Detecting Deepfakes with Self-Blended Images

Kaede Shiohara Toshihiko Yamasaki
The University of Tokyo
{shiohara, yamasaki}@cvm.t.u-tokyo.ac.jp

Abstract

In this paper, we present novel synthetic training data called self-blended images (SBIs) to detect deepfakes. SBIs are generated by blending pseudo source and target images from single pristine images, reproducing common forgery artifacts (e.g., blending boundaries and statistical inconsistencies between source and target images). The key idea behind SBIs is that more general and hardly recognizable fake samples encourage classifiers to learn generic and robust representations without overfitting to manipulation-specific artifacts. We compare our approach with state-of-the-art methods on FF++, CDF, DFD, DFDC, DFDCP, and FFIW datasets by following the standard cross-dataset and cross-manipulation protocols. Extensive experiments show that our method improves the model generalization to unknown manipulations and scenes. In particular, on DFDC and DFDCP where existing methods suffer from the domain gap between the training and test sets, our approach outperforms the baseline by 4.90% and 11.78% points in the cross-dataset evaluation, respectively. Code is available at https://github.com/mapooon/SelfBlendedImages.

1. Introduction

The recent rapid advancement of generative adversarial networks [10, 25, 31, 32, 45, 51, 63] (GAN) in computer vision has made it possible to generate realistic facial images. In particular, techniques called deepfake manipulating the identity, expression, or attributes of a subject are used for entertainment purposes, *e.g.*, smartphone applications or movies; however, they can also be used for malicious purposes, *e.g.*, to create fake news or to falsify evidence. Therefore, the vision community is keenly working on deepfake detection techniques.

Most previous detection methods [8, 16, 26, 30, 36, 48, 53, 64] perform well on the in-dataset scenario where they detect forgeries they learned in training; however, some studies [15, 21, 33, 61] have found that the detection performance significantly drops in the cross-dataset scenario where fake samples are forged by unknown manipulations.

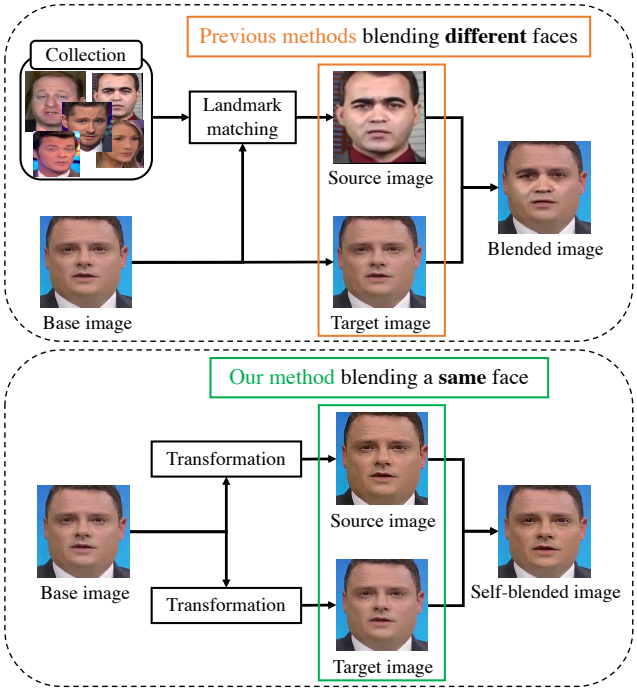
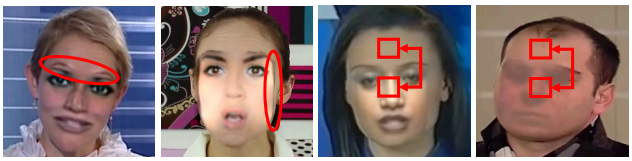


Figure 1. **Overview of fake sample synthesis.** Previous methods blend two distinct faces and generate artifacts based on a gap between selected source and target images. By contrast, our method blends slightly changed faces from a single image and generate artifacts actively by transformations. In this example, we apply a color jitter, sharpening, resize, and translation to the source image and no transformations to the target image.

One of the most effective solutions to this problem is to train models with synthetic data, which encourages models to learn generic representations for deepfake detection. For example, facial regions are blurred to reproduce a quality degradation of GAN-synthesized source images [41], blended images are generated from pairs of two pristine images to reproduce blending artifacts [39, 65]. However, the quality of deepfakes has improved over the years, which has caused the former method to fail on recent benchmarks [42, 52]. Although the latter methods perform well on some datasets [2, 42], low-quality videos in more challenging datasets [19, 20] where artifacts are hardly recog-



(a) Landmark (b) Boundary (c) Color (d) Frequency

Figure 2. **Typical artifacts on forged faces.** We classify artifacts into four types, (a) landmark mismatch, (b) blending boundary, (c) color mismatch, and (d) frequency inconsistency.

nizable owing to the high compression or extreme exposure lead them to unacceptable detection performance.

In this paper, we propose novel synthetic training data called self-blended images (SBIs) to detect deepfakes. The overviews of our method and previous methods [39, 65] are shown in Fig. 1. The key idea is that more hardly recognizable fake samples that contain common face forgery traces encourage models to learn more general and robust representations for face forgery detection. We analyze forged faces and define four typical artifacts motivated from previous works (*e.g.*, blending boundaries [39], source feature inconsistencies [65], and statistical anomalies in frequency domain [13]) as shown in Fig. 2. To synthesize these artifacts based on our key idea, we develop a source-target generator (STG) and mask generator (MG). STG generates pairs of pseudo source and target images from single pristine images using simple image processing, and MG generates various blending masks from facial landmarks of the input images. By blending the source and target images with the masks, we obtain SBIs. Training with SBIs encourages the models to learn generic representations because models learn the forgery traces we actively generate in STG. Moreover, our method improves training efficiency in terms of computational cost. Whereas successful previous works [39, 65] use landmark nearest search for source-target pair selection, which is computationally expensive, SBIs are generated without this process. Therefore, our method does not suffer from the large dataset size problem.

