by different face recognition models, aiming to promote the stability of the identity embedding for better identity consistency.

In the aspect of the expression, the previous methods either use the landmarks or the implicit attribute constraints to ensure the consistency of expressions, but their methods may generate some inconsistent expressions, as shown in the Figure 3 and Figure 4. The reason can be that the current expression representation approaches are not capable of capturing subtle facial movements and complicated expression. To improve and enforce expression consistency, our work adopts a novel expression embedding technique [9] representing the facial expressions in a continuous space. To the best of our knowledge, this is the first face swapping method that addresses the expression consistency issue and considers expression similarity measurement.

In general, as illustrated in Figure 1, our face swapping framework is built upon a generative adversarial network (GAN). It is first trained on human face dataset and then fine-tuned on a game character dataset that we collect. Apart from

the encoder-decoder structure and skip connection design, we further extract identity embeddings from two face recognition models [7], [8] and allow the decoder to refine the robust identity information from two identity embeddings. Furthermore, we design an expression embedding loss for constraining the swapped face image and the target image within a continuous expression space. In summary, our main contributions are as follows:

- Our work proposes a new method for game character face swapping. By collecting a game character face dataset and applying a fine-tuning strategy, we manage to translate the face swapping model from the real human face to the game face.
- Our work proposes an identity compound embedding to improve the identity consistency between the source and the swapped face images, while preserving the subject's attributes to be harmonic with the game character.
- Our work introduces a new expression consistency metric for the face swapping task. By our designed expression embedding loss, it can enforce the generated faces to keep expression similarities with the target faces as close as possible.
- Our work conducts full experiments on both human and game character faces. The quantitative and qualitative experiments demonstrate that our results outperform the previous face swapping methods in terms of cross-style translation capability, identity consistency and expression similarities.

II. Related Work

This section reviews some literature in face swapping and expression representation areas which are closely related to our research topic.

Previous works on face swapping can be divided into three categories: pixel-based methods, 3DMM-based methods, GAN-based methods. The most straightforward solution is to replace the inner face part in pixel space [10]–[12]. However, the manipulated image patches usually suffer from attribute mismatch. 3DMM-based methods [13]–[15] generate the face region by 3D fitting and then the source faces and target backgrounds are blended via inverse rendering. More recently, there occurs many GAN-based methods [1]–[5], [16]–[23]. Specifically, the most popular methods like Deepfakes [1] and its variants [2], [16] need to be trained pairwise. FSGAN [19] first animates the source face by reenactment and then blend it into the background with an in-painting and blending network. FaceShifter [3] generates a swapped face with high-fidelity and handles the occlusions with a second-stage refinement network. SimSwap [4] proposes a weak feature matching loss to improve the facial attributes consistency. FaceController [5] proposes a unified framework for identity swapping and attribute editing and is the first work that uses 3D parameters and the identity embedding to represent facial identity. Later, HifiFace [21] solves shape inconsistency problem in face swapping by 3D shape-aware identity to control the face shape with geometric supervision. MegaFace [22] proposes the first megapixel level method and achieves 1024×1024 face swapping. The methods mentioned above are all based on human data so they can't be applied to the game data directly.

Most methods [3]–[5], [21], [22] transfer identity with a single identity embedding which may be affected by other facial attributes such as facial expressions. In addition, implicit attribute constraints [3], [4] or facial landmark loss [5], [21], [22] are not able to capture subtle expressions, causing the problem of expression consistency.

A. Expression Representation.

Facial expression plays a vital role in human social communication. However, due to its complicated natural and subtle movement, it is non-trivial to represent the accurate expressions of human faces, and thus prevent the downstream tasks such as face editing and manipulation. The commonly-used categorical expressions are still a block to many fine-grained expression-related applications [9], [24]–[27], being inadequate to characterize all the facial expressions and distinguish those facial expressions labeled in the same category. In some facial expression translation tasks [28], [29], facial action units (AUs) are also used as a representation of facial expressions. But the low accuracy of AU detection and AU intensity estimation makes it difficult to represent expressions accurately. Some talking face generation tasks [30]–[33] have no explicit supervision of facial expressions since ground truth images are available in their tasks and the expression transfer is achieved implictly (eg. pixel-level reconstruction loss). Since pixel-level ground truth images in our task is not available, we can only model and supervise expressions explicitly.

In this work, we turn to address the expression consistency manner into face swapping framework and adopt a novel expression representation [34] which extracts a continuous space based on expression similarities.

III. Method

This section introduces the proposed method including the framework overview, the game faces collection, the identity transfer and the expression consistency. In the end, we describe the loss functions we used during training.

