DPL: Cross-quality DeepFake Detection via Dual Progressive Learning

Dongliang Zhang, Yunfei Li, Jiaran Zhou, Yuezun Li*

School of Computer Science and Technology,
Ocean University of China, Qingdao, China

Abstract. Real-world DeepFake videos often undergo various compression operations, resulting in a range of video qualities. These varying qualities diversify the pattern of forgery traces, significantly increasing the difficulty of DeepFake detection. To address this challenge, we introduce a new Dual Progressive Learning (DPL) framework for cross-quality DeepFake detection. We liken this task to progressively drilling for underground water, where low-quality videos require more effort than high-quality ones. To achieve this, we develop two sequential-based branches to "drill waters" with different efforts. The first branch progressively excavates the forgery traces according to the levels of video quality, *i.e.*, time steps, determined by a dedicated CLIP-based indicator. In this branch, a Feature Selection Module is designed to adaptively assign appropriate features to the corresponding time steps. Considering that different techniques may introduce varying forgery traces within the same video quality, we design a second branch targeting forgery identifiability as complementary. This branch operates similarly and shares the feature selection module with the first branch. Our design takes advantage of the sequential model where computational units share weights across different time steps and can memorize previous progress, elegantly achieving progressive learning while maintaining reasonable memory costs. Extensive experiments demonstrate the superiority of our method for cross-quality DeepFake detection.

Keywords: Cross-quality DeepFake detection · Multimedia Forensics

1 Introduction

The rapid advancement of deep learning technologies has significantly propelled the evolution of *Deepfakes*, a generative technique capable of creating highly realistic faces within videos. Misusing DeepFake technique poses serious societal concerns, including forging fake news, economic fraud, and making illusions of public figures [2, 8, 14]. In response to this problem, numerous forensics methods have emerged to detect DeepFakes [5, 18, 30, 32]. These methods, typically developed on deep neural networks (DNNs), have shown promising and even

* Corresponding author (liyuezun@ouc.edu.cn).

near-perfect performance on standarded datasets [12, 21, 34, 46], seemingly indicating that the DeepFake problem has been successfully addressed.

However, real-world DeepFake videos often undergo various compression operations, resulting in varying video quality. Social platforms (*e.g.*, TikTok, Instagram) and video platforms (*e.g.*, YouTube, YouKu) usually compress the uploaded videos to reduce broadcasting overhead. These compression operations disturb the pattern of forgery traces, significantly degrading the performance of existing methods. Thus, exploring the practical feasibility of DeepFake detection against quality variation is pressing.

To address this concern, several efforts have been proposed [3,17,18]. Noticing that existing attempts are usually validated on high-quality DeepFake videos, some methods [3, 18] have shifted their focus to low-quality DeepFake videos. However, these methods do not investigate the generalization detection across various qualities. More recently, QAD [17] has extended to achieve cross-quality DeepFake detection, by employing specific learning strategies (*e.g.*, contrastive learning and collaborative learning) to capture generic forgery traces across different qualities. Nonetheless, when the quality discrepancy is substantial, these models may reach their capacity limits and struggle to capture the generic forgery traces. While ensumbling multiple models may overcome these limits, it introduces significant computational overhead, hindering their practical applications.

In this paper, we rethink this problem and propose a Dual Progressive Learning (DPL) framework for cross-quality DeepFake detection. Our method is designed based on a novel perspective: we liken forgery traces to underground water and the process of DeepFake detection to drilling for water. In high-quality videos, forgery traces are likely "near the surface" and thus easily "drilled". However, in low-quality videos, these traces are greatly disturbed by compression, *i.e.*, "concealed deeper", requiring more effort to reveal them (see Fig. 1). Thus, a natural question arises "**Can we design a dynamic model that adaptively assigns efforts for different DeepFake videos?**"

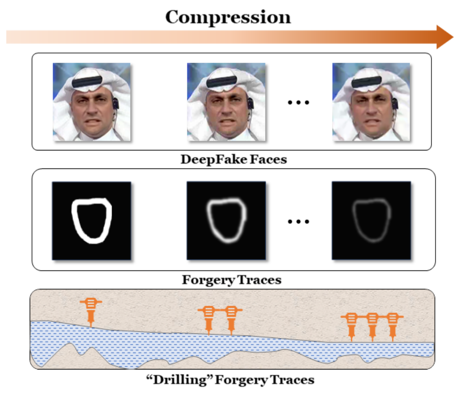

Fig. 1: Different image quality

To achieve this, we first develop a Visual Quality Guided Progressive Learning (VQPL) branch. This branch is designed based on a sequential model, which first estimates the video quality levels as different time steps and progressively engages computational unit (*e.g.*, GRU [9]) to excavate forgery traces, with lower quality requiring more time steps and vice versa. Specifically, the video quality level is estimated by a new CLIP-based Visual Quality Indicator (VQI). Considering that each time step in progressive learning likely focuses on different views, *i.e.*, "the depth of drilling", we propose a Feature Selection Module (FSM), which assigns appropriate features for computational units at different time steps. The benefit of this design is that the sequential model is inherently dynamic with weight-sharing computational units and can memorize the previous progress, reducing the computational cost while increasing the model diversity, similar to ensumble multiple models.

Besides varying video quality, the level of forgery identifiability is another implicit obstacle hindering the detection performance. As shown in Fig. 2, while these DeepFake faces have the same compression level, their forgery identifiability differs. This challenges the proposed VQPL branch. To compensate, we introduce another Forgery Identifiability Guided Progressive Learning (FIPL) branch, which borrows the same spirit of VQPL to handle varying levels of forgery identifiability. This branch shares the same feature selection module and employs computational units controlled by a new CLIP-based Forgery Identifiability Indicator (FII). We then fuse the features of these two branches for DeepFake identification.

To effectively train the proposed DPL framework, we introduce a two-stage pipeline with a Proximal Policy Optimizer (PPO) algorithm [35], that is applied on dedicated designed objective functions. Extensive experiments are conducted on many public datasets and compared to several state-of-the-art methods. The results show that our method outperforms others, demonstrating the efficacy of our method in cross-quality DeepFake detection.

