# FreqBlender: Enhancing DeepFake Detection by Blending Frequency Knowledge

**Hanzhe Li**[1,*]  **Jiaran Zhou**[1,*] **Yuezun Li**[1,†] **Baoyuan Wu**[2]  **Bin Li**[3]  **Junyu Dong**[1]

[1] School of Computer Science and Technology, Ocean University of China
[2] School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
[3] Guangdong Provincial Key Laboratory of Intelligent Information Processing,
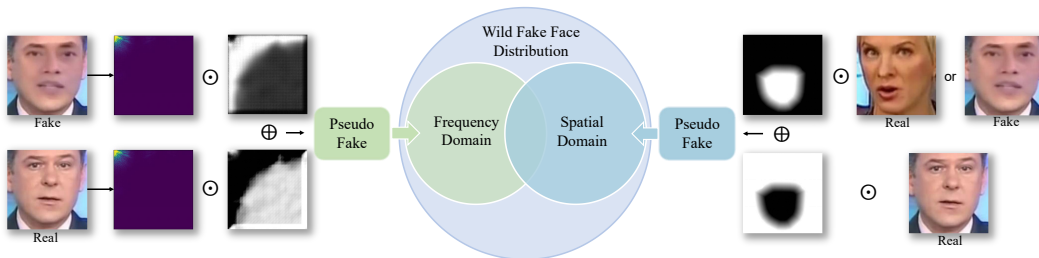Shenzhen University, Shenzhen, China

Figure 1: Overview of our method. In contrast to the existing spatial-blending methods (right part), our method explores face blending in frequency domain (left part). By leveraging the frequency knowledge, our method can generate pseudo-fake faces that closely resemble the distribution of wild fake faces. Our method can complement and work in conjunction with existing spatial-blending methods.

## Abstract

Generating synthetic fake faces, known as pseudo-fake faces, is an effective way to improve the generalization of DeepFake detection. Existing methods typically generate these faces by blending real or fake faces in spatial domain. While these methods have shown promise, they overlook the simulation of frequency distribution in pseudo-fake faces, limiting the learning of generic forgery traces in-depth. To address this, this paper introduces *FreqBlender*, a new method that can generate pseudo-fake faces by blending frequency knowledge. Concretely, we investigate the major frequency components and propose a Frequency Parsing Network to adaptively partition frequency components related to forgery traces. Then we blend this frequency knowledge from fake faces into real faces to generate pseudo-fake faces. Since there is no ground truth for frequency components, we describe a dedicated training strategy by leveraging the inner correlations among different frequency knowledge to instruct the learning process. Experimental results demonstrate the effectiveness of our method in enhancing DeepFake detection, making it a potential plug-and-play strategy for other methods.

## 1  Introduction

DeepFake refers to face forgery techniques that can manipulate facial attributes, such as identity, expression, and lip movement [1]. The recent advancement of deep generative models [2, 3] has

---

[*]Equal contribution
[†]Corresponds to Yuezun Li (`liyuezun@ouc.edu.cn`)

greatly sped up the evolution of DeepFake techniques, enabling the creation of highly realistic and visually imperceptible manipulations. However, the misuse of these techniques can pose serious security concerns [4], making DeepFake detection more pressing than ever before.

There have been many methods proposed for detecting DeepFakes, showing their effectiveness on public datasets [5, 6, 7, 8, 9]. However, with the continuous growth of AI techniques, new types of forgeries constantly emerge, posing a challenge for current detectors to accurately expose unknown forgeries. To address this challenge, recent efforts [10, 11, 12, 13] have focused on improving the generalizability of detection, *i.e.*, the ability to detect unknown forgeries based on known examples. One effective approach to address this problem is to enhance the training data by generating synthetic fake faces, known as *pseudo-fakes* [14, 15, 16]. The intuition behind this approach is that the DeepFake generation process introduces artifacts in the step of blending faces, and these methods generate pseudo-fake faces by simulating various blending artifacts. By training on these pseudo-fake faces, the models can be driven to learn corresponding artifacts. However, existing methods concentrate on simulating the ***spatial*** aspects of face blending (see Fig. 1 (right)). While they can make the pseudo-fake faces resemble the distribution of wild fake faces in the spatial domain, they do not explore the distribution in the ***frequency*** domain. Thus, current pseudo-fake faces lack frequency-based forgery clues, limiting the models to learn generic forgery features.

In this paper, we shift our attention from the spatial domain to the frequency domain and propose a new method called *FreqBlender* to generate pseudo-fake faces by blending frequency knowledge (see Fig. 1 (left)). To achieve this, we analyze the composition of the frequency domain and accurately identify the range of forgery clues falling into. Then we replace this range of real faces with the corresponding range of fake faces to generate pseudo-fake faces. However, identifying the frequency range of forgery clues is challenging due to two main reasons: 1) this range varies across different fake faces due to its high dependence on face content, and 2) forgery clues may not be concentrated on a single frequency range but could be an aggregation of various portions across multiple ranges. Thus, general low-pass, high-pass, or band-pass filters are incapable of precisely pinpointing the distribution.

To address this challenge, we propose a Frequency Parsing Network (FPNet) that can adaptively partition the frequency domain based on the input faces. Specifically, we hypothesize that the faces are composed of three frequency knowledge, which represents *semantic information, structural information, and noise information*, respectively, and the forgery traces are likely hidden in structural information. This hypothesis is validated in our preliminary analysis (refer to Sec. 3 for details). Based on this footstone, we design the network consisting of a shared encoder and three decoders to extract corresponding frequency knowledge. The encoder transforms the input data into a latent frequency representation, while the decoders estimate the probability map of the corresponding frequency knowledge.

Training this network is non-trivial since no ground truth of frequency distribution is provided. Therefore, we propose a novel training strategy that leverages the inner correlations among different frequency knowledge. To be specific, we describe dedicated-crafted objectives that are performed on various blending combinations of the output from each decoder and emphasize the properties of each frequency knowledge. The experimental results demonstrate that the network successfully parses the desired frequency knowledge within the proposed training strategy.

Once the network is trained, we can parse the frequency component corresponding to the structural information of a fake face, and blend it with a real face to generate a pseudo-fake face. It is important to note that our method is not in conflict with existing spatial blending methods, but rather complements them by addressing the defect in the frequency domain. Our method is validated on multiple recent DeepFake datasets (*e.g.*, FF++ [5], CDF [6], DFDC [8], DFDCP [7], FFIW [9]) and compared with many state-of-the-art methods, demonstrating the efficacy of our method in improving detection performance.

The contributions of this paper are summarized in three-fold: **1)** To the best of our knowledge, we are the first to generate pseudo-fake faces by blending frequency knowledge. Our method pushes pseudo-fake faces closer to the distribution of wild fake faces, enhancing the learning of generic forgery features in DeepFake detection.**2)** We propose a Frequency Parsing Network that can adaptively partition the frequency components corresponding to semantic information, structural information, and noise information, respectively. Since no ground truth is provided, we design dedicated objectives

to train this network. **3)** Extensive experimental results on several DeepFake datasets demonstrate the efficacy of our method and its potential as a plug-and-play strategy for existing methods.

