DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection

Supplementary Material

Liming Jiang[1] Ren Li[2] Wayne Wu[1,2] Chen Qian[2] Chen Change Loy[1†]
[1]Nanyang Technological University [2]SenseTime Research
liming002@ntu.edu.sg tomo.blade.lee@hotmail.com
wuwenyan@sensetime.com qianchen@sensetime.com ccloy@ntu.edu.sg

Abstract

This document provides supplementary information which is not elaborated in our main paper due to the constraints of space: Section A illustrates the detailed derivations and experimental results of the proposed DF-VAE framework; Section B describes the supplementary information of our real-world face forgery detection benchmark; Section C presents more examples of our source video data collection; Section D shows diverse perturbations in DeeperForensics-1.0 to better simulate real-world scenarios.

A. Method Details

In order to improve the obvious low *quality* problems of the previous datasets, a new learning-based end-to-end face swapping framework, DeepFake Variational Auto-Encoder (DF-VAE), is proposed as our dataset construction method.

In the main paper, we give a brief and intuitive introduction of DF-VAE. Three key points (*i.e.*, *disentanglement*, *style matching*, *temporal continuity*) have been discussed. In this section, we will elaborate on the details of DF-VAE (see Figure 1 for the main framework of DF-VAE).

A.1. Disentangled Module

To be consistent with our main paper, we refer to the identity in the driving video as the "target" face and the identity of the face which is swapped onto the video as the "source" face.

The disentanglement of structure and appearance is rather difficult because appearance contains too much information. Besides, structure and appearance representation are far from being independent. As we claim in the main paper, face swapping can be considered as a subsequent step of face reenactment with suitable fusion modules for the reenacted face and the background. Thus, in the disentangled module, our main task is to animate the source face with similar expression as the target face, *i.e.* face reenactment, without any paired data.

Let $\mathbf{x}_{1:T} \equiv \{x_1, x_2, ..., x_T\} \in X$ be a sequence of source face video frames, and $\mathbf{y}_{1:T} \equiv \{y_1, y_2, ..., y_T\} \in Y$ be the sequence of corresponding target face video frames. We first simplify our problem and only consider two specific snapshots at time t, x_t and y_t. Let $\tilde{x}_t$, $\tilde{y}_t$, d_t represent the reconstructed source face, the reconstructed target face, and the reenacted face, respectively.

Consider the reconstruction procedure of the source face x_t. Let s_x denote structure representation and a_x denote appearance information. The face generator can be depicted as the posteriori estimate $p_\theta(x_t|s_x, a_x)$. The solution of our reconstruction goal, marginal log-likelihood $\tilde{x}_t \sim \log p_\theta(x_t)$, by a common Variational Auto-Encoder (VAE) [16] can be written as:

$$\begin{aligned}\log p_\theta(x_t) = D_{KL}\left(q_\phi(s_x, a_x|x_t) \| p_\theta(s_x, a_x|x_t)\right) \\ + L(\theta, \phi; x_t),\end{aligned} \tag{1}$$

where q_ϕ is an approximate posterior to achieve the evidence lower bound (ELBO) in the intractable case, and the second RHS term $L(\theta, \phi; x_t)$ is the variational lower bound *w.r.t.* both the variational parameters ϕ and generative parameters θ. Since the first RHS term KL-divergence is non-negative, we get:

$$\begin{aligned}&\log p_\theta(x_t) \geq L(\theta, \phi; x_t) \\ &= \mathbb{E}_{q_\phi(s_x, a_x|x_t)}\left[-\log q_\phi(s_x, a_x|x_t) + \log p_\theta(x_t, s_x, a_x)\right],\end{aligned} \tag{2}$$

where $L(\theta, \phi; x_t)$ can also be written as:

$$\begin{aligned}L(\theta, \phi; x_t) = &- D_{KL}\left(q_\phi(s_x, a_x|x_t) \| p_\theta(s_x, a_x)\right) \\ &+ \mathbb{E}_{q_\phi(s_x, a_x|x_t)}\left[\log p_\theta(x_t|s_x, a_x)\right],\end{aligned} \tag{3}$$

and we need to optimize $L(\theta, \phi; x_t)$ *w.r.t.* ϕ and θ.

In Eq. 1, we assume that both s_x and a_x are latent priors computed by the same posterior x_t. However, the separation of these two variables in the latent space is rather difficult without additional conditions. Therefore, we employ a

† Corresponding author.