A. Framework Overview

The framework is shown in Figure 1. Given one source face image I_s and one target face image I_t, it performs face swapping and generates I_o that reflects the attribute information (expression, skin color, etc.) of I_t but the identity information of I_s. Specially, our framework contains five components: facial image encoder(E_f), facial image decoder(D_f), identity embedding module(E_{id}), expression embedding module(E_{exp}) and a multi-scale discriminator [35]. In face swapping, E_f extracts attribute-related $f_{attr} = \{f^1_{attr}, f^2_{attr}, f^3_{attr}, f^4_{attr}\}$ from a target image I_t. Next, E_{id} extracts two different identity embeddings $\{f^1_{id}, f^2_{id}\}$ from a source image I_s. Then, D_f is fed with f_{attr} and $\{f^1_{id}, f^2_{id}\}$, and render facial semantics into the swapped face I_o.

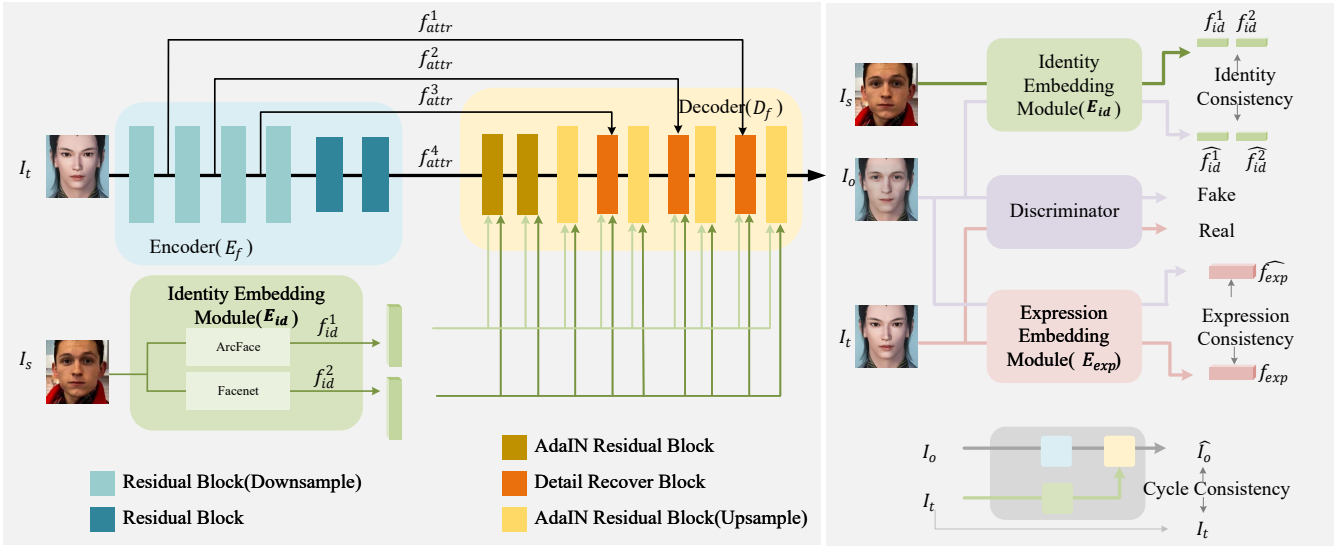


Fig. 1: Architecture of our proposed framework. The framework mainly consists of five components: facial image encoder(E_f), facial image decoder(D_f), identity embedding module(E_{id}), expression embedding module(E_{exp}) and a multi-scale discriminator [35].

B. Game Faces Collection

The game faces are collected from two sources. On one hand, we render face images using a computer graphic engine. We collect 30 3D models of game characters. Each model is driven to perform 1000 different facial expressions by traversing in its expression blendshape and pose parameters. The generated expressions are then rendered into the corresponding images separately by the Unity game engine [36]. In this way, we get about 30,000 rendered game face images with desired expression diversity.

On the other hand, since the rendered images lack identity diversity, we crawl more game images from the Internet. To reduce labor costs, we design an automated filtering method based on face detection and only the images with detected faces are retained. To improve the detection accuracy, we use two different face detection models [37], [38] for cross-validation. Finally, we obtain about 80,000 game face images with better identity diversity.

Finally, we collect a game face dataset containing 110,000 images. These images are used to finetune and evaluate our model for cross-domain face swapping.

C. Identity Transfer

To , we propose to utilize an identity compound embedding instead of using a single identity embedding to provide the identity information of I_s. The identity compound embedding are offered by two pre-trained face recognition models. They provide raw and sufficient identity information from different identity recognition models for the decoder D_f. This allows for avoiding the bias from a specific identity recognition model. Therefore, the compound strategy allows for refining identity information through D_f. Section IV-D provides more analysis and discussion about the bias of an identity embedding.