## 2 Related Works

The rapid progress of AI generative models has spawned the development of DeepFake detection methods. These methods mainly rely on deep neural networks to identify the inconsistency between real and fake faces using various features, including biological signals [17], spatial artifacts [18, 14, 15, 16], frequency abnormality [13, 19, 20], auto-learned clues from dedicated-designed models [21, 22]. These methods have shown promising results on public datasets. However, some of their performance significantly deteriorates when confronted with unknown DeepFake faces due to the large distribution discrepancy resulting from limited training datasets. To tackle this issue, many methods have been proposed to improve their generalizability by learning the generic DeepFake traces, *e.g.*, [10, 14, 15, 16, 12, 23, 24]. One effective approach is to create synthetic fake faces during training, known as pseudo-fake faces, *e.g.*, [10, 14, 15, 16]. FWA [10] is a pioneering method that conducts self-blending to simulate fake faces. Several extended variants (Face X-ray [14], PCL [15], SBI [16], BiG-Arts [25]) have been proposed to blend faces using curated strategies, further improving detection performance. By increasing the diversity of training faces, the gap in the distribution of wild fake faces can be reduced, allowing the models to learn the invariant DeepFake traces across different distributions. To generate the pseudo-fake faces, existing methods usually design spatial blending operations to combine different faces. This involves extracting the face region from a source image and blending it into a target image. However, these methods overlook the distribution of wild fake faces in the frequency domain. While the synthetic faces may resemble the spatial-based distribution, the lack of consideration for frequency perspective hinders the models from learning the fundamental generic DeepFake traces.

## 3 Preliminary Analysis

We perform a statistical analysis of the frequency distribution of real and fake faces and present preliminary results for the main frequency components corresponding to semantic information, structural formation, and noise information, respectively.

**Inspiration and Verification.** The investigation in previous works [26, 27] has indicated that the forgery traces mainly exist in high-frequency areas. However, the precise range of these areas has not been described, driving us to re-investigate the frequency distribution of forgery traces.

Specifically, we conduct verification experiments using FaceForensics++ (FF++) [5] datasets. We extract the frames from all videos and randomly select $3,000$ real images and $3,000$ fake images for each manipulation method (*e.g.*, DF, F2F, FS, and NT). Then we crop out the face region in these selected images using a face detector [28] and apply DCT [29] to generate frequency maps. For analysis, we sum up all frequency maps of real and fake images and adopt the visualization process of azimuthal average described in previous work [30, 27]. This process involves logarithmic transformation and the calculation of azimuthally-averaged flux in circular annuli apertures. By placing the center of the circular annuli aperture at the top-left corner of the frequency map, we can obtain a one-dimensional array representing the spectrum diagram. The visual results of their distribution are shown in Fig. 2 (top). It can be observed that this figure is consistent with the results in [26]. However, when we directly plot their distribution differences without logarithmic operation, the results do not match the previous figure. It can be seen that the disparities in high-frequency regions are not as substantial as expected, while the differences in the lower range become more noticeable (see Fig. 2 (bottom)). This is because the logarithmic operation mitigates the degree of differences in lower frequency ranges, causing the illusion that only the high-frequency range exhibits differences between real and fake faces. Therefore, we conjecture that the forgery traces may not only be concentrated in a very high-frequency range but could possibly spread to the low-frequency range.

**Hypothesis and Validation.** As shown in Fig. 2, the most significant difference can be observed in the range of very low frequency. Given the significant dissimilarity in appearance between real and fake faces, we hypothesize that the semantic information is mainly represented in this low-frequency band. Moreover, we hypothesize that the mid-to-high frequency components capture the structural information, making them more susceptible to containing forgery traces. Furthermore, we hypothesize
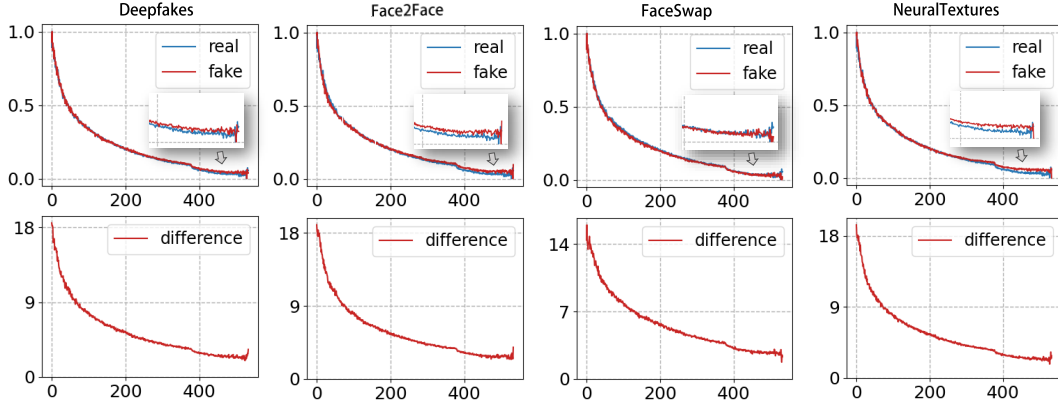
3

Figure 2: Statistics of frequency distribution. The top part shows the frequency distribution of real and fake faces using algorithms in [30, 27]. The bottom part shows the frequency difference between real and fake. The values on the vertical axis are logarithmic with 2.
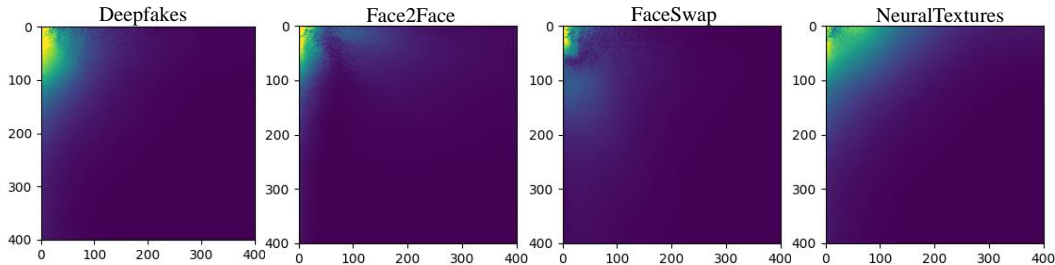


Figure 3: Visualization of the frequency difference between real and fake faces. The lighter color indicates the larger difference.

that the highest frequency components likely correspond to the noise introduced by various video preprocessing operations, such as compression, decompression, and encoding.

To validate our hypothesis, we directly visualize the difference between real and fake faces on their frequency maps in Fig. 3. By observing these results, we empirically split the frequency map into three non-overlap bands. The split operations follow the general band-pass filters. Denote the position in the frequency map as $(x, y)$, where $(0, 0), (1, 1)$ denotes the top-left corner and bottom-right corner. Specifically, we identify the region where $x + y \leq 1/16$ as containing semantic information, the region where $1/16 < x + y \leq 1/2$ as containing structural information, and the region where $x + y > 1/2$ as containing noise information. The corresponding results are visualized in Fig. 4, validating that these three ranges provide empirical evidence that aligns with our frequency distribution hypothesis.



Figure 4: Image visualization corresponding to different frequency components.

## 4   FreqBlender

We describe a new method to create pseudo-fakes by blending specific frequency knowledge. The motivation is that existing methods only focus on spatial domain blending, which overlook the disparity between real and fake faces in the frequency domain. By considering the frequency distribution, the pseudo-fakes can closely resemble the fake faces. To achieve this, we propose

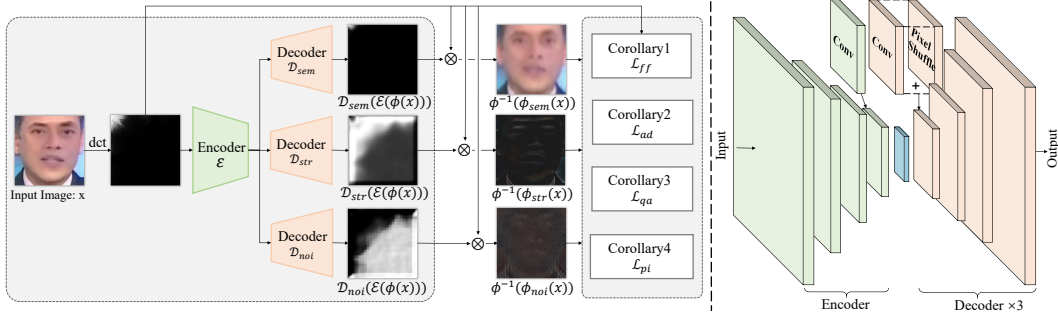Figure 5: Overview of the proposed Frequency Parsing Network (FPNet). Given an input face image, our method can partition it into three frequency components, corresponding to the semantic information, structural information, and noise information respectively. Since there is no ground truth, we propose four corollaries to supervise the training. The architecture of the encoder and decoders is shown in the right part.

a Frequency Parsing Network (FPNet) to partition the frequency domain into three components, corresponding to semantic information, structural information, and noise information, respectively. We then blend the structural information of fake faces with the real faces to generate the pseudo-fakes. The details of the Frequency Parsing Network are elaborated in Sec. 4.1, and the objective and training process for this network is described in Sec. 4.2. Then we introduce the deployment of our method with existing methods in Sec. 4.3.

