Think Twice Before Adaptation: Improving Adaptability of DeepFake Detection via Online Test-Time Adaptation

Hong-Hanh Nguyen-Le[1] , **Van-Tuan Tran**[2] , **Dinh-Thuc Nguyen**[3] and **Nhien-An Le-Khac**[1]
[1] University College Dublin, Ireland
[2] Trinity College Dublin, Ireland
[3] University of Science, Ho Chi Minh City, Vietnam
hong-hanh.nguyen-le@ucdconnect.ie, tranva@tcd.ie, ndthuc@fit.hcmus.edu.vn, an.lekhac@ucd.ie

Abstract

Deepfake (DF) detectors face significant challenges when deployed in real-world environments, particularly when encountering test samples deviated from training data through either postprocessing manipulations or distribution shifts. We demonstrate postprocessing techniques can completely obscure generation artifacts presented in DF samples, leading to performance degradation of DF detectors. To address these challenges, we propose Think Twice before Adaptation ($\texttt{T}^2\texttt{A}$), a novel online test-time adaptation method that enhances the adaptability of detectors during inference without requiring access to source training data or labels. Our key idea is to enable the model to explore alternative options through an Uncertainty-aware Negative Learning objective rather than solely relying on its initial predictions as commonly seen in entropy minimization (EM)-based approaches. We also introduce an Uncertain Sample Prioritization strategy and Gradients Masking technique to improve the adaptation by focusing on important samples and model parameters. Our theoretical analysis demonstrates that the proposed negative learning objective exhibits complementary behavior to EM, facilitating better adaptation capability. Empirically, our method achieves state-of-the-art results compared to existing test-time adaptation (TTA) approaches and significantly enhances the resilience and generalization of DF detectors during inference.

1 Introduction

Recently, Generative Artificial Intelligence (GenAI) has been used to generate DFs for malicious purposes, such as impersonation[1] and disinformation spread[2], raising concerns about privacy and security. Several DF detection approaches have been proposed to mitigate these negative impacts [Nguyen-Le *et al.*, 2024a]. Despite advances, deploying these systems in real-world environments presents two critical challenges. First, in practice, adversaries can strategically apply previously unknown postprocessing techniques to DF samples **at inference time**, completely obscuring the generation artifacts [Corvi *et al.*, 2023] and successfully bypassing detection systems. Second, real-world applications are frequently exposed to test samples drawn from distributions that deviate substantially from the training data distribution [Pan *et al.*, 2023], leading to performance degradation. To mitigate these challenges, existing approaches require access to source training data and labels for complete re-training [Ni *et al.*, 2022; Shiohara and Yamasaki, 2022], continual learning [Pan *et al.*, 2023] or test-time training [Chen *et al.*, 2022], which is costly and time-consuming.

In this work, we address these limitations by introducing a novel TTA-based method, namely **Think Twice before Adaptation** ($\texttt{T}^2\texttt{A}$), which enhances pre-trained DF detectors without requiring access to source training data or labels. Our approach achieves two key objectives: (1) enhanced resilience through dynamic adaptation to unknown postprocessing techniques; and (2) improved generalization to new samples from unknown distributions. While current TTA approaches commonly employ Entropy Minimization (EM) as the adaptation objective, solely relying on EM can result in confirmation bias caused by overconfident predictions [Zhang *et al.*, 2024] and model collapse [Niu *et al.*, 2023]. To this end, in T^2A, we design a novel Uncertainty-aware Negative Learning adaptation objective with noisy pseudo-labels, allowing the model to explore alternative options (i.e., other classes in the classification problem) rather than becoming overly confident in potentially incorrect predictions. For better adaptation, we incorporate Focal Loss [Ross and Dollár, 2017] into the negative learning (NL) objective to dynamically prioritize crucial samples and propose a gradients masking technique that updates crucial model parameters whose gradients align with those of BatchNorm layers.

Our contributions. To the best of our knowledge, we are the first to present a novel TTA-based method for DF detection. Our contributions include:

- We provide a theoretical and quantitative analysis (Sec. 3) that demonstrates the impacts of postprocessing techniques on the detectability of DF detectors.
- We introduce $\texttt{T}^2\texttt{A}$, a novel TTA-based method specifi-

[1] Finance worker pays out $25 million after video call with deepfake 'chief financial officer'

[2] AI-faked images of Donald Trump's imagined arrest swirl on Twitter

cally designed for DF detection. $\texttt{T}^2\texttt{A}$ enables models to explore alternative options rather than relying on their initial predictions for adaptation (Sec. 4.3). We also theoretically demonstrate that our proposed negative learning objective exhibits complementary behavior to EM. Additionally, we introduce Uncertain Sample Prioritization strategy (Sec. 4.4) and Gradients Masking technique (Sec. 4.5) to dynamically focus on crucial samples and crucial model parameters when adapting.

- We evaluate $\texttt{T}^2\texttt{A}$ under two scenarios: (i) Unknown postprocessing techniques; and (ii) Unknown data distribution and postprocessing techniques. Our experimental results show superior adaptation capabilities compared to existing TTA approaches. Furthermore, we demonstrate that integration of $\texttt{T}^2\texttt{A}$ significantly enhances the resilience and generalization of DF detectors during inference, establishing its practical utility in real-world deployments.

2 Related Work

2.1 Deepfake Detection

DF detection approaches are often formulated as a binary classification problem that automatically learns discriminative features from large-scale datasets [Nguyen-Le *et al.*, 2024b]. Existing approaches can be classified into three categories based on their inputs: (i) Spatial-based approaches that operate directly on pixel-level features [Ni *et al.*, 2022; Cao *et al.*, 2022], (ii) Frequency-based approaches that analyze generation artifacts in the frequency domain [Liu *et al.*, 2021; Frank *et al.*, 2020], and (iii) Hybrid approaches that integrate both pixel and frequency domain information within a unified method [Liu *et al.*, 2023b]. Recent advances have improved the cross-dataset generalization of DF detectors by employing data augmentation (DA) strategies [Ni *et al.*, 2022; Yan *et al.*, 2024], synthesis techniques [Shiohara and Yamasaki, 2022], continual learning [Pan *et al.*, 2023], meta-learning and one-shot test-time training [Chen *et al.*, 2022].

Compared to existing methods, our $\texttt{T}^2\texttt{A}$ offers advantages: (1) $\texttt{T}^2\texttt{A}$ enables DF detectors to be adapted to test data without access to source data (e.g., OST [Chen *et al.*, 2022] requires source data for adaptation); (2) $\texttt{T}^2\texttt{A}$ does not rely on any DA or synthesis techniques to extend the diversity of data; (3) Not only enhance the generalization, $\texttt{T}^2\texttt{A}$ also improves the resilience of DF detectors to unknown postprocessing techniques.Additionally, our method is orthogonal to these works [Fang *et al.*, 2024; Liu *et al.*, 2024; He *et al.*, 2024], which require pre-training on joint datasets (physical and digital attacks) and do not adapt during inference.

