Pattern Recognition 135 (2023) 109179

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Watching the BiG artifacts: Exposing DeepFake videos via Bi-granularity artifacts☆

Han Chen[a,1], Yuezun Li[b,1], Dongdong Lin[a], Bin Li[a,c,*], Junqiang Wu[a]

[a] Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China
[b] Ocean University of China, Qingdao 266000, China
[c] Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518129, China

ARTICLE INFO

Article history:
Received 27 July 2021
Revised 2 October 2022
Accepted 10 November 2022
Available online 13 November 2022

Keywords:
Multimedia forensics
Deepfake detection
Granularity artifacts
Multi-task learning

ABSTRACT

Recent years have witnessed significant advances in AI-based face manipulation techniques, known as DeepFakes, which has brought severe threats to society. Hence, an emerging and increasingly important research topic is how to detect DeepFake videos. In this paper, we propose a new DeepFake detection method based on Bi-granularity artifacts (BiG-Arts). We observe that the most of DeepFake video generation can commonly introduce bi-granularity artifacts: the intrinsic-granularity artifacts and extrinsic-granularity artifacts. Specifically, the intrinsic-granularity artifacts are caused by a common series of operations in model generation such as up-convolution or up-sampling, while the extrinsic-granularity artifacts are introduced by a common step in post-processing that blends the synthesized face to original video. To this end, we formulate DeepFake detection as multi-task learning problem, to simultaneously predict the intrinsic and extrinsic artifacts. Benefiting from the guidance of detecting Bi-granularity artifacts, our method is notably boosted in both within-datasets and cross-datasets scenarios. Extensive experiments are conducted on several DeepFake datasets, which corroborates the superiority of our method. Our method has been contributed as a part of the solution to achieve the Top-1 rank in DFGC competition (https://competitions.codalab.org/competitions/29583).

1. Introduction

With the rapid development of deep generative models [1,2], face forgery techniques [3–6] have shown significant progress. One typical technique known as DeepFake draws increasingly attentions due to its highly-synthesized realism and easy deployment. DeepFake can swap a face of source identity in authentic video with a synthesized face of target identity, while retains the consistent facial attributes such as expression and head pose. Hence, the malicious abuse of DeepFake technique can enable attackers to forge human activities that do not occur in reality [7], causing severe threats to the societal security and trustfulness. Therefore, it is of paramount importance to develop DeepFake detections.

To mitigate the risk, various methods [8–11] have been proposed to expose the DeepFake videos. These methods rely on different clues such as hand-crafted features [12], semantic cues [9,13], or directly based on data-driven [8,11,14] and etc. However, DeepFake detection is still a challenging task due to the following two reasons: (1) the counterfeits are constantly thriving, which results in less subtle distinction between real and fake videos; (2) a severe drop in performance when generalizing the detection method to other datasets, which hinders the application in practical use. As such, seeking more effective clues and boosting the generalization is highly desirable.

A following question to ask is: *What are the clues that generally exist in the majority of DeepFake videos?* Straightforwardly, the clues are more generalized if they are introduced by common steps in DeepFake video generation. To figure it out, we elaborate the pipeline of DeepFake video generation. As shown in Fig. 1, a face of source image is first cropped out and sent to DeepFake model to synthesize a target face. Then the synthesized face is warped and blended to the source image. The steps, *face synthesis* (Fig. 1(a)) and *synthesized face blending* (Fig. 1(b)), are essential and commonly existing in the pipeline of current DeepFake video generation. For face synthesis step, the images generated by deep generative models always go through a series of upconv or upsampling

☆ Han Chen and Yuezun Li contributed equally to this work.
* Corresponding author.
E-mail addresses: 2016130205@email.szu.edu.cn (H. Chen), liyuezun@ouc.edu.cn (Y. Li), dongdonglin8@gmail.com (D. Lin), libin@szu.edu.cn (B. Li), wujq@szu.edu.cn (J. Wu).
[1] Han Chen and Yuezun Li contributed equally to this work

https://doi.org/10.1016/j.patcog.2022.109179
0031-3203/

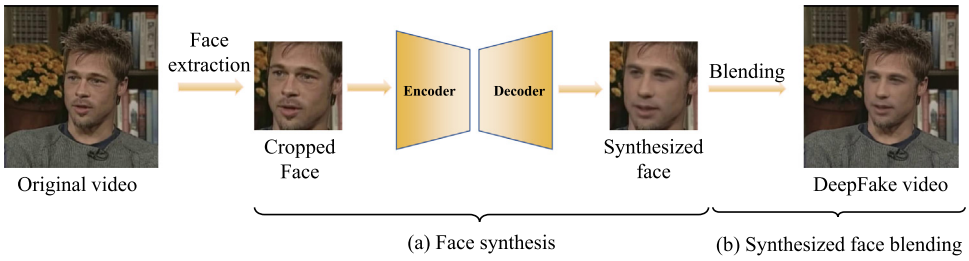

Fig. 1. The overall of DeepFake video generation mainly include: (a)*Face Synthesis* and (b)*Synthesized Face Blending*. As shown in step (a), the encoder-decoder network is employed to synthesize the fake face firstly. Step (b) shows that the synthesized face is warped and then blended to the original video frame.

operations, which results in the statistical discrepancy with real-world images. For synthesized face blending step, as the quality of synthesized face is imperfect, the blending operation can cause texture inconsistency between altered face area and original background, such that a blending boundary is formed.

To this end, we describe a new method, namely *BiG-Arts*, for DeepFake detection in within-dataset and cross-dataset scenarios. Our method is based on Bi-granularity artifacts, which consists of the intrinsic-granularity artifacts (InG-Arts) and the extrinsic-granularity artifacts (ExG-Arts). The InG-Arts and ExG-Arts correspond to the artifacts introduced by face synthesis step and synthesized face blending step, respectively. In our work, we define InG-Arts as a binary mask, where the area of synthesized face is set to 1 and others are 0. Similarly, we also define ExG-Arts as a binary mask, where the area of blending boundary is set to 1 and others are 0.

To fully explore the Bi-granularity artifacts, we formulate DeepFake detection as a multi-task learning problem. Specifically, our method aims to achieve three sub-tasks, which are binary classification, InG-Arts prediction and ExG-Arts prediction, respectively. Here we model the sub-tasks of predicting InG-Arts and ExG-Arts as the auxiliary to enhance the binary classification task. Hence, we design an architecture that contains a main branch and two side branches, where each branch corresponds to a sub-task. For the main branch, it outputs the final binary classification score with the given input. For each side branch, it generates a pixel-level classification map, where the value of a pixel denotes the probability of containing such artifacts. Note that the two branches for Bi-granularity artifacts are managed to improve the performance of main branch. Thus these three sub-tasks can be highly correlated and mutually boost each other. To improve the guidance of Bi-granularity artifacts, we develop connections between the two side branches and the main branch, enabling the main branch to learn more effective clues correlated with Bi-granularity artifacts. Fig. 2 illustrates the overview of our method and several result examples. The three branches are trained simultaneously in an end-to-end fashion, which facilitates the backbone network to learn more representative features. Compared to works [9,10,12,15], which only rely on extrinsic-granularity artifacts, our method considers Bi-granularity artifacts – intrinsic- and extrinsic-granularity artifacts together, which further boosts the generalization performance in cross-dataset scenario. Our method has been contributed as a part of the solution to achieve the Top-1 rank in DFGC competition.

The contribution of this paper is summarized in three-fold:

- We propose Bi-granularity artifacts to expose DeepFake videos. The Bi-granularity artifacts are introduced by the common steps, face synthesis and synthesized face blending, in DeepFake generation pipeline.
- We formulate DeepFake detection as a multi-task learning problem, where one task predicts the binary classification and others predict Bi-granularity artifacts. It is noteworthy that predicting Bi-granularity artifacts serves as auxiliary to further boost the classification.
- We perform extensive experiments on several public datasets (Celeb-DF [16], FF++ [11], DFD [17], etc) and compare our method with several state-of-the-art detection methods, which demonstrates the effectiveness of proposed method in within- and cross-dataset settings. Moreover, we conduct comprehensive ablation studies to investigate the benefit of different components, which provides useful insights for further research.