## 4.1 Frequency Parsing Network

**Overview.** The Frequency Parsing Network (FPNet) is composed of one shared encoder and three independent decoders. The encoder transforms the input faces into frequency-critical features and the decoders aim to decompose the feature from the encoder and extract the respective frequency components.

Denote the encoder as $\mathcal{E}$ and three decoders as $\mathcal{D}_{\mathrm{sem}}, \mathcal{D}_{\mathrm{str}}, \mathcal{D}_{\mathrm{noi}}$ respectively. Given an input face image $\mathbf{x} \in \mathcal{X} \in \{0, 255\}^{h \times w \times 3}$, we first convert this face to the frequency map as $\phi(\mathbf{x}) \in \mathbb{R}^{h \times w \times 3}$, where $\phi$ denotes the operations of Discrete Cosine Transform (DCT). Then we send this frequency map into model and generate three distribution maps as $\mathcal{D}_{\mathrm{sem}}(\mathcal{E}(\phi(\mathbf{x}))) \in [0, 1]^{h \times w}, \mathcal{D}_{\mathrm{str}}(\mathcal{E}(\phi(\mathbf{x}))) \in [0, 1]^{h \times w}$, and $\mathcal{D}_{\mathrm{noi}}(\mathcal{E}(\phi(\mathbf{x}))) \in [0, 1]^{h \times w}$ respectively. Each distribution map indicates the probability of the corresponding frequency component distributed in the frequency map. Given these distribution maps, we can select the corresponding frequency components conveniently. For example, the frequency component corresponding to semantic information can be selected by $\phi_{\mathrm{sem}}(\mathbf{x}) = \phi(\mathbf{x})\mathcal{D}_{\mathrm{sem}}(\mathcal{E}(\phi(\mathbf{x})))$ and the same is for other two frequency components, *i.e.*, $\phi_{\mathrm{str}}(\mathbf{x}) = \phi(\mathbf{x})\mathcal{D}_{\mathrm{str}}(\mathcal{E}(\phi(\mathbf{x})))$ and $\phi_{\mathrm{noi}}(\mathbf{x}) = \phi(\mathbf{x})\mathcal{D}_{\mathrm{noi}}(\mathcal{E}(\phi(\mathbf{x})))$. The overview of FPNet is shown in Fig. 5 (left).

**Network Architecture.** The encoder simply consists of four convolution layers with a kernel size of $3 \times 3$, a stride of 2, and a padding of 1. Each decoder also consists of four layers, and each layer is a combination of a convolutional layer and PixelShuffle operation [31] (see Fig. 5 (right)).

## 4.2 Objective Design for FPNet

The most challenging and crucial aspect of our method is to train the network for frequency parsing, as there is no ground truth available for the different frequency components. Note that the only available resources for supervising the training are the preliminary analysis results in Section 3. Nevertheless, these results are not precise and can not be adaptive to different inputs, which are insufficient for model training. Therefore, we meticulously craft a couple of auxiliary objectives to instruct the learning of networks, allowing for the self-refinement of the network.

These objectives are designed based on the following proposition.

**Proposition 1.** *Each frequency component exhibits the following properties:*

  *1. Semantic information can reflect the facial identity.*

*2. Structural information serves as the carrier of forgery traces.*

*3. Noise information has minimal impact on visual quality.*

*4. The preliminary analysis findings are generally applicable.*

**Corollary 1.** *For a given face $\mathbf{x}$, the transformed face based on its semantic information will retain the same facial identity as $\mathbf{x}$, i.e., $\forall\, \mathbf{x} \in \mathcal{X} \in \{0, 255\}^{h \times w \times 3}$, $\mathcal{F}(\phi^{-1}(\phi_{\mathrm{sem}}(\mathbf{x}))) = \mathcal{F}(\mathbf{x})$, where $\mathcal{F}$ denotes a face recognition model and $\phi^{-1}$ denotes the Inverse Discrete Cosine Transform (IDCT).*

**Facial Fidelity Loss.** We introduce a facial fidelity loss $\mathcal{L}_{\mathrm{ff}}$ to penalize the discrepancy in identity between the input face image and the spatial content represented by semantic information. To measure the identity discrepancy, we employ the MobileNet [32] as our face recognition model and train it using ArcFace [33, 34]. We select MobileNet for its balance between computational efficiency and recognition accuracy. Let $\mathcal{F}$ be the MobileNet and $\mathcal{F}_f(\mathbf{x})$ be the facial features extracted from the face image $\mathbf{x}$. The facial fidelity loss can be defined as

$$\mathcal{L}_{\mathrm{ff}}(\mathbf{x}) = \|\mathcal{F}_f(\phi^{-1}(\phi_{\mathrm{sem}}(\mathbf{x}))) - \mathcal{F}_f(\mathbf{x})\|_2^2. \tag{1}$$

Note that the input face $\mathbf{x}$ can be either real or fake, as the identity information is present in both cases.

**Corollary 2.** *For a given real face $\mathbf{x}_r$, it can be detected as fake if and only if it is inserted the structural information from a fake face $\mathbf{x}_f$, i.e., $\mathcal{D}(\mathbf{x}_r) = 0$ iff $\mathbf{x}_r \leftarrow \mathbf{x}_r \oplus \phi_{\mathrm{str}}(\mathbf{x}_f)$, where $\mathcal{D}$ denotes a Deepfake detector with labels of fake and real in $\{0, 1\}$, $\oplus$ indicates the inserting operation.*

**Authenticity-determinative Loss.** This loss is designed to emphasize the determinative role of structural information. To evaluate the authenticity of faces, we develop a DeepFake detector $\mathcal{D}$, which is implemented using a ResNet-34 [35] trained on real and fake faces. Then we construct two sets of faces by blending frequency components.

The first set contains three types of faces transformed from frequency components corresponding to 1) the semantic information of the real face, 2) the semantic information of the fake face, and 3) the semantic information of the real face blended with the structural information of the real face. We denote this set as $\mathcal{C}_r = \{\phi^{-1}(\phi_{\mathrm{sem}}(\mathbf{x}_r)), \phi^{-1}(\phi_{\mathrm{sem}}(\mathbf{x}_f)), \phi^{-1}(\phi_{\mathrm{sem}}(\mathbf{x}_r) + \phi_{\mathrm{str}}(\mathbf{x}_r))\}$. Since there is no structural information from fake faces in this set, all the faces should be detected as real.