2.2 Test-time Adaptation (TTA)

TTA approaches only require access to the pre-trained model from the source domain for adaptation [Liang *et al.*, 2024]. Unlike source-free domain adaptation approaches [Li *et al.*, 2024], which require access to the entire target dataset, TTA enables online adaptation to the arrived test samples.

TENT [Wang *et al.*, 2020] and MEMO [Zhang *et al.*, 2022] optimized batch normalization (BN) statistics from the test batch through EM. LAME [Boudiaf *et al.*, 2022] adapted only the model's output probabilities by minimizing Kullback–Leibler divergence between the model's predictions and optimal nearby points' vectors. Several methods have studied TTA in continuously changing environments. CoTTA [Wang *et al.*, 2022] implemented weight and augmentation averaging to mitigate error accumulation, while EATA [Niu *et al.*, 2022] developed an efficient entropy-based sample selection strategy for model updates. Inspired by parameter-efficient fine-tuning, VIDA [Liu *et al.*, 2023a] used high-rank adapters to handle domain shifts. However, these methods solely rely on EM as the learning principle, which can present two issues: (1) **Confirmation bias**: EM greedily pushes for confident predictions on all samples, even when predictions are incorrect [Zhang *et al.*, 2024], leading to overconfident yet incorrect predictions; and (2) **Model Collapse**: EM tends to cause model collapse, where the model predicts all samples to the same class, regardless of their true labels [Niu *et al.*, 2023]. The model collapse phenomenon is particularly problematic in DF detection, where the inherent bias toward dominant fake samples in training data [Layton *et al.*, 2024] can exacerbate the collapse.

Focusing on the problem of EM, our $\texttt{T}^2\texttt{A}$ method allows the model to consider alternative options rather than completely relying on its initial prediction during inference through NL with noisy pseudo-labels.

2.3 Negative Learning

Supervised learning or positive learning (PL) directly maps inputs to their corresponding labels. However, when labels are noisy, PL can lead models to learn incorrect patterns. Negative learning (NL) [Kim *et al.*, 2019] addresses this challenge by training networks to identify which classes an input does not belong to. Several loss functions have been proposed by leveraging this concept: NLNL [Kim *et al.*, 2019] combines sequential PL and NL phases, while JNPL [Kim *et al.*, 2021] proposes a single-phase approach through joint optimization of enhanced NL and PL loss functions. Recent work has further integrated NL principles with normalization techniques [Ma *et al.*, 2020] to transform active losses into passive ones [Ye *et al.*, 2023].

Inspired by these advances, we introduce a NL strategy with noisy pseudo-labels to our $\texttt{T}^2\texttt{A}$ method to enable the model to think twice during adaptation, avoiding confirmation bias and model collapse caused by EM.

3 Generation Artifacts Analysis

Artifacts in DFs generated by Generative Adversarial Examples (GANs), which emerge from the upsampling operations in the GANs pipeline, can be revealed in the frequency domain through Discrete Fourier Transform (DFT) [Frank *et al.*, 2020]. In this section, we demonstrate postprocessing techniques can completely obscure these artifacts presented in DF samples, leading to performance degradation of DF detectors.

Definition 3.1. Let an image $x(\cdot, \cdot)$ of size $M \times N$, its DFT $X(\cdot, \cdot)$ is defined as:

$$X(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) e^{-j2\pi(\frac{um}{M} + \frac{vn}{N})}, \quad (1)$$

where $x(m,n)$ represents pixel values at spatial coordinates and $X(u,v)$ denotes the corresponding Fourier coefficient in frequency domain.

Lemma 3.2. *For two images $x_1(\cdot,\cdot)$ and $x_2(\cdot,\cdot)$, their convolution in the spatial domain is equivalent to multiplication of their spectra in the frequency domain:*

$$x_1(m,n) \circledast x_2(m,n) \Leftrightarrow X_1(u,v) \cdot X_2(u,v). \quad (2)$$

This property (Proof in Appendix A) is particularly important for understanding why the upsampling operation leaves artifacts in the frequency domain [Ojha *et al.*, 2023]. For an image $x(\cdot,\cdot)$ convolved with a kernel $c(\cdot,\cdot)$, the output $y(\cdot,\cdot)$ in the spatial domain and its frequency domain form can be expressed as:

$$\begin{aligned} y(m,n) &= x(m,n) \circledast c(m,n) \\ \Leftrightarrow Y(u,v) &= X(u,v) \cdot C(u,v) \end{aligned} \quad (3)$$

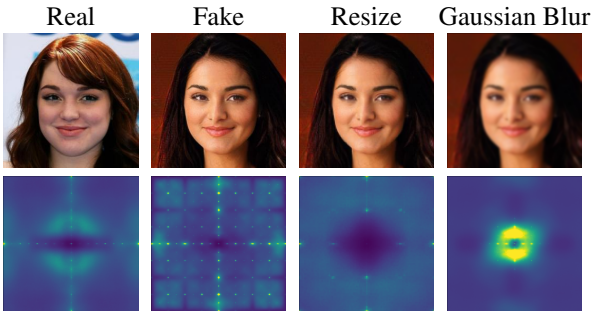

Figure 1: Comparison of frequency domain artifacts across different image processing conditions. Top row: Images in spatial domain. Bottom row: Corresponding frequency spectra. Artifacts as *checkerboard patterns* in (c) and (d) are obscured by postprocessing techniques (i.e., Resize, Gaussian Blur). All fake images are generated by StarGANv2.

When image $x(\cdot,\cdot)$ is upsampled by a factor of 2 in both dimensions, the upsampled image $\tilde{x}(\cdot,\cdot)$ can be expressed as:

$$\tilde{x}(m,n) = \begin{cases} x\left(\frac{m}{2}, \frac{n}{2}\right), & m = 2k, n = 2l \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

where $k = 0, \ldots, M-1$ and $l = 0, \ldots, N-1$. The DFT of the upsampled image is:

$$\tilde{X}(u,v) = \frac{1}{4MN} \sum_{m=0}^{2M-1} \sum_{n=0}^{2N-1} \tilde{x}(m,n) e^{-j2\pi(\frac{um}{2M} + \frac{vn}{2N})} \quad (5)$$

This upsampling operation creates a characteristic periodic structure in the frequency domain, showing that the original image's frequency components appear multiple times in the frequency domain:

$$\tilde{X}(u,v) = \begin{cases} X(u,v), & u \in [0, M-1], v \in [0, N-1] \\ X(u-M,v), & u \in [M, 2M-1], v \in [0, N-1] \\ X(u,v-N), & u \in [0, M-1], v \in [N, 2N-1] \\ X(u-N,v-N), & u \in [M, 2M-1], v \in [N, 2N-1] \end{cases} \quad (6)$$

These duplicated components create distinctive artifacts as *checkerboard patterns* in the frequency domain that distinguishes GAN-generated images from real ones.

