ProFake: Detecting Deepfakes in the Wild against Quality Degradation with Progressive Quality-adaptive Learning

Huiyu Xu[†]
The State Key Laboratory of
Blockchain and Data Security
Zhejiang University
Hangzhou, Zhejiang, P. R. China
huiyuxu@zju.edu.cn

Yaopeng Wang[†]
School of Cyber Science and
Engineering
Southeast University
Nanjing, Jiangsu, P. R. China
yaopengwang@seu.edu.cn

Zhibo Wang[*]
The State Key Laboratory of
Blockchain and Data Security
Zhejiang University
Hangzhou, Zhejiang, P. R. China
zhibowang@zju.edu.cn

Zhongjie Ba
The State Key Laboratory of
Blockchain and Data Security
Zhejiang University
Hangzhou, Zhejiang, P. R. China
zhongjieba@zju.edu.cn

Wenxin Liu
Ant Group
Hangzhou, Zhejiang, P. R. China
wxliu77@gmail.com

Lu Jin
Ant Group
Hangzhou, Zhejiang, P. R. China
lyla.jl@antgroup.com

Haiqin Weng
Ant Group
Hangzhou, Zhejiang, P. R. China
haiqin.wenghaiqin@antgroup.com

Tao Wei
Ant Group
Hangzhou, Zhejiang, P. R. China
lenx.wei@antgroup.com

Kui Ren
The State Key Laboratory of
Blockchain and Data Security
Zhejiang University
Hangzhou, Zhejiang, P. R. China
kuiren@zju.edu.cn

Abstract

Despite the promising advances in deepfake detection on current datasets, detecting visual deepfakes in real-world scenarios (e.g., deepfake videos and live streaming on YouTube) remains a challenge due to the inherent quality degradation such as unpredictable compression employed by social media platforms. Such degradation perturbs discernible forgery clues and diminishes the effectiveness of deepfake detection methods, raising a critical safety concern to the misuse of forgery faces in real-world scenarios. In this paper, we aim to understand the impacts of real-world degradation on the robustness of deepfake detection. Particularly, we investigate the risk of degraded deepfakes towards their detection on two real-world scenarios (i.e., deepfake videos and deepfake live streaming on social media platforms). By measuring the effects of real-world degradations on the performance and representation capabilities of detection models, we reveal that real-world deepfakes can be simulated via common degradation operations (e.g., JPEG compression) as they are perceptually similar to deepfake detectors. By analyzing the training dynamics under different sequences of training samples, we observe that the training order of deepfakes progressing from non-degraded (easy) to heavily degraded (hard) enhances the adaptability of detection models to various degradation in real-world scenarios. Drawing from these observations, we present a novel deepfake detection method ProFake to enhance the robustness of deepfake detection against real-world quality degradations. ProFake enables quality-adaptive learning via progressively degrade, detect and assign weights for the training samples driven by the feedback of model performance and image quality, which ensures that our model gradually focuses on more challenging samples to achieve quality-adaptive deepfake detection. Extensive experiments show that compared with existing methods, ProFake improves deepfake detection accuracy by an average of over 10% in real-world scenarios and by an average of over 30% in heavily degraded scenarios, while maintaining comparable performance in detecting high-quality deepfakes.

CCS Concepts

• **Security and privacy**; • **Computing methodologies** → **Computer vision**;

Keywords

Deepfake Detection; AI Security; Robustness

ACM Reference Format:

Huiyu Xu, Yaopeng Wang, Zhibo Wang, Zhongjie Ba, Wenxin Liu, Lu Jin, Haiqin Weng, Tao Wei, and Kui Ren. 2024. ProFake: Detecting Deepfakes in the Wild against Quality Degradation with Progressive Quality-adaptive Learning. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), October 14–18, 2024, Salt Lake City,*

[†] Huiyu Xu and Yaopeng Wang are co-first authors. An extended version including appendices and supplementary material is available at: CCS-24-ProFake-Google-Drive.
[*] Zhibo Wang is the corresponding author.

CCS '24, October 14–18, 2024, Salt Lake City, UT, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0636-3/24/10
https://doi.org/10.1145/3658644.3690238

UT, USA. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3658644.3690238

1 Introduction

Recently, the ease of access to deepfake tools [2, 22, 24, 55], coupled with their improved realism, has led to a surge in the proliferation of visual deepfakes on social media platforms, including Youtube and Twitter [1, 4]. According to data presented by DeepMedia [3], over 500,000 video deepfakes were circulated on social media in 2023, marking a significant increase from just 14,678 in 2021. However, the malicious use of deepfakes, such as fabricating misleading news about celebrities and orchestrating deceptive live streaming, has emerged as a pressing security concern, with 71% of the population remains unaware of what deepfakes are and only 57% believe they can recognize them [3]. In response to these growing threats, there has been a significant surge in research aimed at developing effective deepfake detection methods [14, 21, 29, 36, 40, 43, 53, 58, 67, 68], which rely on the analysis of different feature signals in the media. The promising detection performance on existing datasets make deepfake detection a mainstream solution for social media platforms to prevent misuse and abuse of deepfakes. As depicted in Figure 1, platforms such as Facebook [6] and YouTube [11] implement built-in deepfake detection mechanisms to tag videos as real or fake prior to public availability. However, once deepfake videos are downloaded, or during live streaming, detecting them becomes challenging as they are subjected to real-world quality degradations induced by lossy operations of platforms. These lossy operations (e.g., compression and downsampling) optimizes transmission speeds, although do not impact human viewing, inadvertently modify the original content of visual deepfakes. These changes, resembling the subtle forgery patterns in deepfakes, causing the detection models fail in identifying degraded deepfakes. Consequently, a safety concern arises, as malicious users may attempt to evade deepfake detection by subjecting their deepfakes to quality degradations introduced by social media platforms, which has not been studied sufficiently.

In this paper, we study the impact of real-world degradation on the robustness of deepfake detection, with a focus on the degradation brought by the social media platforms, where deepfakes undergo various unknown and complex quality degradations in different platforms. Specifically, we systematically evaluate the risk of degraded deepfakes towards their detection under two typical real-world scenarios: deepfake videos and deepfake live streaming. By measuring the effects of the real-world degradation on the performance and representation capabilities of detection models, we find that for the detectors trained by high quality deepfakes, the fine-grained forgery patterns they capture are largely perturbed by real-world quality degradation, causing a significant drop in the performance of detecting deepfakes in the wild. Additionally, to understand how well we can address the real-world degradation challenge with off-the-shelf typical degradations (e.g., JPEG compression), we first measure the perceptual similarity of real-world degrdation and common degradation and reveal that the real-world degradation share great similarity to typical degradation in the representation space of deepfake detectors. Therefore, we can approximate the task of detecting deepfakes in the wild as detecting deepfakes that undergo a wider range of, but already known, common degradations. To further understand why is it difficult to obtain a robust deepfake detector against diverse quality degradations, we explore the impact of inconsistency between deepfakes with varying degradations on detection accuracy. By analyzing the training dynamics of three training sample sequences (i.e., shuffled order, hard-to-easy training, and easy-to-hard training), we find that inconsistent learning dynamics degrade the performance of detecting degraded deepfakes and that the training order from pristine samples to increasingly degraded samples can alleviate this performance degradation.

Upon such understandings, we delineate the following inherent challenges in training robust deepfake detection models against quality degradation: Firstly, the effectiveness of a deepfake detector hinges on its ability to discern the fine-grained discrepancies between real and fake images. However, quality degradation perturbs these subtle clues, thereby compromising the detector's effectiveness. Secondly, existing public deepfake datasets fail to encompass the full spectrum of quality degradation. While manually employing data augmentation technique [58] can somewhat alleviate this problem, there remains a notable gap in diversity. Thirdly, when training samples with different degrees of degradation are incorporated into training, although the overall distribution of the data remains unchanged during training, the inconsistent feedback from samples with varying quality degradation make it difficult for the detector to adapt to different degraded samples. Therefore, the key for enhancing the robustness of deepfake detection models against quality degradation is to harmonize learning signals from samples of varying quality levels to effectively capture the quality-adaptive forgery patterns.

To address the above challenges, we propose ProFake, a novel deepfake detection method that progressively learns the quality-adaptive features from pristine to degraded samples. We achieve this quality-adaptive learning by progressively modifying two key aspects of training samples: the degree of degradation and the weights for learning this sample. To achieve the first key aspect, we design a learnable quality degradation generator to generate degraded samples during training in an adversarial manner under the guidance of the current learning feedback. To achieve the second key aspect, we adjust the learning focus by assigning adaptive weights through an additional learnable adaptive sampling network. The adaptive sampling network determines the weights of training samples by promoting learning of more challenging samples and reducing attention to easy samples, maintaining the optimal learning difficulty of degraded samples in a self-paced manner.

Through progressively degrade and assign weights to the training samples, ProFake enables detectors to maintain the performance for detecting deepfakes in the wild while preserving the performance on high-quality deepfakes. We validate the effectiveness of ProFake through extensive evaluations across three popular deepfake datasets (i.e., FF++ [56], DFDC [26] and Celeb-DF [45]) that undergo a range of quality degradations, as well as real-world deepfake datasets we collected in two scenarios (i.e., deepfake videos and deepfake live streaming on social media platforms). Our results show that ProFake has consistent detection performance for deepfakes of varying quality. To the best of our knowledge, this is the

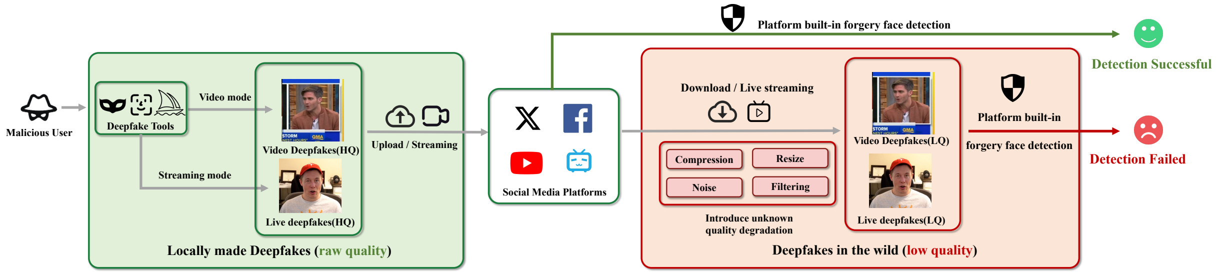


Figure 1: Threats of quality-degraded deepfakes in real-world scenarios to deepfake detectors.

first method for deepfake detection models to satisfy goals in identifying deepfakes under a variety of real-world quality degradations, while also sustaining high performance in detecting high-quality deepfakes.