We evaluate our approach following the two evaluation protocols, cross-dataset evaluation and cross-manipulation evaluation. In the cross-dataset evaluation, we train our model on FF++ [52] and evaluate it on CDF [42], DFD [2], DFDC [19], DFDCP [20], and FFIW [67]. This experimental setting is similar to that in real detection scenarios where defenders are exposed to unseen domains. Our approach surpasses or is at least comparable to the state-of-the-art methods on all test sets despite its simplicity. Especially, on DFDC and DFDCP where previous methods suffer from domain gaps between the training and test sets, our method outperforms the state-of-the-art unsupervised baseline [65] by 4.90% and 11.78% points, respectively. In the cross-manipulation evaluation, we evaluate the generality of our model on unseen manipulation methods of FF++; DF [4], F2F [56], FS [5], and NT [55]. Our approach achieves the AUC of 99.99%, 99.88%, 99.91%, and 98.79% on DF, F2F, FS, and NT, respectively. Although the performance on FF++ becomes saturated, our method still outperforms the state of the art on whole FF++ (99.64% vs. 99.11%).

2. Related Work

Deepfake Detection. Although many detection methods have been introduced, the development of optimal convolutional neural networks (CNNs) has been a primary topic of research (*e.g.,* an efficient shallow network [8], multi-task autoencoders [21, 48], capsule network [49], recurrent convolutional networks [26, 53], and attentional networks [16, 64]). Some studies [23, 37, 43, 44, 50] focus on the frequency domain to capture forgery traces more effectively. These methods achieve impressive performance on highly compressed videos. Another notable direction is focusing on specific representations (*e.g.*, head pose [62], eye blinking [30, 40], mouth movements [27], neuron behaviors [60], optical flow [9], and steganalysis features [24]). Face X-ray [39] introduces a facial representation based on boundaries between altered faces and background images. PCL [65] measures patch-wise similarities of input images to detect inconsistencies between source and target images.

Training Data Synthesis. Although most existing methods perform well in detecting known manipulations, some studies [15, 21, 33, 61] have found that the methods cannot be generalized to fake faces forged by unknown manipulations because they tend to overfit to method-specific artifacts seen in training. One of the most effective approaches to address this problem is training models with synthetic data; this encourages models to learn generic features for face forgery detection. FWA [41] focuses on a quality gap between GAN-synthesized faces and natural faces, and reproduces it on real images by blurring facial regions. However, deepfake techniques have improved over the years and this method fails in detecting forgeries on the recent benchmarks [2, 52]. BI [39] and I2G [65] are introduced to generate blended faces which reproduce blending artifacts from pairs of two pristine images with similar facial landmarks.

These blended images work well as fake samples to train more general detection models; however, some concerns remain. First, because these blending artifacts depend on pairs of source and target images selected by the landmark matching, irregular swaps [57] are sometimes seen in the generated images. It is possible that these easy samples prevent models from learning robust representations. Second, because these methods are introduced to learn the oriented representations *i.e.*, the blending boundary in BI and the source feature consistency in I2G, it is possible that artifacts to be learned for robust deepfake detection are not sufficient only for the artifacts in the blended images.

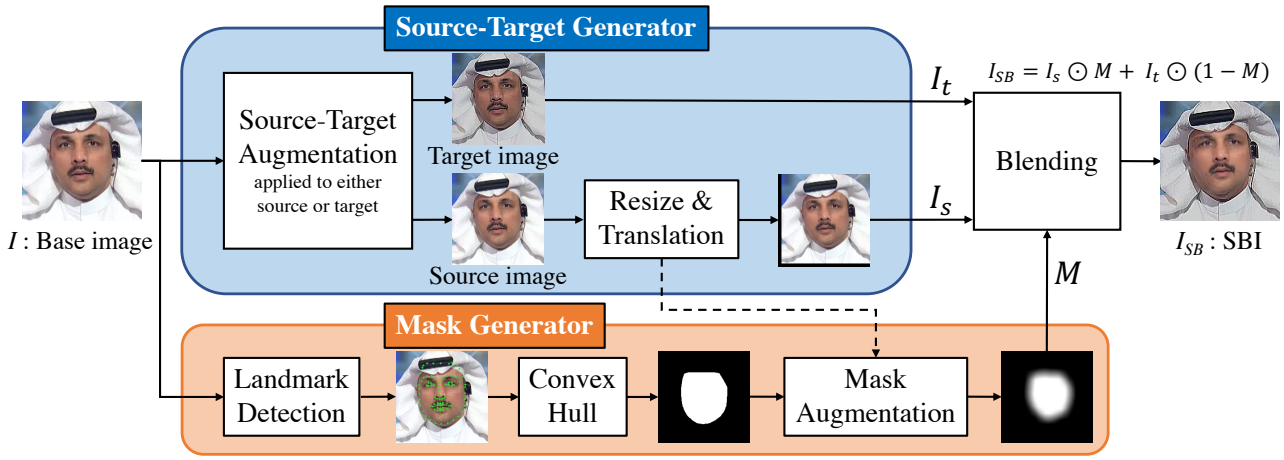


Figure 3. **Overview of generating an SBI.** A base image I is input into the source-target generator (STG) and mask generator (MG). STG generates pseudo source and target images from the base image using some image transformations, whereas MG generates a blending mask from the facial landmarks and deforms it to increase the diversity of the mask. Finally, the source and target images are blended with the mask.