However, these spectral artifacts exhibit vulnerability to various postprocessing operations [Corvi *et al.*, 2023]. As shown in Figure 1(b), the GAN-generated image displays distinctive checkerboard artifacts in its frequency spectrum, but they undergo substantial modifications when subjected to different postprocessing operations (Figures 1(c)-(d)). The magnitude of these artifacts' obscurity correlates directly with the intensity of the applied postprocessing operations, as demonstrated in Figure 3 (Appendix B). Furthermore, the empirical analysis presented in another Figure 2 of Appendix B shows that the performance of existing DF detectors tends to drop significantly when encountering unseen postprocessing techniques with increasing intensities.

4 Methodology

The core principle of $\mathtt{T}^2\mathtt{A}$ lies in its deliberate approach to decision-making, encouraging models to explore alternative options rather than solely relying on their initial predictions. The key steps of $\mathtt{T}^2\mathtt{A}$ are summarized in Algorithm 1.

4.1 Problem Definition

Given a DF detector $f : \mathcal{X} \to \mathbb{R}^2$ parameterized by θ is well-trained on the training data $\mathcal{D}^{train} = \{(x_i, y_i)\}_{i=1}^{N^{train}} \sim P^{train}(x,y)$, where $x \in \mathcal{X}$ is the input and $y \in \mathcal{Y} = \{0,1\}$ is the target label, our goal is to online update parameters θ of f on mini-batches $\{\mathcal{B}_1, \mathcal{B}_2, \ldots\}$ of the test stream $\mathcal{D}^{test} = \{(x_j, y_j)\}_{j=1}^{N^{test}} \sim P^{test}(x,y)$. Note that, in the online TTA setting, $P^{train}(x,y)$ and $\{y_j\}$ are unavailable, and the knowledge learned in previously seen mini-batches could be accumulated for adaptation to the current mini-batch [Liang *et al.*, 2024]. In this work, we consider online TTA in two challenging scenarios of DF detection:

1. **Unseen postprocessing Techniques**: While the test data distribution remains similar to the training distribution $P^{train}(x,y) = P^{test}(x,y)$, the test samples are applied unknown postprocessing operations $\Psi : \mathcal{X} \to \mathcal{X}$. Specifically, given a test sample $x_j \sim P^{test}$, f takes $\Psi(x_j)$ as input, where $\Psi \in \mathfrak{P}$ with $\mathfrak{P}$ denotes a set of unseen postprocessing techniques during training.

2. **Unseen Data Distribution and postprocessing Techniques**: This is a more challenging setting in which test samples come from a different distribution $P^{test} \neq P^{train}$ and are also subjected to unknown postprocessing operations.

Algorithm 1: T^{2}A Algorithm

Input : trained model f_θ, test samples $\mathcal{D}^{test} = \{x_j, y_j)\}_{j=1}^{N^{test}}$
Define: batch size B; loss balancing hyperparameters α, β, gradients alignment threshold ψ; learning rate η

for *mini-batches* $\{x_i\}_{i=1}^{B} \subset \mathcal{D}^{test}$ **do**
- Obtain pseudo-label $\hat{y}_i$ from Eq. 8
- Calculate noisy pseudo-label by Eq. 9
- Calculate entropy of model predictions $\mathcal{L}_{EM}$ follow Eq. 7
- Calculate noise-tolerant negative loss $\mathcal{L}_{NTNL}(x_i, \tilde{y}_i) = \alpha\mathcal{L}_{nn}(x_i, \tilde{y}_i) + \beta\mathcal{L}_p(x_i, \tilde{y}_i)$ follow Equations (11) and (12)
- Optimize the adaptation objective function: $\mathcal{L}_{NTNL} + \mathcal{L}_{EM}$ to obtain the gradient matrix $\nabla_\theta\mathcal{L}$
- Perform Gradient Masking on $\nabla_\theta\mathcal{L}$ by keeping the parameters of those gradients aligned with gradients of BN layers by Eq. 15
- Perform Gradient Descent to adapt the model: $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$

4.2 Revisitting Entropy Minimization (EM)

EM is commonly used to update model parameters by minimizing the entropy of model outputs on test sample x during inference:

$$\mathcal{L}_{EM} = -\sum_{c\in C} p(y=c|x)\log p(y=c|x), \tag{7}$$

where $p(y=c|x)$ is the predicted probability for class c, computed as the softmax output of the model: $p(y=c|x) = \frac{\exp(f_c(x))}{\sum_{c\in C}\exp(f_j(x)}$, where $f_c(x)$ is the logit for class c from the model's forward pass on input x. As discussed in Sec 2.2, EM causes two issues: Confirmation bias and Model collapse. Therefore, besides EM, our T^{2}A method introduces a NL strategy with noisy pseudo-labels (described in Sec. 4.3), allowing models to re-think other potential options before making the final decision.

4.3 Uncertainty-aware Negative Learning

Uncertainty Modelling with Noisy Pseudo-Labels

Given the DF detector f, the pseudo-label $\hat{y} = \hat{y}(x) \in \{0, 1\}$ of input x is defined as:

$$\hat{y} = \text{sign}(f(x) - \tau) = \begin{cases} 1, & f(x) \geq \tau \\ 0, & f(x) < \tau \end{cases}, \tag{8}$$

where $\tau \in [0, 1]$ denotes the classification threshold. Rather than implicitly trusting the model's initial predictions, we enable the model to "doubt" its predictions by introducing noisy pseudo-labels.

We model the uncertainty in pseudo-labels using a Bernoulli distribution. For each input x with pseudo-label $\hat{y}$, we generate a noisy pseudo-label $\tilde{y}$ for input x_i as follows:

$$\tilde{y} = \begin{cases} 1-\hat{y}, & \text{if } X \sim \texttt{Bernoulli}(1-p_{x_i}) = 1 \\ \hat{y}, & \text{otherwise} \end{cases}, \tag{9}$$

where p_{x_i} represents the prediction probability. This indicates that higher confidence predictions have a lower probability of being flipped. When the Bernoulli trial equals 1 (with probability $1-p_{x_i}$), the pseudo-label is flipped to the opposite class; otherwise (with probability p_{x_i}), it remains unchanged. However, directly adapting to noisy pseudo-labels presents two limitations during test-time updates: (1) Without access to source data for regularization, errors from noisy labels can accumulate rapidly; and (2) The stochastic nature of noisy gradients can lead to unstable updates.