Similarly, the second set contains two types of faces: 1) blending the semantic information of the fake face with the structural information of the fake face, and 2) blending the semantic information of the real face with the structural information of the fake face. We denote this set as $\mathcal{C}_f = \{\phi^{-1}(\phi_{\mathrm{sem}}(\mathbf{x}_f) + \phi_{\mathrm{str}}(\mathbf{x}_f)), \phi^{-1}(\phi_{\mathrm{sem}}(\mathbf{x}_r) + \phi_{\mathrm{str}}(\mathbf{x}_f))\}$. Since all blended faces in this set contain the structural information of fake faces, they should be detected as fake. Thus the authenticity-determinative loss $\mathcal{L}_{\mathrm{ad}}$ can be written as

$$\mathcal{L}_{\mathrm{ad}}(\mathbf{x}_r, \mathbf{x}_f) = \frac{1}{|\mathcal{C}_r|} \sum_{\mathbf{x} \in \mathcal{C}_r} \mathrm{CE}(\mathbf{x}, 1) + \frac{1}{|\mathcal{C}_f|} \sum_{\mathbf{x} \in \mathcal{C}_f} \mathrm{CE}(\mathbf{x}, 0) \tag{2}$$

where CE denotes the cross-entropy loss.

**Corollary 3.** *The face should exhibit no visible change if the frequency component of noise information is removed, i.e., $\forall\, \mathbf{x} \in \mathcal{X} \in \{0, 255\}^{h \times w \times 3}$, $\mathbf{x} \approx \mathbf{x} \ominus \phi_{\mathrm{noi}}(\mathbf{x}_f)$, where $\ominus$ indicates the removing operation.*

**Quality-agnostic Loss.** As noise information does not contain decisive details for the overall depiction of the image, the face image is expected to be similar to the face image transformed using the frequency components of semantic and structural information. This similarity can be quantified using the quality-agnostic Loss $\mathcal{L}_{\mathrm{qa}}$, defined as

$$\mathcal{L}_{\mathrm{qa}}(\mathbf{x}) = \|\mathbf{x} - \phi^{-1}(\phi_{\mathrm{sem}}(\mathbf{x}) + \phi_{\mathrm{str}}(\mathbf{x}))\|_2^2, \tag{3}$$

where the face $\mathbf{x}$ can be either real or fake.

**Corollary 4.** *Each frequency component is bound by the preliminary results, i.e., there should be no significant deviation between the predicted frequency component and the approximate frequency distribution in preliminary analysis.*

**Prior and Integrity Loss.** According to our analysis in the Preliminary Analysis section, we have an initial understanding of the approximate frequency distribution. Denote the initial frequency maps for semantic, structural, and noise information as $m_{\text{sem}}, m_{\text{str}}, m_{\text{noi}}$, respectively. These maps are utilized to accelerate the convergence of the model towards the desired direction. Moreover, we add a constraint on the integrity of their distributions, ensuring that their combination covers all elements of the frequency map. This loss $\mathcal{L}_{\text{pi}}$ can be expressed as

$$\mathcal{L}_{\text{pi}} = \|\mathcal{D}_{\text{sem}}(\mathcal{E}(\phi(\mathbf{x}))) - m_{\text{sem}}\|_2^2 + \|\mathcal{D}_{\text{str}}(\mathcal{E}(\phi(\mathbf{x}))) - m_{\text{str}}\|_2^2 + \|\mathcal{D}_{\text{noi}}(\mathcal{E}(\phi(\mathbf{x}))) - m_{\text{noi}}\|_2^2 +$$
$$\|(\mathcal{D}_{\text{sem}}(\mathcal{E}(\phi(\mathbf{x}))) + \mathcal{D}_{\text{str}}(\mathcal{E}(\phi(\mathbf{x}))) + \mathcal{D}_{\text{noi}}(\mathcal{E}(\phi(\mathbf{x})))) - \mathbf{1}\|_2^2,$$

(4)

where $\mathbf{1}$ denotes a mask where all the elements in it is $1$.

**Overall Objectives.** The overall objectives are the summation of all these loss terms, as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{ff}} + \lambda_2 \mathcal{L}_{\text{ad}} + \lambda_3 \mathcal{L}_{\text{qa}} + \lambda_4 \mathcal{L}_{\text{pi}},$$

(5)

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the weights for different loss terms.

### 4.3 Deployment of FreqBlender

Given a fake face $\mathbf{x}_r$ and a real face $\mathbf{x}_f$, we can generate a pseudo-fake face by

$$\mathbf{x}_f' = \phi^{-1}\left(\phi(\mathbf{x}_r)\mathcal{D}_{\text{sem}}(\mathcal{E}(\phi(\mathbf{x}_f))) + \phi(\mathbf{x}_f)\mathcal{D}_{\text{str}}(\mathcal{E}(\phi(\mathbf{x}_f))) + \phi(\mathbf{x}_r)\mathcal{D}_{\text{noi}}(\mathcal{E}(\phi(\mathbf{x}_f)))\right).$$

(6)

🔊 *Note that in our method, **it is not necessary to perform the blending using wild fake faces.** Instead, we can tactfully substitute wild fake faces with the pseudo-fake faces generated by existing spatial face blending methods. It allows us to overcome the limitations in the frequency distribution of existing pseudo-fake faces.*

## 5 Experiments

### 5.1 Experimental Setups

**DataSets.** Our method is evaluated using several standard datasets, including FaceForensics++ [5] (FF++), Celeb-DF (CDF) [6], DeepFake Detection Challenge (DFDC) [8], DeepFake Detection Challenge Preview (DFDCP) [7], and FFIW-10k (FFIW) [9] datasets. Specifically, the FF++ dataset consists of 1000 pristine videos and 4000 manipulated videos corresponding to four different manipulation methods, that are Deepfakes (DF), Face2Face (F2F), FaceSwap (FS), and NeuralTextures (NT). CDF dataset comprises 590 pristine videos and 5639 high-quality fake videos created from DeepFake alterations of celebrity videos available on YouTube. DFDC is a large-scale deepfake dataset, that consists of $100,000$ video clips, and DFDCP is a preview version of DFDC, which is also widely used in evaluation. The FFIW dataset contains 8250 pristine videos and 8250 DeepFake videos with multi-face scenarios. We follow the original training and testing split provided by the datasets for experiments.

**Implementation Details.** Our method is implemented using PyTorch 2.0.1 [36] with a Nvidia 3090ti. In the training stage of FPNet, the image size is set to $400 \times 400$. The batch size is set to 8 and the Adam optimizer is utilized with an initial learning rate of $1e^{-4}$. The training epoch is set to 200. The hyperparameters in the objective function in Eq. (5) are set as follows: $\lambda_1 = 1/12, \lambda_2 = 1, \lambda_3 = 1e^{-3}, \lambda_4 = 1/4$. For DeepFake detection, we employ the vanilla EfficientNet-b4 [37] as our model following [16]. In the training phase, we create pseudo-fake faces on-the-fly. We first generate synthetic faces using the spatial-blending method [16] and then blend them with real faces using our method with a probability of $\alpha = 0.2$. Other training and testing settings are the same as [16]. *More analysis of parameters are provided in Supplementary.*

### 5.2 Results

To showcase the effectiveness of our method, we train our method solely on the FF++ dataset and test it on the other different datasets. We employ the Area Under the Receiver Operating Characteristic Curve (AUC) as the evaluation metric following previous work [16]. Our method is compared with **five** video-based detection methods, including **Two-branch** [38], **DAM** [9], **LipForensics** [1],

Table 1: The cross-dataset evaluation of different methods. Blue indicates best and red indicates second best.

