What to Remember: Self-Adaptive Continual Learning for Audio Deepfake Detection

Xiaohui Zhang[1, 2], **Jiangyan Yi**[1 *], **Chenglong Wang**[1, 4], **Chu Yuan Zhang**[1], **Siding Zeng**[1], **Jianhua Tao**[3]

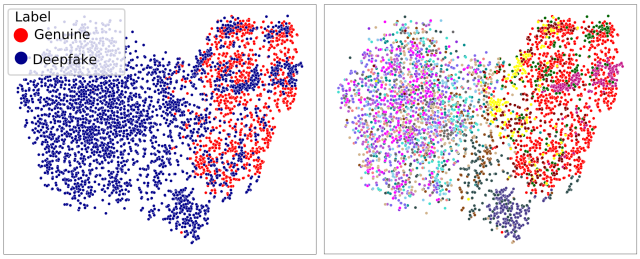
[1] State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Computer and Information Technology, University of Beijing Jiaotong, Beijing, China
[3] Department of Automation, Tsinghua University, Beijing, China
[4] University of Science and Technology of China, Beijing, China.

Abstract

The rapid evolution of speech synthesis and voice conversion has raised substantial concerns due to the potential misuse of such technology, prompting a pressing need for effective audio deepfake detection mechanisms. Existing detection models have shown remarkable success in discriminating known deepfake audio, but struggle when encountering new attack types. To address this challenge, one of the emergent effective approaches is continual learning. In this paper, we propose a continual learning approach called Radian Weight Modification (RWM) for audio deepfake detection. The fundamental concept underlying RWM involves categorizing all classes into two groups: those with compact feature distributions across tasks, such as genuine audio, and those with more spread-out distributions, like various types of fake audio. These distinctions are quantified by means of the in-class cosine distance, which subsequently serves as the basis for RWM to introduce a trainable gradient modification direction for distinct data types. Experimental evaluations against mainstream continual learning methods reveal the superiority of RWM in terms of knowledge acquisition and mitigating forgetting in audio deepfake detection. Furthermore, RWM's applicability extends beyond audio deepfake detection, demonstrating its potential significance in diverse machine learning domains such as image recognition.

Introduction

In recent years, the advancement of speech synthesis and voice conversion technologies has blurred the line between reality and fabrication (Wang et al. 2018, 2021). This has significantly amplified concerns about the potential misuse of audio deepfakes – synthesized audio that closely mimics genuine human speech, posing serious threats to social stability and public interests. Consequently, the pursuit of reliable audio deepfake detection mechanisms has garnered increasing attention across research domains. The landscape of audio deepfake detection has witnessed substantial growth, catalyzed by a series of challenges such as the ASVspoof challenge (Wu et al. 2015; Kinnunen et al. 2017; Todisco et al. 2019; Yamagishi et al. 2021) and the Audio Deep Synthesis Detection (ADD) challenge (Yi et al. 2022, 2023). These competitions have underscored the crucial role of deep neural networks in achieving remarkable success in audio deepfake detection. With the advent of large-scale pre-trained models, audio deepfake detection has experienced significant breakthroughs, boasting impressive performance on publicly available datasets (Tak et al. 2022; Martín-Doñas and Álvarez 2022; Lv et al. 2022; Wang and Yamagishi 2021).



(a) (b)

Figure 1: The t-SNE (Van der Maaten and Hinton 2008) visualization of genuine and various deepfake audio in the ASVspoof2019LA dataset visualized using Linear Frequency Cepstral Coefficients (LFCC) feature (Sahidullah, Kinnunen, and Hanilçi 2015). All sentences are first blocked with a 20 ms Hamming window with a 10 ms shift, and then unify the frame numbers of all features into 100. (a) shows the comparison of feature distribution between genuine and deepfake audio, and (b) is the feature visualization of all audio types, including genuine and various deepfake audio.