Contributions. We summarize the contributions as follows:

- We conduct a systematic evaluation of the effects of real-world quality degradation on deepfake detection, focusing on two scenarios (i.e., deepfake videos and deepfake live streaming on social media platforms). Our results reveal that fine-grained forgery patterns, which deepfake detectors rely on for detection, are disrupted by quality degradations, resulting in a significant performance drop in detecting degraded deepfakes.
- We introduce ProFake, a novel deepfake detection method that progressively adjusts both the applied degradation and weight of the training samples to achieve quality-adaptive deepfake detection. By incorporating training dynamics relevant to quality, we can balance the inconsistent feedback arising from samples of varying quality. With the progressive degradation dynamics of training samples, we enhance the robustness of deepfake detection models against quality degradation.
- Extensive evaluations show that ProFake enhances deepfake detection accuracy by more than 10% in real-world scenarios (i.e., deepfake videos and deepfake live streaming across 6 mainstream social media platforms) and by over 30% in heavily degraded scenarios, while preserving performance on high-quality deepfake detection that are on par with leading state-of-the-art methods.

2 Background and related work

In this section, we first introduce the quality degradation on facial forgery in real-world scenarios. Then we discuss the representative techniques for facial forgery detection methods.

2.1 Facial Forgery in the Wild

Facial Forgery, commonly known as deepfakes [15, 51, 59, 74], aims to manipulate the appearance and actions of individuals in the media (video or image), often without their consent, which can lead to potential misuse in various malicious contexts, including telecommunications fraud [7] and smearing celebrities [8]. Deepfake techniques can be divided into four categories [13, 63]: face synthesis [23, 46], attribute manipulation [50, 72], face swapping [39, 49], and face reenactment [17, 28]. Among them, face synthesis technology cannot generate high-quality, temporally coherent videos, making them easily recognizable by humans, and attribute manipulation technology primarily aims to beautify the target image, thus posing a limited threat. Therefore, in this paper, we focus on detecting deepfakes created by face-swapping [59, 71] and face reenactment [15, 34], which are widespread across online social platforms with considerable misuse threats.

Both techniques synthesize the required media by combining a target image, which provides identity information, with a driving source that supplies background and motion.

With the rapid advancement of generative models, facial forgery technologies have evolved significantly from the initial use of generative adversarial networks (GANs) [30] to the more recent diffusion models [33], advancing the creation of photorealistic deepfakes. Concurrently, the advent of more accessible real-time forgery capabilities that tolerate acceptable quality degradation has diminished the need for high-quality media materials, and significant computational resources that were previously essential. The individuals can create deepfakes using just a single image and the tool API of online products like SwapFace [2]. Such trend enables cheap deepfake creation, making the technology more adaptable for different users yet increasingly difficult to regulate. Regulators face challenges in detecting low-quality deepfakes, which can be easily created and spread by malicious actors with minimal resources. Simultaneously, they must maintain performance in identifying high-quality deepfakes that are meticulously crafted by well-resourced malicious actors. Moreover, in real-world scenarios, deepfakes often undergo various forms of degradation due to lossy process such as transmission errors, re-encoding and re-compression [54, 69], further strengthening the need to detect deepfakes with complex and unkonwn quality degradation. Below we present the typical methods to characterize and simulate quality degradation.

Typical Degradation Modeling. To estimate the degraded image, typical degradation models apply a series of operations to the raw image, including Gaussian blur, downsampling, Gaussian noise introduction, and JPEG compression. Such process can be formulated as follows: a pristine deepfake image x is first convolved with a blur kernel k, introducing a certain degree of blur. Subsequently, a downsampling operation is applied, reducing the resolution of the image by a scale factor s. Noise n is then introduced to the downsampled image. Finally, JPEG compression is also adopted, further degrading the image quality with a quality factor c. Thus the typical degradation models can be represented as:

$$x' = \text{JPEG}\left((x \oplus k) \downarrow s + n\right)_c , \tag{1}$$

where x' represents the degraded image, $\oplus$ denotes the convolution operation, $\downarrow s$ represents the downsampling operation with scale factor s, and c represents the quality factor of JPEG compression.

Degradation Settings. To facilitate a controlled degradation process, the typical degradation model can be parameterized as degradation setting S. For the blur, we typically model it as a convolution with a Gaussian blur filter, which can be characterized by kernel size k. For the downsampling, we consider the nearest-neighbor interpolation and characterize the process using s, which represents the scaling factor for downsampling. For the noise, we consider the commonly-used zero-mean additive white Gaussian noise. Such noise can be described using α to represent the intensity of the noise as a percentage, which determines the standard deviation of the Gaussian distribution. For JPEG compression, the quality of compressed images is determined by a compression factor c. Thus, the degradation setting S can be represented as illustrated in Equation 2, encompassing the four degradation types discussed.

$$S = (k, s, \alpha, c). \tag{2}$$

Real-world Degradation Modeling. Modeling real-world quality degradations poses a significant challenge and remains unexplored due to the difficulty of estimating unknown degradation. In this paper, we focus on the quality degradation induced by the transition via social media platforms, represented by social networking platforms such as Facebook [6] and online video platforms such as Youtube [11]. Many studies [54, 61, 69] have shown that different social media platforms employ platform-specific lossy operations, including format conversion, resizing, enhancement filtering, and compression, aiming to reduce bandwidth consumption and storage space for multimedia content. While specific mechanism employed by platforms are not publicly available, research [54] that investigates the image compression practices on platforms like Facebook has revealed some of its potential mechanisms (e.g., Facebook resizes images when their longest dimension exceeds 2048 pixels).

2.2 Facial Forgery Detection

Facial forgery detection techniques aim to identify whether the media (image or video) contain forgery faces, relying on the analysis of various feature signals extracted from the media [40, 41, 58, 69], and is commonly implemented by social media platforms to avoid malicious spread of deepfakes. In response to the escalating security threats associated with deepfakes, a variety of detection techniques have been proposed to effectively identify them [16, 19–21, 27, 35, 37, 44, 47, 60, 68, 78]. Most of these studies aim to improve the performance against forgery types that are not represented in the training dataset [27, 31, 73]. They mainly focus on capturing universal semantic visual artifacts in deepfakes within the spatial domain [14, 21, 36, 40, 58], or leveraging frequency information for deepfake detection [29, 43, 53, 67, 68]. Meanwhile, several studies exploit temporal inconsistencies to detect deepfakes, such as inconsistent facial movements [52, 77] and audio-visual discrepancy [32]. However, the lack of consideration of image quality makes such methods sensitive to image degradation (e.g., noise and compression), leading to significant drop in performance when identifying degraded deepfakes in real-world scenarios.

Low-quality Deepfake Detection. To detect low quality deepfakes in real-world scenarios, BZNet [42] designs a multi-scale super-resolution network to enhance the quality of the detected image. In contrast to methods that augment input samples, several studies explore quality-agnostic features of deepfakes. Cao *et al.* [18] propose a dual-branch network to learn common features of paired images at different compression levels. ADD [70] applies frequency domain learning and optimal transport theory to knowledge distillation, assisting the student model in learning discriminative features from low-quality images to enhance the detection rate of low-quality deepfake images. QAD [41] introduces an intra-model collaborative learning framework that maximize the dependency between intermediate representations of images from varying quality levels to improve the performance on the detection of low quality deepfakes. However, the aforementioned methods show high performance within a single, specific quality degradation and struggle with diverse or unknown degradations. In this paper, we examine the susceptibility of deepfake detectors to complex, unknown real-world quality degradations (see Section 3.2 to 3.3) and investigate how to preserve detector performance across a spectrum of quality degradations (see Section 3.4).

3 Risk of degraded deepfake detection

To better understand the risk of real-world quality degradations towards deepfake detection, we first introduce the threat model for degraded deepfake detection in Section 3.1. Then we present a measurement study aiming to address the three key research questions (RQs):

RQ1: How does real-world quality degradation impact the performance of deepfake detection methods? (Section 3.2)

RQ2: Can we simulate unknown real-world deepfakes with typical degradations? (Section 3.3)

RQ3: Why is it difficult to obtain a robust deepfake detector against diverse quality degradations? (Section 3.4)

3.1 Threat Model

Adversary Goals. As depicted in Figure 1, the adversary aims to perform real-world quality degradations on created or collected deepfakes via real-time transmission or uploading to social media platforms in order to bypass the deepfake detection and maliciously spread deepfakes. To increase the evasion rate, the adversary can also add common degradations (i.e., JPEG compression, Gaussian noise, Gaussian blur, and downsampling).

Adversary Capabilities. Given the high computational resources and extensive media materials required to train a deepfake model for a specific individual, we posit that the adversary is limited to using publicly available deepfake models or accessing online deepfake tools via application programming interface (API). Furthermore, the internal mechanisms of quality degradation enforced by social media platforms, such as specific image size thresholds that trigger compression, remain undisclosed to the public. Consequently, the adversary is unable to manipulate the input to specifically trigger specific degradation mechanisms on any given social media platforms. The adversary's actions are confined to a single upload or transmission, and they lack the capability to add sophisticated adversarial noise aimed at evading deepfake detection models. Instead, their local modifications are restricted to simpler forms of degradation, commonly used in typical degradation models.

3.2 Real-world Degradation Impacts on Deepfake Detection (RQ1)

To evaluate the robustness of the state-of-the-art deepfake detectors against quality degradations in real-world scenarios, we create two deepfake datasets aiming to study two representative scenarios: deepfake videos and deepfake live streaming, covering both the face-swapping and face reenactment deepfakes in real-world scenarios. In these scenarios, deepfakes are subject to quality degradation when uploaded or broadcast to platforms due to transmission limitations. Therefore, we collect deepfakes with these real-world degradation from mainstream platforms through the above operations. With the collected datasets, we then comprehensively evaluate how the degradation effects of real-world social media platforms affect the performance and representation capabilities of deepfake detectors in both in-domain and cross-domain evaluation settings.

Deepfake Detectors. We evaluate four mainstream deepfake detectors: F^3Net [53], MAT [75], EfficientNet-b4 [62] and Xception [56]. Among them, F^3Net aims to identify abnormal frequency components within deepfakes. The other three detectors detect crucial clues in fake images by analyzing spatial features such as textures and patterns introduced by deepfakes. Each detector is trained on the predominant deepfake dataset, FF++ [56], which comprises 1,000 source videos (i.e., real videos) and 5,000 manipulated videos (i.e., fake videos) created using various deepfake methods.