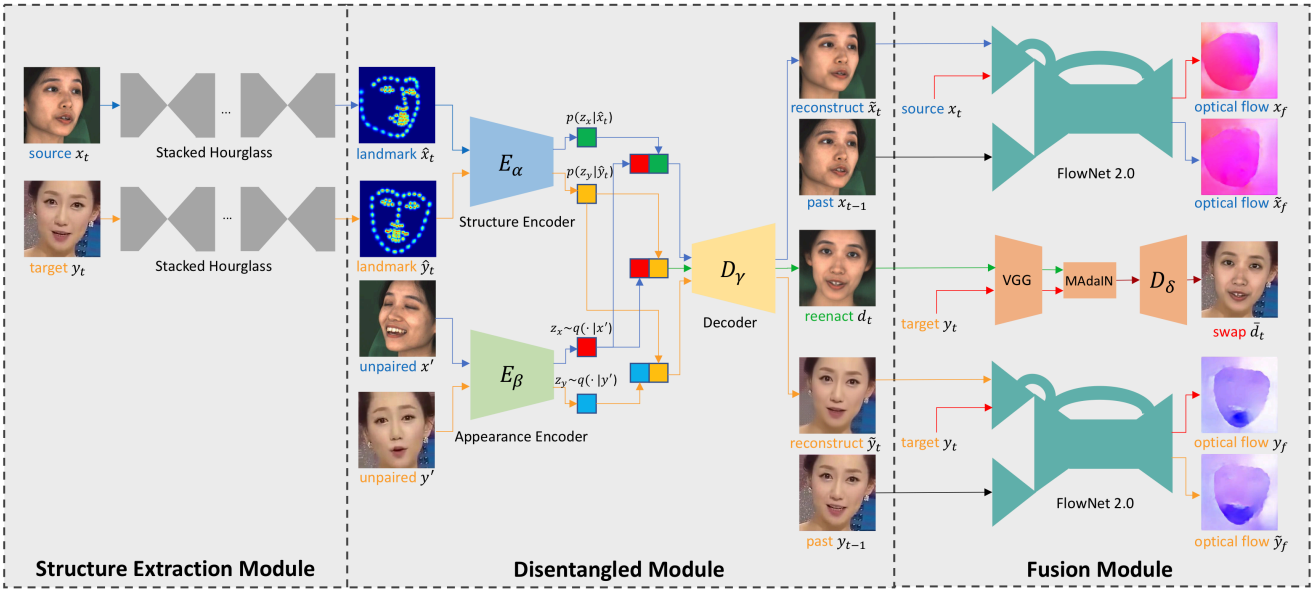

Figure 1: The main framework of DeepFake Variational Auto-Encoder. In training, we reconstruct the source and target faces in blue and orange arrows, respectively, by extracting landmarks and constructing an unpaired sample as the condition. Optical flow differences are minimized after reconstruction to improve temporal continuity. In inference, we swap the latent codes and get the reenacted face in green arrows. Subsequent MAdaIN module fuses the reenacted face and the original background resulting in the swapped face.

simple yet effective approach to disentangle these two variables.

The blue arrows in Figure 1 show the reconstruction procedure of the source face x_t. Instead of feeding a single source face x_t, we sample another source face x' to construct unpaired data in the source domain. To make the structure representation more evident, we use the stacked hourglass networks [17] to extract landmarks of x_t in the structure extraction module and get the heatmap $\hat{x}_t$. Then we feed the heatmap $\hat{x}_t$ to the Structure Encoder E_α, and x' to the Appearance Encoder E_β. We concatenate the latent representations (small cubes in red and green) and feed it to the Decoder D_γ. Finally, we get the reconstructed face $\tilde{x}_t$, *i.e.*, marginal log-likelihood of x_t.

Therefore, the latent structure representation s_x in Eq. 1 becomes a more evident heatmap representation $\hat{x}_t$, which is introduced as a new condition. The unpaired sample x' with the same identity *w.r.t.* x_t is another condition, being a substitute for a_x. Eq. 1 can be rewritten as a conditional log-likelihood:

$$\log p_\theta\left(x_t|\hat{x}_t, x'\right) = D_{KL}\left(q_\phi\left(z_x|x_t, \hat{x}_t, x'\right) \| p_\theta\left(z_x|x_t, \hat{x}_t, x'\right)\right) + L\left(\theta, \phi; x_t, \hat{x}_t, x'\right), \tag{4}$$

similarly,

$$\log p_\theta\left(x_t|\hat{x}_t, x'\right) \geq L(\theta, \phi; x_t, \hat{x}_t, x') = \mathbb{E}_{q_\phi(z_x|x_t, \hat{x}_t, x')}\left[-\log q_\phi\left(z_x|x_t, \hat{x}_t, x'\right) + \log p_\theta\left(x_t, z_x|\hat{x}_t, x'\right)\right], \tag{5}$$

and $L(\theta, \phi; x_t, \hat{x}_t, x')$ can also be written as:

$$L\left(\theta, \phi; x_t, \hat{x}_t, x'\right) = -D_{KL}\left(q_\phi\left(z_x|x_t, \hat{x}_t, x'\right) \| p_\theta\left(z_x|\hat{x}_t, x'\right)\right) + \mathbb{E}_{q_\phi(z_x|x_t, \hat{x}_t, x')}\left[\log p_\theta\left(x_t|z_x, \hat{x}_t, x'\right)\right]. \tag{6}$$

We let the variational approximate posterior be a multivariate Gaussian with a diagonal covariance structure:

$$\log q_\phi\left(z_x|x_t, \hat{x}_t, x'\right) \equiv \log \mathcal{N}\left(z_x; \mu, \sigma^2\mathbf{I}\right), \tag{7}$$

where $\mathbf{I}$ is an identity matrix. Exploiting the reparameterization trick [16], the non-differentiable operation of sampling can become differentiable by an auxiliary variable with independent marginal. In this case, $z_x \sim q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)$ is implemented by $z_x = \mu + \sigma\epsilon$ where ϵ is an auxiliary noise variable $\epsilon \sim \mathcal{N}(0, 1)$. Finally, the approximate posterior $q_\phi(z_x|x_t, \hat{x}_t, x')$ is estimated by the separated encoders, Structure Encoder E_α and Appearance Encoder E_β, in an end-to-end training process by standard gradient descent.

We discuss the whole workflow of reconstructing the source face. In the target face domain, the reconstruction procedure is the same, as shown by orange arrows in Figure 1.

During training, the network learns structure and appearance information in both the source and the target domains. It is noteworthy that even if both y_t and x' belong to arbitrary identities, our effective disentangled module is capable of learning meaningful structure and appearance information of each identity. During inference, we concatenate the

appearance prior of x' and the structure prior of y_t (small cubes in red and orange) in the latent space, and the reconstructed face d_t shares the same structure with y_t and keeps the appearance of x'. Our framework allows concatenations of structure and appearance latent codes extracted from arbitrary identities in inference and permits *many-to-many face reenactment*.