However, existing detection models face a critical challenge, namely degraded performance when dealing with new types of deepfake audio. This challenge underscores the need for strategies to enhance the adaptability and resilience of audio deepfake detection models. To this end, two primary approaches have emerged. (Zhang et al. 2021; Zhang, Jiang, and Duan 2021). The first approach involves extracting more discriminative features and developing robust model architectures to bolster the robustness of detection models against new types of deepfake audio. This strategy proves valuable in scenarios where access to data representing new types of deepfake attacks is not accessible, such

*Corresponding Author

as during the initial stages of encountering an unknown attack. In contrast, the second approach leverages the principles of continual learning, enabling deepfake detection models to sequentially learn from newly collected data (Ma et al. 2021). This method capitalizes on the advantages of maintaining proficiency in detecting known deepfake types while simultaneously enhancing detection accuracy for emerging, unencountered attack.

For those general and widely-used continual learning algorithms, experience replay (Chaudhry et al. 2019; Prabhu, Torr, and Dokania 2020) has demonstrated success across diverse domains. However, its applicability in audio deepfake detection is challenged by the acquisition of old data. Alternatively, regularization-based continual learning methods offer a more flexible approach by obviating the need for prior data. Among these methods, the Detecting Fake Without Forgetting (DFWF) (Ma et al. 2021) approach stands as the pioneering solution tailored specifically for audio deepfake detection. While DFWF exhibits notable strengths in overcoming forgetting, it still deteriorates learning performance in the context of new attack types compared to finetuning.

To address this limitation, we propose a continual learning approach named Radian Weight Modification (RWM) for audio deepfake detection. Most fake audio detection datasets are under clean conditions, where the genuine audio has a more similar feature distribution than the fake audio (Ma et al. 2021), as shown in Fig. 1, and they can be seen as a whole from the same dataset or generated by the experienced-replay method on different tasks. From the view of replay, data replayed on the new task should be trained without any additional modification. Based on the above inference, it is more effective for genuine audio on new datasets to be trained with as little modification as possible. Drawing inspiration from the disparities in feature distribution between the genuine and various types of fake audio, RWM splits all classes into two groups and leverages a self-attention mechanism to enable the model to learn optimal gradient modification directions based on the current input batch. Specifically, the algorithm adapts the gradient direction based on the feature similarity between different tasks. By categorizing classes into two groups—those with compact feature distributions across tasks and those with more disparate distributions—we employ distinct strategies. When confronted with data featuring distinct characteristics across tasks, such as various types of fake audio, the algorithm guides the model to adopt a direction orthogonal to the previous data plane, ensuring preservation of learned knowledge during adaptation to new deepfake algorithms. Conversely, for data exhibiting similar features, exemplified by genuine audio, the algorithm encourages the model to learn a gradient modification direction aligned with the previous data plane, thus minimizing the interference from gradient modification. The experiments conducted on audio deepfake detection demonstrate the superiority of our proposed approach over several mainstream continual learning methods, including Elastic Weight Consolidation (EWC) (Kirkpatrick et al. 2017), Learning without Forgetting (LwF) (Li and Hoiem 2017), Orthogonal Weight Modification (OWM) (Zeng et al. 2019) and DFWF, in terms of knowledge acquisition and mitigating forgetting. In addition, RWM can also be easily generalized to other machine learning fields. Our experiments conducted on image recognition underscore its potential significance across diverse machine learning domains. Furthermore, the utilization of the RWM method obviates the requirement for accessing previously stored data, thereby conferring a wide-ranging applicability in diverse domains of practical significance.

In summary, we make the following contributions.

- We propose a continual learning approach for audio deepfake detection that enables the model to learn discriminative information for classification on each task while autonomously optimizing the gradient direction for continuous learning across different tasks based on the similarity of feature distributions.
- Although our method is inspired by the difference of feature distribution in audio deepfake detection, RWM can be applied to various machine learning fields, such as image recognition, and is not limited to any specific domain.

The code of the RWM has been uploaded in the supplemental material. In the foreseeable future, we plan to make the code of our method publicly available to facilitate its adoption and further research.

Background

The orthogonal weight modification (OWM) algorithm is a valuable approach employed to address the issue of catastrophic forgetting in continual learning. Its primary objective is to modify the weight direction on the new task in such a way that the resulting modified direction $\boldsymbol{P}$ becomes orthogonal to the subspace spanned by all inputs from the previous task. To construct the orthogonal projector, an iterative method resembling the Recursive Least Squares (RLS) algorithm (Shah, Palmieri, and Datum 1992) is utilized, which needs a minimal number of previous samples.