Video Dataset. Considering that real-world degradation is complex and unknown, compared to a single off-the-shelf transformation that can only simulate certain aspects of real-world degradation, we collect deepfake videos from various social media platforms (i.e., Facebook [6], Twitter [9], WeChat [9] and Weibo [10]) through an uploading and downloading process, which undergo actual real-world degradation.

According to [54, 69], social media platforms apply varying degradations that are mainly triggered by resizing strategies for images with large size. For instance, Facebook resizes images when their longest edge exceeds 2048 pixels. Therefore, we stitch samples into upload examples to avoid the uploaded image resolution from being too small to trigger resize strategies of these platforms. We obtained degraded samples by uploading and subsequently downloading these images, after which we split them to align with the unstitched samples that were initially uploaded.

We observe that if the number of stitched images is too large, the images may be severely degraded and lose most of the facial information, resulting in ineffective face detection. For example, when a high-resolution image (5120×5120 pixels, 31.7MB) is uploaded to the Weibo, it is downsized to just 1080×1080 pixels and compressed to a mere 489KB, resulting in a loss of critical facial information that even cannot be discerned by the human eye. Therefore, we need to set rules to determine the number of splicing of uploaded images in collecting our video dataset. We took the change in pixel values to roughly measure the extent of quality degradation after processing by platforms, which is computed as follows: 1) resize the degraded image to match the dimensions of the original image; 2) calculate the pixel difference between the degraded image and the original image; 3) calculate the proportion of non-zero differences to the total number of pixels. We consider images with less than 75% pixel value change retain sufficient facial details for detection. To meet this rules for our uploaded images, We initially tried uploading images with a larger number of stitching, such as a grid of 20×20 tiles. We then analyzed the processed images to measure the percentage of pixel value change for the split image. If the change remains below 75%, we collected the corresponding data, otherwise we reduced the number of tiles to decrease the overall image size until the pixel value change is under 75%. Thus we obtained the size of uploaded images for different platforms as shown in Table 1, where the grey line shows the selected setting for uploaded images.

Table 1: The changes of images of different sizes uploaded and downloaded on multiple platforms. The gray lines represent the upload settings of different platforms used in our dataset.

Platforms	Size of upload images	Size of download images	Percentage of pixel value change	Storage of upload images	Storage of download images
Twitter	**5120x5120**	**4096x4096**	**74.6%**	**31.7MB**	**2.05MB**
Facebook	5120x5120	2048x2048	82.0%	31.7MB	843KB
	1280x2048	**1280x2048**	**65.3%**	**3.04MB**	**321KB**
WeChat	5120x5120	1280x1280	93.0%	31.7MB	204KB
	1280x2048	**1280x2048**	**72.4%**	**3.04MB**	**201KB**
Weibo	5120x5120	1080x1080	94.0%	31.7MB	489KB
	1280x2048	690x431	81.9%	3.04MB	132KB
	1024x1024	**690x690**	**74.5%**	**1.28MB**	**151KB**

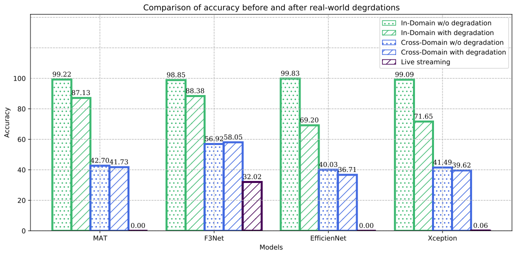


Figure 2: Accuracy of detection models on datasets that before and after real-world quality degradations.

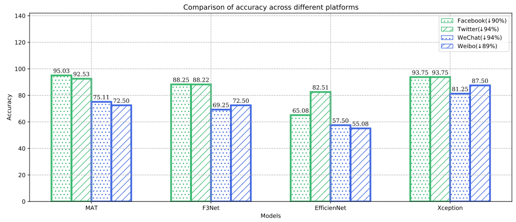


Figure 3: Accuracy of detection models on in-domain setting of the video dataset from different platforms, annotated with the percentage of storage change per platform.

Considering the correlation between the performance of deepfake detectors and their training samples, it is crucial that our dataset minimizes the impact of cross-domain challenges on evaluations. Thus our data for uploading consists of 2000 frames, randomly sampled from four datasets (i.e., FF++, DFDC, Celeb-DF and FaceShifter), each contributing 400 frames with size of 256 × 256. The dataset is then split into two parts: in-domain settings (using the 400 processed frames from FF++), cross-domain settings (using the 1600 processed frames from the other three datasets).

Live Streaming Dataset. To simulate the real-world scenarios in deepfake live streaming, we consider two typical live broadcast forms: recorded broadcast and real-time live broadcast. Real-time live broadcast requires lightweight Deepfake technology, while recorded broadcast usually does not have this limitation. We used

SwapFace [2] as the deepfake tool to create deepfakes locally and then broadcast them on two major live streaming platforms: YouTube [11] and Bilibili [5]. The SwapFace tool provides multiple off-the-shelf face-swapping deepfake models across various identities. To ensure the diversity of our collected live streaming videos, we deployed four individuals as the source faces and pair them with four randomly chosen identities provided by SwapFace as the targeted face templates.

- **Pre-recorded video:** We created video deepfakes using SwapFace's video mode as a recording source and broadcast them to the platform along with 5 videos included in the FF++ dataset to add real-world quality degradation. We then screen-recorded these videos and collected 12,000 frames.
- **Real-time deepfake live streaming:** We created deepfake live broadcast sessions by performing real-time face swapping through camera mode of SwapFace. Then we screen recording these live streaming videos and collected 20,000 frames.

In total, the live streaming dataset contains more than 32,000 frames of deepfakes from two major video live streaming platforms, covering both face swapping and face reenactment deepfakes. Considering that mainstream deepfake detectors do not incorporate deepfakes created by real-world deepfake tools into their training, the Live Streaming Dataset is only evaluated in a cross-domain setting.

Metrics. Aligned with previous studies [18, 25, 41, 44], we adopt the accuracy score to measure the overall ratio of correctly identifying deepfakes and real images.

Results. In Figure 2, we compare the performance degradation of different deepfake detectors on degraded deepfakes under our collected two real-world datasets. We further visualize the latent space of a deepfake detector via t-SNE visualization [64] to demonstrate the degradation effects introduced by real-world social media platforms, as shown in Figure 5 (a).

- **Overall performance drop brought by real-world degradations.** It is evident that the performance of deepfake detectors degrades to varying extents when faced with unknown real-world degradations. The detector MAT and F^3Net exhibit a performance decline of over 10%, while EfficientNet and Xception experience a drop of over 20% in the in-domain setting of the video dataset. From the visualization of latent space as shown in Figure 5 (a), we can observe that real-world quality degradations can disrupt well-trained deepfake detectors. As the clusters representing real and fake samples converge within this space, distinguishing between them becomes more challenging. Therefore, we can conclude that the main challenge in detecting deepfakes in the wild is that **quality degradation perturbs subtle details of the forgery, rendering these patterns undetectable and significantly reducing the effectiveness of deepfake detection methods**.
- **The impact of unseen samples on performance under quality degradation.** We can observe that there is a consistent and significant drop in detection accuracy when models are tested in a cross-domain and degraded quality scenario compared to an in-domain and degraded quality scenario. This indicates that the models are less robust to domain changes and quality degradation simultaneously. Given their already diminished performance on cross-domain video datasets, these models see a relatively smaller decline, with accuracy below 50%. In the case of the more challenge live streaming dataset, all detectors struggle, with none achieving an accuracy higher than 50%.
- **Degradation effects introduced by different platforms.** The results for detection models on different platforms are shown in Figure 3. Storage changes vary between platforms, primarily due to their different compression and resizing strategies. We find that the detection performance is not strictly correlated with changes in storage or pixel values; for instance, Weibo exhibits the smallest storage change yet results in the lowest detection performance among the four platforms.
- **Robustness of different detectors.** We find that methods that exhibit overfitting (as evidenced by models like Xception, which achieve near 100% accuracy on the raw dataset) suffer greater performance declines, exceeding 30% when confronted with real-world quality degradations. Further, we observe that detectors with better cross-domain performance are also more resistant to quality degradations, which indicates **a positive correlation between a detector's ability to generalize across different data domains and its robustness to variations in quality**.

3.3 Perceptual similarity between real-world degradation and typical degradation (RQ2)

To mitigate the performance drop in detecting degraded deepfakes, it is important to incorporate the degradation into training samples to improve the robustness of detectors. However, estimating the real-world degradations employed by social media platforms poses significant challenges, especially given the high cost of acquiring suitable training data from varying platforms to train robust deepfake detectors. Consequently, a more practical approach is to apply data augmentation on the available deepfakes with already known common degradation operations to improve the robustness of detectors, which motivates us to investigate whether these operations can effectively simulate real-world degradations. We make a hypothesis: **from the viewpoint of deepfake detectors, real-world degradation constitutes a subset of the alterations produced by common degradation operations.** That is to say, within the representation space, real-world degradations can be encompassed by carefully crafted common degradations. To verify such hypothesis, our study evaluate the perceptual similarity between real-world degradation and typical degradation (i.e., Gaussian noise, Gaussian blur, JPEG compression and downsampling).

Detection Models. We examine the latent space of the representative deepfake detector Xception, which is trained on FF++ dataset. To obtain the feature representation, we select the final convolutional layer from Xception as the representation layer.

Dataset. Consequently, we apply common degradation to each source following the degradation settings. To compare real-world deepfakes with those affected by common degradations, we randomly selected 200 images from the Video Dataset, representing deepfakes with real-world quality degradations, along with their original source images. Then we randomly generated degradation settings, varying kernel size $k \in 1, 3, 5$, noise scale $\alpha \in [0, 10]$, downsampling scale $s \in [0.5, 1]$, and JPEG compression quality factor $c \in [10, 100]$. These degradation settings were then used to apply common degradations to each source image accordingly.

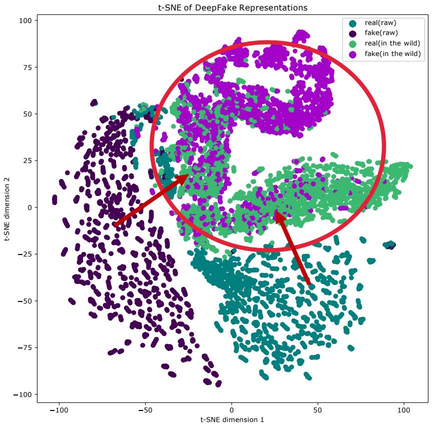


Figure 4: Comparison of latent spaces of deepfake detectors under real-world degraded and non-degraded samples.