Noise-tolerant Negative Loss Function

The goal of the noise-tolerant negative loss (NTNL) is to enable the model to think twice through NL with noisy pseudo-labels.

From Positive to Negative Learning. Negative learning (NL) enables the model to be taught with a lesson that *"this input image does not belong to this complementary label"* [Kim *et al.*, 2019]. In our work, converting from pseudo-labels to noisy versions is equivalent to transforming from positive to negative learning, facilitating the DF model to re-think that "this input image might not belong to this real (fake)/fake (real) label".

Noise-tolerant Negative Loss Function. Inspired by existing works [Zhou *et al.*, 2021; Ma *et al.*, 2020; Ghosh *et al.*, 2017], we start from the fact that any loss function can be robust to noisy labels through a simple normalization operation:

$$\mathcal{L}_{norm} = \frac{\ell(f(x), y)}{\sum_{c\in C}\ell(f(x), c)}. \tag{10}$$

Theorem 4.1. *In the binary classification with pseudo-label $\hat{y} \in \{0, 1\}$, if the normalized loss function $\mathcal{L}_{norm}$ has the local extremum at x^*, the entropy minimization function $\mathcal{L}_{EM}$ also has the local at x^*, and vice versa.*

From Theorem 4.1 (Proof in Appendix A), we demonstrate that simply using pseudo-labels in the normalized loss function could drive the model toward maximizing confidence in its initial predictions $\hat{y}$. This behavior aligns with the EM objective presented in Eq.7. However, we seek to enable the model to explore another option rather than uncritically trusting its initial predictions, which may be incorrect. To do that, we introduce noisy pseudo-labels $\tilde{y}$ in place of the original pseudo-labels $\hat{y}$ within the normalized loss function, in which $\tilde{y}$ is generated by the flipping procedure described previously, effectively transforming normalized loss function (Eq. 10) to a negative one. This normalized negative loss $\mathcal{L}_{nn}$ for adapting with noisy pseudo-labels is defined as:

$$\mathcal{L}_{nn}(x, \tilde{y}) = \frac{\ell(f(x), \tilde{y})}{\sum_{c=\in\{0,1\}}\ell(f(x), c)}. \tag{11}$$

As shown in Figure 4 (Appendix C.2), given a normalized loss function with pseudo label $\mathcal{L}_{norm}(x, \hat{y})$, our normalized negative loss function $\mathcal{L}_{nn}(x, \tilde{y})$ with noisy pseudo-label is the opposite of $\mathcal{L}_{norm}(x, \hat{y})$.

Prior research by [Ma *et al.*, 2020; Ye *et al.*, 2023] has indicated that the normalized loss function suffers from the underfitting problem. This problem is particularly critical in the TTA context where the model only "sees" a few samples during inference. To address this challenge, we incorporate the

passive loss function $\mathcal{L}_p$ [Ye *et al.*, 2023] into TTA, leading to our NTNL which can effectively help the model to adapt to noisy pseudo-labels:

$$\mathcal{L}_{NTNL}(x, \tilde{y}) = \alpha \mathcal{L}_{nn}(x, \tilde{y}) + \beta \mathcal{L}_p(x, \tilde{y}), \tag{12}$$

where $\mathcal{L}_p(x, \tilde{y}) = 1 - \frac{p_0 - \ell(f(x), \tilde{y})}{\sum_{c \in \{0,1\}} p_0 - \ell(f(x), c)}$, p_0 is the minimum value of the model prediction in the current test batch, and α, β are balancing hyperparameters.

Definition 4.2. (Passive loss function). $\mathcal{L}_p$ is a passive loss function if $\forall(x, y) \in \mathcal{D}, \exists k \neq y, \ell(f(x), k) \neq 0$.

4.4 Uncertain Sample Prioritization

To identify which samples should be prioritized during adaptation, we propose a dynamic prioritization strategy that focuses on uncertain samples (i.e., low confidence). Our intuition here is that lower-confidence samples require the model to be considered more carefully. Specifically, we incorporate Focal Loss [Ross and Dollár, 2017] into the NTNL function (Eq. 12). Formally, the loss function $\ell(x, \tilde{y})$ is now defined:

$$\ell(x, \tilde{y}) = -(1 - p(\tilde{y}|x)^\gamma) \log p(\tilde{y}|x), \tag{13}$$

where γ controls the rate at which high-confident samples are down-weighted.

The proposed NTNL with Focal Loss enables the model to explore alternative options beyond its initial predictions while dynamically focusing on uncertain samples during adaptation. When combined with EM, we formulate our final adaptation objective function to enhance the adaptation of DF detectors as follows:

$$\mathcal{L} = \mathcal{L}_{NTNL} + \mathcal{L}_{EM}, \tag{14}$$

where $\mathcal{L}_{EM}$ is the entropy of model predictions defined in Eq. 7. By optimizing this objective, our approach achieves robust adaptation that can effectively handle both unknown postprocessing techniques and distribution shifts during inference.

4.5 Gradients Masking

BatchNorm (BN) adaptation [Schneider *et al.*, 2020] is widely used in existing TTA approaches [Niu *et al.*, 2022; Wang *et al.*, 2020]. BN is a crucial layer that normalizes each feature z during training: $y = \varrho * \left(\frac{(z - \mu^b)}{\sigma^b}\right) + \vartheta$, where μ^b and σ^b are batch statistics, and ϱ, ϑ are learnable parameters. After training, μ^{ema} and σ^{ema}, which are estimated over the whole training dataset via exponential moving average (EMA) [Schneider *et al.*, 2020], are used during inference. When $P^{train}(x, y) \neq P^{test}(x, y)$, BN adaptation replaces EMA statistics (μ^{ema}, σ^{ema}) with statistics computed from test mini-batches ($\hat{\mu}^b$, $\hat{\sigma}^b$). However, this approach is limited by only updating BN layer parameters.

To overcome this limitation, we propose a gradient masking technique that identifies and updates parameters whose gradients align with those of BN layers. Let θ_{BN_i} be the parameter of i-th BN layer, and all BN parameters' gradients are concatenated into a single vector: $u = [\nabla_{\theta_{BN_1}} \mathcal{L}, \nabla_{\theta_{BN_2}} \mathcal{L}, ..., \nabla_{\theta_{BN_L}} \mathcal{L}]$, where N is the number of BN layers and $\nabla_{\theta_{BN_i}} \mathcal{L}$ represents the gradient vector of the loss $\mathcal{L}$ with respect to parameters in the i-th BN layer. For each non-BN parameter's gradient $v_i = \nabla_{\theta_i} \mathcal{L}$ in the model, we compute its cosine similarity with the concatenated BN gradients: $\text{sim}(u, v_i) = \frac{\langle v_i, u \rangle}{||v_i|| \cdot ||u||}$.