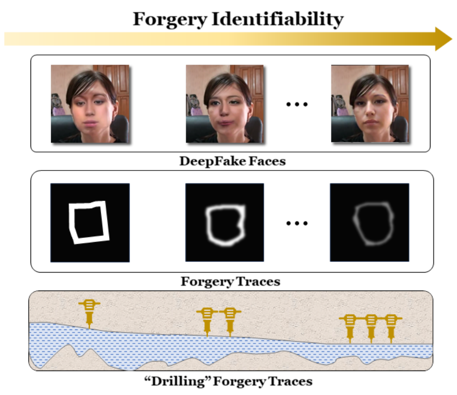

Fig. 2: Different forgery quality

Our contributions can be summarized as follows:

– We propose a new Dual Progressive Learning framework (DPL) for cross-quality DeepFake detection. In contrast to existing methods, we formulate this task as progressively excavating forgery traces according to varying levels of video quality and forgery identifiability.
– For both branches, we introduce a shared Feature Selection Module (FSM) that can flexibly assign features to computational units. For each branch, we design CLIP-based indicators to determine the time steps.
– We describe a two-stage training pipeline with devoted designed objective functions, to effectively train the proposed DPL framework.
– Extensive experiments are conducted on several public datasets, comparing our method to many state-of-the-art methods. We also thoroughly study the effect of each component, revealing interesting insights for future research.

2 Related Work

Deepfakes have become a highly critical issue due to the significant threats they pose to social security and privacy. To counteract the risk, a large number of deepfake detection methods have been proposed [6, 18, 24, 27, 30, 32, 38]. These methods are typically developed on deep neural networks (DNNs) and attempt to uncover the forgery traces concealed in various aspects, such as physiological signals [16, 31, 38], frequency domain [23, 28, 32], blending boundaries [7, 20, 37], and high dimensional feature spaces derived by networks [1, 30, 43]. Although these approaches have demonstrated their promising capacity on public datasets, they still face great challenges when applying them in real-world scenarios.

Unlike standard datasets, the real-world DeepFake videos are more diversified, greatly degrading the performance of existing methods. Several recent methods have been proposed to enhance the generalization ability of detection, mainly concentrating on the scenarios of different manipulations [37, 43, 44] and different datasets [20, 27, 42]. Besides these scenarios, the variation in video quality is also a crucial obstacle in DeepFake detection. For instance, social / video platforms usually apply compression operations to reduce memory costs, resulting in varying forgery traces and degrading detection performance. The works of [3, 17, 18] focus on this problem and attempt to expose DeepFake videos with various qualities. In this paper, we describe a new method based on a novel perspective: we treat this task as "drilling" water and develop a Dual Progressive Learning (DPL) framework to enhance the detection ability under cross-quality scenarios.

3 Method

This paper describes a new Dual Progressive Learning (DPL) framework to detect DeepFakes across different video qualities. Specifically, we develop a Visual Quality Guided Progressive Learning (VQPL) branch to extract forgery traces progressively with the instruction of a Visual Quality Indicator (VQI) (Sec. 3.1). Moreover, we describe a Forgery Identifiability Guided Progressive Learning (FIPL) branch to overcome the difficulties introduced by forgery identifiability (Sec. 3.2). These two branches collaborate to capture the generic forgery traces by fusing the output features of each branch. Then we introduce specific objectives and training procedures devoted to the proposed framework (Sec. 3.3). The overall framework of our method is illustrated in Fig. 3.

3.1 Visual Quality Guided Progressive Learning

This branch consists of four key components: a feature extractor, a feature selection module, a sequential-based detection module, and a video quality indicator. Given an input face image $\mathbf{x} \in [0, 255]^{H \times W \times 3}$, we first extract the feature $f \in \mathbb{R}^{h \times w \times c}$ using a feature extractor Ψ (*i.e.*, backbone network). This feature serves as the input for the sequential-based detection module, a recurrent

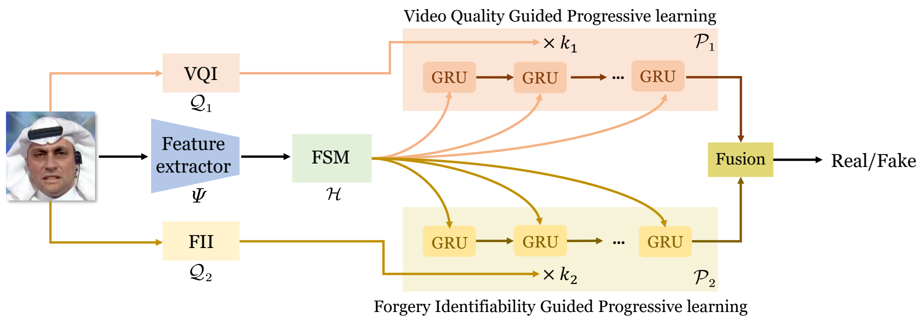

Fig. 3: Overview of proposed DPL framework.

network $\mathcal{P}_1$ with GRUs [9]. The maximum time step of $\mathcal{P}_1$ is represented by the visual quality level of input face image $\mathbf{x}$, estimated by a video quality indicator $\mathcal{Q}_1$. Denote $k_1 = \mathcal{Q}_1(\mathbf{x}) \in \{1, ..., K_1\}$ as the estimated quality level. Then the detection module $\mathcal{P}_1$ is executed k times with the following formulas $h_t = \mathcal{P}_1(f_t, h_{t-1})$, where f_t denotes the input feature at time step $t \in \{1, ...k_1\}$, h_{t-1} and h_t are the hidden state and current output (a hidden state for next time step). Note that h_0 is the initial state and h_{k_1} is the final output of this detection module. Then we fuse the outputs of all time steps as the final feature of this branch $f_{\text{VQ}} = \sum_t h_t$.

Intuitively, the feature f can be employed as the input for all time steps, *i.e.*, $f_t = f, t \in \{1, ...k_1\}$. However, considering the focus on progressive learning likely varies. For example, the early time steps are likely capable of capturing superficial traces while the late time steps should be responsible for more challenging traces. Therefore, we design a Feature Selection Module $\mathcal{H}$, which can process the feature f for different time steps, *i.e.*, $f_t = \mathcal{H}_t(f), t \in \{1, ...k_1\}$.