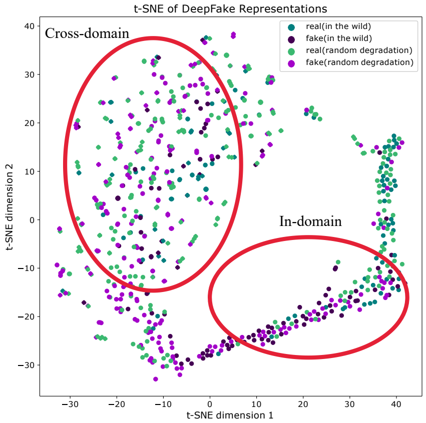


Figure 5: Comparison of latent spaces of deepfake detectors under real-world degraded and random typical samples.

For each source image x, we obtain one corresponding real-world degradation one x_r and 5 random degradation one $x_{d_0}, x_{d_1}, \ldots, x_{d_5}$.
Metrics. To measure the perceptual similarity of deepfakes undergoes real-world degradations and common degradations from the perspective of deepfake detectors, our study focuses on two primary indicators of disparity: similarity and overlap of representations in the latent space of deepfake detectors. We employ cosine similarity as the metric to calculate similarity between each real-world degraded sample and its randomly degraded counterpart, given the widespread use of this metric in tasks related to facial analysis. The cosine similarity is linearly normalized to a range between 0 and 1, with values approaching 1 indicating a higher degree of similarity. For measuring the overlap of representations, we employ t-SNE visualization to examine how extensively the representation clusters from both groups intersect.
Results. Our study depicts the distribution of similarity for varying data source, contrasting real degradation with randomly applied common degradation via cumulative frequency plots in Figure 6. Then we visualize the latent space shown in Figure 5. From Figure 6, it is clear that a small proportion of samples have similarity scores below 0.3. Instead, a substantial majority cluster around 0.6, as highlighted by a sharp incline in the curve at this value, signifying that samples subjected to random degradation possess considerable perceptual resemblance to those degraded by real-world conditions. The latent space visualization of random-degraded and real-world degraded samples reveals that **expanding the range of random degradation encompasses the previously unseen real-world degradation scenarios**. This coverage extends within the latent space of detectors for both in-domain and cross-domain settings.

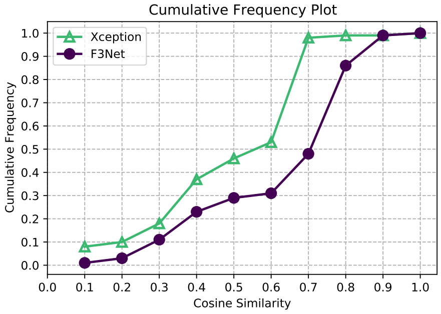


Figure 6: Frequency accumulative plots for the cosine similarity between randomly degraded and real-world degraded samples.

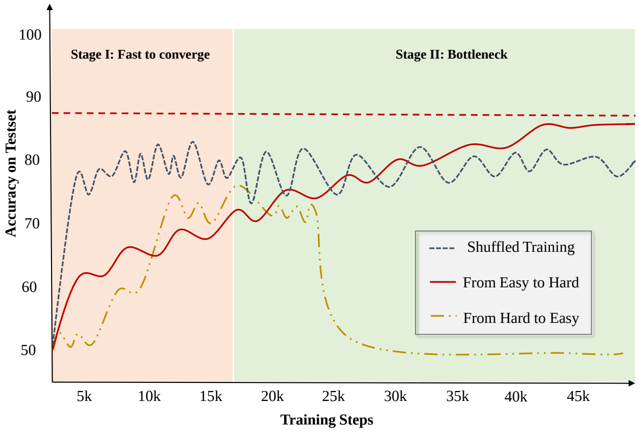


Figure 7: Training dynamics of detection models on degraded deepfake datasets: We train the Xception model with different training orders on a degraded form of the FaceForensics++ dataset [56], which has been subjected to a range of degradation. In this test, we measure three types of training orders: shuffled order, from easy (non-degraded) to hard (heavily degraded) and from hard to easy.

Upon such findings, we verify that **the unknown complex degradation induced by real-world social media platforms can be simulated via common quality degradations** so that our goal of detecting deepfakes in the wild can be switched to detecting deepfakes with a wide range of common but already known degradation.

3.4 Training Dynamics of Deepfake Detectors against Quality Degradations (RQ3)

Upon the findings in Section 3.3, we can examine the degradation effects towards deepfake detectors via analyzing the deepfakes with typical degradations. Thus our goal is to find out why the robust deepfake detector is hard to train against various common quality degradations. Our results in Section 3.2 have shown that deepfake detectors trained on high quality data face challenges in identifying deepfakes when tested across varied quality levels. This observed discrepancy in performance of high and low quality samples implies that the learning signals differ for deepfakes of various qualities, potentially leading to inconsistencies during the training process. Thus, we make a hypothesis that such **inconsistency may hinder the adaptability of deepfake detectors to deepfakes across different quality domains, leading to difficulty in training a robust detector against varying degradations.** Our study aims to explore the impact of such inconsistency on detection accuracy on testsets with a certain degree of common degradations. To incorporate different types of training inconsistencies during training, we tested the impact of different training orders of the data (sorted by quality) on the overall accuracy. By analyzing the training dynamics under different orders (i.e., different levels of inconsistency), we reveal the impact of this inconsistency on robustness.

Detection Models. As outlined in Section 3.3, we adopt Xception as the representative deepfake detector.

Datasets. Taking the FF++ dataset as data source, we impose three types of degradation on the dataset. By assigning different degradation settings to individual training samples, we construct multiple training sequences and arrange them according to the degree of quality degradation, thereby introducing controlled diversity in the training process. We take a mixture of three quality type deepfakes provided by FF++ (i.e., raw quality, high quality and low quality) as the source of test dataset to better reflect the overall performance of the detector in detecting deepfakes under varying quality degradation. Since the above three qualities only include image compression degradation, we further add 3/4 to 1 downsampling operation to the High quality test data set to obtain the medium quality test set, and add 0-10% random noise, 1/2 to 1 downsampling and Gaussian filtering operation with random kernel size to the Low quality test data set. The number of data in the three test sets of raw quality, medium quality and Low quality is the same.

• **Random Degradation (Shuffled Training)**: We replicate the degradation setting outlined in Section 3.3 to evaluate the shuffled training order, where the degraded samples are randomly generated in a predefined range.

• **Degradation from easy to hard**: We increase the level of quality degradation from non-degraded to heavily-degraded samples at a uniform rate during training. Take compression as an instance: the quality factor c is progressively decreased from 100 to 10, with one decrement at every 1/90 *th* interval of the total training steps.

• **Degradation from hard to easy**: Contrary to degradation from easy to hard, this sequences downgrade the level of quality degradation in a uniform rate.

Results. Below we will show the results of the test set after the three qualities are mixed in Figure 7.

• **Stages in Learning a Robust Detector:** We observe that the learning process can be divided into two stages. Initially, the model demonstrates rapid convergence, suggesting an ease in acquiring forgery detection knowledge from high-quality samples. However, as training advances, the model reaches a plateau phase, where further performance improvements become challenging. This phase could potentially lead to unstable learning, and in some cases, catastrophic forgetting of previously acquired knowledge. Based on such observations, we can see that the inconsistency of learning signals between samples with different quality starts to occur in the plateau phase, which determines the final prediction accuracy on deepfakes with diverse quality degradations.

• **Impacts of Training Sequences:** Our observations reveal that while the overall data distribution remains the same, the large variations between different quality levels pose a challenge to shuffle training, leading to instability. However, progressively increasing the difficulty of training samples (from easy to hard) can stabilize learning and address this challenge to some extent, with improved performance compared to shuffle training.

Therefore, we conclude that the core challenge in training robust deepfake detectors to cope with real-world degradation problems lies in the difficulty of alleviating the training inconsistency between degraded and non-degraded samples. We find that organizing the training samples well according to the degree of quality degradation can solve the above inconsistency problem to some extent.

4 ProFake

Based on the above observations, in this section, we propose our method, ProFake, which aims to achieve robust deepfake detection against real-world degradations via progressive quality-adaptive learning. We first introduce the overview of our methodology in Section 4.1. We introduce the details of each stage within ProFake in Sections 4.2 to Section 4.4, and then introduce the overall training objective of ProFake in Section 4.5.

4.1 Overview

Motivated by the observations in Section 3, our design goal is to develop a training method that can alleviate the inconsistency of learning signals caused by differences in the quality of training samples, thereby enhancing the quality adaptability of deepfake detection models. Therefore the core idea of ProFake is to automatically determine the optimal order of degraded samples during training via progressive quality-adaptive learning. We achieve this quality-adaptive learning by gradually modifying two key aspects of the training samples: the degree of degradation and the weights for learning this sample. To achieve the first key aspect, ProFake gradually facilitates a learnable generator to generate samples that are gradually degraded during training in an adversarial manner guided by the current learning feedback (i.e., the model performance and the quality of samples). To achieve the second key aspect, ProFake adjusts the learning focus by assigning adaptive weights through an additional learnable sampling network. Such adaptive weights automatically exploit the current learning feedback to ensure that the model learns samples with appropriate degradation. By automatically adjusting samples from easy to hard based on the above two key designs, the detector gradually learns quality-adaptive features to adapt to a wide range of quality degradation. Since the proposed ProFake focuses on modifying the training samples and their weights, it has good scalability and is not limited to any specific detection method. For conciseness, we will focus on spatial-domain deepfake detectors when introducing our method. If a frequency domain deepfake detector is adopted, a preprocessing step needs to

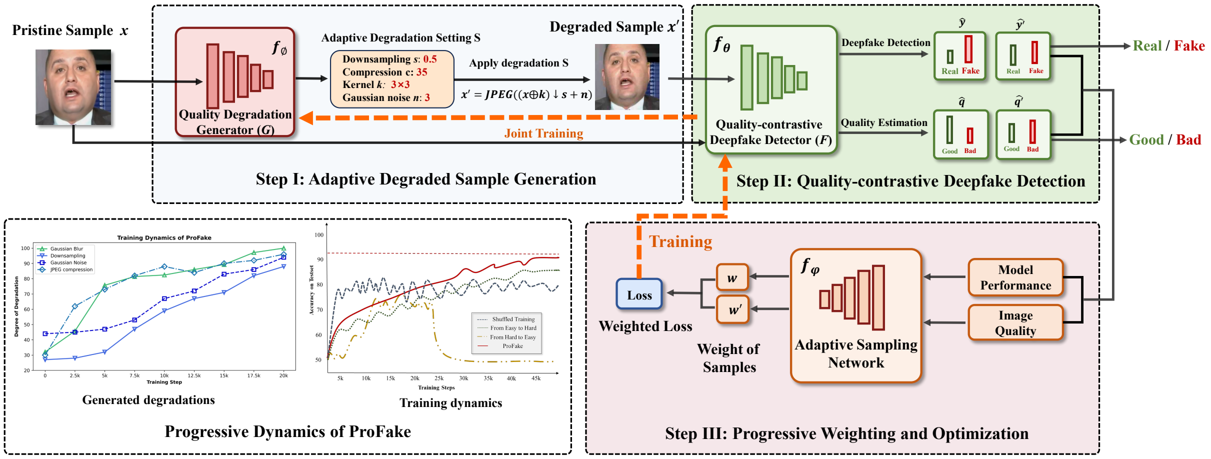


Figure 8: The overall training pipeline of ProFake. The lower-left corner of the figure illustrates the progressive training dynamics of ProFake, showcasing the evolution of degradations generated by G, alongside the detection accuracy on test sets with mixed quality degradations, in contrast with three alternative training sequences.

be performed to transform the image and extract frequency domain information before progressive quality-adaptive learning, such as performing a discrete cosine transform (DCT) [12].