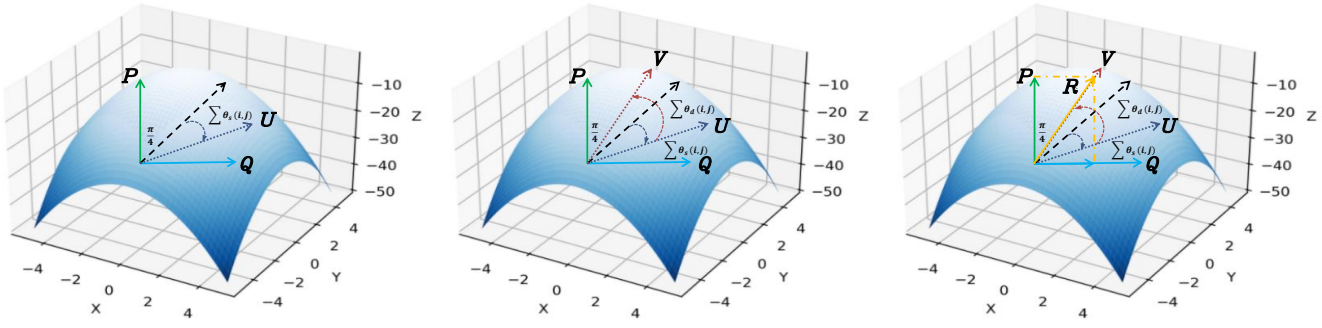
We consider a feed-forward network comprising $L+1$ layers, denoted by the index $l = 0, 1, \cdots, L$, each employing the same activation function $g(\cdot)$. The symbol $\overline{\mathbf{x}}_l(i,j) \in \mathbb{R}^s$ represents the output of the l-th layer corresponding to the mean of the i-th batch inputs obtained from the j-th dataset, with $\overline{\mathbf{x}}_l(i,j)^T$ denoting the transpose matrix of $\overline{\mathbf{x}}_l(i,j)$. The computation of the modified direction $\boldsymbol{P}$ can be expressed as follows:

$$\begin{aligned} \boldsymbol{P}_l(i,j) &= \boldsymbol{P}_l(i-1,j) - \mathbf{k}_l(i,j)\overline{\mathbf{x}}_{l-1}(i,j)^T \boldsymbol{P}_l(i-1,j) \\ \mathbf{k}_l(i,j) &= \frac{\boldsymbol{P}_l(i-1,j)\overline{\mathbf{x}}_{l-1}(i,j)}{\alpha + \overline{\mathbf{x}}_{l-1}(i,j)^T \boldsymbol{P}_l(i-1,j)\overline{\mathbf{x}}_{l-1}(i,j)} \end{aligned} \tag{1}$$

where α represents a hyperparameter that decays with the number of tasks.

Proposed Method

In continual learning, some categories have a more compact feature distribution that has similar features across different tasks. For instance, in audio deepfake detection, genuine audio from different datasets has a more compact feature distribution than fake audio. To better leverage this phenomenon, we can modify the direction of the gradient based



(a) (b) (c)

Figure 2: The calculation process for the gradient modification direction in RWM algorithm. Firstly, we partition all categories into two groups based on their feature similarity across different tasks. The total LRR for all samples in the similar group is represented as $\sum \theta_s(i,j)$, while the total LRR for all samples in the dissimilar group is represented as $\sum \theta_d(i,j)$. As illustrated in Fig 2a and Fig 2b, the RWM algorithm rotates from the direction of $\frac{\pi}{4}$ to the $\boldsymbol{Q}$ direction by $\sum \theta_s(i,j)$ and then towards the $\boldsymbol{P}$ direction by $\sum \theta_d(i,j)$ to obtain the target direction $\boldsymbol{R}$, as shown in Fig 2c. During continual learning, the LRR for all samples is autonomously optimized through a self-attention mechanism.

on whether or not that category shares similar features across tasks. For categories with dissimilar features across tasks, we can modify the gradient for this portion of the data in the direction orthogonal to the data plane of the old task. This ensures that learning from this portion of data in the new task does not disturb the knowledge learned from the old task. For categories with similar features across tasks, we can treat them as replay data generated from the experience-replay algorithm, which means that it is reasonable to minimize the modification of the gradient calculated from these data as much as possible.