Feature Selection Module. The core of this module is another recurrent network Φ, which is employed to recursively process the input feature of $\mathcal{P}_1$ for the current time step. Specifically, given the input feature f_{t-1} at time step $t-1$, we derive a mask based on this feature and perform a masking operation to refine f_{t-1} as f_t. By learning masking operations recursively, the features corresponding to challenging forgery traces will be retained for the later time steps, leading to progressive learning.

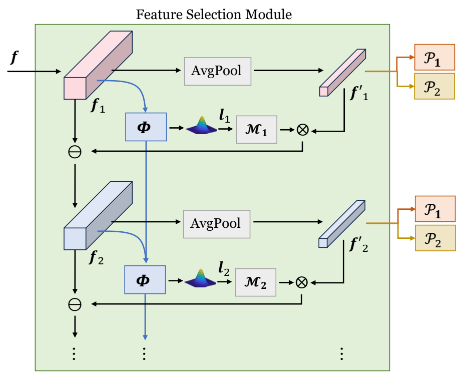

Fig. 4: Overview of the feature selection module

As shown in Fig. 4, the network Φ takes as input the features $f_t \in \mathbb{R}^{h \times w \times c}$ and generate a mean μ_t and a standard variation σ_t to form a Gaussian distribution $\mathcal{N}(\mu_t, \sigma_t)$. This distribution outlines the probability of masking positions. This process can be defined as $\mu_t, \sigma_t, g_t = \Phi(f_t, g_{t-1})$, where g_{t-1} and g_t are the hidden states for the previous and current time step. Then we create a mask $\mathcal{M}_t \in \mathbb{R}^{h \times w \times c}$ with the same size as f_t. Given the distribution $\mathcal{N}(\mu_t, \sigma_t)$, we can sample a starting position l_t and scale it to the range $[0, c]$. Then we let $\mathcal{M}_t^{(:,:,i)} = 0$ if $i \in [l_t, l_t + \delta_t]$, otherwise $\mathcal{M}_t^{(:,:,i)} = 1$. Note that δ_t is a hyper-parameter that represents the size of the retained region. Then we obtain f'_t by performing average pooling on f_t and perform element-wise multiplication with $\mathcal{M}_t$. The results are subtracted from f_t to retain suitable features for the next time step. The process of feature selection can be defined as $f_{t+1} = f_t - \mathcal{M}_t \otimes f'_t$. Then we perform average pooling on f_{t+1} to obtain f'_{t+1}, the input feature for the time step $t+1$ in $\mathcal{P}_1$. Note that the feature at the first time step is $f_1 = f$.

Video Quality Indicator. Benefiting from the powerful semantic prior knowledge of vision-language models (VLMs) [4,33], we develop a CLIP-based video quality indicator $\mathcal{Q}_1$ to estimate the visual quality levels of the input face image, severing as the time steps in sequential-based detection module $\mathcal{P}_1$. Inspired by [40], we adopt the paired text prompt of `[‘‘Good photo.’’, ‘‘Bad photo.’’]` to estimate the image quality. Specifically, given the input face image $\mathbf{x}$, we obtain the visual embedding from the image encoder as $v = \mathcal{Q}_1^{IE}(\mathbf{x})$. Denote the text embedding from the text encoder as $t_1 = \mathcal{Q}_1^{TE}($`[‘‘Good photo.’’]`$)$ and $t_2 = \mathcal{Q}_1^{TE}($`[‘‘Bad photo.’’]`$)$. Then we can obtain the quality score q as

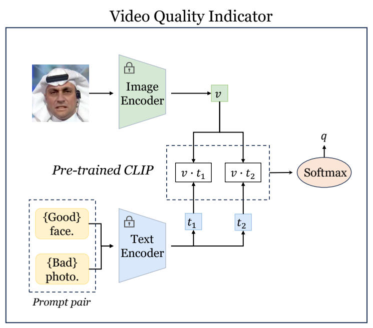

Fig. 5: Overview of VQI.

$$
\begin{aligned}
s_i &= \frac{v \odot t_i}{\|x\| \cdot \|t_i\|}, i \in \{1, 2\} \\
q &= \frac{\exp(s_1)}{\exp(s_1 + s_2)},
\end{aligned} \tag{1}
$$

where s_i is the cosine similarity between visual and corresponding text embedding, which is performed `softmax` as the final score. A larger q indicates the image $\mathbf{x}$ has better quality and vice versa.

Generally, low-quality videos undergo compression operations, disrupting the pattern of forgery traces. Thus they exhibit unnoticeable forgery artifacts, making them more difficult to detect than high-quality videos. To this end, we should assign more steps for low-quality videos. Since the quality scores are continuous, we should discretize them to represent time steps. Specifically, we first estimate the quality scores for all images in the training set and divide the scores into K_1 bins, ensuring that the range of each bin can include a similar number of images. Inspired by Likert-scale [11,45], we establish $K_1 = 5$ quality levels using $k \in \{1, 2, 3, 4, 5\} = \{$“perfect”, “good”, “fair”, “poor”, “bad”$\}$. It is worth noting that a smaller bin index represents better image quality, corresponding to assigning fewer time steps. The overview of VQI is shown in Fig. 5.

3.2 Forgery Identifiability Guided Progressive Learning

This branch is complementary to the VQPL branch, handling the variation in the forgery identifiability. Specifically, it shares the same feature extractor Ψ and feature selection module $\mathcal{H}$. But it employs another sequential-based detection module $\mathcal{P}_2$ to progressively excavate forgery traces based on forgery identifiability levels, estimated by a new CLIP-based Forgery Identifiability Indicator (FII). Denote $k_2 \in \{1, ..., K_2\}$ as the identifiability level of image $\mathbf{x}$. Similar to $\mathcal{P}_1$, the module $\mathcal{P}_2$ is executed k_2 times, described by $w_t = \mathcal{P}_2(f_t, w_{t-1})$, where f_t is the input feature processed by feature selection module $\mathcal{H}$ and w_t is the hidden state. Note that w_k is the final output of this branch. Then we fuse the outputs of all time steps as $f_{\mathrm{FI}} = \sum_t w_t$.