In summary, DF-VAE is a new conditional variational auto-encoder [15] with robustness and scalability. It conditions on two posteriors in different domains. In the disentangled module, the separated design of two encoders E_α and E_β, the explicit structure heatmap, and the unpaired data construction jointly force E_α to learn structure information and E_β to learn appearance information.

A.2. Derivation

The core equations of DF-VAE are Eq. 4, Eq. 5, and Eq. 6. We will provide the detailed mathematical derivations in this section. The previous three equations, Eq. 1, Eq. 2 and Eq. 3, have very similar derivations therefore we will not be repeating them.

Derivation of Eq. 4 and Eq. 5:

$$
\begin{aligned}
&\log p_\theta\left(x_t|\hat{x}_t, x'\right) \\
&= \mathbb{E}_{q_\phi(z_x|x_t,\hat{x}_t,x')}\left(\log p_\theta\left(x_t|\hat{x}_t, x'\right)\right) \\
&= \mathbb{E}_{q_\phi(z_x|x_t,\hat{x}_t,x')}\left[\log \frac{p_\theta\left(x_t, z_x|\hat{x}_t, x'\right)}{p_\theta\left(z_x|x_t, \hat{x}_t, x'\right)}\right] \\
&= \mathbb{E}_{q_\phi(z_x|x_t,\hat{x}_t,x')}\left[\log \frac{q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)}{p_\theta\left(z_x|x_t, \hat{x}_t, x'\right)} \cdot \frac{p_\theta\left(x_t, z_x|\hat{x}_t, x'\right)}{q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)}\right] \\
&= \int q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)\left[\log \frac{q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)}{p_\theta\left(z_x|x_t, \hat{x}_t, x'\right)} + \log \frac{p_\theta\left(x_t, z_x|\hat{x}_t, x'\right)}{q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)}\right] dz_x \\
&= D_{KL}\left(q_\phi\left(z_x|x_t, \hat{x}_t, x'\right) \| p_\theta\left(z_x|x_t, \hat{x}_t, x'\right)\right) + L\left(\theta, \phi; x_t, \hat{x}_t, x'\right),
\end{aligned}
$$

where

$$
L\left(\theta, \phi; x_t, \hat{x}_t, x'\right) = \mathbb{E}_{q_\phi(z_x|x_t,\hat{x}_t,x')}\left[\log \frac{p_\theta\left(x_t, z_x|\hat{x}_t, x'\right)}{q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)}\right],
$$

$$
D_{KL}\left(q_\phi\left(z_x|x_t, \hat{x}_t, x'\right) \| p_\theta\left(z_x|x_t, \hat{x}_t, x'\right)\right) \geq 0.
$$

Derivation of Eq. 6:

$$
\begin{aligned}
&L\left(\theta, \phi; x_t, \hat{x}_t, x'\right) \\
&= \mathbb{E}_{q_\phi(z_x|x_t,\hat{x}_t,x')}\left[\log \frac{p_\theta\left(x_t, z_x|\hat{x}_t, x'\right)}{q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)}\right] \\
&= \mathbb{E}_{q_\phi(z_x|x_t,\hat{x}_t,x')}\left[\log \frac{p_\theta\left(x_t|z_x, \hat{x}_t, x'\right) p_\theta\left(z_x|\hat{x}_t, x'\right)}{q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)}\right] \\
&= \int q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)\left[-\log \frac{q_\phi\left(z_x|x_t, \hat{x}_t, x'\right)}{p_\theta\left(z_x|\hat{x}_t, x'\right)} + \log p_\theta\left(x_t|z_x, \hat{x}_t, x'\right)\right] dz_x \\
&= -D_{KL}\left(q_\phi\left(z_x|x_t, \hat{x}_t, x'\right) \| p_\theta\left(z_x|\hat{x}_t, x'\right)\right) \\
&\quad + \mathbb{E}_{q_\phi(z_x|x_t,\hat{x}_t,x')}\left[\log p_\theta\left(x_t|z_x, \hat{x}_t, x'\right)\right].
\end{aligned}
$$

A.3. Objective

Reconstruction loss. In the reconstruction, the source face and target face share the same forms of loss functions. The reconstruction loss of the source face, L_{recon_x}, can be written as:

$$
L_{recon_x} = \lambda_{r_1} L_{pixel}\left(\tilde{x}, x\right) + \lambda_{r_2} L_{ssim}\left(\tilde{x}, x\right). \quad (8)
$$

L_{pixel} indicates pixel loss. It calculates the Mean Absolute Error (MAE) after reconstruction, which can be written as:

$$
L_{pixel}\left(\tilde{x}, x\right) = \frac{1}{CHW}\|\tilde{x} - x\|_1. \quad (9)
$$

L_{ssim} denotes ssim loss. It computes the Structural Similarity (SSIM) of the reconstructed face and the original face, which has the form of:

$$
L_{ssim}\left(\tilde{x}, x\right) = \frac{\left(2\mu_{\tilde{x}}\mu_x + C_1\right)\left(2\sigma_{\tilde{x}x} + C_2\right)}{\left(\mu_{\tilde{x}}^2 + \mu_x^2 + C_1\right)\left(\sigma_{\tilde{x}}^2 + \sigma_x^2 + C_2\right)}. \quad (10)
$$