2. Related work

2.1. Deepfakes generation

The recent advances in deep generative models significantly improve face manipulation techniques, such as GAN face synthesis [6,18], facial attribution editing [19] face swapping [20] and etc. In particular, one technique known as DeepFake attracts tremendous attention. DeepFake is a face swapping technique that can swap the source face of input image with a synthesized target face while keeping the same facial expression and orientation. Specifically, DeepFake is based on variational auto-encoder (VAE) architecture [21], where the encoder aims to remove identity-related attributes and the decoder aims to recover the appearance of target identity. The overview of DeepFake video generation is shown in Fig. 1. Given an input image, the face is first cropped out and aligned to a standard shape (source). Then the source face is fed into DeepFake model, which can synthesize a new face of target identity (target). In order to fit into the input image, the synthesize face is warped using affine transform and blended with input image. In training phase, an encoder and a decoder is used for reconstructing faces of one identity. Then the same encoder and another decoder is used for reconstructing another identity. After training, a combination of encoder and decoder is selected according to demand in synthesis. To date, many DeepFake tools are released such as FaceSwap [4], FaceSwap-GAN [3] and DeepFaceLab [5]. These tools are the mainstream chooses for users. The FaceSwap follows the classical pipeline. The FaceSwap-GAN additionally utilizes adversarial and perceptual loss. The DeepFaceLab improves the quality in eyes by introducing a more complex architecture. Despite variance exists in different tools, all of them share two common steps – face synthesis and face blending operations as shown in Fig. 1.

2.2. Deepfakes detection

Many DeepFake detection methods have been proposed so far, *e.g.*, [22–25]. We divide these methods into four categories: data-driven, frequency based, artifacts based and consistency based. The data-driven denotes training detector directly using real and DeepFake images. For example, MesoNet [14], XceptionNet [11] and CapsuleNet [22], which trained their advised networks using real and

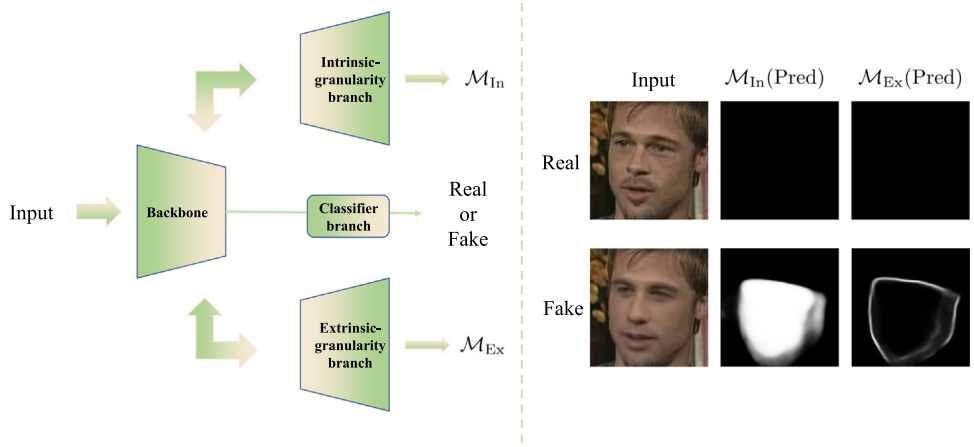

Fig. 2. The overall of our proposed method (left) and several result examples (right). The proposed method is designed as a three-branch architecture for multi-task learning. Specifically, two granularity branches are utilized to predict intrinsic- and extrinsic-granularity artifacts, respectively, and a classifier branch is employed to distinguish real and fake faces.

DeepFake images. MTD-Net [23] proposed Central Difference Convolution (CDC) and Atrous Spatial Pyramid Pooling (ASPP) to further improve the performance. For frequency based methods, Agarwal et al. [26] proposed a novel cross-stitched network to mine the distinguishing features in the spatial and frequency domains. Luo et al. [27] employed the SRM filters to extract the high-frequency noise feature, and proposed a multi-scale high-frequency feature extraction module to capture multi-scale high-frequency signals. The work [24] proposed a Spatial-Phase Shallow Learning (SPSL) method, which combines the spatial information and phase spectrum in the frequency domain. For artifacts based methods, FWA [9] proposed a fake face production pipeline which fabricates face warping artifacts in real face images. Face X-ray [10] also proposed a forgery face generation pipeline using two real faces and predict fake video by detecting blending boundary. Kim and Kim [12] exposed DeepFake video via pixel-level difference in facial boundary area. For consistency based methods, they mainly detect the consistency of given image content. Zhao et al. [28] proposed a novel representation learning algorithm, which optimizes a consistency branch to detect and localize forged faces. Shang et al. [25] proposed the Pixel-region relation network to capture the relation of different pixels in feature map and the relation between the manipulated region and original region. The purpose is to learn whether the foreground face region is inconsistent with the background region. In this work, we analyze the DeepFake artifacts and propose a new method based on Bi-granularity artifacts introduced by face synthesis and face blending respectively.

3. Proposed method

In this section, we describe the proposed method, BiG-Arts, in detail. Section 3.1 describes the details of mining Bi-granularity artifacts – intrinsic- and extrinsic-granularity artifacts, respectively. Section 3.2 introduces the formulation of DeepFake detection using our method. Section 3.3 elaborates the data augmentation strategy used in our method to increase diversity.

3.1. Mining Bi-granularity artifacts

The Bi-Granularity Artifacts utilized in our method consists of the intrinsic-granularity artifacts, which is formed due to the common operations in model generation, and the extrinsic-granularity artifacts, which is introduced by the face blending operations in post-processing. We describe the mining strategy of each artifacts in sequel.

Intrinsic-granularity Artifacts. As shown in Fig. 1(a), the altered face area is synthesized by DeepFake model. Since DeepFake model is either an VAE or GAN, the synthesized face can contain a common pattern of artifacts due to the up-conv/sampling operations [29]. In our case, we extract intrinsic-granularity artifacts using a simple strategy as following. Denote a fake face image as $\mathbf{x}'$ and corresponding real face image as $\mathbf{x}$. We define the intrinsic-granularity artifacts p_{In} as the discrepancy between $\mathbf{x}'$ and $\mathbf{x}$, *i.e.*, $p_{\text{In}} = |\mathbf{x}' - \mathbf{x}|$. Note the p_{In} is highly related with the manipulation method. Thus solely depending on p_{In} may baffle the generalization. Therefore, we simply utilize the convex hull of p_{In} to imply the area containing intrinsic-granularity artifacts. This is to say, for a fake image, the Intrinsic-granularity Artifacts is contained in the area $\mathcal{M}_{\text{In}}$, which is defined as $\mathcal{M}_{\text{In}} = \text{Convex}(p_{\text{In}} > \gamma)$, where γ is a predefined threshold.

Extrinsic-granularity Artifacts. In contrast to intrinsic-granularity artifacts, Extrinsic-granularity Artifacts is introduced by a general blending step in DeepFake generation, see Fig. 1(b). The face blending is to glue altered face area to original image, which can cause inconsistency across boundary [10]. Therefore, we extract the face blending boundary of fake face images as Extrinsic-granularity Artifacts. In our case, we directly extract the boundary of mask $\mathcal{M}_{\text{In}}$ as Extrinsic-granularity Artifacts. Denote $\mathcal{M}_{\text{Ex}}$ as the boundary mask, which can be defined as $\mathcal{M}_{\text{Ex}} = D(\mathcal{M}_{\text{In}}) - E(\mathcal{M}_{\text{In}})$ where D, E denote dilation and eroding operations.

3.2. DeepFake detection using Bi-granularity artifacts

Denote an input face image as $\mathbf{x} \in \mathcal{X} = \{0, \dots, 255\}^{h \times w \times 3}$, where h, w are the height and width of image $\mathbf{x}$ respectively. Denote $\mathbf{y} \in \mathcal{Y} = \{0, 1\}$ as the label of image $\mathbf{x}$, with $\mathbf{y} = 0$ indicating image $\mathbf{x}$ is real, fake otherwise. Let $\mathcal{D}_\theta : \mathcal{X} \to [0, 1]$ denote a CNN classifier with model parameters θ, where $\mathcal{D}_\theta(\mathbf{x})$ denotes the probability vector of image $\mathbf{x}$.