Forgery Identifiability Indicator. This indicator, denoted as $\mathcal{Q}_2$, shares the same CLIP model and estimation pipeline used in VQI. However, we design a different paired text prompt `[“manipulated face.”, “genuine face.”]`, which can represent whether the given image $\mathbf{x}$ looks like being manipulated, reflecting the forgery identifiability levels. Similar to VQI, we also employ the Eq. (1) to calculate the score and use the same strategy to discretize the scores into $K_2 = 5$ levels. A smaller k_2 denotes the forgery is more identifiable, thus requiring less effort to detect, otherwise requiring more effort. The overview of VQI is shown in Fig. 6.

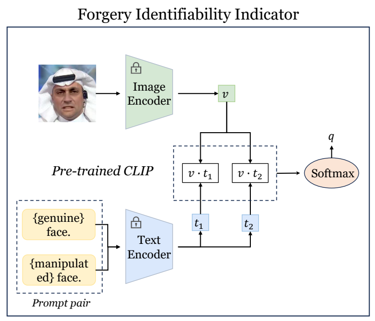

Fig. 6: Overview of FII.

3.3 Objective and Training

At the beginning of training, the framework likely struggle to identify forgery traces, making the recurrent network Φ in feature selection module difficult to converge. Thus, we describe a two-stage training pipeline that trains feature selection module and dual branches respectively. In the first stage, we enhance the feature extraction capacity by training the dual branches without Φ. Once the whole network has developed adequate feature extraction capacity, we shift our focus to training Φ within the feature selection module, keeping the rest of the network fixed during the second stage.

Stage I. In this stage, we replace Φ with a randomizer, where the position l_t for creating the mask is randomly sampled from a uniform distribution, indicating that the input feature for each time step is randomly masked without using Φ.

In training, we employ the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$, enhanced by the Focal Loss [22] to learn the discriminative features between real and fake faces. Moreover, we introduce a regularization loss $\mathcal{L}_{\mathrm{REG}}$ to enhance the learning of hard samples, the samples are often misclassified. At early time steps, the network may easily misclassify these samples due to insufficient learning capacity at the beginning of progressive learning. Therefore, we can make the network refrain from making decisions (*i.e.*, real or fake) in the early time steps, deferring these decisions to the later time steps. Specifically, we use the entropy value to assess whether a decision is made. A larger entropy value denotes the confidence for real and fake classes is similar, indicating the authenticity is not determined. This loss can be defined as

$$\mathcal{L}_{\mathrm{REG}} = -\sum_{\mathbf{x}_i} \mathit{\Pi}(\mathbf{x}_i) \left[\sum_{j\in\{0,1\}} c_i^j \log(c_i^j) \right], \tag{2}$$

where $\mathit{\Pi}(\mathbf{x}_i)$ is an indicator function, which is equal to 1 if $\mathbf{x}_i$ is a hard sample, 0 otherwise, c_i^j denotes the confidence of j-th class on $\mathbf{x}_i$, where $j \in \{0, 1\}$ represents real and fake respectively. Thus, in this stage, the overall objective function can be defined as

$$\mathcal{L}_1 = \mathcal{L}_{\mathrm{CE}} + \mathcal{L}_{\mathrm{REG}}. \tag{3}$$

Stage II. In this stage, we revoke the network $\mathit{\Phi}$ but fix other parameters. To effectively train $\mathit{\Phi}$, we employ the Proximal Policy Optimizer algorithm (PPO) [35] to maximize the total reward. The motivation is that the network $\mathit{\Phi}$ should assist VQPL and FIPL to progressively improve the performance along with the time steps increasing. Denote c_t as the predicted probability of input image $\mathbf{x}$ for the ground truth label and $r_t = c_t - c_{t-1}$ as the rewards. The goal of this stage is to maximize the total rewards as $\max_{\mathit{\Phi}} \sum_{t=2} r_t$. Inspired by [41], we adopt three objective terms for this goal. Specifically, we employ a reward learning term $\mathcal{L}_{\mathrm{REW}}$ [35] which instructs the optimization direction of $\mathit{\Phi}$ as

$$\mathcal{L}_{\mathrm{REW}} = -\mathbb{E}_t \left[\min\left(\frac{p_{\mathit{\Phi}}\left(l_t|f_t\right)}{p_{\mathit{\Phi}'}\left(l_t|f_t\right)}, \mathrm{clip}\left(\frac{p_{\mathit{\Phi}}\left(l_t|f_t\right)}{p_{\mathit{\Phi}'}\left(l_t|f_t\right)}, 1-\epsilon, 1+\epsilon \right) \right) \mathcal{A}_t \right], \tag{4}$$

where $p_{\mathit{\Phi}}\left(l_t|f_t\right)$ represents the probability of sampling l_t from $\mathit{\Phi}$, while $p_{\mathit{\Phi}'}\left(l_t|f_t\right)$ denotes the probability of sampling l_t from $\mathit{\Phi}'$. Note that $\mathit{\Phi}$ and $\mathit{\Phi}'$ indicate the state after and before the update. $\mathcal{A}_t$ is an indicator at time step t, measuring the increase of reward compared to a standard value. $\mathcal{A}_t > 0$ indicates the current state of l_t can increase the reward, thus increasing $p_{\mathit{\Phi}}\left(l_t|f_t\right)$ and vice versa.

Specifically, $\mathcal{A}_t = -\mathcal{V}(f_t) + \mathcal{R}_t$, where $\mathcal{R}_t = \sum_t r_t$ denotes the sum of rewards at all steps and $\mathcal{V}(f_t)$ is the standard obtained using a sub-network $\mathcal{V}$ with a squared error loss $\mathcal{L}_{\mathrm{SE}}$ as

$$\mathcal{L}_{\mathrm{SE}} = \|\mathcal{V}(f_t) - \mathcal{R}_t\|^2. \tag{5}$$

Minimizing this term can allow the sub-network $\mathcal{V}$ to learn the practical rewards. Moreover, in case the l_t is only sampled in a small range, we describe a term $\mathcal{L}_{\mathrm{EN}}$ to measure the entropy of the distribution, where a larger value denotes the distribution is flatter. The overall objective in this stage is defined as

$$\mathcal{L}_2 = \mathcal{L}_{\mathrm{REW}} + \mathcal{L}_{\mathrm{SE}} - \mathcal{L}_{\mathrm{EN}}. \tag{6}$$

4 Experiments

4.1 Experimental Settings

Datasets. Our method is validated on widely used deepfake datasets: FaceForensics++ (FF++) [34], CelebDFv2 (CDFv2) [21], FaceShifter (FSH) [19] and FFIW10K [46]. FF++ datasets comprise 1000 original videos and 4000 fake videos produced in four manipulation methods, Deepfakes (DF), Face2Face (F2F), FaceSwap (FS), and NeuralTextures (NT), with two widely used quality versions: c23 (lightly compressed), and c40 (heavily compressed). CDFv2 [21] is a recent high-quality dataset with more than 5600 DeepFake videos. FSH [19] contains 1000 original videos and 1000 fake videos. FFIW10k consists of 10000 fake videos with an average of three human faces in every frame.