3. Self-Blended Images (SBIs)

Our goal is to detect statistical inconsistencies between altered faces and background images on deepfakes. To train more general and robust detectors, we generate synthetic fake samples that consist of common forgery traces, and are difficult to recognize. Our key observation is that, if deepfake generation techniques continue to improve, GAN-synthesized source images will be even closer to pristine target images in their properties, *e.g.*, facial landmarks and pixel statistics. Therefore, we develop a synthetic data generation pipeline where a fake image is generated by blending pseudo source and target images from a single image to give models a more general and difficult task for face forgery detection.

To achieve this, we introduce self-blended images (SBIs). As shown in Fig. 3, an SBI is generated by three steps; (1) A **source-target generator** generates pseudo source and target images for blending. The source and target images are augmented to generate statistical inconsistencies (*e.g.*, color and frequency) between them. The source image is also resized and translated to reproduce blending boundaries and landmark mismatches. (2) A **mask generator** generates a gray-scale mask image with some deformations. (3) We **blend** the source and target images with the mask to obtain an SBI. Although the general flow of an SBI generation is illustrated in Fig. 3, we show the pseudocode in Alg. 1 where the procedure is slightly different from that in Fig. 3 for training efficiency (*e.g.*, facial landmarks are extracted in preprocess but not in training). Our pipeline to generate an fake sample has a constant running time regardless of the dataset size while previous methods [39, 65] have a running time of $O(NK)$ in the preprocess due to the pair selection for source and target images,

Algorithm 1 Pseudocode for SBIs Generation

Input: Base image I of size $(H, W, 3)$, facial landmarks L of size $(81, 2)$
Output: Self-blended image I_{SB} of size $(H, W, 3)$

def $\mathcal{T}(I)$ **:** ▷ Source-Target Augmentation
  $I \leftarrow \text{ColorTransform}(I)$
  $I \leftarrow \text{FrequencyTransform}(I)$
  return I
if $\text{Uniform}(\min = 0, \max = 1) < 0.5$ **:**
  $I_s, I_t \leftarrow \mathcal{T}(I), I$
else :
  $I_s, I_t \leftarrow I, \mathcal{T}(I)$
$I_s, p \leftarrow \text{RandomResizeTranslate}(I_s)$ ▷ p : Parameter
$L \leftarrow \text{LandmarkTransform}(L)$
$M \leftarrow \text{ConvexHull}(L)$
$M \leftarrow \text{ParameterizedResizeTranslate}(M, p)$
$M \leftarrow \text{MaskDeform}(M)$
$r \leftarrow \text{Uniform}(\{0.25, 0.5, 0.75, 1, 1, 1\})$
$M \leftarrow rM$
$I_{SB} \leftarrow I_s \odot M + I_t \odot (1 - M)$

where N and K are the number of videos and the number of frames of each, respectively [1].

3.1. Source-Target Generator (STG)

Given an input image I, STG initializes pseudo source and target images by copying I. To generate statistical inconsistencies between source and target images, STG randomly applies some image transformations to either of them. Here, we randomly shift the values of rgb channels, hue, saturation, value, brightness, and contrast of input

[1] Because the official source code of [39, 65] is not publicly available, we only discuss qualitatively.

Figure 4. **Samples of pristine images (top row) and their SBIs (bottom row).**

images as color transformations. Then we downsample or sharpen input images as frequency transformations.

To reproduce blending boundaries and landmark mismatches, STG **resizes** the source image. Let the height and width of I be H and W, respectively. We define the height H_r and width W_r of the resized image as $H_r = u_h H$ and $W_r = u_w W$, where u_h and u_w are sampled independently from a continuous uniform distribution $U_{[u_{\text{min}},\ u_{\text{max}}]}$ in the range $[u_{\text{min}},\ u_{\text{max}}]$. The resized image is zero-padded or center-cropped to have the same size as the original. Then, STG **translates** the resized source image. We define a translation vector $\boldsymbol{t} = [t_h, t_w]$ as $t_h = v_h H$ and $t_w = v_w W$, where v_h and v_w are sampled independently from $U_{[v_{\text{min}},\ v_{\text{max}}]}$.

3.2. Mask Generator (MG)

MG provides a gray scale mask image to blend source and target images. To perform this, MG applies a landmark detector to the input image to predict a facial region and initializes a mask by calculating convex hull from predicted facial landmarks. Then the mask is deformed with the landmark transformation as used in BI [39]. To increase the diversity of blending masks, the shape of masks and blending ratio are randomly changed. First, the mask is deformed by elastic deformation as adopted in [65]. Second, the mask is smoothed by two Gaussian filters with different parameters. After the first smoothing, the pixel values less than 1 are changed to 0. This means that the mask is eroded if the kernel size of the first Gaussian filter is larger than that of the second one and is dilated in the opposite case. Finally, MG varies the blending ratio of the source image. This can be achieved by multiplying the mask image by a constant $r \in (0, 1]$. Here, we uniformly sample r from $\{0.25, 0.5, 0.75, 1, 1, 1\}$.

3.3. Blending

By blending the source image I_s and the target image I_t with the blending mask M, we obtain the self-blended image I_{SB} as

$$I_{SB} = I_s \odot M + I_t \odot (1 - M). \quad (1)$$

We show some representative examples of SBIs in Fig. 4. Although the purpose of SBIs is not for counterfeiting, they contain artifacts seen in forged faces.

