# Exploiting Fine-grained Face Forgery Clues via Progressive Enhancement Learning

**Qiqi Gu**[1 2*] **Shen Chen**[2*] **Taiping Yao**[2*] **Yang Chen**[2] **Shouhong Ding**[2†] **Ran Yi**[1 3†]

[1]Shanghai Jiao Tong University, [2]Youtu Lab, Tencent, [3]MoE Key Lab of Artificial Intelligence, SJTU
miemie@sjtu.edu.cn, {kobeschen, taipingyao, wizyangchen, ericshding}@tencent.com, ranyi@sjtu.edu.cn

## Abstract

With the rapid development of facial forgery techniques, forgery detection has attracted more and more attention due to security concerns. Existing approaches attempt to use frequency information to mine subtle artifacts under high-quality forged faces. However, the exploitation of frequency information is coarse-grained, and more importantly, their vanilla learning process struggles to extract fine-grained forgery traces. To address this issue, we propose a progressive enhancement learning framework to exploit both the RGB and fine-grained frequency clues. Specifically, we perform a fine-grained decomposition of RGB images to completely decouple the real and fake traces in the frequency space. Subsequently, we propose a progressive enhancement learning framework based on a two-branch network, combined with self-enhancement and mutual-enhancement modules. The self-enhancement module captures the traces in different input spaces based on spatial noise enhancement and channel attention. The Mutual-enhancement module concurrently enhances RGB and frequency features by communicating in the shared spatial dimension. The progressive enhancement process facilitates the learning of discriminative features with fine-grained face forgery clues. Extensive experiments on several datasets show that our method outperforms the state-of-the-art face forgery detection methods.

## 1 Introduction

Over the past few years, face forgery technology has made significant progress. With public accessible tools such as deepfakes (Tora 2018) or face2face (Thies et al. 2016), people can easily manipulate the expression, attribution or identity of faces in images. Various powerful algorithms are utilized to generate realistic forged faces, which are hardly discerned by human eyes. These realistic forged faces have been used in pornography or political rumors, which are harmful to the community and evoke social panic. Under this background, the face forgery detection task was born and has attracted more and more attention recently.

Many significant works (Chollet 2017; Li et al. 2020a; Dang et al. 2020; Chen et al. 2021; Qian et al. 2020; Zhao et al. 2021; Gu et al. 2021) have made great contributions to this task. Prior works use hand-crafted features (*e.g.*, eye blinking (Li, Chang, and Lyu 2018), head inconsistency (Yang, Li, and Lyu 2019)) or semantic features extracted by universal CNN (Afchar et al. 2018; Chollet 2017) to conduct binary classification. Recently, two typical manners for mining subtle task-specific artifacts have been explored. On the one hand, some works (Dang et al. 2020; Li et al. 2020a; Chen et al. 2021; Zhao et al. 2020; Wang et al. 2020) utilize auxiliary supervision such as blending boundary (Li et al. 2020a) or forged mask (Dang et al. 2020; Chen et al. 2021). However, the mask ground-truth they rely on is difficult to access in most cases. On the other hand, some attempt to employ the frequency-aware features to assist classification, using DCT (Qian et al. 2020), local frequency (Chen et al. 2021) or SRM (Luo et al. 2021). Although these works have made considerable performance, the frequency features extracted by their strategies are relatively coarse, which burdens the network to assemble discriminative feature patterns. Moreover, all these frequency-aware approaches either directly concatenate RGB and frequency features at the end of the network or fuse them only at a shallow layer. This makes it difficult for them to fully utilize the decomposed frequency information and limits the possibility of discovering fine-grained face forgery cues.

To address the above issues, we propose a novel Progressive Enhancement Learning (PEL) framework aiming at exploiting face forgery clues. Specifically, we employ the patch-wise Discrete Fourier Transform (Xu et al. 2020) with the sliding window to decompose the RGB input into fine-grained frequency components. Frequency domain information is assisted and complementary to the RGB features. And different from conventional frequency components, fine-grained frequency components can produce more combinations of features to discover potential artifacts.

To fully exploit RGB image and fine-grained frequency components, we design a two-stream network structure with two novel enhancement modules that progressively enhance the forgery clues. One is the self-enhancement module. It captures the traces in different input spaces separately based on spatial noise enhancement and channel attention. The other is the mutual-enhancement module, which achieves concurrent enhancement of both the RGB and frequency branches through feature communication in the shared spatial dimension. We insert these two enhancement modules

*Equal contribution.
†Corresponding authors.

after each convolutional block of the network. The earlier enhancements facilitate feature extractions in the later layers, and the forgery cues hidden in the fake faces are sufficiently excavated and magnified.

Extensive experiments are conducted to demonstrate the effectiveness of our method. With the progressive representation enhancement, better results are obtained on high-quality forged images. Our method exceeds the other comparison methods and achieves state-of-the-art performance on the common benchmark FaceForensics++ dataset and the newly published WildDeepfake dataset. The cross-dataset evaluations on three additional challenging datasets prove the generalization ability, while the perturbed evaluations prove the robustness of our method.

Our contributions can be summarized as follows:

- We propose a progressive enhancement learning network aiming at exploiting the combined fine-grained frequency components and RGB image.
- We first perform fine-grained decomposition in the frequency domain, and then propose two novel enhancement modules, *i.e.*, self-enhancement and mutual-enhancement modules, which progressively enhance the feature to capture the subtle forgery traces.
- Extensive experiments and visualizations reveal the significance of feature enhancement in face forgery detection, and demonstrate the effectiveness of our proposed method against the state-of-the-art competitors.

## 2 Related work

Due to the rapid development of face forgery methods, face forgery detection has recently attracted more attention in computer vision communities. Great efforts have been made to explore the authenticity of human faces.