Implementation Details. We employ ConvNeXt-T [26] as our feature extractor. During the training phase, we employ random JPEG compression for data augmentation. The size of the retained region is set $\delta_t = c/3$. The DPL framework is trained using Adam optimizer [15] with a learning rate of 5e-5. The batch size is 32 and the total training epoch is 10, where 5 epochs are used in stage I and 5 epochs are used in stage II. The input image size is 224×224.

4.2 Results

Our method is compared to eight state-of-the-art methods, including F3Net (ECCV'20) [32], SRM (CVPR'21) [28], SPSL (CVPR'21) [23], SIA (ECCV'22) [39], RECCE (CVPR'22) [5], UIA-ViT (ECCV'22) [47], QAD (ICCV'23) [17], UCF (ICCV'23) [44]. Note that all methods are trained on FF++(c23) and evaluated by the Area Under Curve (AUC) metric for a fair comparison.

Cross-quality Evaluation. In this setting, the methods are evaluated on the test set for each manipulation method DF, F2F, FS, and NT, under random JPEG compression. The results are presented in Tab. 1, showing that our method achieves superior performance compared to other methods in most cases and performs the best on average, improving the AUC by approximately 1% compared to the second-best method QAD.

Moreover, we further evaluate these methods in a more challenging setting, where the train set remains FF++(23) and the test set is FF++(40). The test set is subjected to the same random JPEG compression. The results, shown in Tab. 2, indicate that our method consistently outperforms the others, achieving the highest average score and improving the AUC by about 1% over the second-best method UIA-ViT.

Cross-quality with Cross-manipulation Evaluation. We also perform a coss-quality cross-manipulation evaluation on the FF++ dataset. Each method is trained on one manipulation type and tested across all four types (NT, F2F, FS, and DF). Given that QAD and UCF are among the most recent and effective methods, we compare them with our method. As shown in Tab. 3, our method

Table 1: Cross-quality Evaluation (AUC %). Train set: FF++(**c23**). Test set: FF++(**c23**) with random JPEG compression. The **best** and <u>second-best</u> scores are highlighted in bold and underlined.

Methods	DF	F2F	FS	NT	Avg.
Random JPEG compression on c23 test set					
F3Net (ECCV'20) [32]	94.93	91.75	93.22	83.47	90.84
SRM (CVPR'21) [28]	96.16	<u>95.09</u>	95.67	84.93	92.96
SPSL (CVPR'21) [23]	95.60	95.00	93.93	<u>85.96</u>	92.62
SIA (ECCV'22) [39]	93.86	94.06	93.73	81.73	90.83
RECCE (CVPR'22) [5]	95.31	94.17	95.18	84.14	92.20
UIA-ViT (ECCV'22) [47]	95.94	91.40	93.02	80.33	90.17
QAD (ICCV'23) [17]	<u>97.56</u>	94.71	**97.11**	84.55	<u>93.48</u>
UCF (ICCV'23) [44]	96.87	94.56	93.70	80.94	91.52
DPL (Ours)	**98.14**	**95.16**	<u>96.63</u>	**87.70**	**94.41**

Table 2: Cross-quality evaluation (AUC %). Train set: FF++(**c23**). Test set: FF++(**c40**) with random JPEG compression.

Methods	DF	F2F	FS	NT	Avg.
Random JPEG compression on c40 test set					
F3Net (ECCV'20) [32]	90.41	85.26	89.02	<u>73.80</u>	84.62
SRM (CVPR'21) [28]	90.87	85.97	90.89	68.24	83.99
SPSL (CVPR'21) [23]	89.03	86.82	88.50	70.03	83.60
SIA (ECCV'22) [39]	89.05	<u>86.96</u>	89.57	68.16	83.43
RECCE (CVPR'22) [5]	90.09	85.45	90.51	68.97	83.75
UIA-ViT (ECCV'22) [47]	92.85	85.73	90.40	**74.03**	<u>85.75</u>
QAD (ICCV'23) [17]	90.46	84.30	<u>91.83</u>	64.21	82.70
UCF (ICCV'23) [44]	<u>92.92</u>	**87.28**	89.33	68.61	84.53
DPL (Ours)	**94.22**	86.26	**92.88**	72.09	**86.36**

still outperforms others on average, demonstrating its effectiveness in detecting varying qualities under different scenarios.

Cross-quality with Cross-dataset Evaluation. Furthermore, we perform a cross-quality cross-dataset evaluation, where all methods are trained on FF++(c23) and directly evaluated on the test set of the unseen dataset with random JPEG compression. Tab. 4 shows the results under this setting. It can be seen that our method achieves the best performance on all datasets and averagely improves the AUC score by 0.58% compared to the second-best method UIA-ViT. This experiment further corroborates the practical feasibility of our method even across various datasets.

4.3 Ablation Study

This section studies the effect of different settings in our method. Note that we apply random JPEG compression on the test set in each setting as in the main experiments.

Table 3: Cross-quality with cross-manipulation evaluation (AUC %). Train set: FF++(c23). Test set: both FF++(c23) and FF++(c40) with random JPEG compression.

Methods	Train	NT	F2F	FS	DF	Avg.
Random JPEG Compression on c23 and c40 test set						
QAD (ICCV'23) [17]	NT	84.10	63.70	45.73	72.34	<u>68.11</u>
	F2F	55.31	96.07	56.28	66.62	
	FS	48.66	58.49	98.34	62.45	
	DF	62.18	63.74	56.39	99.50	
UCF (ICCV'23) [44]	NT	81.54	60.24	46.42	64.14	66.56
	F2F	53.80	94.87	57.37	64.78	
	FS	49.06	55.15	97.37	67.77	
	DF	58.15	58.80	56.69	99.03	
DPL (Ours)	NT	84.72	62.41	47.72	70.14	**68.97**
	F2F	54.32	95.33	60.95	69.04	
	FS	47.15	58.40	98.10	74.20	
	DF	60.47	63.16	58.27	99.18	

Table 4: Cross-quality with cross-dataset evaluation (AUC %). Train set: FF++(c23). Test set: individual test with random JPEG compression.