λ_{r_1} and λ_{r_2} are two hyperparameters that control the weights of two parts of the reconstruction loss. For the target face, we have the similar form of reconstruction loss:

$$
L_{recon_y} = \lambda_{r_1} L_{pixel}\left(\tilde{y}, y\right) + \lambda_{r_2} L_{ssim}\left(\tilde{y}, y\right). \quad (11)
$$

Thus, the full reconstruction loss can be written as:

$$
L_{recon} = L_{recon_x} + L_{recon_y}. \quad (12)
$$

KL loss. Since DF-VAE is a new conditional variational auto-encoder, reparameterization trick is utilized to make the sampling operation differentiable by an auxiliary variable with independent marginal. We use the typical KL loss in [16] with the form of:

$$
L_{KL}\left(q_\phi\left(z\right), p_\theta\left(z\right)\right) = \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\left(\left(\sigma_j\right)^2\right) - \left(\mu_j\right)^2 - \left(\sigma_j\right)^2\right), \quad (13)
$$

where J is the dimensionality of the latent prior z, μ_j and σ_j are the j-th element of variational mean and s.d. vectors, respectively.

MAdaIN loss. The MAdaIN module is jointly trained with the disentangled module in an end-to-end manner. We apply MAdaIN loss for this module, in a similar form as described in [10]. We use the VGG-19 [20] to compute MAdaIN loss to train Decoder D_δ:

$$
L_{\text{MAdaIN}} = L_c + \lambda_{ma} L_s. \quad (14)
$$

L_c denotes the content loss, which is the Euclidean distance between the target features and the features of the swapped face. L_c has the form of:

$$
L_c = \|o - c\|_2, \quad (15)
$$

where $o = m_t^b \cdot \overline{d}_t$, $c = m_t^b \cdot d_t$. m_t^b is the blurred mask described in the main paper.

L_s represents the style loss, which matches the mean and standard deviation of the style features. Like [10], we match the IN [22] statistics instead of using Gram matrix loss which can produce similar results. L_s can be written as:

$$L_s = \sum_{i=1}^{L} \|\mu(\Phi_i(o)) - \mu(\Phi_i(s))\|_2 + \sum_{i=1}^{L} \|\sigma(\Phi_i(o)) - \sigma(\Phi_i(s))\|_2, \tag{16}$$

where $o = m_t^b \cdot \overline{d}_t$, $s = m_t^b \cdot y_t$. m_t^b is the blurred mask. Φ_i denotes the layer used in VGG-19 [20]. Similar to [10], we use `relu1_1`, `relu2_1`, `relu3_1`, `relu4_1` layers with equal weights.

λ_{ma} is the weight of style loss to balance two parts of MAdaIN loss.

Temporal loss. We have given a detailed introduction of temporal consistency constraint in our main paper. The form of temporal loss is the same as the newly proposed temporal consistency constraint. We will not repeat it here.

Total objective. DF-VAE is an end-to-end many-to-many face swapping framework. We jointly train all parts of the networks. The problem can be described as the optimization of the following total objective:

$$L_{total} = \lambda_1 L_{recon} + \lambda_2 L_{KL} + \lambda_3 L_{MAdaIN} + \lambda_4 L_{temporal}, \tag{17}$$

where λ_1, λ_2, λ_3, λ_4 are the weight hyperparameters of four types of loss functions introduced above.

A.4. Implementation Details

The whole DF-VAE framework is end-to-end. We use the pretrained stacked hourglass networks [17] to extract landmarks. The numbers of stacks and blocks are set to 4 and 1, respectively. We exploit FlowNet 2.0 network [11] to estimate optical flows. The typical AdaIN network [10] is applied to our style matching and fusion module. The learning rate is set to 0.00005 for all parts of DF-VAE. We utilize Adam [14] and set $\beta_1 = 0.5$, $\beta_2 = 0.999$. All the experiments are conducted on NVIDIA Tesla V100 GPUs.

A.5. User Study of Methods

In addition to user study based on datasets to examine the quality of DeeperForensics-1.0 dataset, we also carry out a user study to compare DF-VAE with state-of-the-art face manipulation methods. We will present the user study of methods in this section.

Baselines. We choose three learning-based open-source methods as our baselines: DeepFakes [1], faceswap-GAN [2], and ReenactGAN [25]. These three methods are representative, which are based on different architectures. DeepFakes [1] is a well-known method based on Auto-Encoders (AE). It uses a shared encoder and two separated decoders to perform face swapping. faceswap-GAN [2] is based on Generative Adversarial Networks (GAN) [6], which has a similar structure as DeepFakes [1] but also uses a paired discriminators to improve face swapping quality. ReenactGAN [25] makes a boundary latent space assumption and uses a transformer to adapt the boundary of source face to that of target face. As a result, ReenactGAN can perform many-to-one face reenactment. After getting the reenacted faces, we use our carefully designed fusion method to obtain the swapped faces. For a fair comparison, DF-VAE utilizes the same fusion method when compared to ReenactGAN [25].

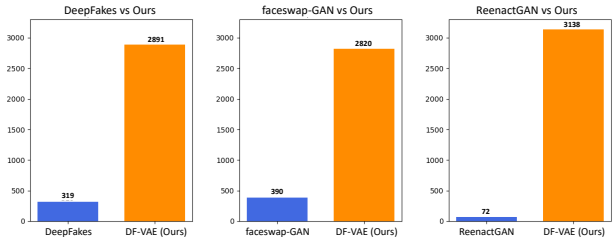

Figure 2: Results of user study comparing methods. The bar charts show the number of users who give preference in each compared pair of manipulated videos.