Early works use hand-crafted features to seek for the artifacts in forged faces. Fridrich *et al.* (Fridrich and Kodovsky 2012) extract steganalysis features and train a linear Support Vector Machine (SVM) classifier to distinguish authenticity. After that, Cozzolino *et al.* (Cozzolino, Poggi, and Verdoliva 2017) cast the features mentioned above to a convolution neural network. Matern *et al.* (Matern, Riess, and Stamminger 2019) analyze the facial regions which are likely to generate artifacts and design specific descriptors to capture them. Yang *et al.* (Yang, Li, and Lyu 2019) estimate the inconsistency of head poses between adjacent faces. With the rapid progress of deep learning, some CNN-based works (Bayar and Stamm 2016; Afchar et al. 2018; Chollet 2017) emerge to extract high-level semantic information for classification. However, their vanilla structures are unsuitable for the fast-growing and realistic forgery faces.

To further mine the heuristic forgery cues, two typical manners are widely used. One is resorting to auxiliary supervisions. For example, Nguyen *et al.* (Nguyen et al. 2019) incorporate the classification and segmentation task into one framework simultaneously. Dang *et al.* (Dang et al. 2020) further highlight the manipulated area via attention mechanism. Face X-ray (Li et al. 2020a) focuses on the blending boundary induced by the image fusion process. Besides, some works (Chen et al. 2021; Zhao et al. 2020) resort to the similarity between local regions and achieve superior performance on benchmark datasets. However, the manipulated mask above methods relied on training is not applicable to some wild datasets (e.g., Wild Deepfake dataset (Zi et al. 2020)), limiting practical application.

The others notice the artifacts hidden in the frequency domain that are difficult to observe directly. Singhal *et al.* (Singhal et al. 2020) extracts frequency features through the residuals of image median filtering for detecting manipulated faces. Frank *et al.* (Frank et al. 2020) reveal the peculiar frequency spectrum pattern of GAN-generated deepfake images. $F^3$-Net (Qian et al. 2020) employ the frequency information in two ways: image components which are transformed from a certain frequency band, and local frequency statistics calculated by frequency spectrum of patches. Chen *et al.* (Chen et al. 2021) use high-frequency components of the image to assist in calculating patch-similarities. Luo *et al.* (Luo et al. 2021) use SRM filter to guide RGB features. FDFL (Li et al. 2021) utilize frequency information by DCT and adaptive frequency information mining block. Although these methods have achieved remarkable performances, the exploitation of frequency information is coarse-grained. More importantly, their vanilla learning process is insufficient to make full use of the subtle artifacts in the frequency domain. Different from existing works, we propose a novel progressive enhancement learning network that exploits the fine-grained frequency components.

## 3 Method

Fig. 1 illustrates the proposed PEL framework. We transform the RGB input into the fine-grained frequency component and fed them together into a two-stream network with EfficientNet as the backbone. Each convolutional block of the backbone is followed by a self-enhancement module and a mutual-enhancement module to progressively enhance the fine-grained clues in an intra- and inter-stream manner.

### 3.1 Fine-grained Clues

As shown in the leftmost of Fig. 1, we apply domain transformation that decomposes the input RGB image into frequency components, revealing fine-grained forgery cues hidden in the frequency space.

Without loss of generality, let $x^{rgb} \in \mathbb{R}^{3\times H\times W}$ denotes the RGB input, where $H$ and $W$ are the height and width of image. First, the RGB input $x^{rgb}$ is transformed to YCbCr space (denoted by $x^{ycbcr} \in \mathbb{R}^{3\times H\times W}$) that coincides with the widely-used JPEG compression in the forged video. Then, we slice $x^{ycbcr}$ via a sliding window to obtain a set of $8 \times 8$ patches. $p^c_{(i,j)} \in \mathbb{R}^{8\times 8}$ denotes the patch sliced by the ($i$ th,$j$ th) sliding window of a certain color channel. Each patch $p^c_{(i,j)}$ is processed by DCT into $8 \times 8$ frequency spectrum $d^c_{(i,j)} \in \mathbb{R}^{8\times 8}$, where each value corresponds to the intensity of a certain frequency band. We flatten the frequency spectrum and group all components of the same frequency into one channel to form a new input, following the patch location in the original input:

$$x^{freq}[:][i][j] = flatten(d_{(i,j)}), \tag{1}$$

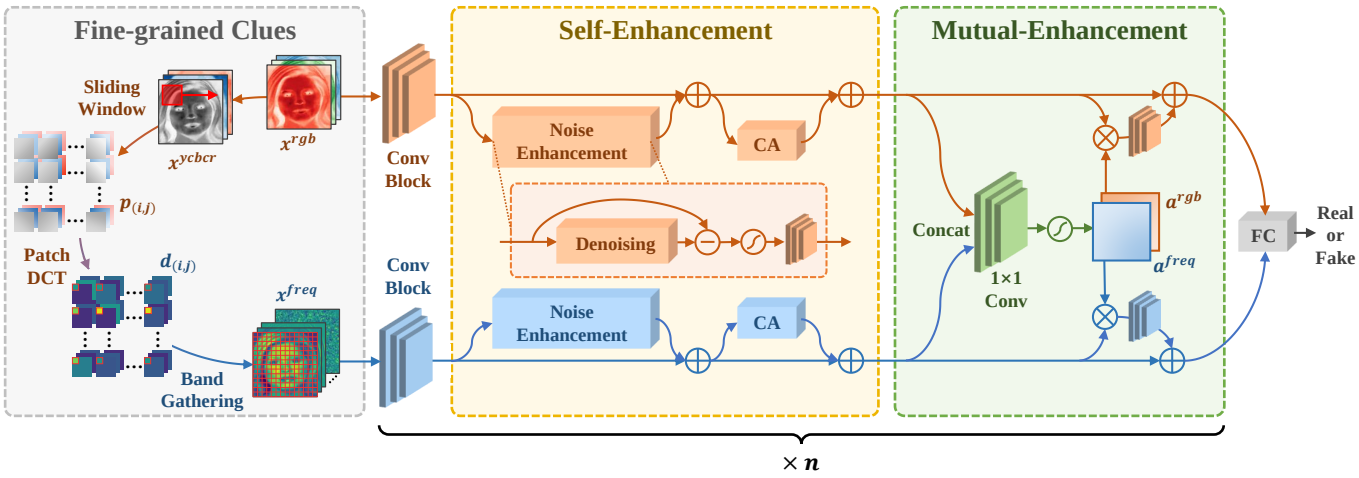


Figure 1: The proposed PEL framework. The RGB input is decomposed into frequency components to mine fine-grained clues hidden in frequency space. In the self-enhancement module, the features of each stream are enhanced separately using a noise enhancement block and a channel attention block. In the mutual enhancement, both features are enhanced in the shared spatial dimension. The two enhancement modules are selectively inserted after each convolutional block of the network.

where $x^{freq} \in \mathbb{R}^{192\times H_2\times W_2}$ is the newly formed input, $H_2$ and $W_2$ are the vertical and horizontal step numbers of sliding window, and $d_{(i,j)} \in \mathbb{R}^{3\times 8\times 8}$ denotes the concatenation of all the $d^c_{(i,j)}$. In this way, the original color input is decomposed and recombined to fine-grained frequency-aware data while maintaining the spatial relationship to match the shift-invariance of CNN. Finally, a $1\times 1$ convolution layer is applied to the frequency data to adaptively extract the most informative frequency bands, forming the fine-grained frequency input $x^{freq} \in \mathbb{R}^{64\times H\times W}$.

### 3.2 Self-Enhancement Module

The self-enhancement module consists of two mountable sub-modules: a noise enhancement block and a channel attention block, which enhance the features of each stream separately. The visual artifacts in newly forged faces have significantly been eliminated (Li et al. 2019b), pushing subtle noise to a more critical role in the face forgery detection task. To this end, we design a noise enhancement block to excavate the artifact hidden in the feature noise, especially that of the fine-grained frequency input.

Given the RGB input $x^{rgb}$ and the frequency input $x^{freq}$, let $f^{rgb}_{in} \in \mathbb{R}^{c\times h\times w}$ and $f^{freq}_{in} \in \mathbb{R}^{c\times h\times w}$ denote the feature maps extracted by intermediate convolutional block of RGB stream and frequency stream, where $c$ is the number of channels, $h$ and $w$ denote the height and width of the feature maps, respectively. Since the RGB and frequency streams have the same network structure except for the first convolutional layer, we use the superscript $s \in \{rgb, freq\}$ to indicate different stream.

For noise enhancement, we first extract the noise of input feature maps $f_{in}$ as follows:

$$f^s_{noi} = f^s_{in} - \mathcal{M}(f^s_{in}), \tag{2}$$

where $\mathcal{M}$ is the median filter with a kernel size of $3\times 3$ that can filter noise such as salt to denoise at the feature level. By subtracting the input features and denoising features spatially, $f_{noi}$ manifests the noise information of the input feature.

Then we apply a *Sigmoid* function (denoted by $\sigma$) to amplify the subtle noises and suppress the exaggerated value of noise to prevent polarization. After that, a depth-wise $1\times 1$ convolution layer is applied to the feature noise to adjust the amplitude of noise enhancement. Finally, the feature noise is added to the input feature $f_{in}$ to obtain the noise-enhanced feature $f_{ne}$, which is used as input of the subsequent block. The above process can be summarized as follows:

$$f^s_{ne} = f^s_{in} + Conv(\sigma(f^s_{noi})). \tag{3}$$

Since the noise enhancement block considers the low-level noise of each channel separately, we further utilize a channel attention block to facilitate inter-channel interactions within each stream. Specifically, the channel attention block enhances features along the depth dimension, which combines the information from all channels and determines the significance of each channel, and enhances them accordingly. The output feature $f_{out}$ is obtained as:

$$f^s_{out} = f^s_{ne} + f^s_{ne} \otimes \sigma(MLP(GAP(f^s_{ne}) + GMP(f^s_{ne}))), \tag{4}$$

where $MLP$ is a multi-layer perceptron, $GAP$ denotes the global average pooling, and $GMP$ is the global max pooling.

In each stream of the network, the noise enhancement module is inserted after the first two convolutional blocks. This is because high-level features capture semantic information within a large reception field and are not suitable to extract the noise with local concepts. And the channel attention block is empirically mounted after the second to the penult convolutional blocks. With the self-enhancement module, features are enhanced within the stream without changing their size, providing more helpful features for the following mutual-enhancement module.

### 3.3 Mutual-Enhancement Module

As the features of two streams are originated from different input spaces, they provide distinct cues that can complement each other. Thus, we design a mutual-enhancement module to comprehensively integrate the dual-stream knowledge along spatial dimension to enhance the features.