Methods	FSH	CDFv2	FFIW10k	Avg
Random JPEG Compression on individual test set				
F3Net (ECCV'20) [32]	70.37	67.41	65.85	67.78
SRM (CVPR'21) [28]	66.76	66.40	66.81	65.98
SPSL (CVPR'21) [23]	65.80	69.34	65.74	65.74
SIA (ECCV'22) [39]	64.37	64.78	63.68	63.90
RECCE (CVPR'22) [5]	66.92	68.25	64.64	65.72
UIA-ViT (ECCV'22) [47]	<u>72.60</u>	70.52	67.55	<u>70.11</u>
QAD (ICCV'23) [17]	67.51	<u>70.58</u>	<u>68.16</u>	67.30
UCF (ICCV'23) [44]	66.75	66.89	64.92	65.64
DPL (Ours)	**74.91**	**71.00**	**68.77**	**70.69**

Effect of Each Component. This part studies the effect of each component. Specifically, we study five settings: 1) **Baseline**: only employing the backbone network without DPL. 2) **w/o VQPL**: disabling the VQPL branch. 3) **w/o FIPL**: disabling the FIPL branch. 4) **w/o Train Stage II**: only applying random sampling l_t in FSM. 5) **w/o $\mathcal{L}_{\mathbf{REG}}$**: only using the cross-entropy loss $\mathcal{L}_{\text{CE}}$ without regularization loss $\mathcal{L}_{\text{REG}}$ in Stage I. As shown in Tab. 5, without either VQPL or FIPL, the performance drops at FF++(c40) and FF++(c23), demonstrating their role in dual progressive learning branches. Moreover, omitting train stage II also drops the performance, corroborating the efficacy of network Φ in the feature selection module. Additionally, without $\mathcal{L}_{\text{REG}}$ introduces a certain performance drop, showing the effectiveness of this objective.

Table 5: Effect of each component.

Setting	FF++(c40)	FF++(c23)
Baseline	84.65	92.82
w/o VQPI	85.68	92.83
w/o FIPL	86.31	93.42
w/o Train Stage II	86.32	94.39
w/o $\mathcal{L}_{\text{REG}}$	85.26	93.32
DPL	**86.36**	**94.41**

Table 6: Effect of different backbones.

Backbone	FF++(c40)	FF++(c23)
ConvNeXt-B	86.43	93.61
ConvNeXt-B + DPL	86.78 (+0.35)	94.61 (+1.00)
ConvNeXt-S	85.05	93.35
ConvNeXt-S + DPL	86.56 (+1.51)	94.20 (+0.85)
Swin-T	85.98	93.20
Swin-T + DPL	86.12 (+0.14)	93.44 (+0.24)
Res-50	84.38	93.01
Res-50 + DPL	84.86 (+0.48)	93.36 (+0.35)

Table 7: Effect of different fusion strategies.

Setting	FF++(c40)	FF++(c23)
Add	86.36	94.41
Concat	85.43	94.26
Attention [10]	86.54	93.46

Table 8: Effect of different settings of applying $\mathcal{L}_{\text{REG}}$

Setting	FF++(c40)	FF++(c23)
$\mathcal{L}^{*}_{\text{REG}}$	85.16	92.41
$\mathcal{L}^{\dagger}_{\text{REG}}$	86.20	94.13
$\mathcal{L}^{\ddagger}_{\text{REG}}$	86.35	94.41

Effect of Different Backbones. Tab. 6 shows the comparability of DPL with four different backbone networks: ConvNeXt-B [26], ConvNeXt-S [26], Swin-T [25], ResNet-50 [13]. The results show that the DPL improves the performance from 0.41% to 1.18% on average on two sets compared to the baselines.

Various Fusion Strategies for Dual Branches. This part studies three fusion strategies in Tab. 7: 1) **Add**: directly adding the features from the VQPL branch and FIPL branch. 2) **Concat**: concatenating the features from the VQPL branch and FIPL branch. 3) **Attention**: merge the features from both branches using an attention module AFF [10]. It can be seen that the strategy of directly adding features performs best from the overall effect.

Different Settings of Applying $\mathcal{L}_{\text{REG}}$. For the regularization loss $\mathcal{L}_{\text{REG}}$, applying it to the output features at different time steps results in varying effects. Tab. 8 presents three settings: 1) applying $\mathcal{L}_{\text{REG}}$ only to the output features of the first time step, denoted as $\mathcal{L}^{*}_{\text{REG}}$. 2) applying $\mathcal{L}_{\text{REG}}$ only to the output features of the maximum effective time step, denoted as $\mathcal{L}^{\dagger}_{\text{REG}}$. 3) applying $\mathcal{L}_{\text{REG}}$ to the output features of the maximum effective time step including all preceding time steps, denoted as $\mathcal{L}^{\ddagger}_{\text{REG}}$. The results show that the third strategy achieves the best performance.

4.4 Further Analysis

Robustness Analysis. This part analyzes the robustness of our method to unseen perturbations. Specifically, we employ four perturbations: Saturation (color saturation change), Contrast (color contrast change), Blockwise (local block-wise distortion), Noise (white Gaussian noise in color components). Each perturbation has five severity levels. Fig. 7 shows the performance of different method against perturbations. It can be seen that our method consistently outperforms other methods across all scenarios.

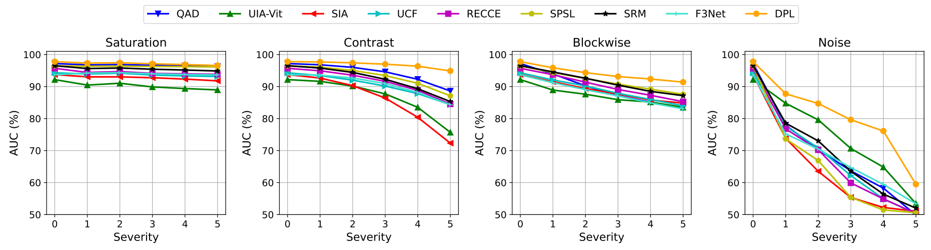

Fig. 7: Performance of different methods under four perturbations.