Vanilla Learning in DeepFake Detection. The preliminary goal of many existing DeepFake detection methods [11,14] is to increase the probability of ground-truth label as

$$\min_{\theta} \sum_{\mathbf{x},\mathbf{y}} \mathcal{L}_{Cls}(\mathcal{F}_\theta(\mathbf{x}), \mathbf{y}), \tag{1}$$

where $\mathcal{L}_{Cls}$ is the regression loss for training (*e.g.*, cross-entropy loss). However, a single task learning may not be able to capture desired features due to the lack of proper guidance. Hence, we utilize the Bi-granularity artifacts as a guidance to formulate Deep-

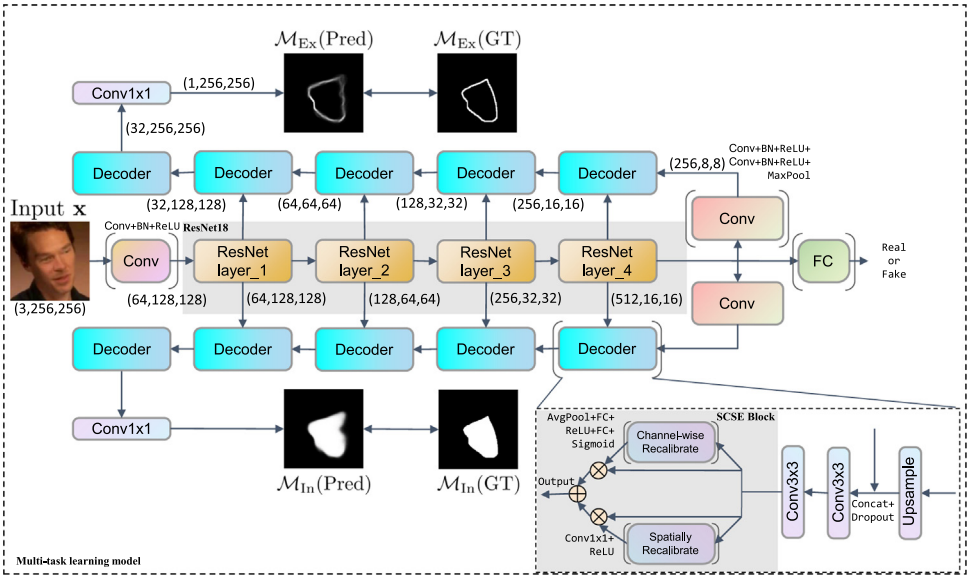

Fig. 3. The overall of proposed architecture. The ResNet-18 [30] is used as the backbone feature extractor. The bottom-right shows a decoder block which takes features both from the residual block of backbone and the output of the previous decoder block. (C, H, W) indicates the channel, height, and width of a feature map. See Section 3.2 for the details of the architecture.

Fake detection as multi-task learning problem, as demonstrated in Fig. 3.

Multi-Task Learning in DeepFake Detection. We design a three-branch architecture for multi-task learning, where two branches are used to predict Intrinsic- and Extrinsic-granularity artifacts and one branch is for classification prediction.

1) Architecture: we employ ResNet-18 [30] as the backbone architecture and two decoders for intrinsic- and extrinsic-granularity branches. One branch follows backbone with a Fully Connected (FC) layer for classification. Then we design two decoder structures for predicting intrinsic- and extrinsic-granularity artifacts, respectively. Specifically, each auxiliary branch contains five decoder blocks and one convolution layer with 1×1 kernel. The decoder block contains SCSE (Spatial and Channel Squeeze & Excitation) modules [31], which aims to enhance the important feature and ignore the unimportant feature. Specifically, the channel-wise recalibrate is composed by an average pooling and two FC layers, and the spatial-wise recalibrate contains a convolution layer with kernel 1×1. To encourage the guidance of Bi-granularity artifacts to DeepFake detection, the Decoder block is designed to take features from Residual block of backbone and the output of Decoder block. The Conv block before FC layer is a composition of two convolution layers and one maxpooling layer. Note the backbone can be other more complex architectures (*e.g.*, ResNet-101, XceptionNet etc), we use ResNet-18 here for a better balance between running efficiency and resource occupancy.

2) Objective function: To train the proposed architecture, we introduce three loss terms for corresponding branch, which are Intrinsic-granularity Artifacts Localization loss $\mathcal{L}_{In}$, Extrinsic-granularity Artifacts Localization loss $\mathcal{L}_{Ex}$ and classification loss $\mathcal{L}_{Cls}$ respectively. Denote the training dataset as $\mathcal{K}$. Let $\{\mathbf{x}, \mathcal{M}_{\mathrm{In}}, \mathcal{M}_{\mathrm{Ex}}, \mathbf{y}\} \in \mathcal{K}$ be a training sample, where $\mathbf{x}, \mathcal{M}_{\mathrm{In}}, \mathcal{M}_{\mathrm{Ex}}, \mathbf{y}$ are input image, intrinsic artifacts Ground Truth (GT) mask, extrinsic artifacts GT mask and class label, respectively. Given the input image $\mathbf{x}$, the output of each branch can be denoted as $\mathcal{O}_{In}(\mathbf{x}), \mathcal{O}_{Ex}(\mathbf{x}), \mathcal{O}_{Cls}(\mathbf{x})$. The whole objective function can be defined as

$$\sum_{\{\mathbf{x}, \mathcal{M}_{\mathrm{In}}, \mathcal{M}_{\mathrm{Ex}}, \mathbf{y}\} \in \mathcal{K}} (\lambda_{In}\mathcal{L}_{In} + \lambda_{Ex}\mathcal{L}_{Ex} + \lambda_{Cls}\mathcal{L}_{Cls}). \tag{2}$$

Concretely, the Intrinsic-granularity Artifacts Localization loss $\mathcal{L}_{In}$ is defined as

$$\mathcal{L}_{In} = \sum_{i,j} \left(\mathcal{M}_{In}^{i,j} \log(\mathcal{O}_{In}(\mathbf{x})^{i,j}) + (1 - \mathcal{M}_{In}^{i,j}) \log(1 - \mathcal{O}_{In}(\mathbf{x})^{i,j}) \right), \tag{3}$$

where i, j denote the pixel localization on mask. Similarly, the Extrinsic-granularity Artifacts Localization loss $\mathcal{L}_{Ex}$ is defined as

$$\mathcal{L}_{Ex} = \sum_{i,j} \left(\mathcal{M}_{Ex}^{i,j} \log(\mathcal{O}_{Ex}(\mathbf{x})^{i,j}) + (1 - \mathcal{M}_{Ex}^{i,j}) \log(1 - \mathcal{O}_{Ex}(\mathbf{x})^{i,j}) \right). \tag{4}$$

Then we use cross-entropy loss for classification, which is defined as

$$\mathcal{L}_{Cls} = \mathbf{y} \log(\mathcal{O}_{Cls}(\mathbf{x})) + (1 - \mathbf{y}) \log(1 - \mathcal{O}_{Cls}(\mathbf{x})). \tag{5}$$

As such, minimizing the Eq. (2) can learn the three tasks end-to-end in a multi-task learning fashion, which can facilitate the classification branch to learn the distinguishable feature of BiG artifacts.

3.3. Forgery face augmentation

To further boost the generalization, we propose an augmentation strategy to increase the diversity of forgery faces. Specifically, we create two types of forgery face by blending original real and fake images in dataset. Let (A, B) denotes blending source image A to image B. Then the two types of forgery face can be denoted as (Real, Real), (Fake, Real). The details are described as following:

1. (Real, Real): We first select a source image A of one identity from real videos and extract its facial landmarks using Dlib [32]. Then we look into real videos of other identities. We extract the facial landmarks in frames of these real videos and calculate the Euclidean distance with the facial landmarks of A. The image with least Euclidean distance serves as target image B. Then we calculate the convex hull of image A as blending mask $\mathcal{M}$. To increase the diversity, we utilize 2D piece-wise affine transform to randomly distort the mask. Hence, the final mask is defined as $\bar{\mathcal{M}} = T(\mathcal{M})$. The blended image can defined as $\bar{\mathcal{M}} \odot A + (1 - \bar{\mathcal{M}}) \odot B$, where $\odot$ specifies the element-wise

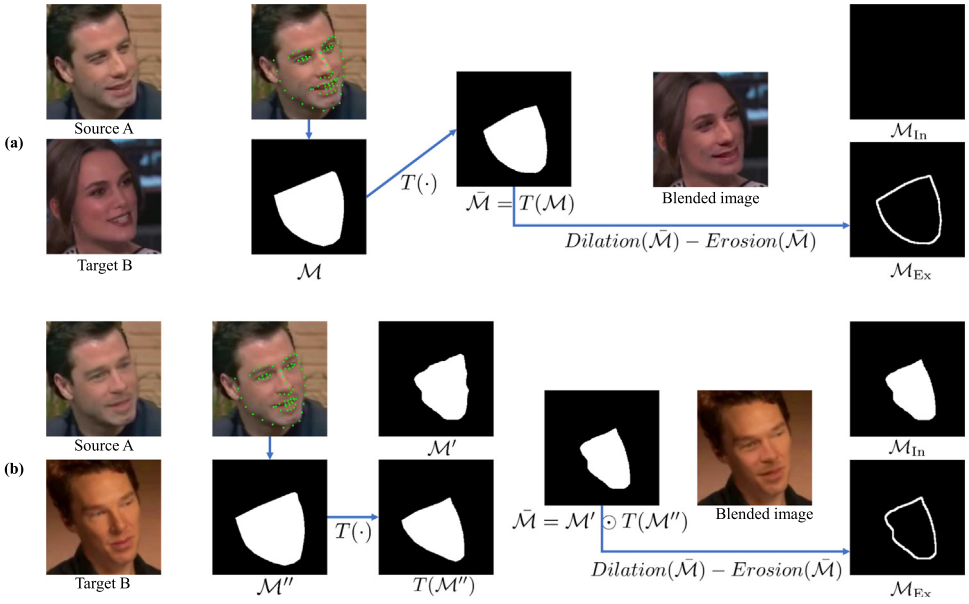

Fig. 4. Illustration of forgery face augmentation for (a) (Real, Real) and (b) (Fake, Real). $\mathcal{M}_{\text{In}}$ corresponds to the mask map of intrinsic-granularity artifacts, and $\mathcal{M}_{\text{Ex}}$ corresponds to the mask map of extrinsic-granularity artifacts. Note that since there is no intrinsic-granularity artifact in (a), its mask map $\mathcal{M}_{\text{In}}$ is black. The $\mathcal{M}_{\text{In}}$ in (b) is obtained from $\bar{\mathcal{M}}$. More details about the face augmentation process can be referred to Section 3.3.

multiplication. Note the forgery face is created only using real images, and it has no intrinsic-granularity artifacts. Fig. 4(a) illustrates the process of creating forgery face by real and real image.