Concretely, the dual feature maps are concatenated and go through a point-wise convolution and a *Sigmoid* function to get two spatial attention maps for two streams, respectively:

$$A = \sigma(Conv(Cat(f_{in}^{rgb}, f_{in}^{freq}))), \tag{5}$$

where $A \in \mathbb{R}^{2\times h\times w}$ is the spatial attention maps, $Conv$ denote the point-wise convolution layer, and $Cat$ concatenate the features along with the depth. The input feature of each stream is multiplied with its attention map (*i.e.* $a^{rgb}$ and $a^{freq}$ which are the two channels of $A$), and then go through a depth-wise $1 \times 1$ convolution layer to adjust the enhancing intensity. Finally, we add the enhanced feature with the input feature as follows (the feature size remains the same):

$$f_{out}^{s} = f_{in}^{s} + Conv(f_{in}^{s} \otimes a^{s}). \tag{6}$$

The mutual-enhancement module is inserted after the second to the penult convolutional blocks and placed behind the self-enhancement module. In this way, we can highlight forged regions that benefit from the other stream while maintaining stream-specific fine-grained forgery clues, which result in more discriminative features.

### 3.4 Loss Function

In the preceding blocks, the two features enhanced by the mutual-enhancement module are fed into the next convolution blocks. In the last block, the outputs of two streams are concatenated for classification. We adopt the widely-used Binary Cross-Entropy loss as the objective function:

$$\mathcal{L} = -\frac{1}{n}\sum(y_{gt}\times log(y_{pred}) + (1-y_{gt})\times(log(1-y_{pred})), \tag{7}$$

where $y_{gt}$ is the ground truth label, and $y_{pred}$ denotes the prediction of the network. The parameters of the network are updated via the back-propagation algorithm.

## 4 Experiment

In this section, we first introduce the overall experimental setup, then present extensive experimental results to demonstrate the effectiveness and robustness of our approach, and finally visualize the enhanced fine-grained face forgery clues and class activation map.

### 4.1 Experimental Setup

**Datasets.** We adopt five widely-used public datasets in our experiments, *i.e.*, FaceForensics++ (Rossler et al. 2019), WildDeepfake (Zi et al. 2020), Celeb-DF (Li et al. 2020b), DeepfakeDetection (Dufour and Gully 2019), DeepfakeDetectionChallenge (Dolhansky et al. 2020), in which the former two are used for both training and evaluation, while the latter three for cross-dataset evaluation only.

- **FaceForensics++** (FF++) is a large-scale benchmark dataset containing 1000 original videos from youtube and corresponding fake videos which are generated by four typical manipulation methods: *i.e.* Deepfakes (DF) (Tora 2018), Face2Face (F2F) (Thies et al. 2016), FaceSwap (FS) (Kowalski 2018) and NeuralTextures (NT) (Thies, Zollhöfer, and Nießner 2019). Different levels of compressions are used on raw videos, and two counterparts are generated, *i.e.* high quality (HQ) and low quality (LQ). We follow the official splits by using 720 videos for training, 140 videos for validation, and 140 videos for testing.
- **WildDeepfake** is a dataset with 7,314 face sequences entirely collected from the internet, in which 6,508 sequences are for training and 806 are for testing. The contents in this dataset are diverse in many aspects, *e.g.* activities, scenes, and manipulation methods, which makes this dataset more challenging and closer to the real-world face manipulation scenario.
- **Celeb-DF** is composed of 590 real videos based on 59 celebrities from youtube, and the corresponding 5639 fake videos tampered with the improved DeepFake algorithm (Li et al. 2020b).
- **DeepfakeDetection** (DFD) is a large dataset that contains over 3000 manipulated videos in various scenes using publicly available deepfake generation methods.
- **DeepfakeDetectionChallenge** (DFDC) is a prominent face swap video dataset, with over 100,000 total clips sourced from 3,426 paid actors, produced with several Deepfake, GAN-based, and non-learned methods.

**Implementation detail.** We implement the proposed framework via open-source PyTorch (Paszke et al. 2017). We sample 50 frames per video at equal interval, then use the state-of-the-art face extractor DSFD (Li et al. 2019a) to detect faces and resize them to $320 \times 320$ (training of FF++) or $224 \times 224$ (training on WildDeepfake). The EffcientNet-B4 (Tan and Le 2019) pre-trained on ImageNet was adopted as the backbone of our network, which is trained with Adam optimizer with the learning rate of $2 \times 10^{-4}$, the weight decay of $1 \times 10^{-5}$, and the batch size of 32. The stride of the sliding window is set to 2 in all experiments. The results of comparisons are obtained from their paper, and we specify our implementation by $^{\dagger}$ otherwise.

**Evaluation Metrics.** Following the convention (Rossler et al. 2019; Qian et al. 2020; Dang et al. 2020), we apply Accuracy score (Acc), Area Under the Receiver Operating Characteristic Curve (AUC) and Equal Error Rate (EER) as our evaluation metrics.

**Comparing Methods.** We compare our method with several advanced methods: C-Conv(Bayar and Stamm 2016), CP-CNN(Rahmouni et al. 2017), MesoNet (Afchar et al. 2018), Xception (Chollet 2017), Face X-ray (Li et al. 2020a), Two-branch RN (Masi et al. 2020), Add-Net (Zi et al. 2020), $F^3$-Net (Qian et al. 2020), FDFL (Li et al. 2021) and Multi-Att (Zhao et al. 2021).

| Methods | FF++ (HQ) | | FF++ (LQ) | | WildDeepfake | |
|---|---|---|---|---|---|---|
| | Acc | AUC | Acc | AUC | Acc | AUC |
| C-Conv (Bayar and Stamm 2016) | 82.97% | - | 66.84% | - | - | - |
| CP-CNN (Rahmouni et al. 2017) | 79.08% | - | 61.18% | - | - | - |
| MesoNet (Afchar et al. 2018) | 83.10% | - | 70.47% | - | - | - |
| Xception (Chollet 2017) | 95.73% | - | 86.86% | - | 79.99%† | 0.8886† |
| Face X-ray (Li et al. 2020a) | - | 0.8735 | - | 0.6160 | - | - |
| Two-branch RN (Masi et al. 2020) | 96.43% | 0.9870 | 86.34% | 0.8659 | - | - |
| Add-Net (Zi et al. 2020) | 96.78% | 0.9774 | 87.50% | 0.9101 | 76.31%† | 0.8328† |
| $F^3$-Net (Qian et al. 2020) | 97.52% | 0.9810 | 90.43% | 0.9330 | 80.66%† | 0.8753† |
| FDFL (Li et al. 2021) | 96.69% | 0.9930 | 89.00% | 0.9240 | - | - |
| Multi-Att (Zhao et al. 2021) | 97.60% | 0.9929 | 88.69% | 0.9040 | 81.99%† | 0.9057† |
| Ours | **97.63%** | **0.9932** | **90.52%** | **0.9428** | **84.14%** | **0.9162** |