Results. We randomly choose 30 real videos from DeeperForensics-1.0 as the source videos and 30 real videos from FaceForensics++ [18] as the target videos. Thus, each method generates 30 fake videos. Same as the user study based on datasets, we conduct the user study based on methods among 100 professional participants who specialize in computer vision research. Because there are corresponding fake videos, we let the users directly choose their preferred fake videos between those generated by other methods and those generated by DF-VAE. Finally, we got 3210 answers for each compared pair. The results are shown in Figure 2. We can see that DF-VAE shows an impressive advantage over the baselines, underscoring the *high quality* of DF-VAE-generated fake videos.

A.6. Quantitative Evaluation Metrics

Fréchet Inception Distance (FID) [8] is a widely exploited metric for generative models. FID evaluates the *similarity* of distribution between the generated images and the real images. FID correlates well with the visual quality of the generated samples. A lower value of FID means a better quality.

Inception Score (IS) [19] is an early and somewhat widely adopted objective evaluation metric for generated images. IS evaluates two aspects of generation quality: *articulation* and *diversity*. A higher value of IS means a better quality.

Table 1 shows the FID and IS scores of our method compared to other methods. DF-VAE outperforms all the three baselines in quantitative evaluations by FID and IS.

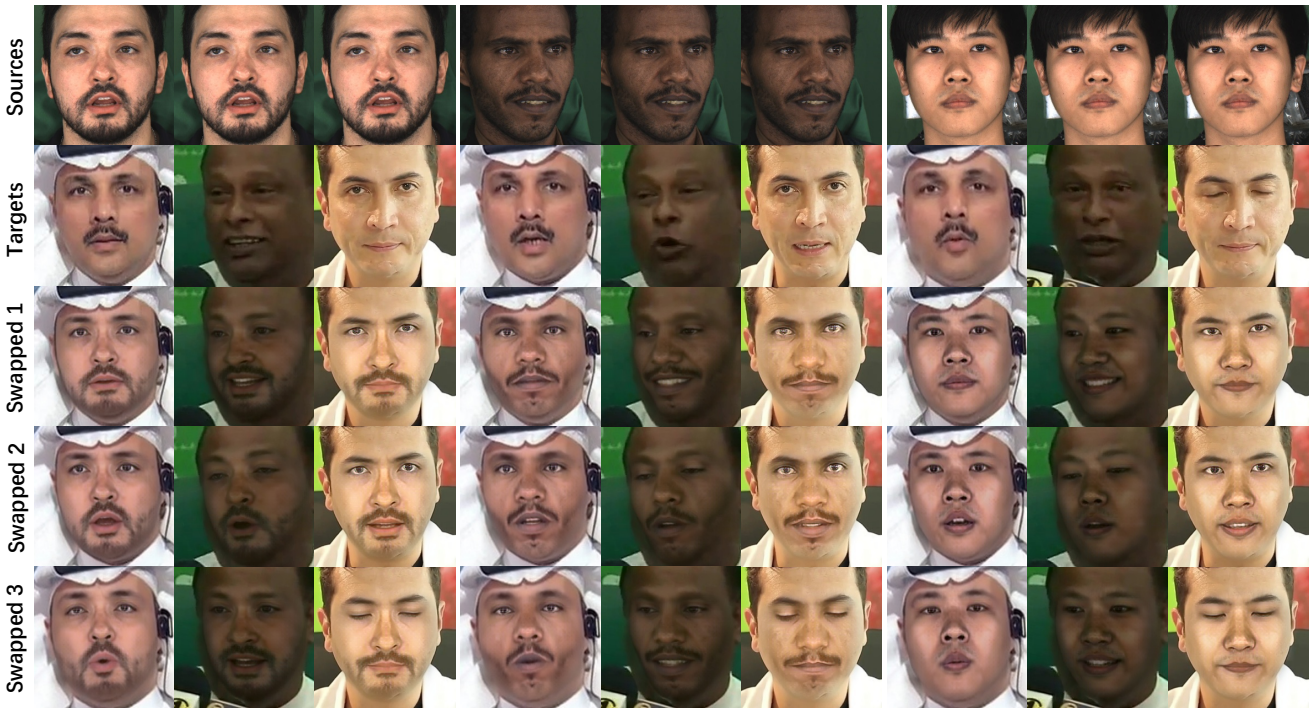

Figure 3: Many-to-many (three-to-three) face swapping by a **single** model with obvious reduction of style mismatch problems. This figure shows the results between three source identities and three target identities. The whole process is end-to-end.

Table 1: The FID and IS scores of DeepFakes [1], faceswap-GAN [2], ReenactGAN [25], and DF-VAE (Ours).

Method	FID	IS
DeepFakes [1]	25.771	1.711
faceswap-GAN [2]	24.718	1.685
ReenactGAN [25]	26.325	1.690
DF-VAE (Ours)	**22.097**	**1.714**

A.7. Many-to-Many Face Swapping

By a *single* model, DF-VAE can perform *many-to-many face swapping* with obvious reduction of style mismatch and facial boundary artifacts (see Figure 3 for the face swapping between three source identities and three target identities). Even if there are multiple identities in both the source domain and the target domain, the quality of face swapping does not degrade.

A.8. Ablation Studies

Ablation study of temporal loss. Since the swapped faces do not have the ground truth, we evaluate the effectiveness of temporal consistency constraint, *i.e.*, temporal loss, in a self-reenactment setting. Similar to [13], we quantify the re-rendering error by Euclidean distance of per pixel in RGB channels ([0, 255]). Visualized results are shown in Figure 4. Without the temporal loss, the re-rendering error is higher, hence demonstrating the effectiveness of temporal consistency constraint.