Note that, since parameter gradients and BN gradient vectors have different dimensions, zero-padding is applied to align dimensions before computing similarity. The final gradient masking is then applied as:

$$\nabla_{\theta_i} \mathcal{L} = \begin{cases} v_i & \text{if sim}(v_i, u) > \psi \\ 0 & \text{otherwise} \end{cases}, \tag{15}$$

where ψ is a threshold to control the selection of parameters for updating. This technique brings more capacity for adaptation as more model parameters are updated compared to approaches that only update BN parameters during inference [Niu *et al.*, 2022; Wang *et al.*, 2020].

5 Experiments

In this section, we demonstrate the effectiveness of our $\texttt{T}^2\texttt{A}$ method when comparing it with state-of-the-art (SoTA) TTA approaches and DF detectors. We also provide an ablation study for our method in Appendix D.1 and an analysis of running time compared to other TTA methods in Appendix D.4.

5.1 Setup

Datasets and modeling

We use Xception [Chollet, 2017] as the source model, which as commonly used as the backbone in DF detectors. The training set is FaceForensics++ (FF++) [Rossler *et al.*, 2019]. To evaluate the adaptability of our $\texttt{T}^2\texttt{A}$ method, we use six more datasets at inference time, including CelebDF-v1 [Li *et al.*, 2020b], CelebDF-v2 [Li *et al.*, 2020b], DeepFakeDetection (DFD) [Google, 2019], DeepFake Detection Challenge Preview (DFDCP) [Dolhansky, 2019], UADFV [Li *et al.*, 2018], and FaceShifter (FSh) [Li *et al.*, 2020a]. The dataset implementations are provided by [Yan *et al.*, 2023] and more details are described in Appendix C.

Metrics

We use three evaluation metrics: accuracy (ACC), the area under the ROC curve (AUC), and average precision (AP). For each metric, higher values show better results. Notably, in the DF detection context, datasets inherently exhibit significant class imbalance with fake samples substantially dominating real ones [Layton *et al.*, 2024], the AUC metric is more important as it remains robust to this problem.

Postprocessing Techniques

Following [Chen *et al.*, 2022], we employ four postprocessing techniques: Gaussian blur, changes in color saturation, changes in color contrast, and resize: downsample the image by a factor then upsample it to the original resolution. At the inference time, test samples are applied to these operations with the intensity level increasing from 1 to 5. Details of postprocessing techniques and intensity levels are provided in Appendix C. Note that these postprocessing techniques are unknown to all models.

Method	Postprocessing Techniques														
	Color Contrast			Color Saturation			Resize			Gaussian Blur			Average		
	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP
Source	0.7891 ± 0.04	0.8696 ± 0.03	0.9639 ± 0.01	0.8074 ± 0.04	0.8195 ± 0.06	0.9432 ± 0.02	0.8120 ± 0.03	0.8767 ± 0.02	0.9669 ± 0.01	0.8431 ± 0.01	0.8423 ± 0.04	0.9523 ± 0.01	0.8129 ± 0.01	0.8520 ± 0.02	0.9566 ± 0.01
TENT	0.8745 ± 0.01	0.9043 ± 0.01	0.9732 ± 0.01	0.8408 ± 0.03	0.8510 ± 0.05	0.9562 ± 0.01	**0.8517** ± 0.01	0.8837 ± 0.02	0.9680 ± 0.01	0.8622 ± 0.01	0.8844 ± 0.02	0.9676 ± 0.01	0.8573 ± 0.01	0.8808 ± 0.01	0.9663 ± 0.01
MEMO	0.8288 ± 0.01	0.8612 ± 0.01	0.9603 ± 0.01	0.8268 ± 0.01	0.8244 ± 0.04	0.9482 ± 0.01	0.8348 ± 0.01	0.8611 ± 0.02	0.9620 ± 0.01	0.8334 ± 0.01	0.8676 ± 0.02	0.9626 ± 0.01	0.8310 ± 0.01	0.8536 ± 0.01	0.9583 ± 0.01
EATA	0.8740 ± 0.01	0.9044 ± 0.01	0.9733 ± 0.01	0.8402 ± 0.03	0.8507 ± 0.05	0.9561 ± 0.01	0.8511 ± 0.01	0.8839 ± 0.02	0.9681 ± 0.01	0.8625 ± 0.01	0.8846 ± 0.02	0.9676 ± 0.01	0.8570 ± 0.01	0.8809 ± 0.01	0.9663 ± 0.01
CoTTA	0.8548 ± 0.01	0.8706 ± 0.02	0.9596 ± 0.01	0.8214 ± 0.01	0.8256 ± 0.01	0.9481 ± 0.01	0.8445 ± 0.01	0.8618 ± 0.02	0.9618 ± 0.01	0.8517 ± 0.01	0.8664 ± 0.02	0.9622 ± 0.01	0.8431 ± 0.01	0.8561 ± 0.01	0.9579 ± 0.01
LAME	0.7882 ± 0.03	0.8185 ± 0.05	0.9393 ± 0.01	0.8088 ± 0.03	0.7594 ± 0.05	0.9096 ± 0.03	0.7957 ± 0.01	0.8113 ± 0.02	0.9311 ± 0.01	0.8065 ± 0.01	0.7519 ± 0.06	0.9035 ± 0.02	0.7998 ± 0.01	0.7853 ± 0.02	0.9209 ± 0.01
VIDA	0.8517 ± 0.01	0.8794 ± 0.01	0.9647 ± 0.01	0.8168 ± 0.02	0.8210 ± 0.05	0.9446 ± 0.01	0.8385 ± 0.01	0.8668 ± 0.03	0.9617 ± 0.01	0.8448 ± 0.01	0.8631 ± 0.02	0.9596 ± 0.01	0.8380 ± 0.01	0.8576 ± 0.01	0.9576 ± 0.01
COME	0.8660 ± 0.01	0.8983 ± 0.01	0.9716 ± 0.01	0.8391 ± 0.02	0.8502 ± 0.05	0.9568 ± 0.02	0.8528 ± 0.02	0.8781 ± 0.03	0.9654 ± 0.01	0.8622 ± 0.01	0.8812 ± 0.02	0.9665 ± 0.01	0.855 ± 0.01	0.877 ± 0.02	0.9651 ± 0.01
T^{2}A (Ours)	**0.8745** ± 0.01	**0.9044** ± 0.02	**0.9733** ± 0.01	**0.8437** ± 0.03	**0.8519** ± 0.05	**0.9566** ± 0.01	0.8502 ± 0.02	**0.8840** ± 0.02	**0.9681** ± 0.01	**0.8642** ± 0.01	**0.8847** ± 0.02	**0.9676** ± 0.01	**0.8582** ± 0.01	**0.8813** ± 0.01	**0.9664** ± 0.01

Table 1: Comparison with SoTA TTA methods on FF++ with different unknown postprocessing techniques. The results for each postprocessing technique are averaged across 5 intensity levels. Bold values denote the best performance for each metric.