2. (Fake, Real): To further increase the diversity, we create forgery faces using fake and real images. We first select an source image *A* from fake videos and extract the facial landmarks. Then we use the same strategy as in (Real, Real) to find a landmark-matched target image *B* from real videos. To obtain the manipulation area, we first calculate the difference of image *A* with its original real image and expand the difference to a mask $\mathcal{M}'$. However, the expansion may cause mask $\mathcal{M}'$ crossing the manipulation boundary. To bound $\mathcal{M}'$ inside a reasonable area, we then calculate the convex hull of facial landmarks in image *A* as $\mathcal{M}''$, which is the upper bound of $\mathcal{M}'$. Similarly, we then increase the mask randomness by affine transform. Hence, the final mask is obtained by $\bar{\mathcal{M}} = \mathcal{M}' \odot T(\mathcal{M}'')$. Then we can use the same blending equation to generate forgery face. Fig. 4(b) illustrates the process of creating forgery face by fake and real image.

With the forgery face augmentation, we eventually have four type of training images: the original real and fake images, and two type of augmentation forgery images. Several examples are shown in Fig. 5.

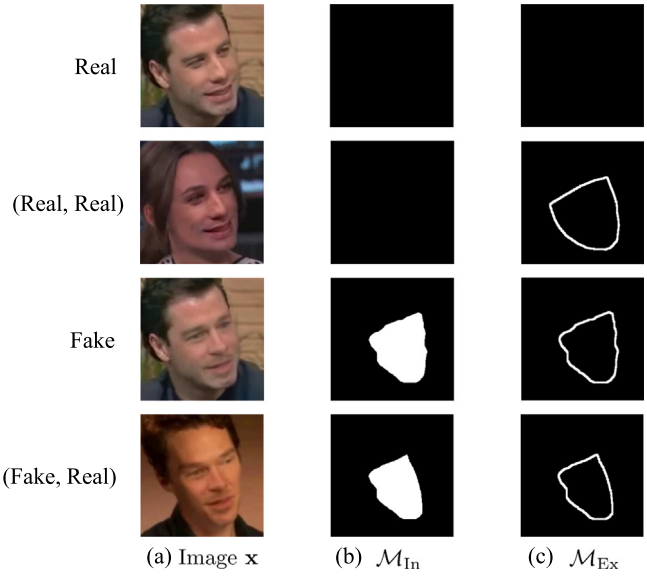

Fig. 5. Four types of face images and corresponding BiG artifacts $\mathcal{M}_{\text{In}}$ and , $\mathcal{M}_{\text{Ex}}$. From top to down are original real, augmentation (Real, Real), original fake and augmentation (Fake, Real), respectively.

4. Experiments

This section describes the experiments of our method on public datasets. Specifically, Section 4.1 describes the experimental setup. Section 4.3 describes the ablation study on the effect of different components. Section 4.2 demonstrates the effectiveness of our method compared with others and Section 5 discuss the robustness of our method against compression, blurring and additive noise etc.

4.1. Experimental setup

DeepFake datasets. We utilize the following datasets to demonstrate the efficacy of our method, which are Celeb-DF, FaceForensics++ (FF++), DFD, DFDC preview (DFDC-P), UADFV, DF-TIMIT, and WildDeepFake respectively.

1. Celeb-DF [16]. This dataset contains 5639 high-quality DeepFakes videos and 890 corresponding real videos, covering 59 identities with various gender, age and etc. This dataset is split using 6011 videos for training and 518 videos for testing.
2. FaceForensics++ (FF++) [11]. This dataset includes 1000 original videos and fake videos generated by different manipulation methods, such as DeepFakes, Face2Face, FaceSwap, NeuralTextures and FaceShifter. All of these videos have three variants, *i.e.*, raw, Low Quality (LQ) with compressed factor c40 and High Quality (HQ) with compressed factor c23. Since our method focuses on DeepFake manipulation, we only use Deepfakes videos and corresponding pristine videos in HQ version. The split of

this dataset is 740 videos for training, 140 videos for validation and 140 videos for testing.
3. DFD [17]. The Google & Jigsaw released the DeepFake detection dataset which has 3068 Deepfakes videos and 363 pristine videos in three variants, *i.e.*, raw, LQ and HQ. This dataset does not split training and testing set.
4. DFDC-P [33]. This dataset is the preview version of DeepFake detection challenge [34] that contains 4113 Deepfakes videos and 1131 pristine videos.
5. UADFV [13]. This dataset contains 49 pristine videos and 49 Deepfakes videos without training and testing split.
6. DF-TIMIT [35]. The dataset contains two quality subsets: DF-TIMIT-LQ and DF-TIMIT-HQ. Each set has 320 real and 320 fake videos. This dataset does not has training and testing split.
7. WildDeepFake [36]. This dataset contains 3805 pristine face sequences and 3509 fake Deepfakes sequences collected from the internet, including 6508 videos for training and 806 videos for testing.

In our experiment, we train our method using Celeb-DF dataset and evaluate the performance within- and cross-datasets. Since Celeb-DF does not provide validation set, we split the original training set to a new training set and validation set with the ratio 8:2. Thus, the final training set contains 4871 videos, the validation set contains 1140 videos. In testing, we perform our method on the testing set of Celeb-DF, FF++ and WildDeepFake dataset, and entire videos for other datasets.

Following works [9,10], we employ Area Under the Receiver Operating Characteristic Curve (AUC) as our evaluation metrics.

Implementation details. Here we describe the implementation detail regarding data preparation and training and testing setup in sequel.

1) Data preparation. To prepare the training and testing data, we use MTCNN [37] to crop face area and Dlib tool [32] to extract the landmarks of face area. In training, we employ on-the-fly scheme to dynamically augment forgery faces in each training batch. Specifically, for one training batch, 50% are real images and 50% are fake images. For fake images in a batch, 33% are directly sampled from the fake videos, 33% are sampled using (Real, Real) approach and rest are sampled using (Fake, Real). For data augmentation, we only use JPEG compression. The compression quality is randomly selected from $[75, 100]$.

2) Training and testing setup. Our method is implemented by PyTorch on CentOS 7.2.1511 with one Tesla P100 GPU. In training, the size of input images is set to 256×256, the batch size is set to 32 and the total number of epochs is set to 1200. We use Adam optimizer [38] with a start learning rate of 0.001, betas of 0.9 and 0.999 respectively. We use CosineAnnealingLR in pytorch and set T_max to 40. The weighing factor in loss function is set as $\lambda_{In} = \lambda_{Ex} = 15, \lambda_{Cls} = 1$.

In testing, we sample one frame every ten frames in each video and average the scores of these frames as the score of video.

4.2. Comparison with recent works

Quantitative Results. The comparison of our method with recent works is shown in Table 1. Note all methods are trained on Celeb-DF dataset. The first five rows in table show the performance of many well-known base networks trained in an end-to-end fashion. The following rows, XceptionNet, MesoInception-4, Capsule, Face X-ray, RFM and GSRM, are recent DeepFake detection methods. For fair comparison, we retrain the detection model of these methods on Celeb-DF using their released codes. Since Face X-ray does not release the training code, we implement Face X-ray using our method by pruning the intrinsic-granularity artifacts branch. From the results we can observe that the vanilla training of base networks can achieve competitive performance on within-dataset scenario, but server performance drop on cross-dataset scenario, which indicates the necessity of developing specialized methods. The XceptionNet, MesoInception-4 and Capsule are data-driven methods which are trained using real and fake images on intentionally designed architectures. Similarly, we can see these methods perform well on within-dataset scenario but can only achieve decreased performance on other datasets, especially on DT-HQ, FF++ and DFD. More recent methods Face X-ray and GSRM are designed to improve the generalization performance by capturing the high-frequency artifacts. Specifically, Face X-ray relies on detecting face blending artifacts and GSRM targets on the high-frequency features extracted by SRM filters. RFM employs an attention-based data augmentation method to reduce overfitting. These three methods exhibits a notable improvement on cross-datasets scenario. Compared to these methods, our method shows more superior performance on cross-dataset scenario thanks to the proposed bi-granularity artifacts, *i.e.*, achieving 0.91 AUC score averagely on all datasets, which outperforms other methods at least 2%.