| Method | Input Type | Training Set | | Test Set AUC (%) | | | |
|---|---|---|---|---|---|---|---|
| | | Real | Fake | CDF | DFDC | DFDCP | FFIW |
| Two-branch (ECCV'20) [38] | Video | ✓ | ✓ | 76.65 | - | - | - |
| DAM (CVPR'21) [9] | Video | ✓ | ✓ | 75.3 | - | 72.8 | - |
| LipForensics (CVPR'21) [1] | Video | ✓ | ✓ | 82.4 | 73.50 | - | - |
| FTCN (ICCV'21) [39] | Video | ✓ | ✓ | 86.9 | 71.00 | 74.0 | 74.47 |
| SST (CVPR'24) [24] | Video | ✓ | ✓ | 89.0 | - | - | - |
| DSP-FWA (CVPRW'19 [10]) | Frame | ✓ | ✓ | 69.30 | - | - | - |
| Face X-ray (CVPR'20) [14] | Frame | ✓ | - | - | - | 71.15 | - |
| Face X-ray (CVPR'20) [14] | Frame | ✓ | ✓ | - | - | 80.92 | - |
| F3-Net (ECCV'20) [29] | Frame | ✓ | ✓ | 72.93 | 61.16 | 81.96 | 61.58 |
| LRL (AAAI'21) [40] | Frame | ✓ | ✓ | 78.26 | - | 76.53 | - |
| FRDM (CVPR'21) [41] | Frame | ✓ | ✓ | 79.4 | - | 79.7 | - |
| PCL+I2G (ICCV'21) [15] | Frame | ✓ | - | 90.03 | 67.52 | 74.37 | - |
| DCL (AAAI'22) [42] | Frame | ✓ | ✓ | 82.30 | - | 76.71 | 71.14 |
| SBI* (CVPR'22) [16] | Frame | ✓ | - | 92.94 | 72.08 | 85.51 | 85.99 |
| SBI (CVPR'22) [16] | Frame | ✓ | - | 93.18 | 72.42 | 86.15 | 84.83 |
| TALL-Swin (ICCV'23) [22] | Frame | ✓ | ✓ | 90.79 | 76.78 | - | - |
| UCF (ICCV'23) [12] | Frame | ✓ | ✓ | 82.4 | 80.5 | - | - |
| BiG-Arts (PR'23) [25] | Frame | ✓ | ✓ | 77.04 | - | 80.48 | - |
| F-G (CVPR'24) [43] | Frame | ✓ | ✓ | 74.42 | 61.47 | - | - |
| LSDA (CVPR'24) [23] | Frame | ✓ | ✓ | 83.0 | 73.6 | 81.5 | - |
| **FreqBlender (Ours)** | Frame | ✓ | - | **94.59** | 74.59 | **87.56** | **86.14** |

**FTCN** [39] and **SST** [24]. Moreover, we involve **thirteen** frame-level state-of-the-art methods for comparison, which are **DSP-FWA** [10], **Face X-ray** [14], F3-Net [29], **LRL** [40], **FRDM** [41], **PCL** [15], **DCL** [42], **SBI** [16], **TALL-Swin** [22], **UCF** [12], **SST** [24], **F-G** [43], **LSDA** [23],respectively.

**Cross-dataset Evaluation.** We evaluate the cross-dataset performance of our method compared to other counterparts in Table 1. The best performance is highlighted in blue and the second-best is marked by red. It should be noted that our method operates on pseudo-fake faces generated by SBI, thus we do not need fake faces. In comparison to video-level methods, our method achieves the best performance, which outperforms all the methods by a large margin.

When compared to frame-level methods, our method still outperforms the others. For example, our method improves upon the performance of the most relevant counterpart SBI by $1.41\%$, $2.17\%$, $1.41\%$, $1.31\%$ on CDF, DFDC, DFDCP, and FFIW respectively. This improvement can be attributed to the incorporation of frequency knowledge in pseudo-fake faces, enhancing the generalization of detection models. Note that the performance of the compared methods (except SBI* and F3-Net) is extracted from their original papers. SBI* denotes the performance obtained using the officially released codes, and F3-Net is reproduced using the codes implemented by others [3]. The results closely align with the reported scores, which verifies the correctness of our configuration of their codes. In subsequent experiments, we employ their release codes for comparison.

**Cross-manipulation Evaluation.** Since SBI is the most recent and effective method, we compare our method with it for demonstration. Specifically, we compare our method with two variants of the SBI method. The first is trained using the raw set of real videos in the FF++ dataset, while the second is trained using the c23 set. According to the standard protocols, all methods are tested on c23 videos. The results are shown in Table 2. It can be seen that our method outperforms SBI-raw by $3.62\%$ and SBI-c23 by $3.89\%$, demonstrating the efficacy of our method on cross-manipulation scenarios.

### 5.3 Analysis

**Effect of Each Objective Term.** This part studies the effect of each objective term on CDF, DFDC, and DFDCP datasets. The results are shown in Table 3. Note that Baseline denotes only using prior and integrity loss $\mathcal{L}_{\text{pi}}$ and "w/o" denotes without. It can be seen that without one certain objective

---
[3]F3-Net:https://github.com/Leminhbinh0209/F3Net

Table 2: The cross-manipulation evaluation of different methods.

| Method | FF++ | | | | Avg |
|---|---|---|---|---|---|
| | DF | F2F | FS | NT | |
| SBI-raw [16] | 98.35 | 91.07 | 96.92 | 83.69 | 92.51 |
| SBI-c23 [16] | 98.60 | 92.60 | 95.44 | 82.30 | 92.24 |
| **FreqBlender (Ours)** | 99.18 | 96.76 | 97.68 | 90.88 | **96.13** |

Table 3: Effect of each objective term.

| Setting | CDF | DFDC | DFDCP | Avg |
|---|---|---|---|---|
| Baseline | 91.69 | 72.69 | 86.67 | 83.68 |
| w/o $\mathcal{L}_{ff}$ | 94.01 | 74.30 | 86.86 | 85.06 |
| w/o $\mathcal{L}_{ad}$ | 93.31 | 72.97 | 86.32 | 84.20 |
| w/o $\mathcal{L}_{qa}$ | 94.28 | 74.42 | 87.25 | 85.32 |
| w/o $\mathcal{L}_{pi}$ | 93.78 | 74.03 | 85.99 | 84.60 |
| All | **94.59** | **74.59** | **87.56** | **85.58** |

Table 4: Effect of our method complementary to spatial-blending methods.

| Method | FF++ | CDF | DFDCP | FFIW | Avg |
|---|---|---|---|---|---|
| DSP-FWA [10] | 48.14 | 62.91 | 60.74 | 40.65 | 53.11 |
| DSP-FWA [10] + Ours | 49.46 | 65.47 | 56.18 | 41.81 | **53.23** |
| I2G [15] | 59.56 | 53.55 | 48.02 | 46.75 | 51.97 |
| I2G [15] + Ours | 63.84 | 48.89 | 49.53 | 48.96 | **52.81** |
| Face X-ray [14] | 82.26 | 67.99 | 65.00 | 63.65 | 69.73 |
| Face X-ray [14] + Ours | 84.03 | 76.05 | 63.90 | 67.24 | **72.81** |

Table 5: The effect of our method on different networks.

| Method | CDF | FF++ | DFDCP | FFIW | Avg |
|---|---|---|---|---|---|
| ResNet-50 [35] + SBI | 84.82 | 95.39 | 73.51 | 81.67 | 83.85 |
| ResNet-50 [35] + Ours | 85.44 | 94.61 | 76.16 | 86.32 | **85.63** |
| EfficientNet-b1 [37] + SBI | 90.25 | 94.66 | 87.54 | 82.55 | 88.75 |
| EfficientNet-b1 [37] + Ours | 90.53 | 94.65 | 87.70 | 83.76 | **89.16** |
| VGG16 [44] + SBI | 78.22 | 93.05 | 74.13 | 87.26 | **83.16** |
| VGG16 [44] + Ours | 78.38 | 93.10 | 73.47 | 87.63 | 83.15 |
| Xception [45] + SBI | 87.00 | 91.40 | 75.68 | 70.24 | 81.08 |
| Xception [45] + Ours | 90.52 | 93.32 | 76.07 | 70.43 | **82.59** |
| ViT [46] + SBI | 85.85 | 96.09 | 87.71 | 86.05 | 88.92 |
| ViT [46] + Ours | 86.34 | 96.10 | 87.17 | 86.88 | **89.12** |
| F3-Net [29] + SBI | 84.94 | 93.42 | 79.29 | 73.42 | 82.77 |
| F3-Net [29] + Ours | 88.10 | 95.16 | 84.32 | 74.49 | **85.52** |
| GFFD [41] + SBI | 81.34 | 91.81 | 77.19 | 65.53 | 78.97 |
| GFFD [41] + Ours | 86.71 | 92.18 | 78.25 | 77.45 | **83.65** |

term, the performance drops on all datasets, which demonstrates that different objective terms have distinct impacts, and their collective contributes most to our method.