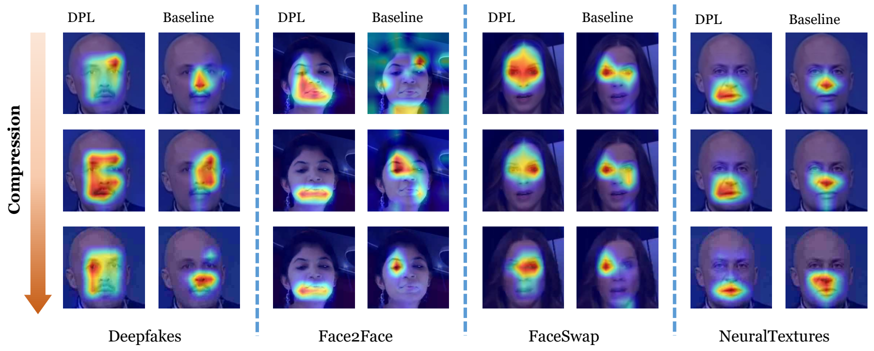

Fig. 9: Grad-CAM visualization of DPL and baseline.

Integrating CLIP for Detection. Since CLIP contains powerful semantic priors, we explore integrating it for feature enhancement. Specifically, we investigate three settings: 1) Concatenating feature from CLIP image encoder with backbone Ψ. 2) Adding feature from CLIP image encoder with backbone Ψ. 3) Adding feature from CLIP image encoder with dual branches $\mathcal{P}_1, \mathcal{P}_2$. However, as shown in Tab. 9, integrating CLIP into DPL achieves inconspicuous improvement.

Table 9: Performance of integrating CLIP image encoder for detection.

Setting	FF++(c40)	FF++(c23)
Concat(CLIP, Ψ)	85.68	92.96
Add(CLIP, Ψ)	85.11	94.01
Add(CLIP, $\mathcal{P}_1, \mathcal{P}_2$)	85.12	94.36

Attention Visualization. We further use GradCAM [36] to localize the activated regions to detect forgery. The visualization results on Deepfakes, Face2Face, FaceSwap, and NeuralTextures are shown in Fig. 9. It can be observed that, compared to the baseline, our method shows relatively stable attention regions on the forgery artifacts, demonstrating superiority against the compression.

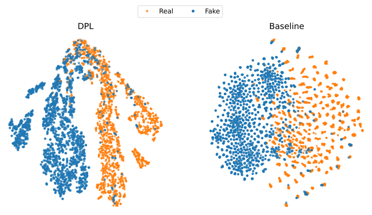

Fig. 8: t-SNE visualization of DPL and baseline.

Feature Visualization. We visualize the fused feature from dual branches using t-SNE algorithm [29]. Fig. 8 shows the visual comparison between DPL

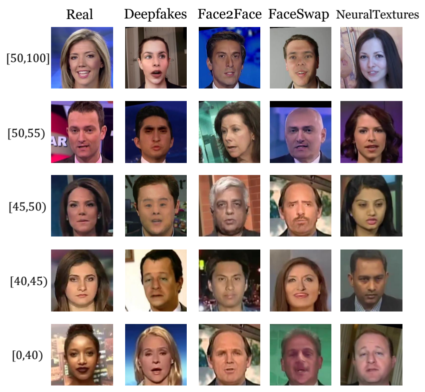

Fig. 10: Analysis Result of VQI on FF++ (c23) dataset

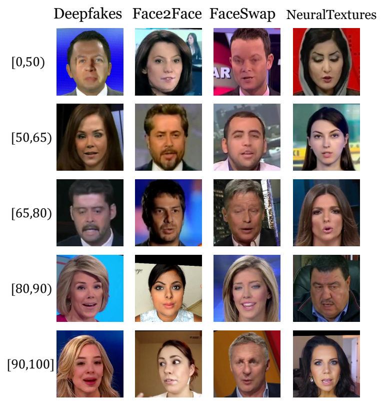

Fig. 11: Analysis Result of FII on FF++ (c23) dataset

and baseline method on FF++ (c23), demonstrating that our method effectively partition these samples into two different clusters while retaining less overlap.

More Analysis on VQI. We show several examples of different ranges of video quality in Fig. 10. Note that we directly employ the prompt text in [40]. Specifically, we divide VQI score into different intervals: [50,100], [50,55), [45,50), [40,45), [0,40), with lower scores indicating worse image quality. These intervals correspond to the K_1 quality labels described in Sec. 3.1. It can be observed that VQI can effectively distinguish images of different quality.

More Analysis on FII. We also show several examples of forgery identifiability in Fig. 11. The FII score is divided into five intervals as well, which are [0,50), [50,65), [65,80), [80,90), [90,100]. Moreover, we have attempted several paired text prompts, and inspected if the indication score matches the degree of manipulation.

5 Conclusion

This paper introduces a new Dual Progressive Learning (DPL) framework for cross-quality DeepFake detection. In contrast to existing methods, we analog this task to the progressive effort of "drilling for underground water". Our method involves two sequential-based branches designed to excavate the forgery traces based on the levels of video quality and forgery identifiability, estimated by dedicated CLIP-based indicators. These branches share a feature selection module that prepares tailored features for different time steps. To train this framework, we propose a two-stage pipeline using the PPO optimization with dedicated objectives. Experimental results validate the effectiveness of our method in detecting DeepFakes across varying video qualities.

Acknowledgement: This work is supported in part by the National Natural Science Foundation of China (No.62402464), Shandong Natural Science Foundation (No.ZR2024QF035), China Postdoctoral Science Foundation (No.2021TQ0314, No.2021M703036).