Ablation study of different components. We conduct further ablation studies *w.r.t.* different components of our DF-VAE framework under many-to-many face swapping setting (see Figure 5). The source and target faces are shown in Column 1 and Column 2. In Column 3, our full method, DF-VAE, shows high-fidelity face swapping results. In Column 4, style mismatch problems are very obvious if we remove the MAdaIN module. If we remove the hourglass (structure extraction) module, the disentanglement of structure and appearance is not very thorough. The swapped face will be a mixture of multiple identities, as shown in Column 5. When we perform face swapping without constructing unpaired data in the same domain (see Column 6), the disentangled module will completely reconstruct the faces on the side of E_β, thus the disentanglement is not established at all. Therefore, the quality of face swapping will degrade if we remove any component in DF-VAE framework.

B. Supplementary Information of Benchmark

In this section, we will provide some supplementary information of our *real-world* face forgery detection benchmark. First, we will elaborate on the basic information of five baselines in our benchmark in Section B.1. Then, we will provide evaluation results of face forgery detection model performance *w.r.t.* dataset size in Section B.2.

Table 2: The binary detection accuracy of the baseline methods on DeeperForensics-1.0 standard test set (std) / hidden test set (hidden) when trained on the standard training set with *different dataset sizes*. FS denotes the full dataset size of the standard training set reported in the main paper. $FS \div x$ indicates the reduction of dataset size to $1/x$ of the full dataset size.

Method	C3D [21]		TSN [23]		I3D [4]		ResNet+LSTM [7, 9]		XceptionNet [5]	
Test (acc)	std	hidden	std	hidden	std	hidden	std	hidden	std	hidden
FS (Full Size)	98.50	74.75	99.25	77.00	100.00	79.25	100.00	78.25	100.00	77.00
$FS \div 2$	97.50	75.00	98.75	78.50	100.00	78.13	100.00	78.88	100.00	76.75
$FS \div 4$	96.00	75.25	99.25	76.50	99.50	77.00	100.00	78.25	100.00	76.50
$FS \div 8$	92.25	72.13	91.25	70.88	95.00	73.50	98.25	75.63	100.00	76.13
$FS \div 16$	84.25	66.88	84.75	69.50	95.25	74.25	98.50	76.25	100.00	74.88
$FS \div 32$	62.50	54.50	68.00	58.25	80.50	67.88	95.50	74.13	95.00	71.38
$FS \div 64$	61.25	53.88	52.75	50.00	62.00	57.25	90.75	70.25	88.50	67.75

B.1. Details of Benchmark Baselines

We elaborate on five baselines used in our face forgery detection benchmark in this section. Our benchmark contains four video-level face forgery detection methods, C3D [21], Temporal Segment Networks (TSN) [23], Inflated 3D ConvNet (I3D) [4], and ResNet+LSTM [7, 9]. One image-level detection method, XceptionNet [5], which achieves the best performance in FaceForensics++ [18], is evaluated as well.

- **C3D** [21] is a simple but effective method, which incorporates 3D convolution to capture the spatiotemporal feature of videos. It includes 8 convolutional, 5 max-pooling, and 2 fully connected layers. The size of the 3D convolutional kernels is $3 \times 3 \times 3$. When training C3D, the videos are divided into non-overlapped clips with 16-frames length, and the original face images are resized to 112×112.
- **TSN** [23] is a 2D convolutional network, which splits the video into short segments and randomly selects a snippet from each segment as the input. The long-range temporal structure modeling is achieved by the fusion of the class scores corresponding to these snippets. In our experiment, we choose BN-Inception [12] as the backbone and only train our model with the RGB stream. The number of segments is set to 3 as default, and the original images are resized to 224×224.
- **I3D** [4] is derived from Inception-V1 [12]. It inflates the 2D ConvNet by endowing the filters and pooling kernels with an additional temporal dimension. In the training, we use 64-frame snippets as the input, whose starting frames are randomly selected from the videos. The face images are resized to 224×224.
- **ResNet+LSTM** [7, 9] is based on ResNet [7] architecture. As a 2D convolutional framework, ResNet [7] is used to extract spatial features (the output of the last convolutional layer) for each face image. In order to encode the temporal dependency between images, we place an LSTM [9] module with 512 hidden units after ResNet-50 [7] to aggregate the spatial features. An additional fully connected layer serves as the classifier. All the videos are downsampled with a ratio of 5, and the images are resized to 224×224 before feeding into the network. During training, the loss is the summation of the binary entropy on the output at all time steps, while only the output of the last frame is used for the final classification in inference.
- **XceptionNet** [5] is a depthwise-separable-convolution based CNN, which has been used in [18] for image-level face forgery detection. We exploit the same XceptionNet model as [18] but without freezing the weights of any layer during training. The face images are resized to 299×299. In the test phase, the prediction is made by averaging classification scores of all frames within a video.

B.2. Evaluation of Dataset Size

We include additional evaluation results of face forgery detection methods *w.r.t.* dataset size, as shown in Table 2. The experiments are conducted on DeeperForensics-1.0 in this setting. All the models are trained on the standard training set with different dataset sizes.

The reduction of dataset size hurts the accuracy of all the baselines on either standard test set (std) or hidden test set (hidden). This justifies that a larger dataset size can be helpful for detection model performance. It is noteworthy that accuracy drops little if we do not reduce the dataset size significantly. These observations concerning dataset size are in line with that of [18, 24].

In contrast to previous works, we also find that the image-level detection method is less sensitive (*i.e.*, has a stronger ability to remain high accuracy) to dataset size than pure video-level method. One possible explanation for this is that despite a significant reduction of video samples, there still exist enough image samples (frames in total) for the image-level method to achieve good performance.