To inject the identity information into the decoder D_f, we replace the normalization layer in the original residual block [39] with the adaptive instance normalization (AdaIN) [40] and then the identity information (embedding) is mapped to two modulation vectors (γ_{id} and β_{id}) of the normalization layer in AdaIN with two fully connected layers. The difference is that the identity information in our method is from two different models, so we first compound the two identity embeddings with a multi-layer perceptron. The formulation is written as:

$$c_{id}^{i} = P^{i}([f_{id}^{1}, f_{id}^{2}]), \tag{1}$$

where P^i represents the perceptron in the i_{th} block of the decoder. Then the compound identity c_{id}^{i} is then injected into the intermediate feature map with AdaIN:

$$Z_{id}^{i} = F_{\gamma}^{i}(c_{id}^{i}) \left(\frac{Z^{i} - \mu(Z^{i})}{\sigma(Z^{i})} \right) + F_{\beta}^{i}(c_{id}^{i}), \tag{2}$$

where F_{*}^{i} are two fully connected layers in the i_{th} block of the decoder and Z^i is the input feature map of the i_{th} block.

To recover the details lost due to downsampling, we design a *Detail Recover Block* (DRB) following [3]. In the i_{th} Detail Recover Block, we first obtain the identity-injected feature Z_{id}^{i} through the injection method in the AdaIN residual block, and then the corresponding attribute feature f_{attr}^{i} is used to merge with Z_{id}^{i} adaptively with attention. The difference with [3] is that we not only use spatial attention but also channel attention. Specifically, we first inject the f_{attr}^{i} into Z^i with spatially-adaptive normalization (SPADE) [41].

$$Z^i_{attr} = T^i_\gamma(f^i_{attr}) \left(\frac{Z^i - \mu(Z^i)}{\sigma(Z^i)} \right) + T^i_\beta(f^i_{attr}), \quad (3)$$

where T^i_* are two convolutional layers used to compute modulation parameters γ_{attr} and β_{attr} of the normalization layer in SPADE. But unlike the vector form parameters in AdaIN, γ_{attr} and β_{attr} here are tensors with the same spatial dimension as Z^i.

Then, we generate the spatial attention mask M^i_s and the channel attention maskM^i_c from the original input Z^i with a convolutional block attention module (CBAM) [42].

$$(M^i_s, M^i_c) = CBAM^i(Z^i). \quad (4)$$

Finally, the attribute-injected feature Z^i_{attr} and the identity-injected feature Z^i_{id} are fused using the two attention masks:

$$\hat{Z}_i = M^i_s * M^i_c * Z^i_{id} + (1 - M^i_s * M^i_c) * Z^i_{attr}, \quad (5)$$

where $\hat{Z}_i$ is the output of the i_{th} block.

D. Expression Consistency

Some previous methods [3], [4] treat the expression as the same as other attributes and achieve consistency of expressions through an implicit constraint on attribute features, some other methods use facial landmarks [5], [19], [22] to characterize and constrain expressions. We argue that attribute features cannot obtain some subtle expressions since they contain many other attributes like pose, skin color, etc. And the facial landmark is related to the identity which may harm the identity consistency.

To avoid problems in previous methods and achieve better expression consistency, we leverage a novel expression embedding technique called DLN [34] in E_{exp} to compute the expression loss between I_o and I_t during training. The expression embedding provided by DLN can provide a fine-grained representation for facial expression and the embedding is well disentangled from identity. This allows for estimating expression similarity within a continuous compact space with no impact on identity. Therefore, expression embedding is used to enforce expression consistency.

E. Loss Function

This section details the supervision in the training, including reconstruction loss, identity loss, expression loss, and cycle consistency loss.

Reconstruction Loss: During training, we make I_s and I_t the same with a certain probability and expect the generated image I_o as same as the input. So we introduce a pixel-wise reconstruction loss following [3]. The reconstruction loss is written as

$$\mathcal{L}_{rec} = \|I_o - I_t\|_2, \quad (6)$$

where $\|*\|_2$ denotes the euclidean distance. In our experiment, we set the probability that I_s and I_t are the same as 0.25.

Identity Loss: An identity loss is usually used in face swapping tasks. The loss enforces D_f to acquire identity information from the injected identity embedding. Due to the identity compound embedding for injection, the identity loss is also based on two face recognition models (ArcFace and FaceNet):

$$\mathcal{L}_{id} = \sum_{k=1}^{K} \lambda_k (1 - cos(E_{id}(I_o), E_{id}(I_s)), \quad (7)$$

where λ_k represents the relative weight of each face recognition model and $cos(*,*)$ denotes the cosine similarity of two identity embeddings. In our experiments, we set K=2 and λ_1 = 10, λ_2=5 for ArcFace and FaceNet respectively.