For comprehensive comparison, we also evaluate our method using some common metrics, including Accuracy, Precision, Recall, F1 and Equal Error Rate (EER). Table 2 shows the performance of our method which is trained and tested both on Celeb-DF. We can observe that our method performs well under these metrics.

To further evaluate the performance against real-world DeepFakes, we conduct cross-dataset evaluation experiment on WildDeepFake dataset. The results are reported in Table 3. Note that WildDeepFake employs various post-processing operations on images, thus the methods such as Face X-ray which solely rely on Extrinsic-granularity artifacts are degraded. In contrast, our method considers Bi-granularity artifacts, which improves the performance by around 6% compared to Face X-ray.

Moreover, we also train all methods on FF++ dataset and evaluate on all datasets. The results are shown in Table 4, we can see that our method can still achieve impressive performance under this setting, which demonstrates that our method is effective on using different datasets in training.

Visualizations. To better understand the effectiveness of our method, we employ Grad-CAM [40] to visualize the attention regions on different datasets. Specifically, we average the faces and corresponding Grad-CAM maps by sampling five frames per video. As shown in Fig. 6, the Grad-CAM between fake and real images is significantly different on all of these datasets. Since UADFV, DT-LQ and DT-HQ datasets are proposed early, the visual quality of these datasets are compromised. We can observe that the Grad-CAM maps are mainly concentrated on the tampered face regions. In contrast, the recently proposed FF++, DFD, and DFDC-P datasets have better visual quality, which consequently increase the difficulty of detection. Thus the Grad-CAM maps on these datasets have the trend of widespread to the regions outside faces. Note that the WildDeepFake dataset is applied with many post-processing operations to increase DeepFake diversity. Thus detecting the forgeries in this dataset is more challenging compared to other datasets. It can also be reflected by the Grad-CAM maps, as which are more diffused than others. Moreover, several visual results of Bi-granularity artifacts prediction are shown in Fig. 7. The results show that our method can well locate the Bi-granularity artifacts in DeepFake images.

4.3. Ablation study

This section describes the ablation study on different settings of our method, including the effect of forgery face augmentation, the effect of auxiliary branches and the performance of our method on partial face manipulation.

Table 1
The AUC performance of our method (BiG-Arts) and other recent methods. The best performance is highlighted in **bold**. Note these methods are trained on Celeb-DF dataset.

Methods	UADFV [13]	DT-LQ [35]	DT-HQ [35]	FF+ [11]	DFD [17]	DFDC-P [33]	Celeb-DF [16]	Avg
AlexNet	0.8573	0.7437	0.4941	0.6705	0.6709	0.7062	0.9810	0.7319
VGG16	0.8898	0.9078	0.6350	0.7783	0.7739	0.7546	0.9939	0.8190
Densenet121	0.9107	0.9080	0.5874	0.7146	0.7242	0.7394	0.9979	0.7974
Resnet50	0.9032	0.8818	0.6302	0.7361	0.7399	0.7423	0.9972	0.8043
Resnet18	0.9226	0.8829	0.6049	0.7627	0.7283	0.7496	0.9974	0.8069
XceptionNet [11]	0.9610	0.9550	0.6539	0.7551	0.7706	0.7401	**0.9985**	0.8334
MesoInception-4 [14]	0.7750	0.8133	0.5609	0.7296	0.6707	0.7674	0.9242	0.7487
Capsule [22]	0.8751	0.8519	0.6162	0.7219	0.6676	0.7056	0.9900	0.7754
Face X-ray [10]	0.9305	0.9899	0.8917	0.8185	**0.8253**	0.7535	0.9984	0.8868
GSRM [27]	**0.9645**	0.9709	0.6749	0.8029	0.8048	0.7924	0.9962	0.8580
RFM [39]	0.9110	0.9569	0.6582	0.8288	0.7870	0.7678	0.9973	0.8439
BiG-Arts (Ours)	0.9404	**0.9952**	**0.9377**	**0.8523**	0.8185	**0.8189**	0.9980	**0.9087**

Table 2
The performance of our method (BiG-Arts) and other recent methods. The best performance is highlighted in **bold**. Note these methods are trained and tested on Celeb-DF dataset.

Methods	Accuracy	Precision	Recall	F1	AUC	EER
AlexNet	0.9336	0.9160	0.9876	0.9504	0.9810	0.0642
VGG16	0.9517	0.9335	0.9960	0.9638	0.9939	0.0367
Densenet121	0.9753	0.9683	0.9943	0.9811	0.9979	0.0228
Resnet50	0.9744	0.9694	0.9916	0.9804	0.9972	0.0259
Resnet18	0.9725	0.9627	0.9959	0.9790	0.9974	0.0245
XceptionNet [11]	0.9830	0.9766	0.9975	0.9869	**0.9985**	**0.0157**
MesoInception-4 [14]	0.8541	0.8963	0.8751	0.8856	0.9242	0.1568
Capsule [22]	0.9546	0.9427	0.9898	0.9657	0.9900	0.0468
Face X-ray [10]	**0.9835**	**0.9794**	0.9954	**0.9873**	0.9984	0.0167
GSRM [27]	0.9768	0.9672	**0.9979**	0.9823	0.9962	0.0196
RFM [39]	0.9702	0.9579	0.9977	0.9773	0.9973	0.0259
BiG-Arts (Ours)	0.9775	0.9724	0.9932	0.9827	0.9980	0.0214

Table 3
The performance of our method (BiG-Arts) and other recent methods. The best performance is highlighted in **bold**. Note these methods are trained on Celeb-DF dataset and tested on WildDeepFake dataset.

Methods	Frame-level		Sequence-level	
	AUC	EER	AUC	EER
AlexNet	0.6518	0.3898	0.6358	0.3965
VGG16	0.6671	0.3712	0.6674	0.3763
Densenet121	0.6532	0.3902	0.6651	0.3813
Resnet50	0.6651	0.3851	0.6461	0.3914
Resnet18	0.6458	0.3923	0.6607	0.3813
XceptionNet [11]	0.6720	0.3759	0.6962	0.3535
MesoInception-4 [14]	0.6377	0.3994	0.6415	0.4040
Capsule [22]	0.6395	0.4029	0.7096	**0.3232**
Face X-ray [10]	0.6098	0.4227	0.6841	0.3586
GSRM [27]	0.6774	0.3769	0.6992	0.3484
RFM [39]	**0.7313**	**0.3311**	**0.7457**	0.3359
BiG-Arts (Ours)	0.6609	0.3898	0.7272	0.3510

Effect of Forgery Face Augmentation. To study the effect of forgery face augmentation, we conduct experiments with different settings: 1) No augmentation (None), 2) only using (Real, Real) augmentation ((Real, Real)), 3) only using (Fake, Real) augmentation ((Fake, Real)) and 4) using all augmentations (All). Table 5 shows the performance of each augmentation setting. Note all settings are trained on Celeb-DF and tested on all datasets. We observe using no augmentation performs the best when testing on Celeb-DF. It is because the augmentation can disturb the learning of a specific distribution, which can slightly improve the performance on a same dataset. However, from another perspective, the augmentation can notably reduce the overfitting phenomenon by increasing the data diversity. Therefore, it can generalize the performance across different datasets. As shown in Table 5, our method with all augmentations (All) achieves the best average performance on all datasets.

Table 4
The AUC performance of our method (BiG-Arts) and other recent methods. The best performance is highlighted in **bold**. Note these methods are trained on FF++ dataset.