As depicted in Figure 8, from a high-level overview, ProFake contains three stages: Adaptive Degraded Sample Generation, Quality-contrastive Deepfake Detection, and Progressive Weighting and Optimization. Here is the training loop of ProFake:

(i) **Adaptive Degraded Sample Generation.** Given a pristine sample x, the Quality Degradation Generator G generates a degradation setting S. By applying S to x, we obtain the degraded sample x'.

(ii) **Quality-contrastive Deepfake Detection.** The image pair $\{x, x'\}$ is fed into the Quality-contrastive Deepfake Detector F for pairwise learning. F performs dual tasks of deepfake detection and quality estimation, yielding two outputs: the predicted probabilities $\{\hat{y}, \hat{y'}\}$ determining whether the image is real or fake, and the quality scores $\{\hat{q}, \hat{q'}\}$ evaluating the overall quality of the image.

(iii) **Progressive Weighting and Optimization.** The Adaptive Sampling Network collects the feedback from the detector's performance $\{\hat{y}, \hat{y'}\}$, and the image quality $\{\hat{q}, \hat{q'}\}$. Utilizing this feedback, the Adaptive Sampling Network employs a convolutional network f_φ to determine the sample weight w. This weight is then used to compute a weighted loss, which facilitates subsequent updates to both the Quality Degradation Generator G and the Quality-contrastive Deepfake Detector F. The lower-left corner of Figure 8 shows the progressive learning behaviours during training. As the training progresses, the degradation level generated by the Quality Degradation Generator G (for clarity, degradation is linearly normalized between 0 and 100) and the performance on testset with diverse quality show a progressive improvement. After the training, we obtain a quality-adaptive deepfake detector F, which is then employed for inference.

4.2 Adaptive Degraded Sample Generation

To enable diversify and controllable degraded sample generation, ProFake employs a Quality Degradation Generator (G) implemented by a convolutional neural network parameterized by f_ϕ, to facilitate controlled and adaptive degraded sample generation. Specifically, G processes the pristine sample x to determine the degradation setting S. To avoid generating overly degraded samples that are unlikely to occur in practical scenarios, we constrain the kernel size k to the set $\{1, 3, 5\}$, the downsampling scale s to $[0.5, 1]$, the noise scale α to $[0, 10]$, and the compression scale c to $[10, 100]$. Concurrently, we initialize f_ϕ with a relatively low value by using a small-scale normal distribution, ensuring that the outputs remain close to zero in the early training, which facilitates a more effective starting point for the generation of non-degraded samples. The network f_ϕ is designed to generate specific values for s, α, and c. For the kernel size k, f_ϕ yields three-class probabilities, corresponding to the kernel sizes of 1, 3 and 5, respectively. Subsequently, the Quality Degradation Generator applies degradation operations to the pristine training sample x according to the generated degradation setting S, yielding the degraded sample x' as formulated in Eq. 1. For Gaussian blur degradation operations, we employ a differentiable technique by generating a suite of Gaussian kernels corresponding to all potential values of k. The predicted labels are then utilized to compute a weighted sum of these kernels, which serves as the effective, differentiable kernel for the sample.

4.3 Quality-contrastive Deepfake Detection

The Quality-contrastive Deepfake Detector (F) identifies the degradation in input samples and performs deepfake detection on them implemented by a backbone network parameterized by f_θ. By effectively capturing the degradation, F can adaptively discern deepfakes across varying image quality. Specifically, F extracts representations $f_\theta(x), f_\theta(x')$ from x and their degraded counterparts x', respectively. These representations are then fed into a classification head to predict deepfake labels $\hat{y}, \hat{y'}$ and quality head to estimate the image quality labels $\hat{q}, \hat{q'}$, where the deepfake and image quality labels are both binary labels. Note that the predicted quality label refers to the detector's judgment on whether the image has experienced quality degradation, where a pristine image has a ground truth label of 1 and a degraded one is labeled as 0. We derive a quality score $s_q = p(1|q)$ to reflect the confidence in predicting high quality, where the higher the score, the better the image quality.

4.4 Progressive Weighting and Optimization

The Adaptive Sampling Network (W) collects feedback regarding the model performance and the image quality to assign adaptive weights that guide the optimization of both the Degradation Generator G and the Quality-contrastive Deepfake Detector F, modifying the training process in a progressive manner. To measure the model performance on training samples, we calculate the absolute difference between the ground-truth label y and the predicted label $\hat{y}$ for each prediction to measure the model performance on training samples, using the metric $\Delta y = |y - \hat{y}|$. This metric provides a direct measure of the model's prediction error for each sample, with smaller values indicating higher confidence and larger values indicating greater uncertainty or error. To obtain feedback on image quality, the Adaptive Sampling Network utilizes the quality score s_q from the detector as the quality metric. Both feedback metrics Δy and q are normalized to the range $[0, 1]$ using the min-max normalization technique to ensure consistency and comparability. Given the normalized model performance feedback Δy and image quality feedback q, W employs a convolutional neural network parameterized by f_φ to generate adaptive weights for the training samples. The concatenated normalized feedback is processed through embedding layers to produce a feedback embedding. Conditioned on this embedding, f_φ predicts the sample weight w, w' for the pristine and its degraded counterpart of each sample, which are constrained to be $(0, 1)$.

4.5 Training Objective

The goal of the optimization in ProFake is to strike a balance: emphasizing learning from challenging samples while not being overly penalized by excessively degraded ones, which is guided by the sample weights generated by Adaptive Sampling Network. Therefore, we first introduce the training objectives of Adaptive Sampling Network, and then introduce the training objectives of Adaptive Degraded Sample Generation and Quality-contrastive Deepfake Detection modules.

The goal of Adaptive Sampling Network is to adjust the learning focus of the deepfake detector to more challenging samples. It adopts a local optimal policy to predict the weights of the training samples in the current iteration of training. Specifically, we optimize f_φ via the error of the weight of sample w_x and the difficulty of prediction Δy, which is formulated as:

$$L_W = (1 - w_x)^2 \Delta y + w_x^2 max(M - \Delta y, 0), \tag{3}$$

where M is a margin hyper-parameter. This loss function is a non-negative saddle-like objective function as it have minima where both sample weight and detection error are either low or high, and maxima where one is high and the other low, which ensures that W assigns higher weights to samples with higher detection errors, thereby promoting a balanced and effective learning process.

The training objectives for Quality-contrastive Deepfake Detection is to identify the quality and detect deepfakes given the training samples. For the goal of deepfake detection, we adopt the AM-Softmax Loss [65] to compute classification loss as with typical deepfake detection methods, which is calculated as:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s(\cos(\theta_{y_i}) - m)}}{e^{s(\cos(\theta_{y_i}) - m)} + \sum_{j \neq y_i} e^{s\cos(\theta_j)}}, \tag{4}$$

where N represents the number of samples, s is a scaling parameter that typically helps in stabilizing the training process. The θ_{y_i} is the angle between the weight vector for the correct class y_i and the feature vector. $\theta_{y_j}$are the angles between the feature vector and all other class weight vectors not equal to y_i. m is the margin parameter added to the cosine of the angle for the correct class.

For the goal of identifying the quality degradation, we build on the observation that degraded samples are naturally inferior to original samples, and exploit this subtle but informative guidance to encourage the detector to discern distinctive features of deepfakes across various quality levels. Specifically, we introduce the quality order loss L_{order}, designed to enforce the constraint that the predicted quality score of a degraded sample $\hat{q'}$ should be lower than that of the pristine sample $\hat{q}$. The loss function is formulated as:

$$L_{order} = \sum_{i=1}^{N} \max(0, \hat{q}_i - \hat{q'}_i + m), \tag{5}$$

where $\hat{q}_i$ and $\hat{q'}_i$ are the predicted quality scores for the pristine and degraded samples, respectively, and m is a hyper parameter that defines the minimum difference required between the two scores to incur zero loss.

The training goal of the Quality Degradation Generator G is to adversarially generate more challenging degraded samples, so we use a negative classification loss to update G.

Given the dynamic weight w and w', we update the Quality Degradation Generator G and the Quality-contrastive Deepfake Detector F by the weighted loss of each training objective, which are delineated by Equation 6 and Equation 7 respectively.

$$L_G = -w_x L_{cls}, \tag{6}$$

$$L_F = w_x(L_{cls} + \lambda L_{order}), \tag{7}$$

where w_x is the weight of the samples, λ is non-negative parameters for loss balancing. The losses L_{cls} and L_{order} represent the classification loss and quality order loss, respectively, and are defined in Equation 4 and Equation 5.

To better demonstrate the changing behavior of ProFake throughout training, we perform a detailed analysis of the training dynamics: Initially, both high-quality and low-quality samples receive high weights due to high classification errors despite how slight the degradation is applied to the samples. As training progresses and the model adapts to high-quality samples, their classification errors decrease, and the adaptive sampling network shifts focus to low-quality samples by assigning them higher weights. When the model's loss on low-quality samples decreases, the degradation generator adversarially generates more challenging degraded samples. This process ensures that the model gradually adapts to increasingly difficult degradations, eventually effectively handling all levels of degradation.