**Complementary to Spatial-blending Methods.** To validate the complementary of our method, we replace the SBI method with other spatial-blending methods and study if the performance is improved. Specifically, we reproduce the pseudo-fake face generation operations in DSP-FWA, I2G, and Face X-ray, and combine them with our method. Note that I2G and Face X-ray have not released their codes, we re-implement them rigorously following their original settings. The results on FF++, CDF, DFDCP, FFIW datasets are presented in Table 4. It can be seen that by combining with DSP-FWA, our method improves $0.12\%$ averagely. A similar trend is also observed in the I2G, which improves $0.84\%$ on average. For Face X-ray, we improve the performance by $3.08\%$ on average. It is noteworthy that I2G and Face X-ray have not released their codes yet. We rigorously follow the instructions as in their papers and run the codes widely used by others[4][5].

**Different Network Architectures.** This part validates the effectiveness of our method on different networks, including ResNet-50 [35], EfficientNet-b1 [37], VGG16 [44], Xception [45], ViT [46], F3-Net [29], and GFFD[41]. We compare our method with SBI on these networks, which are tested on CDF, FF++, DFDCP, and FFIW datasets. The results are shown in Table 5. It can be observed that our method improves the performance by $1.78\%$, $0.41\%$, $1.51\%$, $0.2\%$, $2.75\%$, and $4.68\%$ averagely on ResNet-50, EfficientNet-b1, Xception, Vit networks, F3-Net, and GFFD networks respectively. It is noteworthy that our method slightly reduces the performance of VGG16 by $0.01\%$. It is possibly because the capacity of VGG16 is limited than other networks, and learning spatial pseudo-fake faces almost fills up this capacity, leaving no room for the learning of frequency knowledge.

---

[4]I2G: https://github.com/jtchen0528/PCL-I2G
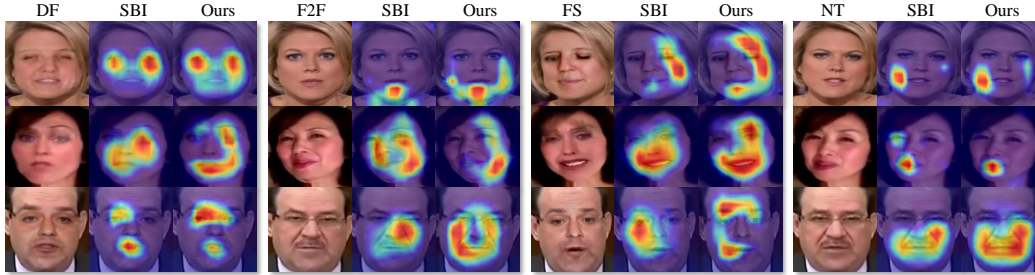[5]Face X-ray: https://github.com/AlgoHunt/Face-Xray

Figure 6: Grad-CAM visualization of SBI and our method on four manipulation types of FF++ dataset. Compared to SBI, our method focuses more on the manipulated structural boundaries.

Table 6: Effect of using wild fake or SP-fake faces.

| SP-fake | CDF | DFDC | DFDCP | FFIW | Avg |
|---------|-------|-------|-------|-------|-------|
| ✗ | 75.79 | 66.30 | 66.30 | 67.70 | 72.95 |
| ✓ | 94.59 | 74.59 | 74.59 | 86.14 | 85.72 |

**Saliency Visualization.** We employ Grad-CAM [47] to visualize the attention of our method compared to SBI on four manipulations in the FF++ dataset. Compared to SBI, our method concentrates more on the structural information, such as the manipulation boundaries. For example, our method highlights the face outline in DF, F2F, and FS, while focusing on the mouth contour in NT.

**Effect of Using Wild Fake or Spatial Pseudo-fake Faces.** As described in Sec. 4.3, our method is performed using real and spatial-blending pseudo-fake (SP-fake) faces. The rationale is that SP-fake faces are greatly diversified, containing more spatial forgery traces. Applying our method to these faces can consider both frequency and spatial traces effectively. To verify this, we directly perform our method on real and wild fake faces. The results in Table 6 indicate a notable performance drop when only wild fakes are used.

**Limitations.** Our method is designed to address the drawbacks of existing spatial-blending methods. Hence, it inherits the assumption that the faces are forged by face-swapping techniques. Further research is needed to validate our performance on other types of forgery operations, such as whole-face synthesis and attribute editing.

## 6 Conclusion

This paper describes a new method called *FreqBlender* that can generate pseudo-fake faces by blending frequency knowledge. To achieve this, we propose a Frequency Parsing Network that adaptively extracts the frequency component corresponding to structural information. Then we can blend this information from fake faces into real faces to create pseudo-fake faces. The extensive Experiments demonstrate the effectiveness of our method and can serve as a complementary module for existing spatial-blending methods.

## References

[1] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5039–5049, 2021.

[2] Yinan He, Bei Gan, Siyu Chen, Yichun Zhou, Guojun Yin, Luchuan Song, Lu Sheng, Jing Shao, and Ziwei Liu. Forgerynet: A versatile benchmark for comprehensive forgery analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 4360–4369, 2021.

[3] Kunlin Liu, Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Wenbo Zhou, and Weiming Zhang. Deepfacelab: Integrated, flexible and extensible face-swapping framework. *Pattern Recognition(PR)*, 141:109628, 2023.

[4] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[5] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 1–11, 2019.

[6] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 3207–3216, 2020.

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

[8] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020.

[9] Tianfei Zhou, Wenguan Wang, Zhiyuan Liang, and Jianbing Shen. Face forensics in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 5778–5788, 2021.

[10] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[11] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 6458–6467, 2021.

[12] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22412–22423, 2023.

[13] Yuan Wang, Kun Yu, Chen Chen, Xiyuan Hu, and Silong Peng. Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7278–7287, 2023.

[14] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 5001–5010, 2020.

[15] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 15023–15033, 2021.

[16] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18720–18729, 2022.

[17] Yipin Zhou and Ser-Nam Lim. Joint audio-visual deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14800–14809, 2021.

[18] Weiming Bai, Yufan Liu, Zhipeng Zhang, Bing Li, and Weiming Hu. Aunet: Learning relations between action units for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24709–24719, 2023.

[19] Changtao Miao, Zichang Tan, Qi Chu, Nenghai Yu, and Guodong Guo. Hierarchical frequency-assisted interactive networks for face manipulation detection. *IEEE Transactions on Information Forensics and Security (TIFS)*, 17:3008–3021, 2022.

[20] Qiqi Gu, Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, and Ran Yi. Exploiting fine-grained face forgery clues via progressive enhancement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 735–743, 2022.

[21] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4113–4122, 2022.

[22] Yuting Xu, Jian Liang, Gengyun Jia, Ziming Yang, Yanhao Zhang, and Ran He. Tall: Thumbnail layout for deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22658–22668, 2023.