Class Regrouping

We first consider a feed-forward network like that described in Sec. background, a deep neural network with $L+1$ layers, where i is the index of the input batch, and j is the index of the current task.

First, we compute the compactness of all categories by the average cosine distance between each two samples across all tasks, as shown in Eq 2:

$$d_r = \frac{1}{N_r} \sum_{m=1}^{N_r} \sum_{n=1}^{N_r} cos_{dis}(x_m, x_n) \quad x_m, x_n \in class_r \tag{2}$$

where $r \in [1, R]$ is the class id, and N_r represents the number of samples in $class_r$ and cos_{dis} is the computing function of cosine distance. while the remaining classes are allocated to group $\mathbb{D}$, as described in Eq 3:

$$\begin{aligned} \mathbb{S} &= \{class_1, class_2, ...class_{r_s}\} \\ \mathbb{D} &= \{class_{r_s}, class_{r_s+1}, ...class_R\} \end{aligned} \tag{3}$$

We can reasonably assume that $d_1 < d_2 < d_3 < ...d_R$. Based on this assumption, we introduce a hyperparameter r_s, which signifies the allocation of the classes with the smallest d_r values up to r_s to group $\mathbb{S}$.

Self-Optimizing Direction Modification

After splitting all classes into two groups, we calculate the modification direction $\boldsymbol{P}$ as Eq 1. The modification direction $\boldsymbol{P}$ is a square matrix, which is orthogonal to the data plane of the old task. Then we introduce another modification direction $\boldsymbol{Q}$, which is also a square matrix and orthogonal to $\boldsymbol{P}$. The new direction $\boldsymbol{Q}$ can be calculated as Eq 4:

$$\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{P}(\boldsymbol{P}^T\boldsymbol{P})^{-1}\boldsymbol{P}^T \tag{4}$$

where the projector $\boldsymbol{P}$, which is orthogonal to the subspace spanned by all previous inputs, can be calculated as Eq 1 and $\boldsymbol{I}$ is an identity matrix. The construction of the orthogonal projector $\boldsymbol{Q}$ is mathematically sound (Haykin 2002; Ben-Israel and Greville 2003; Bengio and LeCun 2007).

To make the model learn the adaptive modification direction automatically, a self-attention (SA) mechanism is then introduced before the classifier to obtain the attention score for each sample in a batch. The attention scores $\delta_t(i,j)$ can be calculated as Eq 5:

$$[\delta_1(i,j), \delta_2(i,j), \delta_3(i,j), ...\delta_b(i,j)] = f_{SA}(h_l(i,j)) \tag{5}$$

where $h(i,j)$ represents the hidden state of this batch before the classifier and b represents the batch size. Then, all attention scores are normalized according to Eq 6.

$$\delta_t(i,j) = \frac{\exp \delta_t(i,j)}{\sum_{t=1}^{b} \exp \delta_t(i,j)} \tag{6}$$

We assume that each attention score δ_t can be expressed as the sine value of an angle θ_t, then according to Eq 7, the sum of all θ_t is greater than 0 and less than $\frac{\pi}{2}$.

$$0 < \sin(\sum_{t=1}^{b} \theta_t(i,j)) < \sum_{t=1}^{b} \sin \theta_t(i,j) \tag{7}$$

where $\sum_{t=1}^{b} \sin \theta_t(i,j) = \sin \frac{\pi}{2}$. Our algorithm adaptively adjusts the gradient modification direction for each sample

based on the attention score. The modification direction can be considered as a direction learned by the model itself, as the attention scores will be continuously optimized during model training. We name the angle θ_t as the learned rotated radians (LRR), which can be calculated according to Eq 8.

$$\theta_t(i,j) = \sin^{-1}(\delta_t(i,j)) \tag{8}$$

For those samples belongs to a class in $\mathbb{S}$, such as genuine audio in audio deepfake detection, RWM first calculates the sum of their LRR in the ith batch of the jth task, denoted as $\sum \theta_s(i,j)$. For those samples belongs to a class in $\mathbb{D}$, such as various types of fake audio, RWM also calculates the sum of their LRR, denoted as $\sum \theta_d(i,j)$.