3.4. Training with SBIs

Once SBIs are generated, we can train any binary classifier, regardless of whether it is designed for deepfake detection or not. Given input images $X = [x_0, x_1, \cdots, x_{N-1}]$ of size $(N, H, W, 3)$ and the corresponding binary labels $T = [t_0, t_1, \cdots, t_{N-1}]$ of size N, a classifier F is optimized on the binary cross-entropy loss L as follows:

$$L = -\frac{1}{N}\sum_{i=0}^{N-1}\{t_i \log F(x_i) + (1+t_i)\log(1-F(x_i))\}, \quad (2)$$

where $F(x)$ is the probability of x being "Fake". We input target images as "Real" instead of using the base images to encourage the models to focus only on artifacts on SBIs. Because MG provides blending masks, we can also adopt mask-based multi-task learning [39, 48, 65].

4. Experiments

4.1. Implementation Details

Preprocess. We adopt Dlib [34] and RetinaFace [18] to extract facial landmarks and bounding boxes from each video frame, respectively. We use an 81 facial landmarks shape predictor [1] in Dlib. For the width and height of the face calculated from the bounding box, the face region is cropped with a random margin of 4–20% for training and a fixed value of 12.5% for inference. Note that the landmarks are not needed during inference; hence we only use RetinaFace at the inference time.

Source-Target Augmentation. For the color and frequency transformations, we adopt RGBShift, HueSaturationValue, RandomBrightnessContrast, Downscale, and Sharpen from a widely used image processing toolbox [11].

Training. We adopt the state-of-the-art convolutional network architecture EfficientNet-b4 [54] (EFNB4) pre-trained on ImageNet [17] as the classifier and train it for 100 epochs with the SAM [22] optimizer. The batch size and learning rate are set to 32 and 0.001, respectively. We sample only eight frames per video for training. If two or more faces are detected in a frame, the face with the largest area of the face bounding box is extracted. Each batch consists of real images and their SBIs, and the same augmentation is applied to each real image and its SBI. We also use some data augmentations, *i.e.*, ImageCompression, RGBShift, HueSaturationValue, and RandomBrightnessContrast.

Model Validation. Considering practical situations, it is important to validate the model without additional evaluation datasets. We use a validation set that consists of real

Method	Input Type	Training Set		Test Set AUC (%)				
		Real	Fake	CDF	DFD	DFDC	DFDCP	FFIW
DSP-FWA [41]	Frame	✓	✓	69.30	-	-	-	-
Face X-ray + BI [39]	Frame	✓		-	93.47	-	71.15	-
Face X-ray + BI [39]	Frame	✓	✓	-	95.40	-	<u>80.92</u>	-
LRL [13]	Frame	✓	✓	78.26	89.24	-	76.53	-
FRDM [44]	Frame	✓	✓	79.4	91.9	-	79.7	-
PCL + I2G [65]	Frame	✓		<u>90.03</u>	**99.07**	67.52	74.37	-
Two-branch [47]	Video	✓	✓	76.65	-	-	-	-
DAM [67]	Video	✓	✓	75.3	-	-	72.8	-
LipForensics [27]	Video	✓	✓	82.4	-	-	-	-
FTCN [66]	Video	✓	✓	86.9	94.40*	<u>71.00</u>*	74.0	<u>74.47</u>*
EFNB4 + SBIs (Ours)	Frame	✓		**93.18**	<u>97.56</u>	**72.42**	**86.15**	**84.83**

Table 1. **Cross-dataset evaluation on CDF, DFD, DFDC, DFDCP, and FFIW.** The results of prior methods are directly cited from the original paper and their subsequences for fair comparison. Bold and underlined values correspond to the best and the second-best value, respectively. * denotes our experiments with the official code. Our method outperforms state-of-the-art methods on CDF, DFDC, DFDCP, and FFIW, and achieves the second best on DFD without any special network architecture for deepfake detection.

videos and their SBIs after each epoch, and select the weight with the highest number of epochs among the five weights with the highest AUC. Therefore, no manipulated images are used even for the model validation in our approach.

Inference Strategy. We sample 32 frames per video for inference. If two or more faces are detected in a frame, the classifier is applied to all faces and the highest fakeness confidence is used as the predicted confidence for the frame. Once the predictions for all frames are obtained, we average them to get the prediction for the video. For fair comparison, we use all videos of all test sets for evaluation by setting the confidences to 0.5 for the videos where no face is detected in all frames.

4.2. Experimental Setting

Datasets. We adopt the widely used benchmark **FaceForensics++** [52] (FF++) for training, following the convention. It contains 1,000 original videos and 4,000 fake videos forged by four manipulation methods, *i.e.*, Deepfakes [4] (DF), Face2Face [56] (F2F), FaceSwap [5] (FS), and NeuralTextures [55] (NT). For our cross-dataset evaluation, we use five recent deepfake datasets. **Celeb-DF-v2** [42] (CDF) applies a more advanced deepfake technique to celebrity videos downloaded from YouTube. **DeepFakeDetection** [2] (DFD) provides thousands of deepfake videos generated with consenting actors. **DeepFake Detection Challenge Preview** [20] (DFDCP) and **DeepFake Detection Challenge** public test set [19] (DFDC), that are released along with the competition [3], contain a lot of disturbed videos, *e.g.*, compression, downsampling, and noise. We further provide a novel cross-dataset baseline on a more recent large scale benchmark **FFIW-10K** [67] (FFIW) which focuses on multi-person scenario. We follow the official train/test splits for all datasets except FFIW where we use the original validation set as our test set because the official test set has not been released yet. Although FaceShifter [38] and DeeperForensics-1.0 [29] provide sophisticated deepfake videos, we do not adopt them in our cross-dataset evaluation because they generate deepfakes from the real videos of FF++ that is the same domain as used in training. See the supplementary material for more statistical details.