Overall, the progressive quality-adaptive learning offers several advantages. Firstly, the training loop is progressively driven by the feedback of model performance and image quality, ensuring that the model starts with better initialization on high-quality deepfake samples, which enables the model to identify subtle manipulations in high-quality deepfakes. As training proceeds, the progressive quality-adaptive learning enables the model to adapt to more challenging degraded samples. Secondly, by leveraging pairwise learning across various degradation settings, our approach effectively mines quality-adaptive representations for detecting deepfakes with varying quality degradations. Thirdly, our model can also estimate the quality of the input samples. This capability empowers the deepfake detector to calibrate its confidence during inference, providing a more comprehensive view on detection results.

5 Evaluation

In this section, we begin by detailing our experimental settings in Section 5.1. Subsequently, we conduct a comprehensive evaluation of the model's performance across various quality degradation scenarios, as presented in Section 5.2. Finally, we present ablation studies to analyse the design choices behind ProFake, which are discussed in Section 5.3.

5.1 Experimental Setup

Datasets. We evaluate ProFake on three public datasets: FaceForensics++ (FF++) [56], Celeb-DF [45] and DFDC [26]. (1) **FF++** includes 1,000 original and 5,000 manipulated videos using diverse methods: DeepFakes (DF), FaceSwap (FS), Face2Face (F2F), NeuralTexture (NT), and FaceShifter (FSH), covering face swapping and reenactment. It offers three quality types: raw quality, high quality (c23), and low quality (c40), simulating real-world scenarios of compression. (2) **DFDC** features over 100,000 deepfake videos, created using a variety of face swapping techniques, including both GAN-based and traditional methods. (3) **Celeb-DF (CDF)** contains 590 source and 5639 deepfake videos of 59 celebrities from diverse backgrounds. The deepfakes are created using an advanced synthesis method with facial augmentation techniques. To validate effectiveness of ProFake in real-world scenarios, we further conduct evaluation on our collected real-world deepfakes (Video dataset and Live streaming dataset as detailed in Section 3.2).

Evaluation Metrics. Following previous studies [18, 25, 41, 44], we utilize the accuracy score to gauge the overall rate of correctly identified deepfakes and authentic images and the Area Under Curve (AUC) as a measure of our model's discriminative power between real and fake images across different thresholds.

Implementation Details. We segment videos from all datasets into frames and use DLIB [57] for face extraction and alignment, resizing the aligned images to 256×256 for both training and testing. Our experiments employ ConvNext-base [48] as the backbone for the Quality-contrastive Deepfake Detector F, with network weights initialized using ImageNet-pretrained ConvNext-base. Our method is implemented in PyTorch and runs on two NVIDIA 3090 GPUs with a batch size of 64. The detection models are trained using AdamW optimizer [38] with a learning rate of 2e-4, weight decay of 5e-4, and 6 training epochs, while the other modules use Adam optimizer, a learning rate of 5e-5, weight decay of 5e-4.

5.2 Cross-quality Evaluation

To evaluate the effectiveness of deepfake detection models under real-world scenarios, we implement image degradation to the samples in our evaluations, including both common degradation as blur, downsampling, Gaussian noise and JPEG compression, as well as real-world degradation. Our evaluation is conducted on the FF++ dataset for in-domain, cross-quality analysis, and extended to the DFDC and CDF datasets for cross-domain, cross-quality analysis.

Baselines. We perform cross-quality comparisons of the proposed ProFake against state-of-the-art methods: 1) Pseudo-based method: SLADD[58] and SBI[21], which enhance the robustness of detection models by designing specific data augmentation techniques for deepfakes. 2) Frequency-based method: F^3Net[53] , which identify abnormal frequency components in deepfakes. 3) Spatial-based method: Xception[56], EfficientNet-b4[62] and MAT[75]. These models detect crucial clues in fake images by analyzing spatial features such as textures and patterns introduced by deepfakes. 4) Quality-agnostic method: QAD[41], which is designed to efficiently detect deepfakes of varying quality by accounting for the discrepancies between pristine and degraded deepfake samples. To ensure fairness across our comparative experiments, we utilize the officially released weights for SBI[21] and QAD[41], in addition to the pre-trained weights for F^{3}Net, MAT, EfficientNet-b4, and Xception available from PyDeepFakeDet[66]. Since SLADD does not have publicly accessible weights, we trained it on the FF++ dataset using a mixture of three quality levels (raw, c23, c40). Moreover, we evaluate the fine-tuned versions of these models that have been trained on randomly degraded data, indicated by * in their model names. By contrast, our model was trained solely on the FF++ (raw).

Results. We divide the degradation settings into **three groups**: **Medium Quality**, **Low Quality**, and **Real-world Quality** for quantitative comparisons.

• **Medium Quality.** In this category, we examine image degradation commonly found on social media, including downsampling and JPEG compression. We set the downsampling scale factor $s = \frac{3}{4}$, resizing images to three-quarters of their original size, and apply JPEG compression with a quality factor $c = 50$ to represent a moderate level of compression. Table 2 presents our method's performance against various baselines. Compared to F^3Net, our method exhibits a modest improvement on the FF++ dataset and achieves an average 10% improvement on other datasets. Significantly, our approach outperforms other baselines, demonstrating a 20% increase in ACC and AUC scores in both in-domain and cross-domain scenarios. The comparison between F^3Net and other baselines suggests that frequency information is less sensitive to image degradation than spatial domain forgery clues, while structured learning significantly enhances performance regardless of whether the method employs frequency or spatial information. We also observe that the augmentation-based methods SLADD and SBI demonstrate better generalization, outperforming other baselines in the CDF dataset, yet they struggle with image degradation in in-domain scenarios.

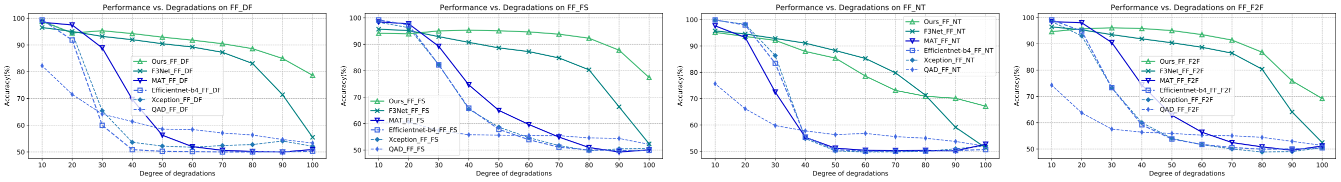
• **Low Quality.** In this category, we simulate severe image degradation typical in social media by combining four types of degradation. We set the kernel size $k = 7 \times 7$ to introduce a significant level of blur to the images. The downsampling scale factor s is reduced

Table 2: Evaluation of classification performance on the Medium Quality datasets. In this setting, we apply $s = \frac{3}{4}$ for downsampling and JPEG compression with quality factor $c = 50$. The detection model with $*$ indicates that the model was trained on dataset with random degradation augmentation. The best results are highlighted in bold.

Model	Test Set															
	NT		DF		F2F		FS		FSH		DFDC		CDF		Avg	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Medium quality datasets with moderate degradation																
F^3Net	**85.27**	**0.94**	89.24	0.96	88.69	0.96	87.25	0.95	60.83	0.74	56.21	0.63	63.54	0.68	75.86	0.84
F^3Net*	72.32	0.82	86.30	0.95	85.00	0.94	88.00	0.96	54.81	0.656	43.86	0.476	60.39	0.68	70.10	0.78
MAT	50.38	0.61	51.98	0.69	56.38	0.71	59.66	0.81	50.10	0.58	14.81	0.57	35.22	0.49	45.50	0.64
MAT*	57.08	0.73	71.92	0.89	69.61	0.86	79.57	0.92	50.37	0.54	11.89	0.45	33.58	0.46	53.43	0.69
Efficientnet-b4	50.03	0.61	50.15	0.69	51.74	0.72	53.91	0.81	50.10	0.56	12.16	0.44	34.96	0.50	43.29	0.62
Efficientnet-b4*	72.28	0.88	90.42	0.97	80.77	0.93	90.86	0.97	50.60	0.51	56.56	0.52	54.19	0.51	70.81	0.76
Xception	49.73	0.49	51.74	0.60	51.64	0.62	54.69	0.72	51.38	0.54	13.73	0.47	36.88	0.48	44.26	0.56
Xception*	67.29	0.85	84.82	0.95	75.69	0.92	80.63	0.94	52.80	0.57	36.71	0.47	35.62	0.428	61.94	0.73
SBI	59.73	0.71	87.75	0.95	68.64	0.80	77.37	0.87	67.87	0.82	34.31	0.71	64.33	0.78	65.71	0.81
SBI*	60.07	0.73	85.22	0.91	70.53	0.82	76.42	0.86	69.32	0.86	36.83	0.76	64.37	0.75	66.10	0.82
SLADD	52.63	0.57	56.25	0.66	54.70	0.60	54.94	0.63	51.42	0.51	46.63	0.56	65.26	0.60	54.55	0.59
SLADD*	58.72	0.63	63.58	0.77	62.73	0.82	63.25	0.71	50.47	0.51	49.73	0.58	67.24	0.68	59.39	0.67
QAD	56.80	0.61	58.39	0.64	55.25	0.60	55.52	0.60	60.30	0.67	27.93	0.63	48.11	0.55	51.76	0.61
QAD*	57.52	0.62	61.12	0.66	59.38	0.64	59.90	0.66	59.51	0.65	48.84	0.61	51.26	0.55	56.79	0.63
ProFake (Ours)	79.57	0.93	**91.80**	**0.97**	**93.51**	**0.98**	**94.69**	**0.98**	**88.39**	**0.98**	**68.65**	**0.79**	**74.54**	**0.83**	**84.45**	**0.92**

Table 3: Evaluation of classification performance ACC (%) and AUC on the Low Quality datasets. In this setting, we apply a Gaussian blur kernel with size $k = 5$, add 10% Gaussian noise, apply $s = \frac{1}{2}$ for downsampling and JPEG compression with quality factor $c = 20$. The detection model with $*$ indicates that the model was trained on dataset with random degradation augmentation. The best results are highlighted in bold.