Expression Loss: To make the expression of the swapped face I_o more consistent with the target face, we adopt an expression loss that penalizes the $\mathcal{L}_2$ distance of two expression embeddings.

$$\mathcal{L}_{exp} = \|E_{exp}(I_o) - E_{exp}(I_t)\|_2, \quad (8)$$

The expression loss encourages the generator to learn to acquire expression-related information from target faces other than some unrelated disturbance like identity.

Cycle Consistency Loss: In addition to expression and identity, it's also important to guarantee that the swapped face properly preserves the attributes of the target face. To do this, we introduce a cycle consistency loss [43]:

$$\mathcal{L}_{cycle} = \left\| \hat{I}_o - I_t \right\|_1, \quad (9)$$

where $\hat{I}_o$= $D_f(E_f(I_o), E_{id}(I_t))$ and $\|*\|_1$ denotes the $\mathcal{L}_1$ distance. This objective encourages the generator to learn to preserve the original attribute of I_t while only changing its identity.

GAN Loss. To make the synthesized facial images more realistic, adversarial training is used in our framework. Specifically, we adopt Hinge loss [44] as the adversarial loss, denote as L_{adv}.

Full objective: Our full objective can be summarised as:

$$\mathcal{L} = \mathcal{L}_{adv} + \mathcal{L}_{id} + \lambda_{exp}\mathcal{L}_{exp} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{cycle}\mathcal{L}_{cycle}, \quad (10)$$

where λ_{exp}, λ_{rec}, λ_{cycle} are hyperparameters for each term.

IV. Experiments

A. Datasets and Settings

We construct two datasets for our task, one is called HFDataset for the human face swapping and the other is called GFDataset for game face swapping. HFDataset is a combination of three public datasets including CelebA-HQ [45], FFHQ [46], and VGGFace2 [47]. As for GFDataset, images are collected by the method described in Section III-B. For each image in the two datasets, we aligned and cropped the face to 256×256 with a face detector [38]. To ensure high-quality training, we have deleted some images that are too blurry or small. HFDataset is used for training, then we randomly choose 10,000 images of GFDataset for evaluation and the rest are used for fine-tuning.

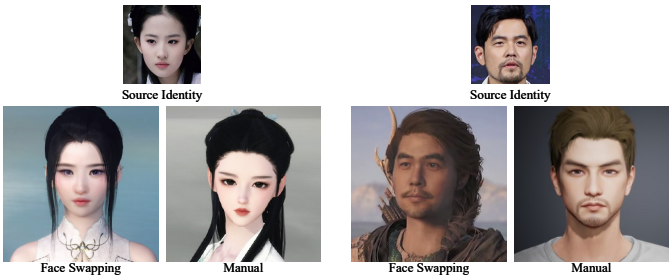


Fig. 2: Comparison of game character results generated by our face swapping method and manual method.

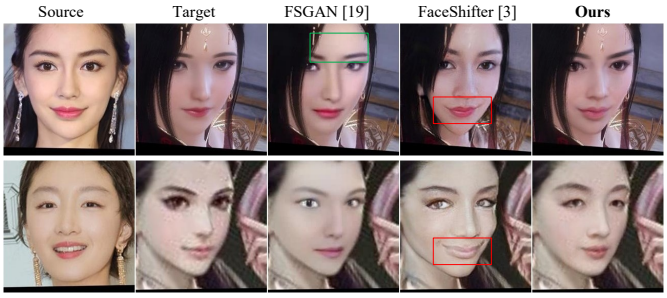


Fig. 3: Game character face swapping comparison with FSGAN and FaceShifter. Some expression errors and occlusion errors are marked with red and green boxes, respectively.

Our model is trained on HFDataset from scratch and fine-tuned on GFDataset, the framework is implemented with PyTorch [48]. We adopt Adam [49] optimizer with β_1=0 and β_2=0.999 and the learning rate is set to 0.0001. We set λ_{exp}=5, λ_{rec}=10, λ_{cycle}=10 for our full pipeline, and our model is trained first about 550K steps and then finetuned about 200K steps with a batch size of 4.

B. Comparison On Game Character Faces

To validate our method on game character faces, we conduct both qualitative and quantitative comparisons with the existing methods.

*1) **Qualitative Comparison**:* We first compare our method with the Manual method. The Manual method usually tasks a skilled game player several hours to edit the hundreds of face parameters to create a character that looks like the source face. As shown in Figure 2, our face swapping method can produce comparable results as the Manual method in less than one second.