Methods	UADFV [13]	DT-LQ [35]	DT-HQ [35]	FF+ [11]	DFD [17]	DFDC-P [33]	Celeb-DF [16]	Avg
AlexNet	0.6926	0.9227	0.7962	0.9894	0.8072	0.7393	0.5723	0.7885
VGG16	0.7378	0.8995	0.8580	0.9955	0.8549	0.7244	0.6945	0.8235
Densenet121	0.7420	0.9172	0.8255	0.9959	0.8520	0.7237	0.6977	0.8220
Resnet50	0.7020	0.8749	0.7939	0.9947	0.8730	0.7204	0.6707	0.8042
Resnet18	0.7498	0.8998	0.7827	0.9947	0.8584	0.6770	0.6316	0.7991
XceptionNet [11]	0.7685	0.9284	0.9050	0.9975	0.8922	0.7166	0.5903	0.8284
MesoInception-4 [14]	0.4413	0.9389	0.8226	0.9868	0.8069	0.7517	0.5719	0.7600
Capsule [22]	**0.9356**	0.9412	0.8694	0.9877	0.8330	0.7291	0.6718	0.8525
Face X-ray [10]	0.8070	0.9315	0.9161	0.9965	0.8786	0.7310	0.7408	0.8574
GSRM [27]	0.8493	0.9910	0.8996	**0.9978**	0.8578	**0.8109**	0.7133	0.8742
RFM [39]	0.8244	0.9530	0.9337	0.9947	0.8747	0.7318	0.7108	0.8604
BiG-Arts (Ours)	0.8814	**0.9940**	**0.9857**	0.9939	**0.8992**	0.8048	**0.7704**	**0.9042**

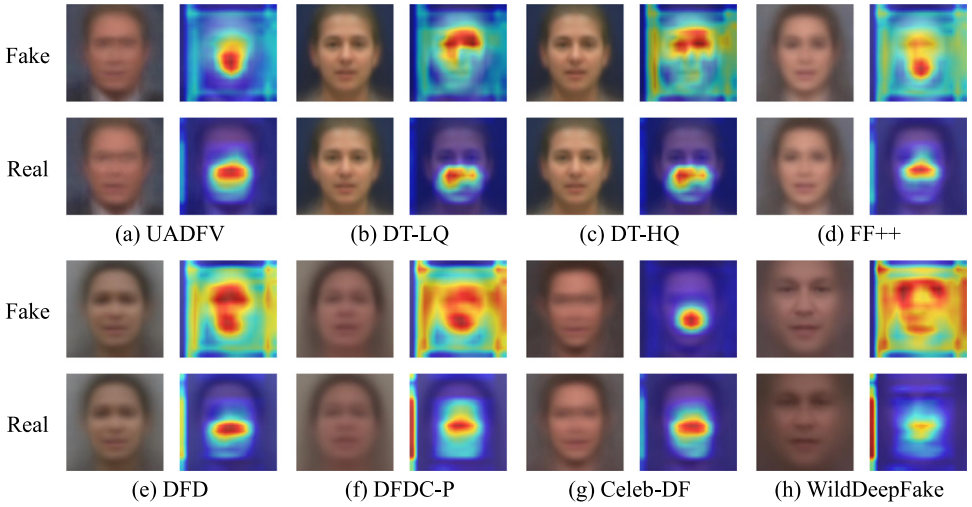

Fig. 6. The averaged faces and the corresponding Grad-CAM maps [40] of our method (BiG-Arts) on different datasets. In each subfigure, the first row shows the averaged fake face and the corresponding Grad-CAM map, and the second row shows the averaged real face and the corresponding Grad-CAM map. Our model is trained on Celeb-DF.

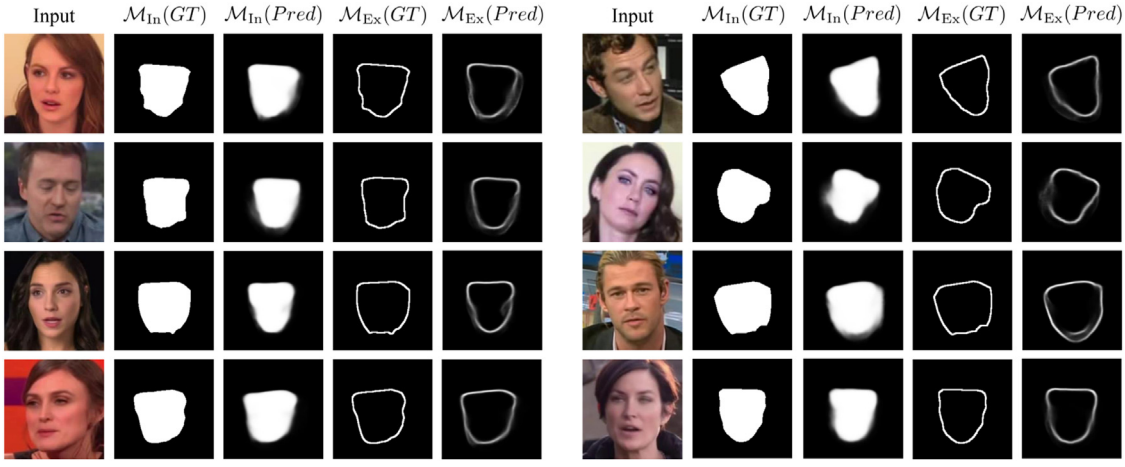

Fig. 7. Visual result of $\mathcal{M}_{\text{Ex}}$ and $\mathcal{M}_{\text{In}}$ on some faces image from Celeb-DF testing dataset. Our model is trained on Celeb-DF.

Table 5
The AUC performance of our method with different forgery face augmentation settings. The best performance is highlighted in **bold**. All settings are trained on Celeb-DF and tested on all datasets.

Settings	UADFV [13]	DT-LQ [35]	DT-HQ [35]	FF+ [11]	DFD [17]	DFDC-P [33]	Celeb-DF [16]	Avg
None	0.9268	0.9665	0.7165	0.8174	0.7914	0.7424	0.9971	0.8511
(Real, Real)	0.9457	0.9576	0.8260	0.8467	0.7779	0.7645	0.9971	0.8736
(Fake, Real)	**0.9609**	0.9913	0.8388	**0.8827**	**0.8445**	0.7880	0.9978	0.9005
All	0.9404	**0.9952**	**0.9377**	0.8523	0.8185	**0.8189**	**0.9980**	**0.9087**

Table 6
The AUC performance of our method using different artifacts. The best performance is highlighted in **bold**. All settings are trained on Celeb-DF and tested on all datasets.

Model	UADFV [13]	DT-LQ [35]	DT-HQ [35]	FF+ [11]	DFD [17]	DFDC-P [33]	Celeb-DF [16]	Avg
None	0.9203	0.9433	0.6561	**0.9270**	0.7173	0.7502	0.9805	0.8421
$\mathcal{M}_{\text{In}}$	0.9169	0.9478	0.7616	0.8506	0.7516	0.7386	0.9954	0.8517
$\mathcal{M}_{\text{Ex}}$	0.9305	0.9899	0.8917	0.8185	**0.8253**	0.7535	**0.9984**	0.8868
$\mathcal{M}_{\text{In}}$ + $\mathcal{M}_{\text{Ex}}$	**0.9404**	**0.9952**	**0.9377**	0.8523	0.8185	**0.8189**	0.9980	**0.9087**

Effect of Bi-granularity Artifacts. We conduct four experiments to demonstrate the efficacy of using Bi-granularity artifacts: 1) using no artifacts (None), *i.e.*, removing two branches for Bi-granularity artifacts prediction; 2) Only using Intrinsic-granularity artifacts ($\mathcal{M}_{\text{In}}$); 3) Only using extrinsic-granularity artifacts ($\mathcal{M}_{\text{Ex}}$) and 4) using Bi-granularity artifacts ($\mathcal{M}_{\text{In}} + \mathcal{M}_{\text{Ex}}$). Table 6 shows the performance of using different artifacts. All settings are trained on Celeb-DF and tested cross datasets. The results indicate using Bi-granularity artifacts outperforms other settings, which demonstrate the effectiveness of our method in cross-datasets scenarios.