Table 1: Quantitative results on FaceForensics++ dataset with different quality settings, and WildDeepfake dataset.

| Training dataset | Methods | Testing dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | WildDeepfake | | Celeb-DF | | DFD | | DFDC | |
| | | AUC | EER↓ | AUC | EER↓ | AUC | EER↓ | AUC | EER↓ |
| FF++ (LQ) | Xception† | 0.6059 | 0.6191 | 0.6005 | 0.4319 | 0.6543 | 0.3933 | 0.5565 | 0.4605 |
| | Add-Net† | 0.5421 | 0.4621 | 0.5783 | 0.4444 | 0.5716 | 0.4531 | 0.5160 | 0.5477 |
| | F3-Net† | 0.6049 | 0.4341 | 0.6795 | 0.3676 | 0.6950 | 0.3539 | 0.5787 | 0.4423 |
| | Multi-Att† | 0.6565 | 0.3965 | 0.6864 | 0.3708 | 0.7418 | 0.3272 | 0.6302 | 0.4098 |
| | Ours | **0.6739** | **0.3825** | **0.6918** | **0.3569** | **0.7586** | **0.3084** | **0.6331** | **0.4043** |
| **Training dataset** | **Methods** | **Testing dataset** | | | | | | | |
| | | FF++(LQ) | | Celeb-DF | | DFD | | DFDC | |
| | | AUC | EER↓ | AUC | EER↓ | AUC | EER↓ | AUC | EER↓ |
| WildDeepfake | Xception† | 0.5920 | 0.4306 | 0.7791 | 0.2944 | 0.8114 | 0.2620 | 0.5702 | 0.4510 |
| | Add-Net† | 0.5388 | 0.4720 | 0.6212 | 0.4151 | 0.6877 | 0.3682 | 0.5337 | 0.4732 |
| | F3-Net† | 0.5595 | 0.4549 | 0.6088 | 0.4276 | 0.7827 | 0.2878 | 0.5240 | 0.4842 |
| | Multi-Att† | **0.6276** | **0.4097** | 0.7695 | 0.2811 | 0.8416 | 0.2374 | 0.5665 | 0.4534 |
| | Ours | 0.6160 | 0.4179 | **0.8294** | **0.2424** | **0.8680** | **0.1997** | **0.5894** | **0.4300** |

Table 2: Quantitative results on unseen datasets.

### 4.2 Experimental Results

**Intra-testing.** Following (Rossler et al. 2019; Qian et al. 2020), we compare our method against several advanced techniques on the FF++ dataset with different quality settings (*i.e.*, HQ and LQ), and further evaluate the effectiveness of our approach on the WlidDeepfake dataset that is closer to the real-world scenario. The quantitative results are shown in Tab.1. It can be observed that our proposed method outperforms all the compared methods in both Acc and AUC metrics across all dataset settings. These gains mainly benefit from the proposed PEL framework that can sufficiently excavate and magnify the fine-grained forgery cues hidden in the fake faces.

**Cross-testing.** As new forgery methods are emerging all the time, the generalizability of detection models directly affects their application under real-world scenarios. To test the generalization ability, we perform cross-dataset evaluation, *i.e.*, training on one dataset while testing on another different dataset. This is more challenging since the testing distribution is different from that of training. We compare our method against four competing methods through cross-dataset evaluations, as shown in Tab. 2. Specifically, we first trained these models on FF++ (LQ) and then evaluated their AUC and EER metrics on WildDeepfake, Celeb-DF, DFD and DFDC, respectively. From Tab. 2 we can observe that our method outperforms all the competitors significantly on all testing datasets. We further trained our model on WildDeepfake and evaluated it on the other four datasets, obtaining similar results that our method substantially outperforms the compared methods in nearly all cases.

**Robustness.** In the process of video acquisition and transmission, various noises such as blur or salt will be introduced to the digital data; hence the robustness towards perturbation of the detection model is essential to the real-world application. Thus, it's important for a model to be robust to all kinds of perturbation. A series of experiments are conducted to verify the robustness of our method. Specifically, we apply Gaussian noise, salt and pepper noise, and Gaussian blur to the test data, and measure the decay of Acc and AUC (denote as $\Delta$Acc and $\Delta$AUC) to assess the robustness of the detection model, as shown in Tab. 3. The results turn out that the performance of our method degrades the least in most cases, which mainly benefits from our novel noise