Mehtod	CelebDF-v1			CelebDF-v2			DFD			FSh			DFDCP			UADFV		
	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP
Source	0.6171	0.5730	0.6797	0.6621	0.6118	0.7337	0.8337	0.5570	0.8891	**0.5370**	0.5587	0.5480	0.6737	0.6553	0.7598	0.6316	0.7109	0.6443
TENT	0.6334	0.6166	0.7028	0.6370	0.6327	0.7475	0.7631	0.6409	0.9258	0.5285	0.5586	0.5540	0.7213	0.6990	0.7763	0.6625	0.7330	0.6674
MEMO	0.6456	0.6216	0.7003	0.6679	0.5937	0.7171	0.8798	0.5884	0.9148	0.5107	0.5619	0.5408	0.7000	0.6892	0.7466	0.6337	0.7295	0.6653
EATA	0.6313	0.6165	0.7029	0.6389	0.6330	0.7474	0.7579	0.6438	0.9276	0.5307	0.5583	0.5532	0.7245	0.7004	0.7758	0.6604	0.7330	0.6685
CoTTA	0.6354	0.6280	0.6975	0.6602	0.6189	0.7380	0.8757	0.6068	0.9222	0.5292	0.5661	0.5528	0.6934	0.6524	0.7384	0.6316	0.7210	0.6532
LAME	0.6211	0.5901	0.6733	0.6505	0.5914	0.7033	0.8935	0.5724	0.9091	0.5007	0.5307	0.5174	0.6475	0.5988	0.6996	0.5102	0.676	0.6284
VIDA	0.6374	0.6057	0.6683	**0.6756**	0.5589	0.6849	**0.8810**	0.5948	0.9230	0.5192	0.5285	0.5337	0.6770	0.6925	0.7692	0.6090	0.6972	0.6149
COME	0.6334	0.6162	0.7041	0.6389	0.6327	0.7465	0.7573	**0.6451**	**0.9286**	0.5292	0.5585	0.5537	0.7262	0.7013	0.7764	0.6625	0.7317	0.6674
T^{2}A (Ours)	**0.6700**	**0.6748**	**0.7299**	0.6718	**0.6430**	**0.7565**	0.7594	0.6438	0.9279	**0.5370**	**0.5728**	**0.5657**	**0.7327**	**0.7320**	**0.7774**	**0.6830**	**0.7623**	**0.7117**

Table 2: Comparison with state-of-the-art TTA methods under the unknown data distributions and postprocessing techniques scenario across six datasets. Bold values denote the best performance for each metric.

Method	Color Contrast			Color Saturation			Resize			Gaussian Blur			Average		
	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP
CORE	0.8154 ± 0.02	0.8245 ± 0.04	0.9349 ± 0.02	0.8237 ± 0.03	0.8067 ± 0.06	0.9395 ± 0.02	0.8360 ± 0.02	0.8628 ± 0.03	0.9598 ± 0.01	0.8334 ± 0.02	0.8265 ± 0.05	0.9409 ± 0.02	0.8271 ± 0.01	0.830 ± 0.02	0.9438 ± 0.01
CORE + T^{2}A	0.8605 ± 0.01	0.8744 ± 0.02	0.9604 ± 0.01	0.8414 ± 0.02	0.8497 ± 0.04	0.9447 ± 0.01	0.8425 ± 0.01	0.8897 ± 0.03	0.9511 ± 0.01	0.849 ± 0.01	0.8662 ± 0.02	0.9539 ± 0.01	0.8491 ± 0.01	0.8725 ± 0.02	0.9525 ± 0.01
Effi.B4	0.6980 ± 0.07	0.8464 ± 0.04	0.9531 ± 0.01	0.8491 ± 0.02	0.7973 ± 0.07	0.9262 ± 0.03	0.8314 ± 0.02	0.8458 ± 0.04	0.9526 ± 0.01	0.8380 ± 0.02	0.7929 ± 0.06	0.9286 ± 0.03	0.8041 ± 0.02	0.8206 ± 0.02	0.9401 ± 0.01
Effi.B4 + T^{2}A	0.8531 ± 0.02	0.8638 ± 0.02	0.9542 ± 0.01	0.8271 ± 0.03	0.8311 ± 0.05	0.9372 ± 0.02	0.8302 ± 0.02	0.8355 ± 0.04	0.9485 ± 0.01	0.8442 ± 0.01	0.8670 ± 0.03	0.9515 ± 0.01	0.8382 ± 0.01	0.8592 ± 0.02	0.9478 ± 0.01
F3Net	0.8037 ± 0.03	0.8306 ± 0.05	0.9438 ± 0.02	0.8542 ± 0.02	0.8196 ± 0.07	0.9413 ± 0.02	0.8551 ± 0.03	0.8681 ± 0.03	0.9575 ± 0.01	0.8360 ± 0.02	0.8136 ± 0.05	0.9374 ± 0.02	0.8284 ± 0.01	0.8387 ± 0.02	0.9491 ± 0.01
F3Net + T^{2}A	0.8605 ± 0.01	0.8879 ± 0.02	0.9641 ± 0.01	0.8617 ± 0.02	0.8737 ± 0.04	0.9599 ± 0.02	0.8142 ± 0.01	0.8723 ± 0.03	0.9632 ± 0.01	0.8417 ± 0.02	0.8489 ± 0.02	0.9524 ± 0.01	0.8547 ± 0.01	0.8776 ± 0.01	0.9621 ± 0.01
RECCE	0.8080 ± 0.03	0.8189 ± 0.04	0.9386 ± 0.02	0.8348 ± 0.02	0.7915 ± 0.06	0.9283 ± 0.02	0.8137 ± 0.03	0.8338 ± 0.04	0.9484 ± 0.01	0.8360 ± 0.02	0.8136 ± 0.04	0.9374 ± 0.01	0.8231 ± 0.01	0.8144 ± 0.02	0.9382 ± 0.01
RECCE + T^{2}A	0.8502 ± 0.01	0.8698 ± 0.02	0.9587 ± 0.01	0.8291 ± 0.02	0.8432 ± 0.05	0.9406 ± 0.02	0.8408 ± 0.01	0.8426 ± 0.03	0.9495 ± 0.01	0.8417 ± 0.01	0.8689 ± 0.02	0.9524 ± 0.01	0.8405 ± 0.01	0.8561 ± 0.01	0.9503 ± 0.01

Table 3: Improvement of DF detectors to unknown postprocessing techniques. All these methods undergo five levels of intensity of postprocessing techniques.