Frame-Level Baselines. We refer to five state-of-the-art frame-level detection methods, including: (1) **DSP-FWA** [41] proposed a training data generation method based on the degradation of the GAN-synthesized source image quality. (2) **Face X-ray** [39] detects deepfakes via segmenting blending boundaries between source and target images. The model is trained with synthetic fake samples called **BI** generated by blending two images from different videos. (3) **Local relation learning** [13] (LRL) and (4) **Fusion + RSA + DCMA + Multi-scale** [44] (FRDM) fuse two different representations from RGB and frequency domains. (5) **Pair-wise self-consistency learning** [65] (PCL) detects deepfakes via measuring consistencies between patches of input images. The model is trained with **inconsistency image generator** (I2G) that is similar to BI [39].

Video-Level Baselines. We further compare our method with video-level methods that output a single scalar fakeness for some video frames. Unlike frame-level methods, video-level methods can detect incoherence across frames, although they require multiple frames of the subject at regular intervals. We refer to four state-of-the-art methods, including: (1) **Two-branch** [47] proposes Laplacian of Gaus-

Method	Test Set AUC (%)				
	DF	F2F	FS	NT	FF++
Face X-ray + BI [39]	99.17	98.57	98.21	98.13	98.52
PCL + I2G [65]	**100**	98.97	99.86	97.63	99.11
EFNB4 + SBIs (Ours)	99.99	**99.88**	**99.91**	**98.79**	**99.64**

Table 2. **Cross-manipulation evaluation on FF++.** Our method achieves state-of-the-art results on F2F, FS, NT, and whole FF++.

sian kernels to enhance the frequency component of the input image. (2) **Discriminative attention model** [67] (DAM) proposes an attention [59]-based network for multi-person scenarios. (3) **LipForensics** [27] detects temporal inconsistencies of mouth movements using a pretrained lip-reading model [46]. (4) **Fully temporal convolution network** [66] (FTCN) enhances temporal representations by reducing spatial kernel size of CNN to 1.

Evaluation Metrics. We report the video-level area under the receiver operating characteristic curve (AUC) to compare with prior works. Typically, frame-level predictions are averaged over video frames. We additionally provide average precision (AP) in the supplementary material.

4.3. Cross-Dataset Evaluation

To show the generality of our method, we conduct a cross-dataset evaluation where models are trained on FF++ and evaluated on other datasets. Although many researchers have considered this task, the test sets used by each of them in their experiments vary from work to work, making comprehensive comparisons difficult. We, therefore, examine the experimental settings in previous works carefully and compile them into Table 1.

Comparison with Frame-Level Methods. Here, we compare our method with other frame-level methods [13, 39, 41, 44, 65]. Our approach outperforms the state-of-the-art methods on CDF, DFDC, and DFDCP by 6.08%, 5.17%, and 5.23% points, respectively, and improves the baseline by 4.58% points on average (87.33% vs. 82.75%). Our result is comparable with PCL + I2G [65] on DFD (97.56% vs. 99.07%), where a forged face is sometimes placed with some other pristine faces in a manipulated frame, and the percentage of frames that subject is throughout manipulated videos is smaller than other test sets. Therefore, our method can be improved by incorporating any object tracking process into our inference strategy as in PCL + I2G [65], instead of extracting frames from the video at equal intervals as in our simple strategy.

Comparison with Video-Level Methods. We then compare our method with video-level methods [27, 47, 66, 67]. For more comprehensive comparison, we conduct additional experiments for FTCN [66] on unconsidered test sets, *i.e.*, DFD, DFDC, and FFIW with officially released code [6]. The results are denoted as * in Table 1. Our method still outperforms the state of the art by 6.28%, 3.16%, 1.42%, 12.15%, and 10.36% points on CDF, DFD, DFDC, DFDCP, and FFIW, respectively, and improves the baseline by 6.68% points on average (86.83% vs. 80.15%). We also evaluate our method on a subset of DFDC which is used in an experiment for LipForensics [27], outperforming the competitor (76.78% vs. 73.5%). The video list is available at the author's repository [7].

Method	Test Set AUC (%)				
	DF	F2F	FS	NT	FF++
Xception + BI [39]	98.95	97.86	89.29	97.29	95.85
Xception + SBIs (Ours)	**99.99**	**99.90**	**98.79**	**98.20**	**99.22**

Table 3. **AUC comparison with BI [39].**

Method	Test Set AUC (%)			
	CDF	DFDC	DFDCP	Avg
ResNet-34 + I2G [65]	78.18	51.72	69.93	66.61
ResNet-34 + SBIs (Ours)	**87.04**	**66.41**	**82.16**	**78.54**

Table 4. **AUC comparison with I2G [65].**

4.4. Cross-Manipulation Evaluation

In real detection situations, the defenders generally are not aware of the attacker's forgery methods. For this reason, it is important to verify the model generalization to various forgery methods. Following the evaluation protocol used in [39, 65], we evaluate our model on four manipulation methods of FF++, *i.e.*, DF, F2F, FS, and NT. We use the raw version for evaluation as well as the competitors.

Table 2 presents our cross-manipulation evaluation result on FF++. Our method outperforms or nearly equals the existing methods on four manipulations (99.99% on DF, 99.88% on F2F, 99.91% on FS, and 98.79% on NT) and achieves the best performance on the whole FF++ (99.64% vs. 99.11%). This result shows that our method works well not only on deepfakes but also on other face manipulations.