As for face swapping methods, we compare our method with FSGAN [19] and FaceShifter [3]. We first obtain the official pre-trained model of FSGAN and then reproduce the first stage of FaceShifter. As shown in Figure 3, FSGAN suffers from unpleasant illumination and face color since FSGAN adopts a blending model to fuse the swapped face with the background and when the source and target face have huge differences in texture, lighting, or skin color (just like the difference between a game character face and a human face), such a fusion method will cause this attribute mismatch. And the swapped faces of FSGAN look less similar to the source face than our method. FaceShifter also has problems in cross-domain face swapping and the expression is affected by the source face, it can also be observed that FaceShifter without its refinement network can not handle the occlusions well but our method can even if we are a one-stage method.

TABLE I: Quantitative Comparison on Game Character Faces.

Method	ID Retrieval Accuracy (%) ↑			FID ↓	Expression Error ↓		
	CosFace	ArcFace2	SphereFace		DFER	FECNet	DLN
FSGAN	49.21	52.37	55.61	71.32	4.13	0.28	0.45
FaceShifter	95.73	98.54	97.72	66.49	4.09	0.35	0.41
Ours	**98.76**	**99.68**	**99.15**	**28.95**	**3.69**	**0.21**	**0.25**

TABLE II: Subjective Comparison Results.

Method	ID.(%)	Exp.(%)	Realism(%)
FSGAN	11.4	28.9	19.6
FaceShifter	42.5	16.4	38.2
Ours	**46.1**	**54.7**	**42.2**

*2) **Quantitative Comparison**:* We further perform the quantitative comparison with FSGAN and FaceShifter on the game character faces. We construct a test set that contains 10K human-game face pairs for human-to-game face swapping. Three types of evaluation metrics are taken into account including identity retrieval accuracy, expression error and Frechét inception distance [50].

ID retrieval accuracy is used to estimate whether the identity of the swapped face is consistent with the source face. We adopt three face recognition models including CosFace [51], [52], ArcFace2 [7], [53] and SphereFace [6], [54] for the evaluation. And identity retrieving [3] is performed in the corresponding test set.

As shown in Table I, our method obtains the highest accuracy. The validated performance in ID retrieval accuracy means that our method maintains identity consistency and guarantees robustness in identity transferring. It indicates that the use of the identity compound embedding contributes to identity consistency in face swapping.

Expression error is used to evaluate the expression distance between the swapped and the target faces. We metric this error by computing the euclidean distance between the swapped face expression embedding and the target face expression embedding. We adopt three facial expression recognition models for the evaluation including DFER [55], DLN [34] and FECNet [9]. As shown in Table I, our method obtains the lowest expression errors in three expression metrics, illustrating our superiority in expression consistency.

Frechét inception distance is used to measure the discrepancy between two sets of images. We use the final average pooling features of an pretrained Inception-V3 [56] to compute FID. As observed from Table I, our method obtains lower FID than FSGAN and FaceShifter. This proves that our method better preserves the game domain feature.

*3) **Subjective Comparison**:* To further illustrate the effectiveness of our method, we conduct a user study on our game test set (10K human-game face pairs) with FSGAN

TABLE III: Quantitative Comparison on FF++.

Method	ID Retrieval Accuracy (%) ↑			Expression Error ↓		
	CosFace	ArcFace2	SphereFace	DFER	FECNet	DLN
Deepfakes	83.70	81.79	87.18	5.02	0.56	0.73
FaceSwap	71.45	64.04	77.01	4.35	0.42	0.58
FSGAN	48.90	49.37	53.85	4.02	0.29	0.42
FaceShifter	86.83	90.77	81.37	4.03	0.36	0.49
Ours	**97.66**	**98.84**	**98.31**	**3.61**	**0.21**	**0.28**

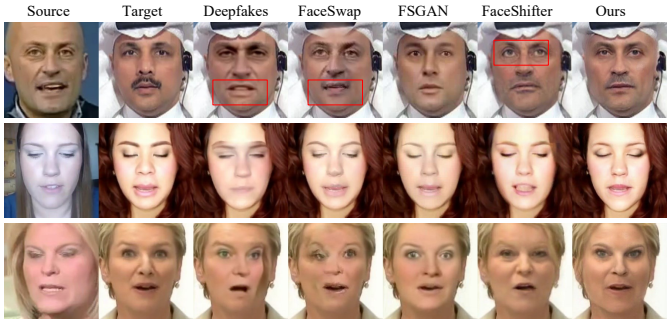


Fig. 4: Comparison with Deepfakes [1], FaceSwap [2], FSGAN [19] and FaceShifter [3] on FaceForensics++ [57]. Some expression errors are marked with red boxes.

and FaceShifter. Thirty participants are asked to complete the questionnaire in terms of identity consistency, expression consistency, or image realism. All these participants have gaming experience so that they can provide more accurate judgements. Each metric contains 30 questions and each participant need to choose the best result under each metric.