For those samples $\in \mathbb{S}$, as we mentioned above, we should reduce the impact of the gradient modification on them. Therefore, the gradient modification direction starts with $\frac{\pi}{4}$ and rotates towards $\boldsymbol{Q}$ direction by $\sum \theta_s(i,j)$, obtaining a new direction $\boldsymbol{U}$, as shown in Fig 2a. Next, we consider those samples that have large differences in features across different tasks. The gradient modification direction starts from $\boldsymbol{U}$ and rotates towards the $\boldsymbol{P}$ direction by $\sum \theta_d(i,j)$, obtaining a new direction $\boldsymbol{V}$, as shown in Fig 2b. Here, direction $\boldsymbol{P}$ is orthogonal to the data plane of the old task. Thus, the closer the modification direction is to $\boldsymbol{P}$, the less interference will cause to the already learned knowledge when training on new dataset. Conversely, the closer the modification direction is to $\boldsymbol{Q}$, which is orthogonal to the direction $\boldsymbol{P}$, the smaller modification will be introduced during learning on a new dataset, making this process more similar to a common gradient backpropagation. After all direction modifications, we obtain the final gradient modification represented by the final LRR $\theta_f(i,j)$ as:

$$\theta_f(i,j) = \frac{\pi}{4} + \frac{\sum \theta_s(i,j) - \sum \theta_d(i,j)}{2} \tag{9}$$

where $\theta_f(i,j)$ will be optimized during the training process, so it can be viewed as a modification direction learned by the model itself. Here, we use $\frac{(\ldots)}{2}$ to ensure that the value range of final LRR $\theta_f(i,j)$ is greater than 0 and less than $\frac{\pi}{2}$, where the trigonometric functions are monotonous.

After calculating the final LRR, the final gradient modification direction $\boldsymbol{R}$ can be easily computed based on trigonometric functions. From the Fig. 2c, the final LRR is the angle between the direction matrix $\boldsymbol{P}$ and $\boldsymbol{R}$, so the direction $\boldsymbol{R}$ can be calculated as Eq 10.

$$\boldsymbol{R} = u(\frac{\boldsymbol{P}}{||\boldsymbol{P}||} + \beta\frac{\boldsymbol{Q}}{||\boldsymbol{Q}||}) \quad where\, u = ||\boldsymbol{P}|| \tag{10}$$

In Eq 10, $||\boldsymbol{P}||$ and $||\boldsymbol{Q}||$ represent the norms of $\boldsymbol{P}$ and $\boldsymbol{Q}$, respectively. The parameter β is defined as the tangent value of LRR, as shown in Eq 11:

$$\beta = \tan\theta_f \tag{11}$$

and the BP process of RWM can be written as Eq 12:

$$\begin{aligned} \boldsymbol{W}_l(i,j) &= \boldsymbol{W}_l(i-1,j) + \gamma(i,j)\Delta\boldsymbol{W}_l^{BP}(i,j) \qquad j=1 \\ \boldsymbol{W}_l(i,j) &= \boldsymbol{W}_l(i-1,j) + \gamma(i,j)\boldsymbol{G}_l(i,j) \qquad j>1 \\ \boldsymbol{G}_l(i,j) &= \boldsymbol{R}_l(i,j)\Delta\boldsymbol{W}_l^{BP}(i,j) \end{aligned} \tag{12}$$