| Datasets | Methods | + GaussianNoise | | + SaltPepperNoise | | + GaussianBlur | |
|---|---|---|---|---|---|---|---|
| | | ΔAcc | ΔAUC | ΔAcc | ΔAUC | ΔAcc | ΔAUC |
| FF++ (LQ) | Xception (Chollet 2017) | -2.65% | -0.0397 | -32.44% | -0.3330 | **-6.22%** | -0.1994 |
| | AddNet† (Zi et al. 2020) | -41.51% | -0.2862 | -11.28% | -0.3445 | -11.28% | -0.3445 |
| | F3Net† (Qian et al. 2020) | -9.86% | -0.0838 | -31.08% | -0.3891 | -11.08% | -0.2077 |
| | Multi-att† (Zhao et al. 2021) | -1.79% | -0.0058 | -49.30% | -0.2494 | -12.23% | -0.2475 |
| | Ours | **-0.10%** | **-0.0031** | **-9.39%** | **-0.2079** | -7.41% | **-0.1274** |
| WildDeepfake | Xception (Chollet 2017) | -0.98% | -0.0082 | -27.80% | -0.1373 | -12.71% | **-0.0664** |
| | AddNet† (Zi et al. 2020) | -11.66% | -0.3327 | -18.21% | -0.3589 | -12.91% | -0.1895 |
| | F3Net† (Qian et al. 2020) | -1.17% | -0.0248 | -43.57% | -0.3407 | -12.43% | -0.1567 |
| | Multi-att† (Zhao et al. 2021) | -0.99% | -0.0139 | -29.47% | -0.1813 | -14.86% | -0.1829 |
| | Ours | **-0.86%** | **-0.0050** | **-4.25%** | **-0.0483** | **-10.88%** | -0.1216 |

Table 3: Robustness evaluation under various types of perturbations.

| RGB | Freq | Self | Mutual | Acc | AUC |
|---|---|---|---|---|---|
| ✓ | | | | 88.59% | 0.9251 |
| | ✓ | | | 88.65% | 0.9250 |
| ✓ | | ✓ | ✓ | 89.15% | 0.9307 |
| | ✓ | ✓ | ✓ | 89.11% | 0.9261 |
| ✓ | ✓ | | | 89.51% | 0.9369 |
| ✓ | ✓ | ✓ | | 89.98% | 0.9390 |
| ✓ | ✓ | | ✓ | 89.98% | 0.9394 |
| ✓ | ✓ | ✓ | ✓ | **90.52%** | **0.9428** |

Table 4: Abalation study on FaceForensics++ (LQ) dataset.

| Denoising methods | Acc | AUC |
|---|---|---|
| Median filter (ks = 3) | **90.52%** | **0.9428** |
| Mean filter (ks = 3) | 89.30% | 0.9346 |
| Non-local means (ks = 3) | 89.48% | 0.9313 |
| Median filter (ks = 5) | 89.25% | 0.9356 |
| Median filter (ks = 7) | 89.32% | 0.9394 |
| Multiple median filter (ks = 3, 5) | 90.12% | 0.9382 |
| Multiple median filter (ks = 3, 5, 7) | 90.02% | 0.9365 |

Table 5: Quantitative results of different denoising methods on FaceForensics++ (LQ) dataset. (ks: kernel size)

enhancement mechanism.

### 4.3 Ablation study

**Components.** As shown in Tab.4, we develop several variants and conduct a series of experiments on the FF++ (LQ) dataset to explore the influence of different components in our proposed method. In the single-stream setting, using only the integral RGB data or the fine-grained frequency data as input leads to similar results, and our proposed two modules can enhance the performance a little (the mutual-enhancement module is replaced with a spatial attention). In the two-stream setting, combining the original two stream can slightly improve the performance, which verifies that the fine-grained frequency input is distinct and complementary to the RGB data. The performance can be improved by adding the proposed self-enhancement module or mutual-enhancement module, reaching the peak when using the overall PEL framework. This shows the effectiveness of each module: the self-enhancement module excavates the essential information within each stream separately, and the mutual-enhancement module enhances the above information by integrating them.

**Denoising methods.** We seek a suitable denoising method by exploring filters with different types, kernel sizes and combinations. Tab. 5 lists the result on FF++ (LQ) dataset. For the filter type, we use the median filter, the mean filter and the non-local means (Buades, Coll, and Morel 2005), in which the median filter beats the other two. For the kernel size, we set it as $3, 5, 7$, respectively, and it turns out that the smallest kernel performs best. We further combine multiple median filters to extract noise with different size regions, followed by $1 \times 1$ group convolutional layer to reduce the number of channels. Based on the above analysis, we choose the single median filter with a kernel size of 3 as the default denoising setting of our method under all experiments.

### 4.4 Visualizations

**Self-enhancement.** Fig. 3 illustrates the enhanced region of feature maps (*i.e.*, the residual between $f_{in}$ and $f_{out}$ in Sec. 3.2) via the first self-enhancement module, where mask denotes the forged region which is obtained from the pixel-level difference between a forged image and its corresponding original image. From Fig. 3, we can observe that although the overall feature distributions of real and fake are similar, the latter manifests a higher enhancement in local manipulated regions, such as eyes or mouth. Such a fine-grained self-enhancement mechanism enables our method to uncover subtle forgery cues.

**Mutual-enhancement.** Fig.4 illustrates the representation residual before and after the last mutual-enhancement module. With the cooperation of the Complementary RGB feature and frequency feature, the manipulated regions on both streams are enhanced simultaneously, increasing the discriminability between real and fake faces. Moreover, both streams suppress the facial region in real images while intensifying that in fake images, enhancing the distinctiveness of fine-grained clues. We also observe different enhanced regions between RGB and frequency streams, *i.e.*, the former fit manipulated mask well, while the latter cover whole face with more subtle cues. The visualization illustrates that RGB