Performance on Partial Face Manipulation. To fully explore the detection ability of our method, we conduct experiments on

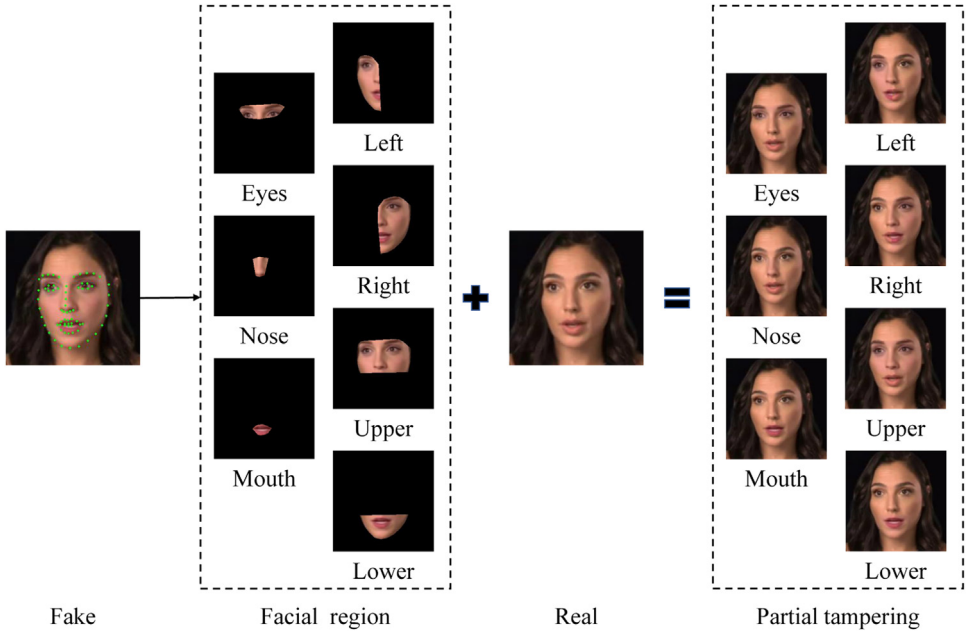

Fig. 8. Illustration of the process of generating partial face manipulation image. The generated fake images include partial tampering such as eyes, nose, mouth, lower, upper, left and right parts of the face.

Table 7
The AUC performance of our method (BiG-Arts) and other recent methods on the partial face manipulation created from Celeb-DF. The best performance is highlighted in **bold**. Note these methods are trained on original Celeb-DF dataset.

Methods	Eye	Nose	Mouth	Lower	Upper	Left	Right
GSRM [27]	0.7120	0.5644	0.5899	0.7720	0.9503	0.8021	0.7661
RFM [39]	0.7088	0.5752	0.6114	**0.8080**	0.9433	0.8115	0.8042
BiG-Arts (Ours)	**0.9204**	**0.8971**	**0.6779**	0.7907	**0.9720**	**0.9561**	**0.9603**

partial face manipulation. Specifically, we utilize facial landmarks to create partial tampering faces. As shown in Fig. 8, we first extract the region on facial components of DeepFake image such as eyes, nose, mouth, lower, upper, left and right parts, and then paste each of them into the corresponding real images to generate different partial tampering images. Table 7 shows the performance of different methods on detecting partial face manipulations created from Celeb-DF. Note these methods are trained on original Celeb-DF dataset. From the results we can see that our method outperforms other methods on most types of partial face manipulation, especially on eye and nose regions.

5. Robustness analysis

In this section, we analyze the robustness of our method against several image post-processing laundry operations such as JPEG compression, video compression, Gaussian blurring, scale resizing and adding additive noise, respectively. Note that our method is trained on Celeb-DF dataset and directly tested on other laundered images of different datasets. **JPEG Compression.** We compress the testing images using the built-in functions in OpenCV[1]. The image quality factors are selected from $[60, 100]$, where 100 denotes the image is not compressed. Fig. 9(a) shows the performance of our method against JPEG compression on all datasets. Note the x-axis is image quality factor (QF), while y-axis is the AUC performance. This figure reveals that the performance of our method on Celeb-DF, UADFV and FF++ datasets are stable with QF reducing, which only drops off $\sim 2\%$ in average. The trend of other datasets is different that the performance drops clearly with the QF reducing, which is probably due to the composition of these datasets are more complex, *e.g.*, containing more low-quality images than others.

Table 8
The AUC performance of our method (BiG-Arts) under video compression with different Constant Rate Factors (CRFs) on Celeb-DF. Our model is trained on Celeb-DF dataset.

CRF	5	10	15	20	25	30
Celeb-DF	0.9979	0.9974	0.9963	0.9929	0.9731	0.8814

Video Compression. Apart from JPEG compression, we also study the video compression, as which is a common operation for videos circulated online. Different with JPEG compression, the video compression considers both spatial and temporal correlations. Concretely, we utilize FFMPEG[2] tool to re-compress the testing videos, using different Constant Rate Factors (CRFs) in $[5, 30]$. Note the less value of CRF, the less compression of the video. Due to the heavy time consuming in video compression, we only test our method on Celeb-DF dataset. Table 8 shows the performance against video compression. Our method only slightly drops with CRF increasing, indicating the resistance of our method against video compression.

Gaussian Blurring. We then test our method under Gaussian blurring operations. The Gaussian kernel is $\mathcal{N}(0, \sigma_{\text{blur}})$ with size 3×3. The standard deviation σ_{blur} is selected from $[0, 0.5]$, where

[1] https://opencv.org/

[2] https://ffmpeg.org/

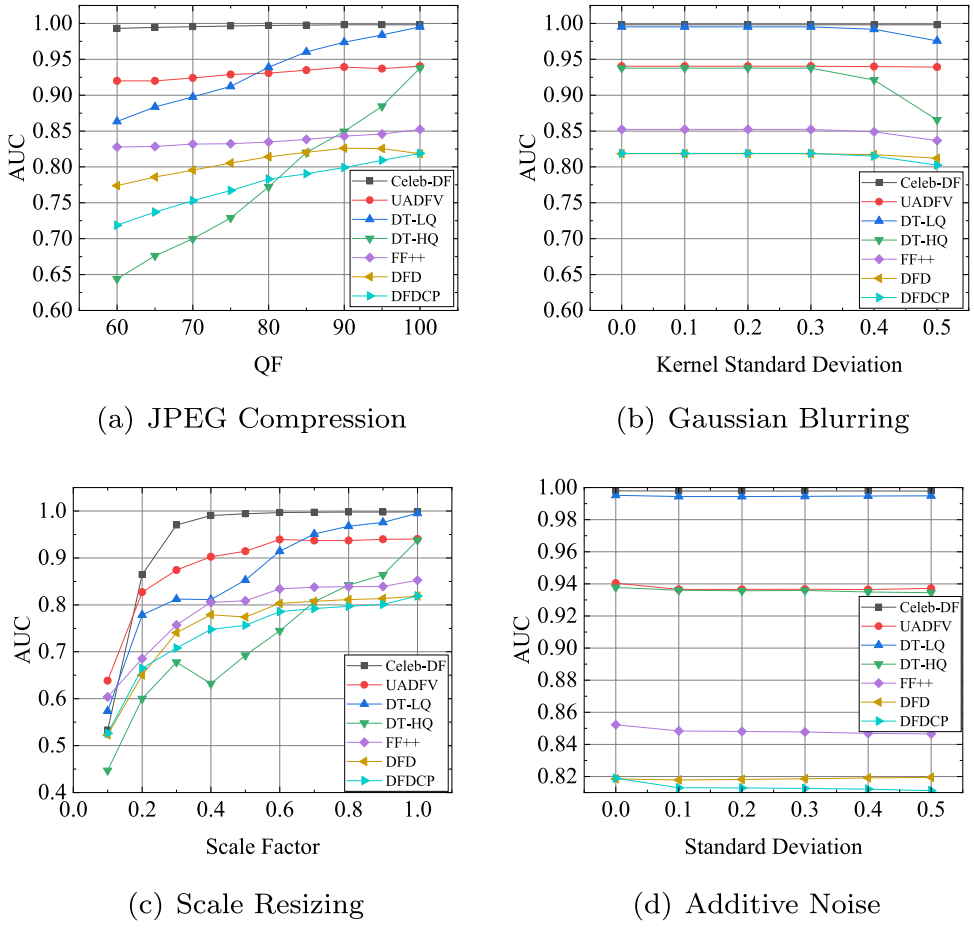

Fig. 9. Robustness analysis tested on all datasets. Our model is trained on Celeb-DF dataset and then tested on other laundered images of different datasets.

0 denotes no blurring is applied. As shown in Fig. 9(b), we can observe our method could resist Gaussian blurring to a certain degree, as the performance slightly goes down with σ_{blur} increasing.

Scale Resizing. We study the effect of scale resizing toward our method. Specifically, we resize testing images using OpenCV functions using scale factors from $[0.1, 1]$, where 1 means no resizing is used. The performance trend is shown in Fig. 9(c). From this figure we can observe the curve drops off slightly when the scale is larger than 0.4, which represents the capacity of resistance to resizing. However, the curve drops quickly with less scale, which is probably because the heavy resizing can greatly alter the original distribution of testing images.

Additive Noise. We employ Gaussian noise $\mathcal{N}(0, \sigma_{\text{noise}})$ in our case and test the performance accordingly. operations. The standard deviation σ_{noise} is selected from $[0, 0.5]$, where 0 denotes no noise is added. The Fig. 9(d) reveals the curve of our method is always flat, which indicates the resistance of our method against additive noise disturbance.