Algorithm 1: Radian Weight Modification

Require: Training data from different datasets, γ (learning rate), r_s (group split proportion rate).
for every class *r* **do**
  $d_r = \frac{1}{N_r}\sum_{m=1}^{N_r}\sum_{n=1}^{N_r} cos_{dis}(x_m, x_n)$
end for
$\boldsymbol{H} = \mathbf{Sort}(d_1, d_2, d_3, ...d_R)$ ▷ $\boldsymbol{H}[0] \leq ...\boldsymbol{H}[R-1]$
$\mathbb{S} = \{class_r \textbf{ for } d_r \textbf{ in } \boldsymbol{H}[:r_s]\}$
$\mathbb{D} = \{class_r \textbf{ for } d_r \textbf{ in } \boldsymbol{H}[r_s:]\}$
for every dataset *j* **do**
  for every batch *i* **do**
    if $j = 1$ **then**
      $\boldsymbol{W}_l(i,j) = \boldsymbol{W}_l(i-1,j) + \gamma(i,j)\Delta\boldsymbol{W}_l^{BP}(i,j)$
    else
      $\mathbf{k}(i,j) = \frac{\boldsymbol{P}_l(i-1)\overline{\mathbf{x}}_{l-1}(i,j)}{\alpha + \overline{\mathbf{x}}_{l-1}(i,j)^T\boldsymbol{P}_l(i-1,j)\overline{\mathbf{x}}_{l-1}(i,j)}$
      $\boldsymbol{P}_l(i,j) = \boldsymbol{P}_l(i-1,j) - \mathbf{k}(i,j)\overline{\mathbf{x}}_{l-1}(i,j)^T\boldsymbol{P}_l(i-1,j)$
      $\boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{P}(\boldsymbol{P}^T\boldsymbol{P})^{-1}\boldsymbol{P}^T$
      $[\delta_1(i,j), \delta_2(i,j), \delta_3(i,j), ...\delta_b(i,j) = f_{SA}(h_l(i,j))$
      $\sum \theta_s(i,j) = 0; = \sum \theta_d(i,j) = 0$
      for every sample *t* **do**
        $\delta_t(i,j) = \frac{\exp \delta_t(i,j)}{\sum_{t=1}^{b} \exp \delta_t(i,j)}$
        $\theta_t(i,j) = sin^{-1}(\delta_t(i,j))$ ▷ $0 < \sum_{t=1}^{b} \theta_t(i,j) < \frac{\pi}{2}$
        if class of $\theta_t(i,j) \in \mathbb{S}$ **then**
          $\sum \theta_s(i,j) += \theta_t(i,j)$
        else
          $\sum \theta_d(i,j) += \theta_t(i,j)$
        end if
      end for
      $\theta_f(i,j) = \frac{\pi}{4} + \frac{\sum \theta_s(i,j) - \sum \theta_d(i,j)}{2}$
      $\boldsymbol{R} = u(\frac{\boldsymbol{P}}{||\boldsymbol{P}||} + \beta\frac{\boldsymbol{Q}}{||\boldsymbol{Q}||})$ ▷ $u = ||\boldsymbol{P}||; \beta = \tan\theta_f$
      $\boldsymbol{W}_l(i,j) = \boldsymbol{W}_l(i-1,j) + \gamma(i,j)\boldsymbol{G}_l(i,j)$
      $\boldsymbol{G}_l(i,j) = \boldsymbol{R}_l(i,j)\Delta\boldsymbol{W}_l^{BP}(i,j)$
    end if
  end for
end for

Demonstrative Analysis

We demonstrate the calculation formula of the direction angle θ_f for both audio deepfake detection and image recognition on the classical continual learning image recognition benchmark, CLEAR(Lin et al. 2021). For audio deepfake detection, the compactness using pre-trained Wav2vec 2.0 (Baevski et al. 2020) feature of genuine audio $d_{genuine}$ and various types of fake audio d_{fake} in the ASVspoof2019LA (Todisco et al. 2019) dataset are 0.010 and 0.062, respectively. Obviously, the r_s is 1. Under this condition, the θ_f can be written as Eq 13.

$$\theta_f(i,j) = \frac{\pi}{4} + \frac{\sum \theta_1(i,j) - \sum \theta_2(i,j)}{2} \tag{13}$$

For image recognition, we calculate the in-class cosine distinance of all categories in the CLEAR dataset. The compactness d_r of all categories are $\{soccer : 0.18, hockey : 0.18, bus : 0.20, baseball : 0.23, cospaly : 0.25, racing : 0.27, dress : 0.27, camera : 0.28, laptop : 0.29, sweater : 0.32, background : 0.40\}$. Therefore, if the hyperparameter

r_s is set as 4, the θ_f for this benchmark can be written as Eq 14.

$$\theta_f(i,j) = \frac{\pi}{4} + \frac{\sum_{c=1}^{4} \sum \theta_c(i,j) - \sum_{c=5}^{11} \sum \theta_c(i,j)}{2} \quad (14)$$

In the Eq 13, $\theta_1(i,j)$ and $\theta_2(i,j)$ represent the LRR of samples belonging to the genuine and fake categories in ith batch of jth task, respectively. In the Eq 14, $\theta_c(i,j)$ ranges from 1 to 4, representing the LRR assigned to samples belonging to $\{soccer, hockey, bus, baseball\}$ and $\theta_c(i,j)$ ranges from 5 to 11, representing the LRR assigned to samples belonging to other classes.