4.5. Data Quality Assessment

Here, we compare our method with state-of-the-art synthetic training data [39, 65], removing influences of the difference of the classifiers. To achieve this, we train the same models and optimizer as the ones competitors use in their original papers. Table 3 presents the comparison with BI [39]. We train Xception [14] with Adam [35] optimizer. Our method outperforms BI [39] on all the manipulation methods of FF++ in terms of AUC. In particular, the baseline on FS is improved from 89.29% to 98.79%. Next, the result of the comparison with I2G [65] is given in Table 4. We train ResNet-34 [28] with Adam optimizer. Our method outperforms I2G [65] on CDF, DFDC, and DFDCP

Process	Test Set AUC (%)				
	FF++	CDF	DFDCP	FFIW	Avg
w/o Source aug.	98.58	**93.59**	78.06	61.11	82.84
w/o Target aug.	99.35	76.61	83.84	82.87	85.67
w/o S-T aug.	89.18	70.68	85.16	**88.31**	83.33
w/o Res.&Trans.	99.58	85.28	81.04	74.69	85.15
SBIs (Ours)	**99.64**	93.18	**86.15**	84.83	**90.95**

Table 5. **Effect of each process of STG.** The skipping of any process causes a fatal performance degradation.

Training Set		Test Set AUC (%)				
Database	#Real	FF++	CDF	DFDCP	FFIW	Avg
FF++	720	99.64	93.18	86.15	84.83	90.95
CDF	622	98.10	93.74	81.10	77.82	87.69
DFDCP	737	98.76	90.79	88.70	81.31	89.89
FFIW	7090	99.72	95.57	78.91	88.07	90.57

Table 6. **Performance of different training datasets.** Our method achieves good results on each training dataset. "#Real" presents the number of real videos of the training set, excluding that of the validation set.

and their average by 8.86%, 14.69%, 12.23%, and 11.93% points, respectively. These results clearly show our method is superior to the competitors as synthetic training data, regardless of the network architecture.

4.6. Ablations

Effect of Each Process of STG. In STG, we use some image processing to generate pseudo source and target images. Conversely, because learned representations are based on the artifacts we actively provide in STG, ablation experiments of the generation process enable the exploration of effective clues on the deepfake benchmarks. Here, we train our model without some processes, *i.e.*, the source augmentation, target augmentation, source-target augmentation, or resize and translation, and evaluate them on FF++, CDF, DFDCP, and FFIW. As shown in Table 5, source and target augmentation is indeed effective in detecting deepfakes, and both of them are necessary for better performance. We also observe that the resize and translation reproduce important artifacts because of the poor performance without them. Through the ablation, it can be concluded that different clues are useful to detectors on different datasets because they have different deepfake generation processes.

Generality to Training Datasets. It is important from a practical standpoint to show that our method can perform well on various real face datasets. We here train models with SBIs from the pristine videos of FF++, CDF, DFDCP, and FFIW. Then we evaluate them on the test sets. On CDF and FFIW, we split the original training sets into the alternative training/validation sets. Table 6 presents the result.

Architecture	Test Set AUC (%)				
	FF++	CDF	DFDCP	FFIW	Avg
ResNet-50	97.77	90.66	82.88	79.30	87.65
ResNet-152	98.33	90.71	85.01	76.43	87.62
Xception	99.26	90.27	78.85	76.72	86.28
EfficientNet-b1	99.10	91.16	84.58	80.23	88.77
EfficientNet-b4	**99.64**	**93.18**	**86.15**	**84.83**	**90.95**

Table 7. **Performance of different network architectures.** An architecture with larger capacity tends to result in better generality.

Our method is generalized to all datasets without a critical performance drop. We observe the large dataset size of FFIW contributes to the model generality. However, the difference of video scene between FFIW and DFDCP leads to a slight performance drop on DFDCP; FFIW consists of videos collected from YouTube, whereas DFDCP consists of videos made by filming recruited subjects. The result also indicates that learning pristine videos can help detect forged faces in the same domain as that in training, even if models did not learn manipulated videos, as indicated by the scores highlighted in brown in Table 6, which supports our not adopting FaceShifter [38] and DeeperForensics-1.0 [29] in the cross-dataset evaluation, as mentioned in Section 4.2.

Choice of Network Architecture. Although we adopt EfficientNet-b4 [54] as our standard classifier, our method can be applied to other network architectures. Here, we investigate the performance of different state-of-the-art architectures, *i.e.*, ResNet-50, -152 [28], Xception [14], EfficientNet-b1, and -b4 [54] trained with SBIs. As shown in Table 7, all architectures achieve good results on FF++, CDF, DFDCP, and FFIW without critical performance degradation. Remarkably, even our method with a vanilla ResNet-50 outperforms all the previous methods on CDF, DFDCP, and FFIW as shown in Tables 1 and 7. We observe larger networks tend to result in greater generality, which indicates SBIs provide a variety of training samples.

4.7. Qualitative Analysis

To obtain qualitative insights, we visualize model saliency maps and feature spaces. Through the analysis, we use two models; one is trained on FF++ (baseline) and the other is trained on SBIs (our model).

Saliency Map. To visualize where the models are paying their attention on the forged faces, we apply Grad-CAM++ [12] to the models on manipulated frames of FF++, *i.e.*, DF, F2F, FS, and NT, as shown in Fig. 5. It can be observed that our method encourages the model to make its attentions sparser than the baseline. This is because our model detects minor artifacts independent of manipulations, e.g., blending boundaries, while the baseline captures method-specific pixel distributions that are widely spread in the forged faces.