C. More Examples of Data Collection

In this section, we show more examples of our extensive source video data collection (see Figure 6). Our high-

quality collected data vary in identities, poses, expressions, emotions, lighting conditions, and 3DMM blendshapes [3]. The source videos will also be released for further research.

D. Perturbations

We also show some examples of perturbations in DeeperForensics-1.0. Seven types of perturbations and the mixture of two (Gaussian blur, JPEG compression) / three (Gaussian blur, JPEG compression, white Gaussian noise in color components) / four (Gaussian blur, JPEG compression, white Gaussian noise in color components, change of color saturation) perturbations are shown in Figure 7. These perturbations are very common distortions existing in real life. The comprehensiveness of perturbations in DeeperForensics-1.0 ensures its *diversity* to better simulate fake videos in real-world scenarios.

References

[1] Deepfakes. https://github.com/deepfakes/faceswap/. Accessed: 2019-08-16. 4, 5

[2] faceswap-gan. https://github.com/shaoanlu/faceswap-GAN/. Accessed: 2019-08-16. 4, 5

[3] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *TVCG*, 20:413–425, 2013. 6, 9

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 6

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 6

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[8] 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. In *NeurIPS*, 2017. 4

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997. 6

[10] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 3, 4

[11] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 4

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 6

[13] Hyeongwoo Kim, Pablo Carrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM TOG*, 37:163, 2018. 5, 8

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980, 2014. 4

[15] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *NeurIPS*, 2014. 3

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*, arXiv:1312.6114, 2013. 1, 2, 3

[17] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 2, 4

[18] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. *arXiv preprint*, arXiv:1901.08971, 2019. 4, 6

[19] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 4

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, arXiv:1409.1556, 2014. 3, 4

[21] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 6

[22] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint*, arXiv:1607.08022, 2016. 4

[23] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 6

[24] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. *arXiv preprint*, arXiv:1912.11035, 2019. 6

[25] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. Reenactgan: Learning to reenact faces via boundary transfer. In *ECCV*, 2018. 4, 5

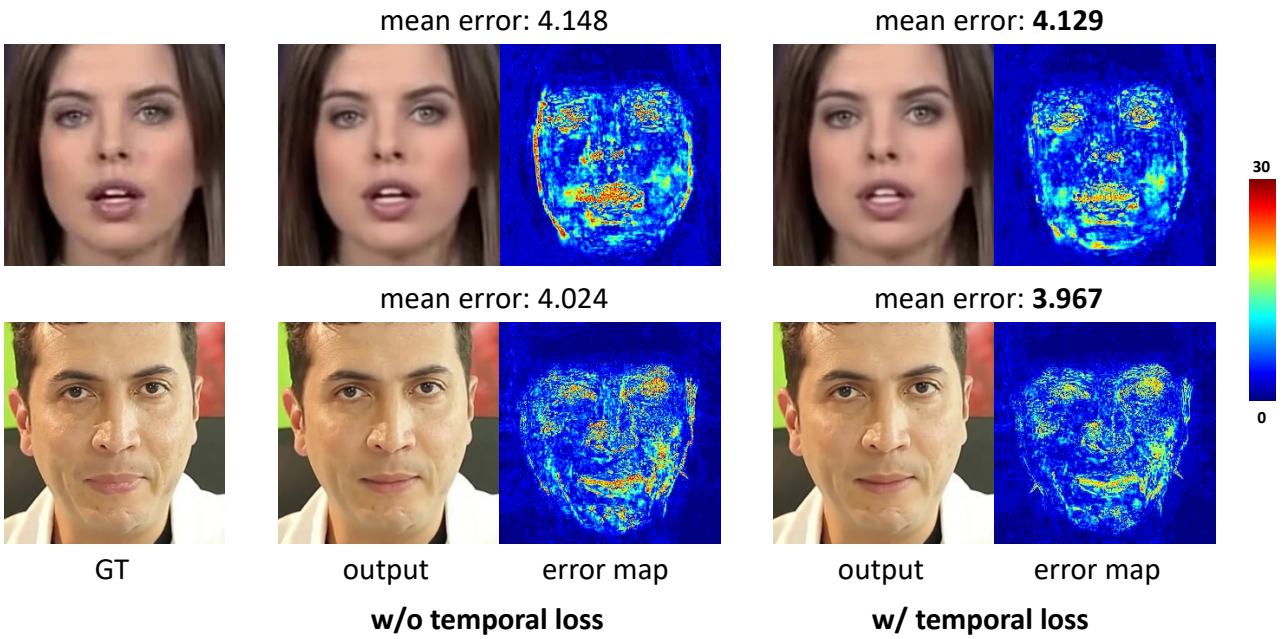

Figure 4: The quantitative evaluation of the effectiveness of temporal loss. Similar to [13], we use the re-rendering error in a self-reenactment setting, where the ground truth is known. The error maps show Euclidean distance of per pixel in RGB channels ([0, 255]). The mean errors are shown above the images. The corresponding color scale *w.r.t.* error values is shown on the right side of the images.