Table II demonstrates the results of the subjective comparison in the user study. Our method outperforms the baselines in terms of identity consistency, expression consistency, and image realism. These results further validate the performance of our method.

C. Comparison On Human Faces

To further validate our contributions on identity consistency and expression consistency, we conduct comparison experiments with more face swapping methods on human faces and report the comparative results as below, including qualitative comparison, quantitative comparison. For a fair comparison, models used in this section are only trained on human data without fine-tuning.

1) ***Qualitative Comparison****:* We first compare with Deepfakes [1], FaceSwap [2], FSGAN [19] and FaceShifter [3] on the FaceForensics++ (FF++) [57] dataset.

As shown in Figure 4, without any constraint on identity or attributes (expression, etc.), the results of Deepfakes and FaceSwap cannot preserve identity well and suffer a very serious mismatch in attributes (expressions, etc.). Results generated by FSGAN loss similarity with the source face and also suffer from inconsistent lighting and skin color. FaceShifter performs very well in terms of image quality and attributes consistency, but cannot preserve the target expressions well such as gaze direction .

TABLE IV: Quantitative ablation study.

Method	ID Retrieval Accuracy (%) ↑			Pose	Expression Error ↓		
	CosFace	ArcFace2	SphereFace	Error ↓	DFER	FECNet	DLN
w/o Exp	98.81	99.59	99.13	3.04	4.24	0.32	0.41
Single ArcFace	95.70	99.45	97.59	2.68	3.63	0.20	0.23
Single FaceNet	76.29	64.60	74.88	**2.46**	3.68	0.21	0.25
Ours	98.76	**99.71**	99.15	2.63	**3.52**	**0.20**	**0.23**

We further compare with FaceController [5] and SimSwap [4] by cropping images from their paper, Figure 5 (a), (b) illustrate the qualitative results, and the comparisons show that besides the comparable image quality, our method preserves the identity of the source image and the subtle expressions of the target image better. As can be seen from the red box in Figure 5, the results of the two compared methods contain some unwanted subtle expressions such as wrong gaze direction and disappearing frown.

A common problem can also be observed from the above comparisons: the swapped faces of the six baselines are affected by the expression of the source face to some extent which is sufficient proof of the point we mentioned that the face identity embedding can be easily affected by facial attribute information.

2) ***Quantitative Comparison****:* The quantitative comparisons only involve the four results-available or codes-available methods Deepfakes, FaceSwap, FSGAN, and FaceShifter. We construct the testset following [3], The quantitative comparisons rely on these five test sets. To metric the effectiveness of our proposed method in identity consistency and expression consistency, we adopt the identity retrieval accuracy and expression error as in Section IV-B2.

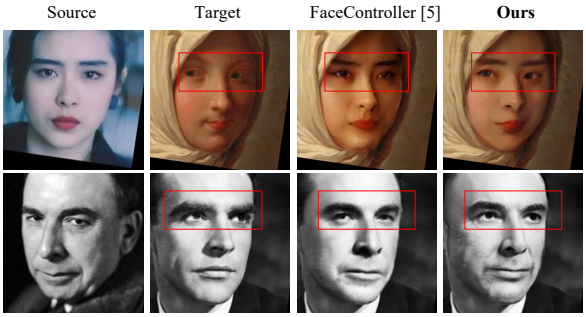
The quantitative results are shown in Table III. Similar to the results of the game character face swapping experiment, we also get the highest identity retrieval accuracy and lowest expression error for human face swapping. This means that our method with compound identity is more robust in identity transferring than single-identity-based methods (FaceShifter) and much better than those methods (Deepfakes, FaceSwap) without any identity constraint. And a fine-grained expression constraint contributes more to expression preservation than implicit constraint methods.

D. Ablation Study

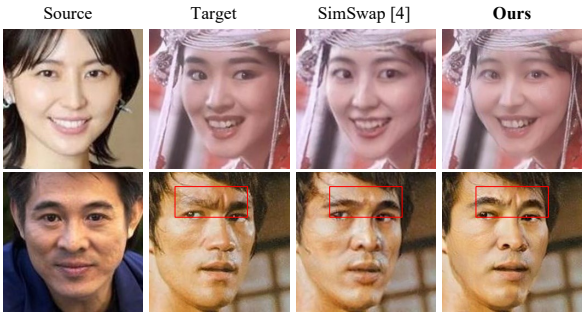
We conduct several ablation settings on the game dataset to demonstrate the effectiveness of our framework.