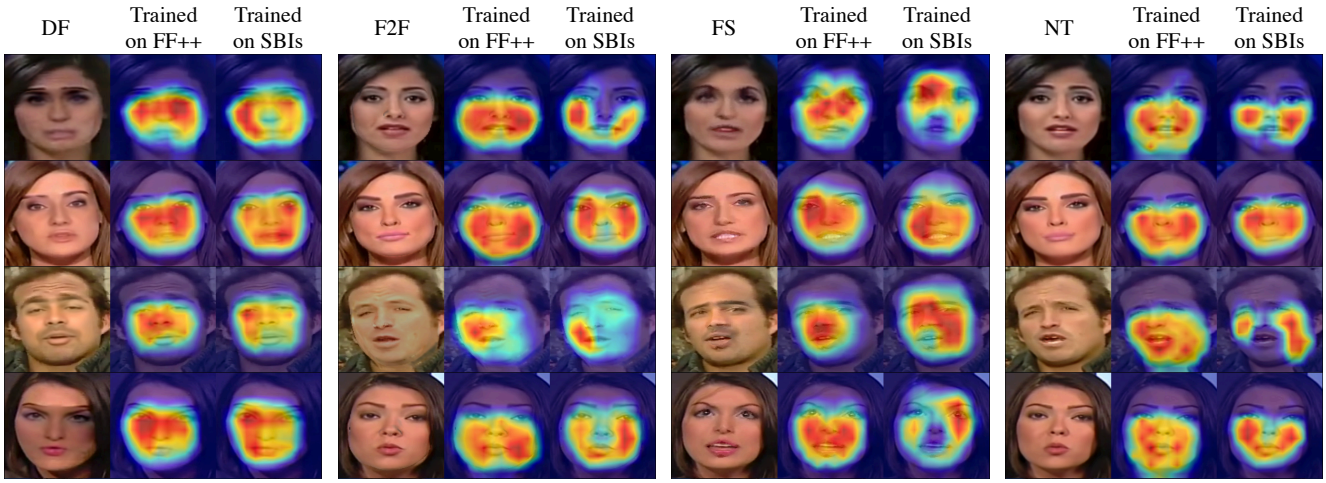
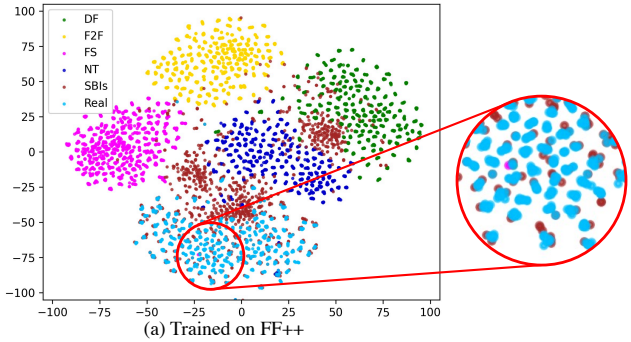
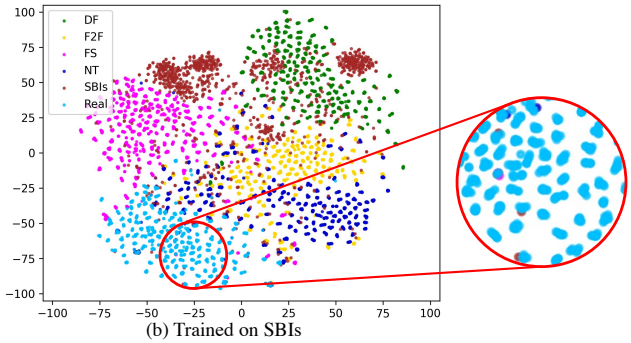


Figure 5. **Saliency map visualization of the baseline and our model.** The baseline captures method-specific artifacts that are widely present across forged faces while our model detects minor artifacts independent of manipulations. Best viewed in color.



(a) Trained on FF++



(b) Trained on SBIs

Figure 6. **Feature space visualization of the baseline (a) and our model (b).** The baseline cannot distinguish real images from SBIs (because the feature vectors fall into the same feature space) while our model succeeds in distinguishing real images from not only SBIs but also forged images. Best viewed in color.

Feature Space. We then apply t-SNE [58] visualization to feature vectors from the last layers of the models. We emphasize again that it is easy for the baseline to recognize the forged faces because they are seen in its training, and that our goal is to separate real faces from others, not to classify types of manipulations. As shown in Fig. 6, the baseline cannot distinguish SBIs from real images although it clusters four manipulations seen in training. On the other hand, our model distinguishes not only SBIs but also forged faces from real ones. We also observe that SBIs are distributed all over the four manipulations in the feature space. These results indicate that SBIs are general synthetic data to train face forgery detectors.

5. Limitations

Although our results in cross-dataset and cross-manipulation evaluations are expected to be beneficial, we observe some limitations of our method. First, similar to other frame-level methods, our model cannot capture temporal inconsistencies across video frames. Therefore, sophisticated deepfake generation techniques with fewer spatial artifacts may pass our detector. Moreover, our method does not perform well on whole-image synthesis because we define a "fake image" as an image where the face region or background is manipulated. We evaluate our model on a 20k image set sampled from FFHQ dataset and StyleGAN [32] synthesis, and its AUC is only 69.11%.

6. Conclusion

In this paper, we proposed a novel synthetic training data, self-blended images (SBIs), based on the idea that more general and hardly recognizable fake samples encourage classifiers to learn more generic and robust representations. SBIs are generated by blending pseudo source and target images that are slightly transformed from single real images to reproduce forgery artifacts. Using SBIs, we could train detectors without forged face images. Extensive experiments show that our method is superior to state-of-the-art methods for unseen manipulations and scenes, and generalized to different network architectures and training datasets.