Baselines

For TTA, we compare our T^{2}A method with SOTA methods, including TENT [Wang *et al.*, 2020], MEMO [Zhang *et al.*, 2022], EATA [Niu *et al.*, 2022], CoTTA [Wang *et al.*, 2022], LAME [Boudiaf *et al.*, 2022], ViDA [Liu *et al.*, 2023a], and COME [Zhang *et al.*, 2024]. For DF detection, we employ the following DF detectors: EfficientNetB4 [Tan and Le, 2019], F3Net [Qian *et al.*, 2020], CORE [Ni *et al.*, 2022], RECCE

Method	CelebDF-v1			CelebDF-v2			DFD			FSh			DFDCP			UADFV		
	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP	ACC	AUC	AP
CORE	0.6517	0.6828	0.7837	0.6467	0.6268	0.7527	0.8515	0.5319	0.8962	0.5050	0.5216	0.5151	0.7016	0.6465	0.7513	0.6090	0.7481	0.7331
CORE + $\texttt{T}^2\texttt{A}$	0.6558	0.6883	0.7599	0.7162	0.6571	0.7576	0.7946	0.6292	0.9291	0.5200	0.5103	0.4985	0.6721	0.6611	0.7565	0.6337	0.7805	0.7692
Effi.B4	0.6313	0.6613	0.7202	0.6428	0.5489	0.6556	0.8743	0.6310	0.9282	0.5292	0.5737	0.5504	0.6344	0.5023	0.6438	0.5576	0.6791	0.6363
Effi.B4 + $\texttt{T}^2\texttt{A}$	0.6415	0.6659	0.7542	0.6351	0.4347	0.7312	0.8259	0.6892	0.9452	0.5450	0.5944	0.5598	0.6475	0.5824	0.7040	0.6152	0.7107	0.6622
F3Net	0.6252	0.6541	0.7614	0.6563	0.6604	0.7681	0.8547	0.5507	0.9012	0.5228	0.5448	0.5644	0.6688	0.6528	0.7443	0.5843	0.7146	0.6866
F3Net + $\texttt{T}^2\texttt{A}$	0.6517	0.6655	0.7531	0.6602	0.6409	0.7283	0.7500	0.6097	0.9244	0.5128	0.5569	0.5647	0.6803	0.6961	0.7831	0.6563	0.7447	0.6877
RECCE	0.5804	0.5689	0.6804	0.6776	0.6175	0.7531	0.8177	0.6256	0.9356	0.5235	0.5367	0.5275	0.6672	0.6358	0.7333	0.6522	0.7194	0.6778
RECCE + $\texttt{T}^2\texttt{A}$	0.6578	0.6508	0.7233	0.6718	0.6725	0.7783	0.7296	0.6521	0.9346	0.5321	0.5512	0.5593	0.7032	0.7184	0.7949	0.7119	0.7910	0.7370

Table 4: Improvement of DF detectors to unknown data distributions and postprocessing techniques across six datasets.

[Cao *et al.*, 2022]. Details are provided in Appendix.

Implementation

For adaptation, we use Adam optimizer with learning rate $\eta = 1e-4$, batch size of 32. Other hyperparameters, including loss balancing ones α, β and gradient masking threshold ψ are selected by a grid-search manner from defined values in Table 5 (Appendix). The γ hyperparameter in Eq. 13 is set to 2.0. Details are provided in Appendix C.2.

5.2 Experimental Results

We design the experiments to assess the effectiveness of our method under two real-world scenarios: (i) unknown postprocessing techniques, and (ii) both unknown data distributions and postprocessing techniques. The primary distinction between these scenarios lies in the underlying data distribution assumptions. In the first scenario, we assume that test samples are drawn from a distribution similar to the training data and focus specifically on evaluating our method's resilience when adversaries intentionally employ unknown postprocessing techniques. The second scenario presents a more challenging setting where test samples stem from unknown distributions, allowing us to evaluate not only the method's resilience to postprocessing techniques but also its broader generalization across different data domains.

Comparison with SoTA TTA Approaches

We compare our $\texttt{T}^2\texttt{A}$ method with existing TTA approaches, with results presented in Table 1 and Table 2. Table 1 reports results when tested with unknown postprocessing techniques. Each technique is tested across five intensity levels, with the results showing averaged performance metrics. Detailed results for individual intensity levels are provided in Appendix D. The *Average* column denotes the mean across all postprocessing techniques, providing a holistic view of adaptation capability. We test our method and other TTA approaches on FF++ samples exposed to unseen postprocessing operations. From Table 1, we can observe that our method outperforms existing TTA approaches. On average, $\texttt{T}^2\texttt{A}$ improves the source DF detector by 2.93% on AUC. For the more challenging scenario - unknown data distributions and postprocessing techniques, Table 2 shows that $\texttt{T}^2\texttt{A}$ achieves SoTA results on 5 out of 6 datasets, including CelebDF-1, CelebDF-2, FSh, DFDCP, and UADFV, and the second-best result on DFD dataset.

Adaptability Improvement over Deepfake Detectors

To further demonstrate the effectiveness of our $\texttt{T}^2\texttt{A}$ method, we evaluate its capability to enhance the adaptability of DF detectors. We test the performance of DF detectors with and without the $\texttt{T}^2\texttt{A}$ method under two scenarios. For the first scenario, Table 3 indicates that: When integrated with $\texttt{T}^2\texttt{A}$, the performance of DF detectors measured by AUC is significantly improved, enhancing the resilience of these detectors against unseen postprocessing techniques. Particularly, our method shows substantial improvements of 4.25% for CORE, 3.86% for EfficientNet-B4, 3.89% for F3Net, and 4.17% for RECCE. Under the more challenging scenario, Table 4 presents results that $\texttt{T}^2\texttt{A}$ consistently enhances the generalization capability of DF detectors over unseen data distributions while maintaining robustness against postprocessing manipulations. For example, on the real-world DF benchmark DFDCP, our method improves the performance of RECCE to 8.26%, EfficientNet-B4 to 8%, F3Net to 4.33%, and CORE to 1.46%.