To verify the effectiveness of the finetune strategy, we train a model without fine-tuning (w/o FT). As shown in Figure 6, it can be observed that the finetune strategy significantly improves the quality of the generated image and FID also drops from 43.56 to 28.95. This proves that the finetune strategy contributes to cross-domain face swapping.

To demonstrate the effectiveness of our expression embedding loss, we conduct an experiment setting without the expression loss (w/o Exp). Quantitative results in Table IV and qualitative results in Figure 7 show that the expression error rises a lot without the expression loss. Observing Figure 7, the swapped faces without the expression loss tend to be



(a) Comparison with FaceController



(b) Comparison with SimSwap

Fig. 5: Comparison with FaceController and SimSwap. These illustrated images are cropped from their published paper. As observed, our method preserves the identity of the source image and the subtle expressions of the target image better than the two methods. Some expression errors are marked with red boxes.

Fig. 6: Ablation results for the finetune strategy.

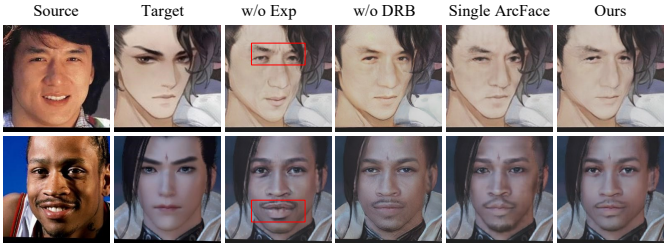


Fig. 7: Ablation study for each component in our framework. Some expression errors and occlusion errors are marked with red and green boxes, respectively. Please zoom in for more details. Please zoom in for more details.

influenced by the expression of the source face (marked in red boxes).

To evaluate the effectiveness of the identity compound embedding, we train another two models called *Single ArcFace* and *Single FaceNet*. As shown in Table IV, the identity compound embedding outperforms the single identity embedding. This validates that compound identity embeddings can alleviate the effect of expression leaked in identity embedding and provide more robust identity information.

V. Conclusion

This work proposes a novel automatic face swapping method for game character face swapping, allowing game players or developers to generate customized identity-specific game CG videos or cutscenes. We mainly focus on three challenges including the style gap between human and game faces, identity preservation, and expression consistency. Specifically, a game face dataset is collected to handle the cross-style gap; an identity compound embedding is proposed to ease the bias existing in the commonly-used ID identifiers and provides a more robust identity representation; a novel expression embedding loss is proposed to enforce the expression consistency between the swapped and target faces. Qualitative, quantitative experiments on both human data and game data show that the proposed method is well adapted to the problem of cross-domain face swapping and outperforms the state-of-the-art methods.

References

[1] DeepFakes, "Deepfakes," https://github.com/deepfakes/faceswap, 2019, Online; Accessed March 1, 2021.

[2] M. MarekKowalski, https://github.com/MarekKowalski/FaceSwap, 2021, accessed March 1, 2021.

[3] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Faceshifter: Towards high fidelity and occlusion aware face swapping," *arXiv preprint arXiv:1912.13457*, 2019.

[4] R. Chen, X. Chen, B. Ni, and Y. Ge, "Simswap: An efficient framework for high fidelity face swapping," in *ACMMM*, 2020, pp. 2003–2011.

[5] Z. Xu, X. Yu, Z. Hong, Z. Zhu, J. Han, J. Liu, E. Ding, and X. Bai, "Facecontroller: Controllable attribute editing for face in the wild," *arXiv preprint arXiv:2102.11464*, 2021.

[6] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017, pp. 212–220.

[7] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019, pp. 4690–4699.

[8] T. Esler, https://github.com/timesler/facenet-pytorch, 2021, accessed March 1, 2021.

[9] R. Vemulapalli and A. Agarwala, "A compact embedding for facial expression similarity," in *CVPR*, 2019, pp. 5683–5692.

[10] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar, "Face swapping: automatically replacing faces in photographs," in *ACM SIGGRAPH*, 2008, pp. 1–8.

[11] Y. Lin, Q. Lin, F. Tang, and S. Wang, "Face replacement with large-pose differences," in *ACMMM*, 2012, pp. 1249–1250.

[12] D. Chen, Q. Chen, J. Wu, X. Yu, and T. Jia, "Face swapping: realistic image synthesis based on facial landmarks alignment," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[13] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel, "Exchanging faces in images," in *Computer Graphics Forum*, vol. 23, no. 3. Wiley Online Library, 2004, pp. 669–676.

[14] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *CVPR*, 2016, pp. 2387–2395.

[15] Y. Nirkin, I. Masi, A. T. Tuan, T. Hassner, and G. Medioni, "On face segmentation, face swapping, and face perception," in *FG*. IEEE, 2018, pp. 98–105.