Model	Test Set															
	NT		DF		F2F		FS		FSH		DFDC		CDF		Avg	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Low quality datasets with strong image degradation																
F^3Net	59.12	0.74	71.41	0.85	64.08	0.80	66.37	0.82	53.79	0.67	53.87	0.61	43.12	0.53	58.82	0.72
F^3Net*	54.26	0.65	71.99	0.85	63.87	0.77	71.70	0.85	52.19	0.60	46.87	0.49	54.63	0.63	59.36	0.69
MAT	50.19	0.53	49.97	0.54	49.52	0.50	49.26	0.58	50.74	0.55	11.50	0.55	39.53	0.52	42.96	0.54
MAT*	53.21	0.66	61.28	0.83	59.41	0.74	59.66	0.76	50.10	0.52	12.55	0.44	33.58	0.46	47.11	0.63
Efficientnet-b4	50.31	0.54	50.00	0.56	50.00	0.54	49.84	0.58	50.18	0.56	11.64	0.46	35.97	0.56	42.56	0.54
Efficientnet-b4*	54.02	0.66	75.79	0.87	62.02	0.75	75.13	0.86	50.03	0.51	**57.67**	0.51	54.05	0.51	61.24	0.67
Xception	50.95	0.52	54.09	0.56	49.03	0.49	50.04	0.52	51.96	0.53	11.61	0.44	42.33	0.51	44.29	0.51
Xception*	54.29	0.65	75.18	0.87	64.90	0.79	63.57	0.78	51.96	0.55	35.22	0.46	35.48	0.41	54.37	0.64
SBI	53.41	0.61	82.34	0.91	56.53	0.69	64.39	0.76	56.74	0.71	23.88	0.68	34.30	0.71	53.08	0.72
SBI*	52.68	0.57	80.07	0.86	57.21	0.70	65.81	0.77	59.39	0.76	30.62	0.71	38.17	**0.75**	54.85	0.73
SLADD	51.35	0.52	54.81	0.57	53.46	0.55	52.88	0.54	51.03	0.48	38.57	0.54	46.63	0.56	49.82	0.54
SLADD*	56.68	0.59	60.74	0.68	58.59	0.58	55.03	0.56	52.60	0.58	46.50	0.53	41.82	0.61	53.14	0.59
QAD	53.73	0.56	54.64	0.59	52.92	0.56	54.39	0.58	55.69	0.60	28.24	0.61	27.93	0.63	46.79	0.59
QAD*	52.47	0.54	57.44	0.61	53.23	0.56	55.02	0.58	56.25	0.60	51.51	0.59	51.65	0.54	53.94	0.57
ProFake (Ours)	**62.14**	**0.83**	**84.96**	**0.95**	**79.49**	**0.91**	**87.78**	**0.95**	**73.29**	**0.88**	54.87	**0.69**	**60.59**	0.67	**71.87**	**0.84**



(a) DF **(b) FS** **(c) NT** **(d) F2F**

Figure 9: Fine-grained cross-quality evaluation of diverse image degradation effects on the FF++ dataset.

to $\frac{1}{2}$, halving the original size of the images. For Gaussian noise, we adjust the noise intensity $\alpha = 10\%$, introducing a deviation equivalent to 10% of the original pixel values. Additionally, we apply JPEG compression with a quality factor of $c = 20$, indicating substantial compression. The evaluation results of our method and baselines are presented in Table 3. It is evident that baseline

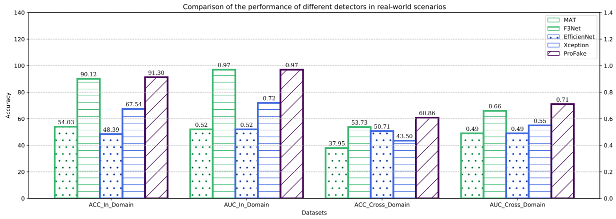


Figure 10: Comparison of the performance of different detectors in real-world scenarios.

models exhibit a significant decline in performance under the Low Quality setting, both in-domain and cross-domain. Considering the poor performance in cross-domain settings, the performance drop of these models in cross-domain settings is smaller than that in the same domain. Notably, some baseline models, such as MAT, SBI, and Xception, demonstrate an accuracy below 50% on certain cross-domain datasets, often classifying real images as fake. Compared to the Medium Quality setting, our method shows only a minor performance decline under the Low Quality setting, indicating that our method has balanced performance across various image degradation levels, even in cross-domain datasets detection.

- **Fine-grained Comparison.** For a more detailed comparison across different degradation levels, we expand our degradation settings to include 10 levels. These levels encompass JPEG compression c ranging from 100 to 10, downsampling scales s from 1 to 0.5, kernel size k from 1×1 to 7×7, and Gaussian noise scales α from 0 to 100 (adding 0% to 10% noise). Figure 9 displays the evaluations across four FF++ datasets at these 10 quality levels, demonstrating that our method robustly maintains high accuracy despite varying quality degradation. Notably, our method sustains an average accuracy of 80% even under degradation degree of 80. While it shows a marginal performance edge over F^3Net at degradation levels below 50, the advantage becomes significant when degradation exceeds 80. Our method consistently outperforms other baselines, with a widening performance gap at higher degradation levels. These baselines become ineffective when the degradation exceeds 30, highlighting their vulnerability to degraded deepfakes.
- **Real-world Quality.** We introduce our comparative analysis on our collected real-world dataset, which includes an evaluation for in-domain videos, as well as live streaming and cross-domain videos for cross-domain comparison. The results are shown in Figure 10. We can see that ProFake significantly enhances the performance of spatial-based detectors for over 30% improvement and even surpasses that of the frequency-based detector $\mathrm{F}^3 Net$. Results demonstrate that ProFake exceeds state-of-the-art performance by over 5% in both ACC and AUC metrics in cross-domain scenarios, and it delivers comparable results in the in-domain setting with performance surpassing 90%.

5.3 Ablation Study

In this section, we present a comprehensive ablation study to assess the effectiveness of each stage of ProFake, in which we focus on the scenario of **the Medium Quality**, as detailed in Section 5.2. The results are presented in Table 4.

Table 4: Ablation study of ProFake with different backbones (Xception and ConvNext-base). The ablation settings are elaborated in Section 5.3.

Model	Xception				ConvNext-base			
	NT	DF	FS	F2F	NT	DF	FS	F2F
staitc degradation	71.93	88.92	88.73	80.58	77.52	86.35	91.84	89.69
only backbone	71.37	91.30	76.45	73.59	73.44	90.37	88.24	90.63
w/o L_{deg}	71.36	88.98	87.39	82.30	76.37	91.76	92.68	93.08
w/o weighting	70.85	81.37	89.03	83.57	76.31	87.63	88.49	92.16
ProFake	72.08	90.09	89.37	82.64	**79.57**	**91.80**	**93.51**	**94.69**

- **Static Progressive Degradation.** To further examine the performance improvement compared to training from easy to hard with static steps (denoted as static degradation in the table), we take only the backbone and the same training set and manually add degradations with static steps. The results show that the progressive training sequence can improve the performance of detecting medium-quality deepfakes compared with only backbone. Notably, our method shows better performance than the static progressive degradation strategy.
- **Adaptive Degraded Sample Generation.** In this component, we eliminate the Quality Degradation Generator and manually degrade the training samples, increasing the degradation intensity by 5 degrees every 1,000 iterations. In contrast to manual degradation, our method self-regulates the pace of degradation, resulting in an average performance improvement of 5%. This underscores the significance of learning to degrade deepfakes as a crucial factor contributing to the success of ProFake.
- **Quality-contrastive Deepfake Detection.** In evaluating the loss function L_{deg}, we omit the quality estimation component. The results indicate a 2% increase in accuracy when quality estimation is incorporated into the detector.
- **Progressive Weighting and Optimization.** We examine the role of dynamic weighting in improving the quality adaptiveness of detection models. By excluding the weighting process, we observe that incorporating weighting enhances the training effectiveness of a quality-adaptive deepfake detector.
- **Selection of Backbone.** Our comparative analysis between the Xception and ConvNext backbones within the detector shows that ConvNext consistently outperforms Xception. This suggests that a higher number of parameters contributes to enhanced robustness of the detector against diverse degradation levels. Our ablation studies demonstrate that simultaneously training on three objectives: degrading, detecting, and assigning weights to training samples can improve the efficacy of detection models, particularly in handling various degrees of image degradation.

6 Discussion

Ethical Consideration. In this work, we conduct a comprehensive security evaluation on degraded risk of deepfake detection in real-world scenarios, which may raise some ethical concerns. While we collected no personal facial data beyond faces of participants recorded by camera. In our evaluation, we mainly use the degraded form of open-source dataset, which is a legitimate and common practice in face-related security research. All generated degraded deepfakes, especially those that exhibit high evasion rates to defeat mainstream deepfake detectors and deceive human perception, are not used outside of the research and have been deprecated following the conclusion ofthe research.

Real-world Applications and Limitations. Since the core of ProFake is to automatically determine the degraded deepfake samples suitable for the current training, there will be no conflict between the training methods and objectives of different detectors. Therefore, our framework is applicable to deepfake detection on various social media platforms, supports multiple backbones and detectors, and does not require actual degraded samples as input. However, ProFake faces cross-domain detection challenges, which can be addressed using compatible methods, such as choosing better network architectures [76] and mining more intrinsic fake cues that different deepfake techniques share [43].

7 Conclusions

In this paper, we explored the effects of real-world quality degradation on the robustness of deepfake detectors across two typical scenarios: deepfake videos and deepfake live streaming on social media platforms. By measuring the performance gap and representation discrepancies introduced by real-world and common degradations to mainstream deepfake detectors, we revealed that the real-world degradations can obscure forgery-related pattern but can be simulated by common degradations such as blur and JPEG compression. Further, through the analysis of training dynamics with varying sequences of sample quality, we observed that train from non-degraded to degraded deepfakes enhances the robustness of the detectors. Based on these insights, we proposed ProFake, a novel deepfake detection method that improves the robustness against quality degradations via progressively learn to degrade, detect and assign weights to the training samples. Extensive experiments demonstrated that ProFake outperforms state-of-the-art deepfake detection methods on deepfake datasets under various degradation conditions including common degradations (average improvement of 30%) and real-world degradations (average improvement of 10%) encountered on 6 social media platforms (i.e., Facebook, Twitter, Weibo, WeChat, YouTube and Bilibili).

ACKNOWLEDGMENTS

This work was supported by Key R&D Program of Zhejiang (Grant No. 2024C01164, 2022C01018), National Natural Science Foundation of China (Grants No. 62122066, U20A20182), National Key R&D Program of China (Grant No. 2021ZD0112803), the Leading Innovative and Entrepreneur Team Introduction Program of HangZhou (Grant No.TD2020001), and CCF-AFSG Research Fund.

References

[1] 2018. It's Getting Harder to Spot a Deep Fake Video. https://www.youtube.com/watch?v=gLoI9hAX9dw. (2018). Accessed: 2023-11-01.

[2] 2021. SwapFace. https://swapface.org/#/home. (2021). Accessed: 2023-11-01.

[3] 2023. DeepfakeStatistics. https://contentdetector.ai/articles/deepfake-statistics. (2023). Accessed: 2023-12-01.