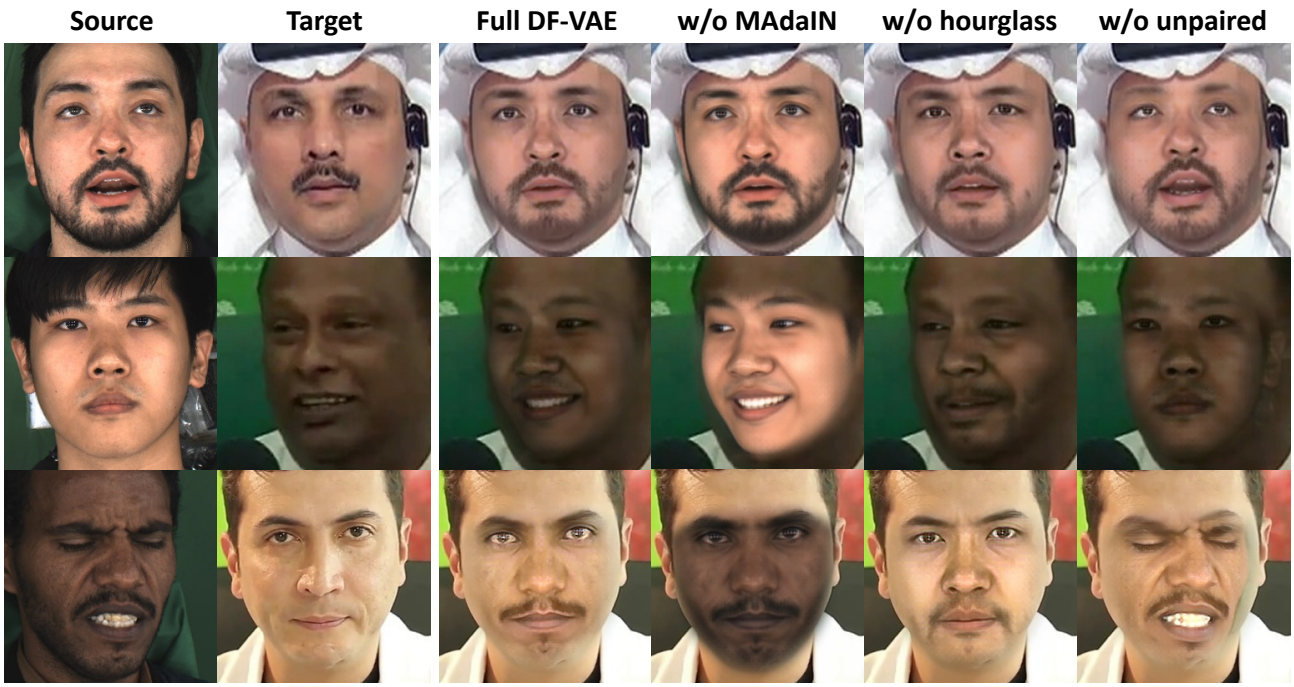

Figure 5: The ablation studies of different components in DF-VAE framework in the many-to-many face swapping setting. Column 1 and Column 2 show the source face and the target face, respectively. Column 3 shows the results of the full method. Column 4, 5, 6 show the results when removing MAdaIN module, hourglass (structure extraction) module, and unpaired data construction, respectively.

Figure 6: More examples of the source video data collection. Our high-quality collected data vary in identities, poses, expressions, emotions, lighting conditions, and 3DMM blendshapes [3].

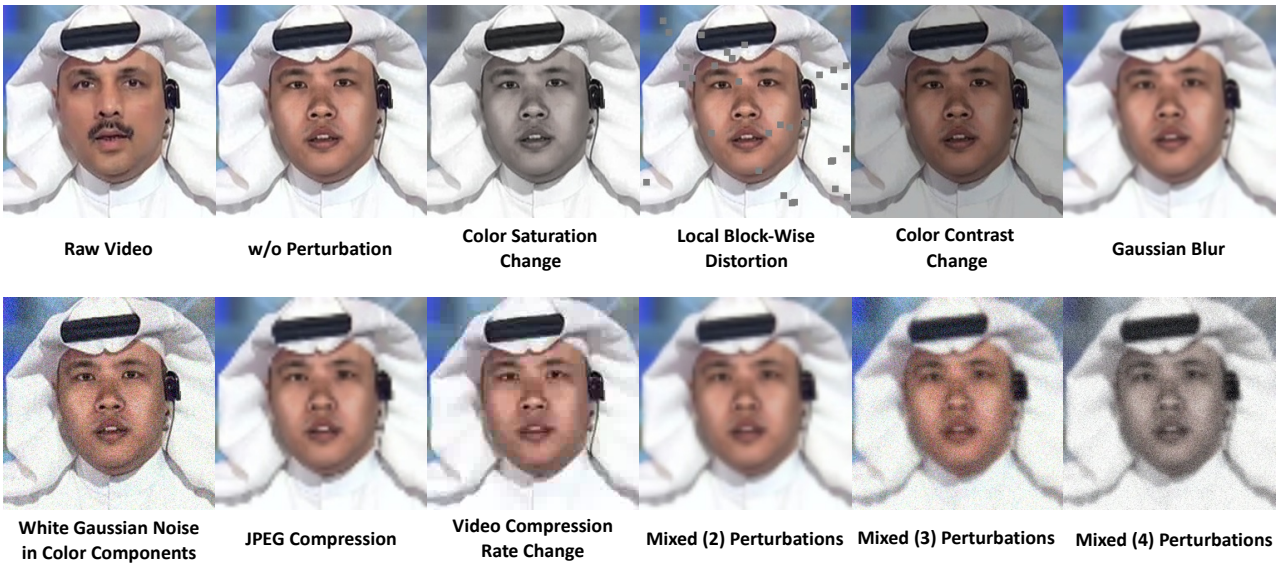

Figure 7: Seven types of perturbations and the mixture of two (Gaussian blur, JPEG compression) / three (Gaussian blur, JPEG compression, white Gaussian noise in color components) / four (Gaussian blur, JPEG compression, white Gaussian noise in color components, change of color saturation) perturbations in DeeperForensics-1.0.