[23] Zhiyuan Yan, Yuhao Luo, Siwei Lyu, Qingshan Liu, and Baoyuan Wu. Transcending forgery specificity with latent space augmentation for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8984–8994, 2024.

[24] Jongwook Choi, Taehoon Kim, Yonghyun Jeong, Seungryul Baek, and Jongwon Choi. Exploiting style latent flows for generalizing deepfake video detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1133–1143, 2024.

[25] Han Chen, Yuezun Li, Dongdong Lin, Bin Li, and Junqiang Wu. Watching the big artifacts: Exposing deepfake videos via bi-granularity artifacts. *Pattern Recognition (PR)*, 135:109179, 2023.

[26] Shuai Jia, Chao Ma, Taiping Yao, Bangjie Yin, Shouhong Ding, and Xiaokang Yang. Exploring frequency adversarial attacks for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4103–4112, 2022.

[27] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 7890–7899, 2020.

[28] Davis King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research (JMLR)*, Dec 2009.

[29] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision (ECCV)*, pages 86–103. Springer, 2020.

[30] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. Unmasking deepfakes with simple features. *arXiv preprint arXiv:1911.00686*, 2019.

[31] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1874–1883, 2016.

[32] Andrew G Howard. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[33] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Cotsia, and Stefanos P Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, page 1–1, Jan 2021.

[34] Jiahao Liang, Huafeng Shi, and Weihong Deng. Exploring disentangled content information for face forgery detection. In *European Conference on Computer Vision (ECCV)*, pages 128–145. Springer, 2022.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[37] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

[38] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 667–684. Springer, 2020.

[39] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 15044–15054, 2021.

[40] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. Local relation learning for face forgery detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1081–1088, 2021.

[41] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 16317–16326, 2021.

[42] Ke Sun, Taiping Yao, Shen Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. Dual contrastive learning for general face forgery detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2316–2324, 2022.

[43] Li Lin, Xinan He, Yan Ju, Xin Wang, Feng Ding, and Shu Hu. Preserving fairness generalization in deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16815–16825, 2024.

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

[45] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1251–1258, 2017.

[46] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[47] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision (IJCV)*, page 336–359, Feb 2020.

[48] Ziqi Huang, Kelvin C.K. Chan, Yuming Jiang, and Ziwei Liu. Collaborative diffusion for multi-modal face generation and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[49] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 4401–4410, 2019.

[50] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 8110–8119, 2020.

[51] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Machine Learning (ICML)*, 2015.

[52] Jiakai Wang. Adversarial examples in physical world. In *IJCAI*, pages 4925–4926, 2021.

[53] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[54] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE symposium on security and privacy (SP)*, pages 39–57. IEEE, 2017.

# A Appendix / supplemental material

**Various Loss Weights** $\lambda_1, \lambda_2, \lambda_3, \lambda_4$. These weights are empirically selected based on experimental results. As shown in Table 7, we evaluate the performance across various set of $\lambda_1, \lambda_2, \lambda_3, \lambda_4$. The results exhibit that our method is not particularly sensitive to the settings of loss weights, with performance variations within approximately $\sim 0.5\%$. For our main experiments, we select the set of $\lambda_1 = 1/12, \lambda_2 = 1, \lambda_3 = 0.001, \lambda_4 = 1/4$ as it yields the best performance.

Table 7: Effect of 3ur method on different loss proportions.

| $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | CDF | FF++ | FFIW | DFDCP | Avg |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 93.16 | 96.30 | 87.82 | 85.32 | 90.65 |
| 0.1 | 1 | 0.1 | 0.5 | 93.59 | 95.60 | 84.78 | 86.60 | 90.14 |
| 0.1 | 1 | 0.01 | 0.5 | 94.27 | 96.11 | 85.54 | 87.81 | 90.93 |
| 0.01 | 1 | 0.0001 | 0.1 | 93.60 | 96.00 | 85.85 | 87.38 | 90.71 |
| 1/12 | 1 | 0.001 | 1/4 | 94.59 | 96.13 | 86.14 | 87.56 | 91.11 |

**More Details of Complementary to Spatial-blending Methods.** Table 8 shows the detailed results of every manipulation in the FF++ dataset. It can be seen that our method improves the performance of all manipulation methods, averaging $1.32\%$ for DSP-FWA, $4.28\%$ for I2G, and $1.77\%$ for Face X-ray. This improvement further demonstrates the effectiveness of our method.

Table 8: Effect of our method complementary to spatial-blending Methods.

| Method | FF++ | | | | |
|---|---|---|---|---|---|
| | DF | F2F | FS | NT | Avg |
| DSP-FWA [10] | 55.48 | 43.79 | 49.26 | 44.05 | 48.14 |
| DSP-FWA [10] + Ours | 56.20 | 45.93 | 53.96 | 41.74 | **49.46** |
| I2G [15] | 47.83 | 82.13 | 60.82 | 47.47 | 59.56 |
| I2G [15] + Ours | 56.90 | 81.26 | 61.66 | 55.52 | **63.84** |
| Face X-ray [14] | 89.38 | 85.02 | 83.45 | 71.17 | 82.26 |
| Face X-ray [14] + Ours | 93.21 | 85.41 | 82.70 | 74.79 | **84.03** |

**Probability** $\alpha$ **in FreqBlender.** As shown in Table 9, we evaluate the effect of using various probability $\alpha$. Note that $\alpha$ denotes the probability of generating a synthetic face whether using spatial-blending or using FreqBlender after spatial-blending. $\alpha = 1$ denotes only using FreqBlender, while $\alpha = 0$ means only using spatial-blending. It can be observed that solely using our method can not perform well, as no color space knowledge is involved, hindering the overall effectiveness of pseudo-fake faces. In contrast, solely using spatial-blending operations can reach a favorable performance. However, when inserting pseudo-fake faces generated by FreqBlender, the generalization performance is further enhanced, as our method can compensate for the loss of frequency knowledge. The optimal effect on CDF evaluation is achieved when $\alpha$ is set to 0.2. This experiment demonstrates the complementary effect of our method to existing spatial-blending methods. More generalization experiments on more datasets have been conducted in the main paper, demonstrating the effectiveness of our method.

Table 9: Effect of our method on different data proportions.

| $\alpha$ | FF++ | | | | | CDF |
|---|---|---|---|---|---|---|
| | DF | F2F | FS | NT | Avg | |
| 1 | 63.18 | 59.85 | 65.79 | 54.61 | 60.86 | 56.34 |
| 0.8 | 98.58 | 94.80 | 97.69 | 89.83 | 95.23 | 90.27 |
| 0.5 | 99.03 | 96.99 | 97.81 | 91.68 | 96.38 | 91.83 |
| 0.3 | 99.12 | 97.15 | 97.93 | 90.91 | 96.28 | 93.76 |
| 0.2 | 99.18 | 96.76 | 97.68 | 90.88 | 96.13 | **94.59** |
| 0.1 | 99.11 | 97.11 | 97.93 | 90.91 | 96.26 | 93.76 |
| 0 | 99.17 | 97.63 | 97.77 | 91.35 | 96.50 | 93.58 |