[4] 2023. the famous youtube star made into scam video by Deepfake. https://twitter.com/MrBeast/status/1709046466629554577. (2023). Accessed: 2023-11-01.

[5] 2024. bilibili. https://www.bilibili.com. (2024). Accessed: 2024-04-01.

[6] 2024. facebook. https://www.facebook.com. (2024). Accessed: 2024-04-01.

[7] 2024. TelecommunicationsFraud. https://www.chinadailyhk.com/hk/article/379805. (2024). Accessed: 2024-04-01.

[8] 2024. the abused deepfake allows threat actors to target individuals for smear campaigns. https://flashpoint.io/blog/what-is-deepfake-technology. (2024). Accessed: 2024-04-01.

[9] 2024. weChat. https://weixin.qq.com. (2024). Accessed: 2024-04-01.

[10] 2024. weibo. https://weibo.com. (2024). Accessed: 2024-04-01.

[11] 2024. youtube. https://www.youtube.com. (2024). Accessed: 2024-04-01.

[12] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. 1974. Discrete cosine transform. *IEEE transactions on Computers* 100, 1 (1974), 90–93.

[13] Zahid Akhtar. 2023. Deepfakes generation and detection: a short survey. *Journal of Imaging* 9, 1 (2023), 18.

[14] Weiming Bai, Yufan Liu, Zhipeng Zhang, Bing Li, and Weiming Hu. 2023. AUNet: Learning Relations Between Action Units for Face Forgery Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 24709–24719.

[15] Stella Bounareli, Christos Tzelepis, Vasileios Argyriou, Ioannis Patras, and Georgios Tzimiropoulos. 2023. HyperReenact: one-shot reenactment via jointly learning to refine and retarget faces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 7149–7159.

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

[17] Meng Cao, Haozhi Huang, Hao Wang, Xuan Wang, Li Shen, Sheng Wang, Linchao Bao, Zhifeng Li, and Jiebo Luo. 2020. Task-agnostic temporally consistent facial video editing. *arXiv preprint arXiv:2007.01466* (2020).

[18] Shenhao Cao, Qin Zou, Xiuqing Mao, Dengpan Ye, and Zhongyuan Wang. 2021. Metric learning for anti-compression facial forgery detection. In *Proceedings of the 29th ACM International Conference on Multimedia.* 1929–1937.

[19] Chaoqi Chen, Jiongcheng Li, Xiaoguang Han, Xiaoqing Liu, and Yizhou Yu. 2022. Compound domain generalization via meta-knowledge encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 7119–7129.

[20] Chaoqi Chen, Luyao Tang, Feng Liu, Gangming Zhao, Yue Huang, and Yizhou Yu. 2022. Mix and Reason: Reasoning over Semantic Topology with Data Mixing for Domain Generalization. *Advances in Neural Information Processing Systems* 35 (2022), 33302–33315.

[21] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. 2022. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 18710–18719.

[22] Renwang Chen, Xuanhong Chen, Bingbing Ni, and Yanhao Ge. 2020. Simswap: An efficient framework for high fidelity face swapping. In *Proceedings of the 28th ACM International Conference on Multimedia.* 2003–2011.

[23] Shu-Yu Chen, Feng-Lin Liu, Yu-Kun Lai, Paul L Rosin, Chunpeng Li, Hongbo Fu, and Lin Gao. 2021. DeepFaceEditing: Deep face generation and editing with disentangled geometry and appearance control. *arXiv preprint arXiv:2105.08935* (2021).

[24] Xuanhong Chen, Bingbing Ni, Yutian Liu, Naiyuan Liu, Zhilin Zeng, and Hang Wang. 2023. SimSwap++: Towards Faster and High-Quality Identity Swapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[25] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. 2020. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition.* 5781–5790.

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

[27] Shichao Dong, Jin Wang, Renhe Ji, Jiajun Liang, Haoqiang Fan, and Zheng Ge. 2023. Implicit Identity Leakage: The Stumbling Block to Improving Deepfake Detection Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 3994–4004.

[28] Michail Christos Doukas, Mohammad Rami Koujan, Viktoriia Sharmanska, Anastasios Roussos, and Stefanos Zafeiriou. 2021. Head2head++: Deep facial attributes re-targeting. *IEEE Transactions on Biometrics, Behavior, and Identity Science* 3, 1 (2021), 31–43.

[29] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. 2020. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning.* 3247–3258.

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[31] Ying Guo, Cheng Zhen, and Pengfei Yan. 2023. Controllable Guide-Space for Generalizable Face Forgery Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 20818–20827.

[32] Alexandros Haliassos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. 2022. Leveraging real talking faces via self-supervision for robust forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 14950–14962.

[33] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[34] Gee-Sern Hsu, Chun-Hung Tsai, and Hung-Yi Wu. 2022. Dual-generator face reenactment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 642–650.

[35] Baojin Huang, Zhongyuan Wang, Jifan Yang, Jiaxin Ai, Qin Zou, Qian Wang, and Dengpan Ye. 2023. Implicit Identity Driven Deepfake Face Swapping Detection. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 4490–4499.

[36] Yihao Huang, Felix Juefei-Xu, Qing Guo, Yang Liu, and Geguang Pu. 2022. Fake-Locator: Robust localization of GAN-based face manipulations. *IEEE Transactions on Information Forensics and Security* 17 (2022), 2657–2672.

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

[38] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[39] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. 2017. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision.* 3677–3685.

[40] Nicolas Larue, Ngoc-Son Vu, Vitomir Struc, Peter Peer, and Vassilis Christophides. 2023. SeeABLE: Soft Discrepancies and Bounded Contrastive Learning for Exposing Deepfakes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 21011–21021.

[41] Binh M Le and Simon S Woo. 2023. Quality-Agnostic Deepfake Detection with Intra-model Collaborative Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 22378–22389.

[42] Sangyup Lee, Jaeju An, and Simon S Woo. 2022. BZNet: Unsupervised Multi-scale Branch Zooming Network for Detecting Low-quality Deepfake Videos. In *Proceedings of the ACM Web Conference 2022.* 3500–3510.

[43] Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. 2021. 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.* 6458–6467.

[44] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. 2020. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 5001–5010.

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

[46] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. 2021. Anycost gans for interactive image synthesis and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 14986–14996.

[47] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. 2021. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 772–781.

[48] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 11976–11986.

[49] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. 2018. Rsgan: face swapping and editing using face and hair representation in latent spaces. *arXiv preprint arXiv:1804.03447* (2018).

[50] Xin Ning, Shaohui Xu, Fangzhe Nan, Qingliang Zeng, Chen Wang, Weiwei Cai, Weijun Li, and Yizhang Jiang. 2022. Face editing based on facial recognition features. *IEEE Transactions on Cognitive and Developmental Systems* 15, 2 (2022), 774–783.

[51] Youxin Pang, Yong Zhang, Weize Quan, Yanbo Fan, Xiaodong Cun, Ying Shan, and Dong-ming Yan. 2023. Dpe: Disentanglement of pose and expression for general video portrait editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 427–436.

[52] Hua Qi, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Wei Feng, Yang Liu, and Jianjun Zhao. 2020. Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In *Proceedings of the 28th ACM international conference on multimedia.* 4318–4327.

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

[54] Zuomin Qu, Zuping Xi, Wei Lu, Xiangyang Luo, Qian Wang, and Bin Li. 2024. DF-RAP: A Robust Adversarial Perturbation for Defending against Deepfakes in Real-world Social Network Scenarios. *IEEE Transactions on Information Forensics and Security* (2024).

[55] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 10684–10695.

[56] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 1–11.

[57] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 2016. 300 faces in-the-wild challenge: Database and results. *Image and vision computing* 47 (2016), 3–18.

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

[59] Kaede Shiohara, Xingchao Yang, and Takafumi Taketomi. 2023. BlendFace: Re-designing Identity Encoders for Face-Swapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 7634–7644.

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

[61] Weiwei Sun, Jiantao Zhou, Yuanman Li, Ming Cheung, and James She. 2020. Robust high-capacity watermarking over online social network shared images. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 3 (2020), 1208–1221.

[62] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning.* 6105–6114.

[63] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. 2020. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion* 64 (2020), 131–148.

[64] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[65] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. 2018. Additive Margin Softmax for Face Verification. *IEEE Signal Processing Letters* 25, 7 (July 2018), 926–930. https://doi.org/10.1109/lsp.2018.2822810

[66] Junke Wang, Zuxuan Wu, Wenhao Ouyang, Xintong Han, Jingjing Chen, Yu-Gang Jiang, and Ser-Nam Li. 2022. M2tr: Multi-modal multi-scale transformers for deepfake detection. In *Proceedings of the 2022 international conference on multimedia retrieval.* 615–623.

[67] Yuan Wang, Kun Yu, Chen Chen, Xiyuan Hu, and Silong Peng. 2023. 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.* 7278–7287.

[68] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, and Houqiang Li. 2023. AltFreezing for More General Video Face Forgery Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4129–4138.

[69] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Mengkai Song, Siyan Zheng, Qian Wang, and Ben Niu. 2020. Towards compression-resistant privacy-preserving photo sharing on social networks. In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing.* 81–90.

[70] Simon Woo et al. 2022. ADD: Frequency attention and multi-view based knowledge distillation to detect low-quality compressed deepfake images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 122–130.

[71] Chao Xu, Jiangning Zhang, Miao Hua, Qian He, Zili Yi, and Yong Liu. 2022. Region-aware face swapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 7632–7641.

[72] Zhiliang Xu, Xiyu Yu, Zhibin Hong, Zhen Zhu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. 2021. Facecontroller: Controllable attribute editing for face in the wild. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 3083–3091.

[73] Kelu Yao, Jin Wang, Boyu Diao, and Chao Li. 2023. Towards Understanding the Generalization of Deepfake Detectors from a Game-Theoretical View. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2031–2041.

[74] Chenxu Zhang, Chao Wang, Yifan Zhao, Shuo Cheng, Linjie Luo, and Xiaohu Guo. 2024. DR2: Disentangled Recurrent Representation Learning for Data-Efficient Speech Video Synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 6204–6214.

[75] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. 2021. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2185–2194.

[76] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. 2021. Multi-attentional Deepfake Detection. (2021). arXiv:cs.CV/2103.02406 https://arxiv.org/abs/2103.02406

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

[78] Wanyi Zhuang, Qi Chu, Zhentao Tan, Qiankun Liu, Haojie Yuan, Changtao Miao, Zixiang Luo, and Nenghai Yu. 2022. UIA-ViT: Unsupervised inconsistency-aware method based on vision transformer for face forgery detection. In *European Conference on Computer Vision.* 391–407.