6 Conclusion

In this work, we introduce $\texttt{T}^2\texttt{A}$, which improves the adaptability of DF detectors across two challenging scenarios: unknown postprocessing techniques and data distributions during inference time. Instead of solely relying on EM, $\texttt{T}^2\texttt{A}$ enables the model to explore alternative options before decision-making through NL with noisy pseudo-labels. We also provide a theoretical analysis to demonstrate that the proposed objective exhibits complementary behavior to EM. Through experiments, we show that $\texttt{T}^2\texttt{A}$ achieves higher adaptation performance compared to SoTA TTA approaches. Furthermore, when integrated with $\texttt{T}^2\texttt{A}$, the resilience and generalization of DF detectors can be significantly improved without requiring additional training data or architectural modifications, making it particularly valuable for real-world deployments. However, since our method is based on backpropagation for updating parameters at inference time, it only works with end-to-end DF detectors that allow gradient flow throughout the model.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183.

References

[Boudiaf *et al.*, 2022] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8344–8353, 2022.

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

[Chen *et al.*, 2022] Liang Chen, Yong Zhang, Yibing Song, Jue Wang, and Lingqiao Liu. Ost: Improving generalization of deepfake detection via one-shot test-time training. *Advances in Neural Information Processing Systems*, 35:24597–24610, 2022.

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

[Corvi *et al.*, 2023] Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 973–982, 2023.

[Dolhansky, 2019] B Dolhansky. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019.

[Fang *et al.*, 2024] Hao Fang, Ajian Liu, Haocheng Yuan, Junze Zheng, Dingheng Zeng, Yanhong Liu, Jiankang Deng, Sergio Escalera, Xiaoming Liu, Jun Wan, et al. Unified physical-digital face attack detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 749–757, 2024.

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

[Ghosh *et al.*, 2017] Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[Google, 2019] Google. Contributing data to deepfake detection research, 2019. Accessed on 11 December 2024.

[He *et al.*, 2024] Xianhua He, Dashuang Liang, Song Yang, Zhanlong Hao, Hui Ma, Binjie Mao, Xi Li, Yao Wang, Pengfei Yan, and Ajian Liu. Joint physical-digital facial attack detection via simulating spoofing clues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 995–1004, 2024.

[Kim *et al.*, 2019] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 101–110, 2019.

[Kim *et al.*, 2021] Youngdong Kim, Juseung Yun, Hyounguk Shon, and Junmo Kim. Joint negative and positive learning for noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9442–9451, 2021.

[Layton *et al.*, 2024] Seth Layton, Tyler Tucker, Daniel Olszewski, Kevin Warren, Kevin Butler, and Patrick Traynor. {SoK}: The good, the bad, and the unbalanced: Measuring structural limitations of deepfake media datasets. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1027–1044, 2024.

[Li *et al.*, 2018] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International workshop on information forensics and security (WIFS)*, pages 1–7. Ieee, 2018.

[Li *et al.*, 2020a] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5074–5083, 2020.

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

[Li *et al.*, 2024] Jingjing Li, Zhiqi Yu, Zhekai Du, Lei Zhu, and Heng Tao Shen. A comprehensive survey on source-free domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[Liang *et al.*, 2024] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, pages 1–34, 2024.

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

[Liu *et al.*, 2023a] Jiaming Liu, Senqiao Yang, Peidong Jia, Renrui Zhang, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang. Vida: Homeostatic visual domain adapter for continual test time adaptation. In *International Conference on Learning Representations*, 2023.

[Liu *et al.*, 2023b] Jiawei Liu, Jingyi Xie, Yang Wang, and Zheng-Jun Zha. Adaptive texture and spectrum clue mining for generalizable face forgery detection. *IEEE Transactions on Information Forensics and Security*, 2023.

[Liu *et al.*, 2024] Ajian Liu, Shuai Xue, Jianwen Gan, Jun Wan, Yanyan Liang, Jiankang Deng, Sergio Escalera, and Zhen Lei. Cfpl-fas: Class free prompt learning for generalizable face anti-spoofing. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 222–232, 2024.

[Ma *et al.*, 2020] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning*, pages 6543–6553. PMLR, 2020.

[Nguyen-Le *et al.*, 2024a] Hong-Hanh Nguyen-Le, Van-Tuan Tran, Dinh-Thuc Nguyen, and Nhien-An Le-Khac. Deepfake generation and proactive deepfake defense: A comprehensive survey. *Authorea Preprints*, 2024.

[Nguyen-Le *et al.*, 2024b] Hong-Hanh Nguyen-Le, Van-Tuan Tran, Dinh-Thuc Nguyen, and Nhien-An Le-Khac. Passive deepfake detection across multi-modalities: A comprehensive survey. *arXiv preprint arXiv:2411.17911*, 2024.

[Ni *et al.*, 2022] Yunsheng Ni, Depu Meng, Changqian Yu, Chengbin Quan, Dongchun Ren, and Youjian Zhao. Core: Consistent representation learning for face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12–21, 2022.

[Niu *et al.*, 2022] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022.

[Niu *et al.*, 2023] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *The Eleventh International Conference on Learning Representations*, 2023.

[Ojha *et al.*, 2023] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023.

[Pan *et al.*, 2023] Kun Pan, Yifang Yin, Yao Wei, Feng Lin, Zhongjie Ba, Zhenguang Liu, Zhibo Wang, Lorenzo Cavallaro, and Kui Ren. Dfil: Deepfake incremental learning by exploiting domain-invariant forgery clues. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8035–8046, 2023.

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

[Ross and Dollár, 2017] T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2980–2988, 2017.

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

[Schneider *et al.*, 2020] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in neural information processing systems*, 33:11539–11551, 2020.

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

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

[Wang *et al.*, 2020] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[Wang *et al.*, 2022] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[Yan *et al.*, 2023] Zhiyuan Yan, Yong Zhang, Xinhang Yuan, Siwei Lyu, and Baoyuan Wu. Deepfakebench: A comprehensive benchmark of deepfake detection. *arXiv preprint arXiv:2307.01426*, 2023.

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

[Ye *et al.*, 2023] Xichen Ye, Xiaoqiang Li, Tong Liu, Yan Sun, Weiqin Tong, et al. Active negative loss functions for learning with noisy labels. *Advances in Neural Information Processing Systems*, 36:6917–6940, 2023.

[Zhang *et al.*, 2022] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

[Zhang *et al.*, 2024] Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. Come: Test-time adaption by conservatively minimizing entropy. *arXiv preprint arXiv:2410.10894*, 2024.

[Zhou *et al.*, 2021] Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. Asymmetric loss functions for learning with noisy labels. In *International conference on machine learning*, pages 12846–12856. PMLR, 2021.