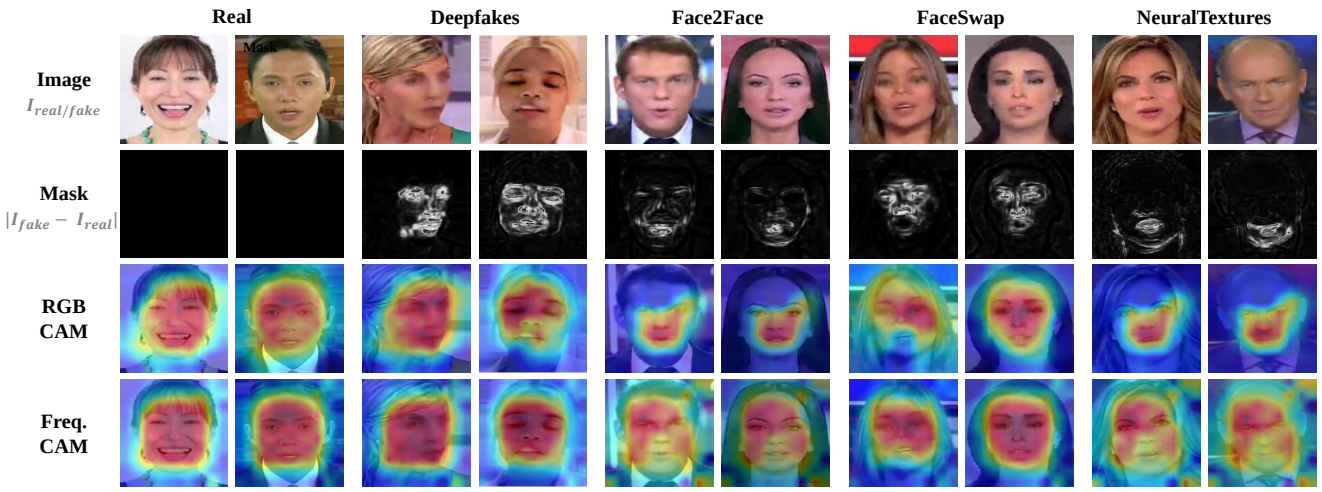


Figure 2: The attention maps for different kinds of faces.

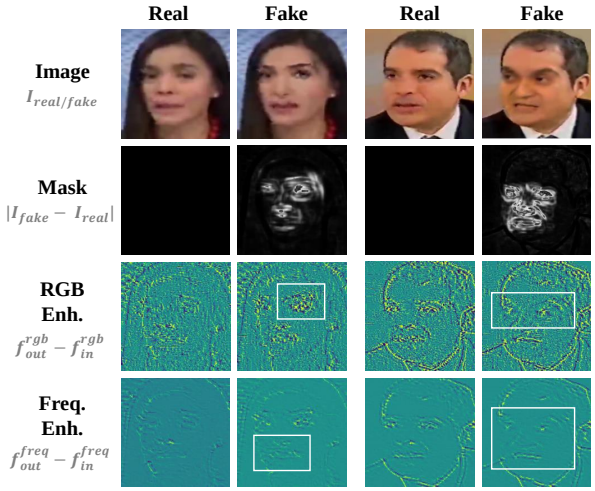


Figure 3: The self-enhancement of feature maps for the RGB stream and the frequency stream.

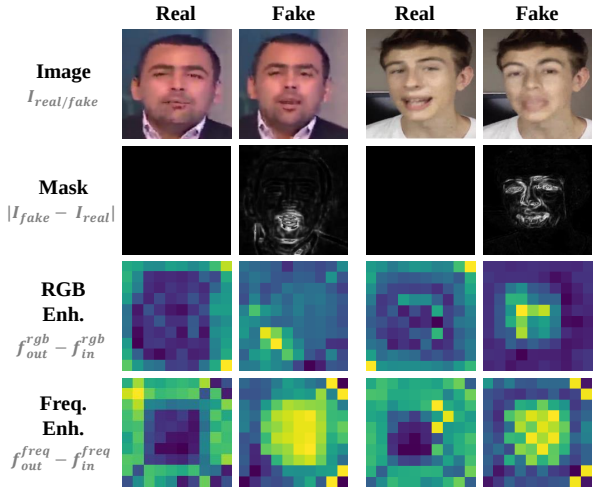


Figure 4: The mutual-enhancement of feature maps for the RGB stream and the frequency stream.

and frequency information are independent but can benefit from each other.

**Class Activation Mapping.** To explore the region of interest for different face types, we visualized the attention maps of several samples by class activation mapping (CAM) (Zhou et al. 2016). As shown in Fig. 2, both RGB and frequency streams can locate the artifacts, but with different emphasis. The RGB stream process inputs in the original image space with relatively focused attention maps: for the forged images, the focus area mainly coincides with the mask of forged regions; while for real images, the attention is evenly focused. The frequency stream considers a larger area (*i.e.* the entire face), which is often complementary to the focus area of the RGB stream, owing to the fine-grained frequency components' ability to reveal more subtle artifacts. In this manner, two streams can contribute to each other for a better forgery clues extraction.

## 5 Conclusion

In this paper, we propose a Progressive Enhancement Learning (PEL) framework to conduct the discriminative representation enhancement based on fine-grained frequency information. Firstly, we decompose the image into fine-grained frequency components and fed it into a two-stream network together with the RGB input, which aids in the subsequent two enhancement modules. The self-enhancement module captures the traces in different input spaces based on spatial noise enhancement and channel attention. The mutual enhancement module achieves concurrent enhancement of both branches through feature communication in the shared spatial dimension. The PEL framework can fully exploit the fine-grained clues in high-quality forgery faces and achieves state-of-the-art performance on both seen and unseen datasets. Visualizations of feature maps and class activation mappings reveal the inner mechanism and explain the effectiveness of our method.

## Acknowledgement

This work is supported by National Key Research and Development Program of China (No. 2019YFC1521104), Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0102), Shanghai Science and Technology Commission (No. 21511101200), Zhejiang Lab (No. 2020NB0AB01), National Natural Science Foundation of China (No. 61972157 and No. 72192821) and Art major project of National Social Science Fund (No. I8ZD22).