References

1. Agarwal, A., Ratha, N.: Deepfake catcher: Can a simple fusion be effective and outperform complex dnns? In: CVPR (2024)
2. Aïmeur, E., Amri, S., Brassard, G.: Fake news, disinformation and misinformation in social media: a review. Soc. Netw. Anal. Min. **13**(1), 30 (2023)
3. Binh, L.M., Woo, S.S.: ADD: frequency attention and multi-view based knowledge distillation to detect low-quality compressed deepfake images. In: AAAI (2022)
4. Bordes, F., Pang, R.Y., Ajay, A., Li, A.C., Bardes, A., Petryk, S., Mañas, O., Lin, Z., Mahmoud, A., Jayaraman, B., Ibrahim, M., Hall, M., Xiong, Y., Lebensold, J., Ross, C., Jayakumar, S., Guo, C., Bouchacourt, D., Al-Tahan, H., Padthe, K., Sharma, V., Xu, H., Tan, X.E., Richards, M., Lavoie, S., Astolfi, P., Hemmat, R.A., Chen, J., Tirumala, K., Assouel, R., Moayeri, M., Talattof, A., Chaudhuri, K., Liu, Z., Chen, X., Garrido, Q., Ullrich, K., Agrawal, A., Saenko, K., Celikyilmaz, A., Chandra, V.: An introduction to vision-language modeling (2024)
5. Cao, J., Ma, C., Yao, T., Chen, S., Ding, S., Yang, X.: End-to-end reconstruction-classification learning for face forgery detection. In: CVPR (2022)
6. Chen, H., Lin, Y., Li, B.: Exposing face forgery clues via retinex-based image enhancement. In: ACCV (2022)
7. Chen, L., Zhang, Y., Song, Y., Liu, L., Wang, J.: Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In: CVPR (2022)
8. Chesney, R., Citron, D.: Deepfakes and the new disinformation war: The coming age of post-truth geopolitics. Foreign Affairs (2018)
9. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP (2014)
10. Dai, Y., Gieseke, F., Oehmcke, S., Wu, Y., Barnard, K.: Attentional feature fusion. In: WACV (2021)
11. Ghadiyaram, D., Bovik, A.C.: Massive online crowdsourced study of subjective and objective picture quality. IEEE TIP **25**(1), 372–387 (2016)
12. Guarnera, L., Giudice, O., Guarnera, F., Ortis, A., Puglisi, G., Paratore, A., Bui, L.M., Fontani, M., Coccomini, D.A., Caldelli, R., et al.: The face deepfake detection challenge. Journal of Imaging **8**, 263 (2022)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
14. Kietzmann, J., Lee, L.W., McCarthy, I.P., Kietzmann, T.C.: Deepfakes: Trick or treat? Business Horizons **63**, 135–146 (2020)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
16. Korshunov, P., Marcel, S.: Deepfakes: a new threat to face recognition? assessment and detection (2018)
17. Le, B.M., Woo, S.S.: Quality-agnostic deepfake detection with intra-model collaborative learning. In: ICCV (2023)
18. Lee, S., An, J., Woo, S.S.: Bznet: Unsupervised multi-scale branch zooming network for detecting low-quality deepfake videos. In: WWW (2022)
19. Li, L., Bao, J., Yang, H., Chen, D., Wen, F.: Advancing high fidelity identity swapping for forgery detection. In: CVPR (2020)
20. Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., Guo, B.: Face x-ray for more general face forgery detection. In: CVPR (2020)

21. Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S.: Celeb-df: A large-scale challenging dataset for deepfake forensics. In: CVPR (2020)
22. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. IEEE TPAMI **42**(2), 318–327 (2020)
23. Liu, H., Li, X., Zhou, W., Chen, Y., He, Y., Xue, H., Zhang, W., Yu, N.: Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain. In: CVPR (2021)
24. Liu, J., Wang, J., Zhang, P., Wang, C., Xie, D., Pu, S.: Multi-scale wavelet transformer for face forgery detection. In: ACCV (December 2022)
25. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
26. Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: CVPR (2022)
27. Liu, Z., Wang, H., Wang, S.: Cross-domain local characteristic enhanced deepfake video detection. In: ACCV (2022)
28. Luo, Y., Zhang, Y., Yan, J., Liu, W.: Generalizing face forgery detection with high-frequency features. In: CVPR (2021)
29. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)
30. Nguyen, H.H., Yamagishi, J., Echizen, I.: Capsule-forensics: Using capsule networks to detect forged images and videos. In: ICASSP (2019)
31. Qi, H., Guo, Q., Juefei-Xu, F., Xie, X., Ma, L., Feng, W., Liu, Y., Zhao, J.: Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In: ACM MM (2020)
32. Qian, Y., Yin, G., Sheng, L., Chen, Z., Shao, J.: Thinking in frequency: Face forgery detection by mining frequency-aware clues. In: ECCV (2020)
33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
34. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Faceforensics++: Learning to detect manipulated facial images. In: ICCV (2019)
35. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017)
36. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
37. Shiohara, K., Yamasaki, T.: Detecting deepfakes with self-blended images. In: CVPR (2022)
38. Stefanov, K., Paliwal, B., Dhall, A.: Visual representations of physiological signals for fake video detection (2022)
39. Sun, K., Liu, H., Yao, T., Sun, X., Chen, S., Ding, S., Ji, R.: An information theoretic approach for attention-driven face forgery detection. In: ECCV (2022)
40. Wang, J., Chan, K.C.K., Loy, C.C.: Exploring CLIP for assessing the look and feel of images. In: AAAI (2023)
41. Wang, Y., Lv, K., Huang, R., Song, S., Yang, L., Huang, G.: Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. In: NeurIPS (2020)
42. Wang, Z., Bao, J., Zhou, W., Wang, W., Li, H.: Altfreezing for more general video face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4129–4138 (June 2023)

43. Yan, Z., Luo, Y., Lyu, S., Liu, Q., Wu, B.: Transcending forgery specificity with latent space augmentation for generalizable deepfake detection. In: CVPR (2024)
44. Yan, Z., Zhang, Y., Fan, Y., Wu, B.: UCF: uncovering common features for generalizable deepfake detection. In: ICCV (2023)
45. Zhang, W., Zhai, G., Wei, Y., Yang, X., Ma, K.: Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In: CVPR (2023)
46. Zhou, T., Wang, W., Liang, Z., Shen, J.: Face forensics in the wild. In: CVPR (2021)
47. Zhuang, W., Chu, Q., Tan, Z., Liu, Q., Yuan, H., Miao, C., Luo, Z., Yu, N.: Uia-vit: Unsupervised inconsistency-aware method based on vision transformer for face forgery detection. In: ECCV (2022)