[16] I. Petrov, D. Gao, N. Chervoniy, K. Liu, S. Marangonda, C. Umé, J. Jiang, L. RP, S. Zhang, P. Wu *et al.*, "Deepfacelab: A simple, flexible and extensible face swapping framework," *arXiv preprint arXiv:2005.05535*, 2020.

[17] R. Natsume, T. Yatagawa, and S. Morishima, "Fsnet: An identity-aware generative model for image-based face swapping," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 117–132.

[18] Natsume, Ryota and Yatagawa, Tatsuya and Morishima, Shigeo, "Rsgan: face swapping and editing using face and hair representation in latent spaces," *arXiv preprint arXiv:1804.03447*, 2018.

[19] Y. Nirkin, Y. Keller, and T. Hassner, "Fsgan: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7184–7193.

[20] H. Zhu, C. Fu, Q. Wu, W. Wu, C. Qian, and R. He, "Aot: Appearance optimal transport based identity swapping for forgery detection," in *NeurIPS*, 2020.

[21] Y. Wang, X. Chen, J. Zhu, W. Chu, Y. Tai, C. Wang, J. Li, Y. Wu, F. Huang, and R. Ji, "Hififace: 3d shape and semantic prior guided high fidelity face swapping," *arXiv preprint arXiv:2106.09965*, 2021.

[22] Y. Zhu, Q. Li, J. Wang, C.-Z. Xu, and Z. Sun, "One shot face swapping on megapixels," in *CVPR*, 2021, pp. 4834–4844.

[23] G. Gao, H. Huang, C. Fu, Z. Li, and R. He, "Information bottleneck disentanglement for identity swapping," in *CVPR*, 2021, pp. 3404–3413.

[24] Y. Chen, J. Wang, S. Chen, Z. Shi, and J. Cai, "Facial motion prior networks for facial expression recognition," in *VCIP*. IEEE, 2019, pp. 1–4.

[25] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao, "Region attention networks for pose and occlusion robust facial expression recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 4057–4069, 2020.

[26] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18–31, 2017.

[27] C. Kervadec, V. Vielzeuf, S. Pateux, A. Lechervy, and F. Jurie, "Cake: Compact and accurate k-dimensional representation of emotion," *arXiv preprint arXiv:1807.11215*, 2018.

[28] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "Ganimation: Anatomically-aware facial animation from a single image," in *ECCV*, 2018, pp. 818–833.

[29] J. Ling, H. Xue, L. Song, S. Yang, R. Xie, and X. Gu, "Toward fine-grained facial expression manipulation," in *ECCV*. Springer, 2020, pp. 37–53.

[30] O. Wiles, A. Koepke, and A. Zisserman, "X2face: A network for controlling face generation using images, audio, and pose codes," in *ECCV*, 2018, pp. 670–686.

[31] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," *Advances in Neural Information Processing Systems*, vol. 32, pp. 7137–7147, 2019.

[32] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *CVPR*, 2019, pp. 9459–9468.

[33] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *CVPR*, 2019, pp. 2377–2386.

[34] W. Zhang, X. Ji, K. Chen, Y. Ding, and C. Fan, "Learning a facial expression embedding disentangled from identity," in *CVPR*, 2021, pp. 6759–6768.

[35] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017, pp. 1125–1134.

[36] Unity, "The leading platform for creating interactive, real-time content," https://unity.com/, 2021, Online; Accessed July 16, 2021.

[37] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[38] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3fd: Single shot scale-invariant face detector," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 192–201.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[40] X. Liu, B. Vijaya Kumar, J. You, and P. Jia, "Adaptive deep metric learning for identity-aware facial expression recognition," in *CVPRW*, 2017, pp. 20–29.

[41] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019, pp. 2337–2346.

[42] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *ECCV*, 2018, pp. 3–19.

[43] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, 2018, pp. 8789–8797.

[44] J. H. Lim and J. C. Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017.

[45] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *CVPR*, 2020, pp. 5549–5558.

[46] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410.

[47] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *FG*. IEEE, 2018, pp. 67–74.

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*. Curran Associates, Inc., 2019, pp. 8024–8035.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[51] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *CVPR*, 2018, pp. 5265–5274.

[52] M. Wang, https://github.com/MuggleWang/CosFace_pytorch, 2018, accessed March 1, 2021.

[53] TreB1eN, https://github.com/TreB1eN/InsightFace_Pytorch, 2018, Online; Accessed March 1, 2021.

[54] W. Liu, https://github.com/wy1iu/sphereface, 2018, accessed March 1, 2021.

[55] WuJie, https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch, 2020, Online; Accessed March 1, 2021.

[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[57] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *International Conference on Computer Vision (ICCV)*, 2019.