6. Conclusion

In this work, we propose a new approach to expose DeepFake videos by detecting Bi-granularity artifacts (BiG-Arts). In particular, Bi-granularity artifacts are made up of intrinsic-granularity artifacts and extrinsic-granularity artifacts, which commonly exist in the face synthesis process and face blending process in DeepFake video generation. To detect these artifacts, we create a new architecture network equipped with three branches, where two branches are respectively designed to detect intrinsic-granularity artifacts and extrinsic-granularity artifacts, and the main classifier branch is used to determine the final authenticity. These branches interact with each other, guiding the learning of the main branch for classification, in a way that leads to a favorable performance in both within-dataset and cross-dataset scenarios. The experiments conducted on public benchmarks with a comparison to many state-of-the-art methods demonstrate the efficacy of our method. Our ablation study on the effect of different components provides insights for the research. We also perform experiments against many pre-processing operations to analyze the robustness.

Future Work. While the proposed method is successful in improving the generalization ability of DeepFake detection, heavy compression is still a challenge case. In the future, the performance of our method may be further improved in following three aspects: 1) improve the backbone to extract more representative features, such as incorporating transformer knowledge; 2) exploiting the temporal correlations between adjacent frames in a video to capture more cues; 3) improving the robustness of our method by utilizing the frequency domain analysis.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The authors do not have permission to share data.

Acknowledgment

This work was supported in part by NSFC (Grant 61872244 and U22B2047), Guangdong Basic and Applied Basic Research Foundation (Grant 2019B151502001), Shenzhen R&D Program (Grant JCYJ20200109105008228), China Postdoctoral Science Foundation (Grant 2021TQ0314 and 2021M703036), Fundamental Research Funds for the Central Universities.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the Conference Neural Information Processing Systems (NeurIPS), 2014, pp. 2672–2680.
[2] L. Gao, D. Chen, Z. Zhao, J. Shao, H.T. Shen, Lightweight dynamic conditional GAN with pyramid attention for text-to-image synthesis, Pattern Recognit. 110 (2021) 107384.
[3] Faceswap-GAN GitHub, Accessed Nov 4, 2019. (https://github.com/shaoanlu/faceswap-GAN).
[4] Faceswap GitHub, Accessed Nov 4, 2019. (https://github.com/deepfakes/faceswap).
[5] DeepFaceLab GitHub, Accessed Nov 4, 2019. (https://github.com/iperov/DeepFaceLab).
[6] N. Liu, T. Zhou, Y. Ji, Z. Zhao, L. Wan, Synthesizing talking faces from text and audio: an autoencoder and sequence-to-sequence convolutional neural network, Pattern Recognit. 102 (2020) 107231.
[7] K.A. Pantserev, The malicious use of ai-based deepfake technology as the new threat to psychological security and political stability, in: Cyber Defence in the Age of AI, Smart Societies and Augmented Humanity, Springer, 2020, pp. 37–55.
[8] S. Tariq, S. Lee, S. Woo, One detector to rule them all: towards a general deepfake attack detection framework, in: Proceedings of The Web Conference 2021 (WWW), 2021, pp. 3625–3637.
[9] Y. Li, S. Lyu, Exposing deepfake videos by detecting face warping artifacts, in: Proceedings of the IEEE International Conference on Computer Vision Workshops (CVPRW), 2019, pp. 46–52.
[10] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, B. Guo, Face x-ray for more general face forgery detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5001–5010.
[11] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, M. Nießner, FaceForensics++: learning to detect manipulated facial images, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 1–11.
[12] D.-K. Kim, K.-S. Kim, Generalized facial manipulation detection with edge region feature extraction, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), 2022, pp. 2828–2838.
[13] X. Yang, Y. Li, S. Lyu, Exposing deep fakes using inconsistent head poses, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 8261–8265.
[14] D. Afchar, V. Nozick, J. Yamagishi, I. Echizen, MesoNet: a compact facial video forgery detection network, in: Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2018, pp. 1–7.
[15] P. Majumdar, A. Agarwal, R. Singh, M. Vatsa, Evading face recognition via partial tampering of faces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.
[16] Y. Li, X. Yang, P. Sun, H. Qi, S. Lyu, Celeb-DF: a large-scale challenging dataset for deepfake forensics, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 3207–3216.
[17] N. Dufour, A. Gully, Contributing data to deepfake detection research, Accessed Nov 4, 2019. (https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html).
[18] Y. Fang, W. Deng, J. Du, J. Hu, Identity-aware cyclegan for face photo-sketch synthesis and recognition, Pattern Recognit. 102 (2020) 107249.
[19] S. Zhao, J. Li, J. Wang, Disentangled representation learning and residual GAN for age-invariant face verification, Pattern Recognit. 100 (2020) 107097.
[20] Y. Nirkin, Y. Keller, T. Hassner, FSGAN: subject agnostic face swapping and reenactment, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 7184–7193.
[21] D.P. Kingma, M. Welling, An introduction to variational autoencoders, arXiv preprint arXiv:1906.02691(2019).
[22] H.H. Nguyen, J. Yamagishi, I. Echizen, Capsule-forensics: using capsule networks to detect forged images and videos, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 2307–2311.
[23] J. Yang, A. Li, S. Xiao, W. Lu, X. Gao, MTD-Net: learning to detect deepfakes images by multi-scale texture difference, IEEE Trans. Inf. Forensics Secur. 16 (2021) 4234–4245.
[24] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, N. Yu, Spatial-phase shallow learning: rethinking face forgery detection in frequency domain, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 772–781.
[25] Z. Shang, H. Xie, Z. Zha, L. Yu, Y. Li, Y. Zhang, PRRNet: pixel-region relation network for face forgery detection, Pattern Recognit. 116 (2021) 107950.
[26] A. Agarwal, A. Agarwal, S. Sinha, M. Vatsa, R. Singh, MD-CSDNetwork: multi-domain cross stitched network for deepfake detection, in: 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), IEEE, 2021, pp. 1–8.
[27] Y. Luo, Y. Zhang, J. Yan, W. Liu, Generalizing face forgery detection with high-frequency features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 16317–16326.
[28] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, W. Xia, Learning self-consistency for deepfake detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2021, pp. 15023–15033.
[29] X. Zhang, S. Karaman, S.-F. Chang, Detecting and simulating artifacts in GAN fake images, in: Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2019, pp. 1–6.
[30] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[31] A.G. Roy, N. Navab, C. Wachinger, Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2018, pp. 421–429.
[32] D.E. King, Dlib-ml: a machine learning toolkit, J. Mach. Learn. Res. 10 (2009) 1755–1758.
[33] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, C.C. Ferrer, The deepfake detection challenge (DFDC) preview dataset, arXiv preprint arXiv:1910.08854(2019).
[34] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, C.C. Ferrer, The deepfake detection challenge dataset, arXiv preprint arXiv:2006.07397(2020).
[35] P. Korshunov, S. Marcel, DeepFakes: a new threat to face recognition? Assessment and detection, arXiv preprint arXiv:1812.08685(2018).
[36] B. Zi, M. Chang, J. Chen, X. Ma, Y.-G. Jiang, WildDeepfake: a challenging real-world dataset for deepfake detection, in: Proceedings of the 28th ACM International Conference on Multimedia (ACM MM), 2020, pp. 2382–2390.
[37] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Process. Lett. 23 (10) (2016) 1499–1503.
[38] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980(2014).
[39] C. Wang, W. Deng, Representative forgery mining for fake face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 14923–14932.
[40] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626.

Han Chen is currently a master student at the Shenzhen University majoring in Information and Communication Engineering. He received the BS degree in Electronic Information Engineering from Shenzhen University, China in 2020. His current research interests include multimedia forensics and deep learning.

Yuezun Li is a lecturer in computer science at Ocean University of China. He received PhD degree in Computer Science from the University of Albany, State University of New York in 2020, and MS degree in Computer Science in 2015 and BS degree in Software Engineering in 2012 both from Shandong University. His research interest is mainly focused on computer vision and multimedia forensics.

Dongdong Lin is a PhD student in Shenzhen University, China. He is now visiting the Visual Information Processing and Protection (VIPP) group, in University of Siena, Italy. He received his MS degree and BS degree both in computer science at the College of Information Engineering, Xiangtan University, China in 2015 and 2018, respectively. His research interests include and multimedia security and adversarial machine learning.

Bin Li is a professor with Shenzhen University, China. He received the BE degree and the PhD degree from Sun Yat-sen University, China, in 2004 and 2009, respectively. He is the Director of Shenzhen Key Laboratory of Media Security. His current research interests include multimedia forensics and pattern recognition.

Junqiang Wu is an engineer at Shenzhen University. He obtained a master's degree in Science of Public Management from Guangxi University for Nationalities in 2016, a bachelor of science in Applied Physics and a bachelor of engineering in Computer Science and Technology from Shenzhen University in 2006. His research interests mainly focused on big data analysis and artificial intelligence.