References

[1] 81 facial landmarks shape predictor. https://github.com/codeniko/shape_predictor_81_face_landmarks. Accessed: 2021-11-13. 4

[2] Contributing data to deepfake detection research. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html. Accessed: 2021-11-13. 1, 2, 5

[3] Deepfake detection challenge. https://www.kaggle.com/c/deepfake-detection-challenge. Accessed: 2021-11-13. 5

[4] Deepfakes. https://github.com/deepfakes/faceswap. Accessed: 2021-11-13. 2, 5

[5] Faceswap. https://github.com/MarekKowalski/FaceSwap/. Accessed: 2021-11-13. 2, 5

[6] Ftcn. https://github.com/yinglinzheng/FTCN. Accessed: 2021-11-13. 6

[7] Lipforensics. https://github.com/ahaliassos/LipForensics. Accessed: 2021-11-13. 6

[8] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *WIFS*, pages 1–7, 2018. 1, 2

[9] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *ICCV*, 2019. 2

[10] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017. 1

[11] Alexander Buslaev, Vladimir Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11:125, 02 2020. 4

[12] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, pages 839–847, 2018. 7

[13] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. Local relation learning for face forgery detection. In *AAAI*, volume 35, pages 1081–1088, 2021. 2, 5, 6

[14] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1251–1258, 2017. 6, 7

[15] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv:1812.02510*, 2018. 1, 2

[16] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *CVPR*, 2020. 1, 2

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 4

[18] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv:1905.00641*, 2019. 4

[19] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge dataset. *arXiv:2006.07397*, 2020. 1, 2, 5

[20] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv:1910.08854*, 2019. 1, 2, 5

[21] Mengnan Du, Shiva Pentyala, Yuening Li, and Xia Hu. Towards generalizable deepfake detection with locality-aware autoencoder. In *ACM CIKM*, pages 325–334, 2020. 1, 2

[22] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021. 4

[23] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *ICML*, pages 3247–3258, 2020. 2

[24] Miroslav Goljan and Jessica Fridrich. Cfa-aware features for steganalysis of color images. *SPIE*, 9409, 2015. 2

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1

[26] D. Güera and E. J. Delp. Deepfake video detection using recurrent neural networks. In *AVSS*, pages 1–6, 2018. 1, 2

[27] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *CVPR*, pages 5039–5049, 2021. 2, 5, 6

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6, 7

[29] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. DeeperForensics-1.0: A large-scale dataset for real-world face forgery detection. In *CVPR*, 2020. 5, 7

[30] T. Jung, S. Kim, and K. Kim. Deepvision: Deepfakes detection using human eye blinking pattern. *IEEE Access*, 8:83144–83154, 2020. 1, 2

[31] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 1

[32] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 1, 8

[33] Ali Khodabakhsh, Raghavendra Ramachandra, Kiran Raja, Pankaj Wasnik, and Christoph Busch. Fake face detection methods: Can they be generalized? In *BIOSIG*, pages 1–6, 2018. 1, 2

[34] Davis E. King. Dlib-ml: A machine learning toolkit. *JMLR*, 10:1755–1758, 2009. 4

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[36] Prabhat Kumar, Mayank Vatsa, and Richa Singh. Detecting face2face facial reenactment in videos. In *WACV*, 2020. 1

[37] Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *CVPR*, pages 6458–6467, 2021. 2

[38] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *CVPR*, 2020. 5, 7

[39] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6

[40] Y. Li, M. Chang, and S. Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *WIFS*, 2018. 2

[41] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *CVPR Workshops*, 2019. 1, 2, 5, 6

[42] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *CVPR*, 2020. 1, 2, 5

[43] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *CVPR*, pages 772–781, 2021. 2

[44] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *CVPR*, pages 16317–16326, 2021. 2, 5, 6

[45] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pages 2813–2821, 2017. 1

[46] Brais Martinez, Pingchuan Ma, Stavros Petridis, and Maja Pantic. Lipreading using temporal convolutional networks. In *ICASSP*, pages 6319–6323, 2020. 6

[47] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. In *ECCV*, pages 667–684, 2020. 5, 6

[48] Huy H. Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *BTAS*, 2019. 1, 2, 4

[49] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Use of a capsule network to detect fake images and videos. *arXiv:1910.12467*, 2019. 2

[50] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, pages 86–103, 2020. 2

[51] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434*, 2015. 1

[52] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, 2019. 1, 2, 5

[53] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and P. Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. In *CVPR Workshops*, 2019. 1, 2

[54] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 4, 7

[55] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM TOG*, 2019. 2, 5

[56] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, pages 2387–2395, 2016. 2, 5

[57] Loc Trinh and Yan Liu. An examination of fairness of ai models for deepfake detection. In *IJCAI*, pages 567–574, 2021. 2

[58] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(86):2579–2605, 2008. 8

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 6

[60] Run Wang, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yihao Huang, Jian Wang, and Yang Liu. Fakespotter: A simple yet robust baseline for spotting ai-synthesized fake faces. In *IJCAI*, pages 3444–3451, 2020. 2

[61] Xinsheng Xuan, Bo Peng, Wei Wang, and Jing Dong. On the generalization of gan image forensics. In *CCBR*, pages 134–141, 2019. 1, 2

[62] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP*, pages 8261–8265, 2019. 2

[63] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, pages 7354–7363, 2019. 1

[64] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *CVPR*, 2021. 1, 2

[65] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *ICCV*, pages 15023–15033, 2021. 1, 2, 3, 4, 5, 6

[66] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *ICCV*, pages 15044–15054, 2021. 5, 6

[67] Tianfei Zhou, Wenguan Wang, Zhiyuan Liang, and Jianbing Shen. Face forensics in the wild. In *CVPR*, 2021. 2, 5, 6