## References

Afchar, D.; Nozick, V.; Yamagishi, J.; and Echizen, I. 2018. MesoNet: A Compact Facial Video Forgery Detection Network. In *IEEE International Workshop on Information Forensics and Security*.

Bayar, B.; and Stamm, M. C. 2016. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*.

Buades, A.; Coll, B.; and Morel, J.-M. 2005. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Chen, S.; Yao, T.; Chen, Y.; Ding, S.; Li, J.; and Ji, R. 2021. Local Relation Learning for Face Forgery Detection. In *Association for the Advancement of Artificial Intelligence*.

Chollet, F. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Cozzolino, D.; Poggi, G.; and Verdoliva, L. 2017. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*.

Dang, H.; Liu, F.; Stehouwer, J.; Liu, X.; and Jain, A. K. 2020. On the Detection of Digital Face Manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Dolhansky, B.; Bitton, J.; Pflaum, B.; Lu, J.; Howes, R.; Wang, M.; and Ferrer, C. C. 2020. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*.

Dufour, N.; and Gully, A. 2019. Contributing Data to Deepfake Detection Research. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html Accessed: 2021-03-29.

Frank, J.; Eisenhofer, T.; Schönherr, L.; Fischer, A.; Kolossa, D.; and Holz, T. 2020. Leveraging frequency analysis for deep fake image recognition. In *Proceedings of the International Conference on Machine Learning*.

Fridrich, J.; and Kodovsky, J. 2012. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*.

Gu, Z.; Chen, Y.; Yao, T.; Ding, S.; Li, J.; Huang, F.; and Ma, L. 2021. Spatiotemporal Inconsistency Learning for DeepFake Video Detection. In *Proceedings of the 29th ACM International Conference on Multimedia*.

Kowalski, M. 2018. FaceSwap. https://github.com/marekkowalski/faceswap. Accessed: 2020-08-01.

Li, J.; Wang, Y.; Wang, C.; Tai, Y.; Qian, J.; Yang, J.; Wang, C.; Li, J.; and Huang, F. 2019a. DSFD: dual shot face detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Li, J.; Xie, H.; Li, J.; Wang, Z.; and Zhang, Y. 2021. Frequency-aware Discriminative Feature Learning Supervised by Single-Center Loss for Face Forgery Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Li, L.; Bao, J.; Yang, H.; Chen, D.; and Wen, F. 2019b. FaceShifter: towards high fidelity and occlusion aware face swapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Li, L.; Bao, J.; Zhang, T.; Yang, H.; Chen, D.; Wen, F.; and Guo, B. 2020a. Face X-Ray for More General Face Forgery Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Li, Y.; Chang, M.-C.; and Lyu, S. 2018. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In *IEEE International Workshop on Information Forensics and Security*.

Li, Y.; Yang, X.; Sun, P.; Qi, H.; and Lyu, S. 2020b. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Luo, Y.; Zhang, Y.; Yan, J.; and Liu, W. 2021. Generalizing Face Forgery Detection with High-frequency Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Masi, I.; Killekar, A.; Mascarenhas, R. M.; Gurudatt, S. P.; and AbdAlmageed, W. 2020. Two-branch recurrent network for isolating deepfakes in videos. In *Proceedings of the European Conference on Computer Vision*.

Matern, F.; Riess, C.; and Stamminger, M. 2019. Exploiting visual artifacts to expose deepfakes and face manipulations. In *IEEE Winter Applications of Computer Vision Workshops*.

Nguyen, H. H.; Fang, F.; Yamagishi, J.; and Echizen, I. 2019. Multi-task Learning for Detecting and Segmenting Manipulated Facial Images and Videos. In *IEEE International Conference on Biometrics Theory, Applications and Systems*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Qian, Y.; Yin, G.; Sheng, L.; Chen, Z.; and Shao, J. 2020. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *Proceedings of the European Conference on Computer Vision*.

Rahmouni, N.; Nozick, V.; Yamagishi, J.; and Echizen, I. 2017. Distinguishing computer graphics from natural images using convolution neural networks. In *IEEE International Workshop on Information Forensics and Security*.

Rossler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; and Nießner, M. 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the International Conference on Computer Vision*.

Singhal, D.; Gupta, A.; Tripathi, A.; and Kothari, R. 2020. CNN-based multiple manipulation detector using frequency domain features of image residuals. *ACM Transactions on Intelligent Systems and Technology*.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*.

Thies, J.; Zollhöfer, M.; and Nießner, M. 2019. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.*

Thies, J.; Zollhofer, M.; Stamminger, M.; Theobalt, C.; and Nießner, M. 2016. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tora, M. 2018. Deepfakes. https://github.com/deepfakes/faceswap/tree/v2.0.0. Accessed 2021-03-29.

Wang, X.; Yao, T.; Ding, S.; and Ma, L. 2020. Face Manipulation Detection via Auxiliary Supervision. In *Neural Information Processing - 27th International Conference*.

Xu, K.; Qin, M.; Sun, F.; Wang, Y.; Chen, Y. K.; and Ren, F. 2020. Learning in the Frequency Domain. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Yang, X.; Li, Y.; and Lyu, S. 2019. Exposing deep fakes using inconsistent head poses. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Zhao, H.; Zhou, W.; Chen, D.; Wei, T.; Zhang, W.; and Yu, N. 2021. Multi-attentional deepfake detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhao, T.; Xu, X.; Xu, M.; Ding, H.; Xiong, Y.; and Xia, W. 2020. Learning to Recognize Patch-Wise Consistency for Deepfake Detection. *arXiv preprint arXiv:2012.09311*.

Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zi, B.; Chang, M.; Chen, J.; Ma, X.; and Jiang, Y.-G. 2020. WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection. In *Proceedings of the ACM Multimedia Conference*.