**Visual Demonstration of FreqBlender.** The goal of our method is to create pseudo-fake faces that resemble the frequency distribution of wild fake faces. To verify this, we conduct a visual experiment on the FF++ dataset. Specifically, we randomly select $3,000$ images from each manipulation method (Deepfakes, FaceSwap, Face2Face, NeuralTextures) and calculate the average frequency map for each manipulation method respectively. Then we create the same number of pseudo-fake faces using
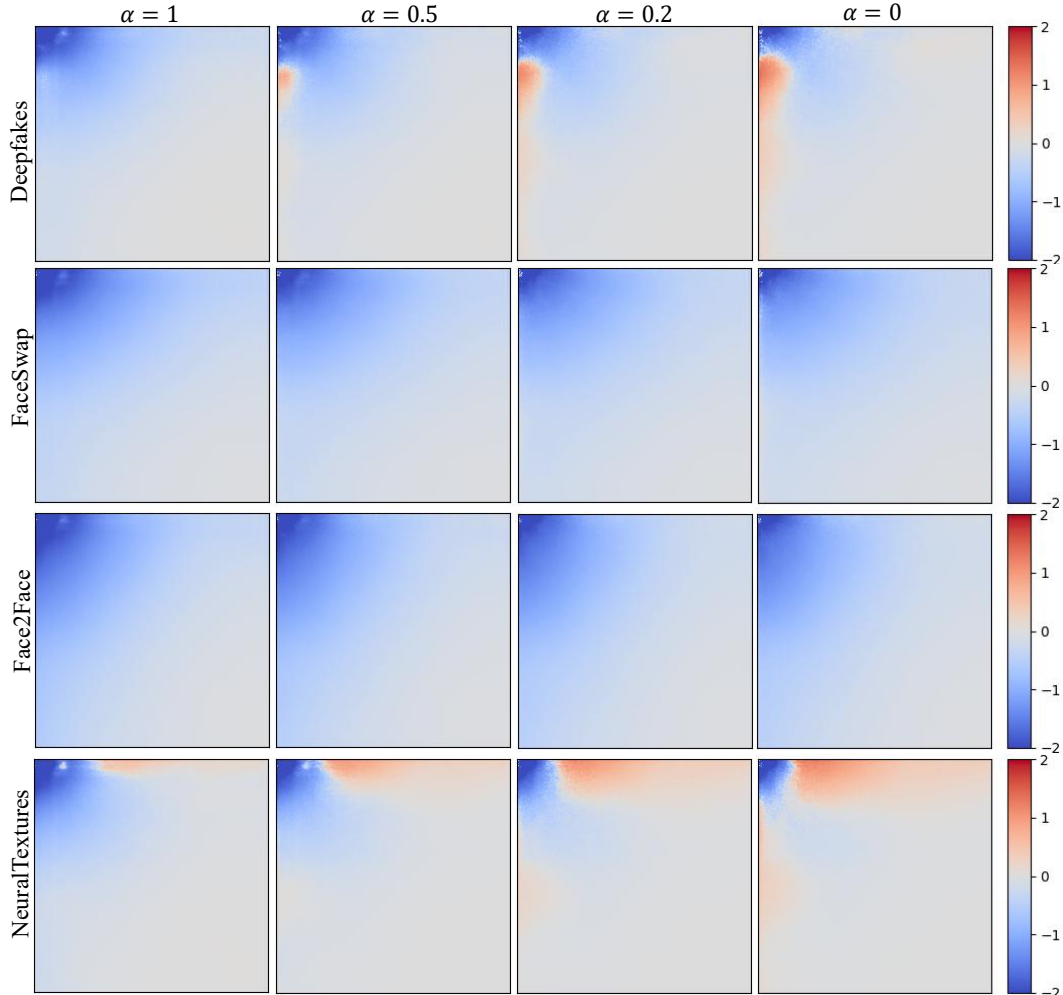
Figure 7: Heatmap visualization of the frequency difference between our method and the wild fake faces (Deepfakes, FaceSwap, Face2Face, NeuralTextures). Note that $\alpha = 0$ denotes that our method is degraded to SBI [16].

our method and calculate their average frequency map. Finally, we visualize the frequency difference between our method and wild fake faces. The results are illustrated in Fig. 7. It is important to note that $\alpha$ represents the probability of applying FreqBlender. Hence, $\alpha = 1$ means all pseudo-fake faces are created using our method, while $\alpha = 0$ denotes our method is degraded to SBI. From this figure, we can see that the difference is minimal when $\alpha = 1$. As $\alpha$ decreases, the difference becomes larger as the pseudo-fake faces are more likely created by SBI. This demonstrates that our method effectively simulates the frequency distribution of wild fake faces.

**Performance on FF++ Low-quality (LQ).** To investigate the performance of our method on low-quality videos, we conduct experiments on the FF++ Low-quality (LQ) set. The results in Table 10 show that while all methods experience substantial performance drops on the LQ set, our method consistently achieves the highest performance, demonstrating better generalization capability on low-quality videos compared to other methods.

**Tentative Validation on Various Synthesized Faces.** In addition to validation on the standard datasets, we investigate a new scenario: Diffusion-based face-swap deepfake detection. In this scenario, we employ a recent diffusion model (Collaborative Diffusion [48]) to synthesize faces, which are then blended into original videos. We create 200 fake faces and evaluate our method in Table 11. It can be seen that our method is effective in detecting such forged faces.

16

Table 10: Performance of our method on FF++ (LQ).

| Method | FF++(LQ) |
|---|---|
| I2G [15] | 52.20 |
| Face X-ray [14] | 65.41 |
| SBI [16] | 76.11 |
| FreqBlender | 77.56 |

Table 11: Performance of Diffusion-based face-swap and Gan-Generated images.

| Method | Diffusion-based [48] | StyleGAN [49] | StyleGAN2 [50] |
|---|---|---|---|
| I2G [15] | 63.51 | 47.89 | 43.86 |
| Face X-ray [14] | 89.81 | 59.11 | 66.54 |
| SBI [16] | 91.70 | 63.99 | 72.88 |
| FreqBlender | 94.74 | 64.39 | 76.70 |

Moreover, although our method is designed for face-swapping techniques, we also test it on face images generated by StyleGAN [49] and StyleGAN2 [50]. The results, also shown in Table 11, reveal a notable performance drop across all methods. However, our method still outperforms the others.

**Performance against Evasion Attacks.** We employ a widely recognized library `TorchAttacks` with four well-known attack methods: FGSM [51], BIM [52], PGD [53] and CW [54]. The attack experiment is conducted on the CDF dataset with six models and the attack configuration is set by default. The results are shown in Table 12. It can be seen without any defense strategies, all models can be easily attacked in the white-box attacking mode, which exactly aligns with our expectations and the discoveries in the papers of FGSM, BIM, etc.

Table 12: AUC (%) performance against evasion attacks.

| Method | Original | Attacked | | | |
|---|---|---|---|---|---|
| | | FGSM | BIM | PGD | CW |
| FWA [10] | 62.91 | 27.43 | 0 | 0 | 0 |
| I2G [15] | 53.35 | 3.26 | 0 | 0 | 0 |
| Face-Xray [14] | 67.99 | 54.66 | 0 | 0 | 0 |
| SBI-c23 [16] | 92.94 | 27.19 | 0 | 0 | 0 |
| SBI-raw [16] | 93.18 | 11.48 | 0 | 0 | 0 |
| Ours | 94.59 | 22.31 | 0 | 0 | 0 |

# NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: After the preliminary analysis section of the paper combined with data analysis, along with our experimental results, we have completed our work comprehensively.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We introduced relevant limitations in the fifth part of our paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We present our assumptions in the Preliminary analysis section of this paper, and our experimental results confirm the validity of our assumptions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: You can perfectly reproduce our results according to our paper, and we will also open-source our code in the future.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code will be thoroughly organized and released after the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In the Experimental Setups section of our paper, we have provided detailed descriptions of our experimental details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: Our paper's experiments do not include error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: In the Experimental Setups section of our paper, we have provided detailed descriptions of our compute resources.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We have reviewed the code of ethics and are confident that we have adhered to this guideline.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Although our paper does not discuss the related impacts, our work enhances the capabilities of deepfake detection, which we believe has positive societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our model is designed to enhance the generalization capabilities of deepfake detection, and there is no risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly attributed the work of others in our paper and have followed licensing and usage terms.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not introduce new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not have this research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not have this research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.