Experiments

A series of experiments were undertaken to evaluate the efficacy of our methodology in both the audio deepfake detection and image recognition domains. In the field of audio deepfake detection, our focus was on detecting fake audio across multiple widely used datasets specifically designed for incremental synthetic algorithms audio deepfake detection. For image recognition, we employed a well-established continual learning benchmark known as CLEAR.

Audio Deepfake Detection

Datasets We evaluate our approach on three fake audio datasets: ASVspoof2019LA ($\mathbf{S}$) (Todisco et al. 2019), ASVspoof2015 ($\mathbf{T_1}$) (Wu et al. 2015), and In-the-Wild ($\mathbf{T_2}$) (Müller et al. 2022). The $\mathbf{S}$ dataset includes attacks from four TTS and two VC algorithms. The bonafide audio is collected from the VCTK corpus (Veaux et al. 2017). The $\mathbf{T_1}$ dataset contains genuine and synthetic speech recordings from 106 speakers. The $\mathbf{T_2}$ dataset contains deep fake and genuine audio from 58 politicians and public figures collected from publicly available sources. We constructed the training set of $\mathbf{T}_2$ by using one-third of the fake audio and an equal number of genuine audio, while the remaining audio was used as the evaluation set. The Equal Error Rate (EER) (Wu et al. 2015), which is widely used for audio deepfake detection, is applied to evaluate the performance. The detailed statistics of the datasets are presented in Table 7 in our supplementary material.

Experimental Setup We employ the Wav2vec 2.0 model (Baevski et al. 2020) as the feature extractor, while the self-attention convolutional neural network (S-CNN) serves as the classifier. The parameters of Wav2vec 2.0 are loaded from the pre-trained model XLSR-53 (Conneau et al. 2020). The S-CNN classifier consists of three 1D-Convolution layers, one self-attention layer, and two fully connected layers in its forward process. The input dimension of the first convolution layer is 256, and all convolution layers have a hidden dimension of 80. A kernel size of 5 and a stride of 1 are applied. The fully connected layers have a hidden dimension of 80 and the output dimension of 2.

Training Details We finetune the XLSR-53 and S-CNN using the Adam optimizer with a learning rate γ of 0.0001 and a batch size of 2. To evaluate the performance of our proposed method for audio deepfake detection, we compared it against three widely used continual learning methods, as well as finetuning and the first continual learning method for audio deepfake detection (DFWF)(Ma et al. 2021). In addition, we present the results of training on all datasets (Replay-All) that are considered to be the lower bound to all continual learning methods we mentioned (Parisi et al. 2019). All results are (re)produced by us and averaged over 7 runs with standard deviations.

Method	$\mathbf{S} \rightarrow \mathbf{T_1}$		$\mathbf{S} \rightarrow \mathbf{T_1} \rightarrow \mathbf{T_2}$		
	$\mathbf{S}$	$\mathbf{T_1}$	$\mathbf{S}$	$\mathbf{T_1}$	$\mathbf{T_2}$
Baseline	0.258	24.532	0.258	24.532	91.473
Replay-All	0.406	0.201	2.344	7.253	1.003
Finetune	7.324	0.510	4.636	28.765	2.543
EWC	2.832	0.570	8.684	12.397	3.722
OWM	2.448	0.540	4.756	10.132	3.647
LwF	3.123	0.343	7.505	9.547	1.540
DFWF	1.849	0.689	6.211	9.672	6.478
RWM (Ours)	**0.438**	**0.212**	**2.896**	**7.693**	**1.161**

(a)

Method	$\mathbf{S} \rightarrow \mathbf{T_2}$		$\mathbf{S} \rightarrow \mathbf{T_2} \rightarrow \mathbf{T_1}$		
	$\mathbf{S}$	$\mathbf{T_2}$	$\mathbf{S}$	$\mathbf{T_2}$	$\mathbf{T_1}$
Baseline	0.258	91.473	0.258	91.473	24.532
Replay-All	2.740	2.160	5.197	13.893	0.842
Finetune	20.976	4.978	13.362	35.368	0.876
EWC	8.039	5.615	7.343	29.516	0.933
OWM	8.130	5.065	6.675	26.619	1.042
LwF	6.453	4.998	10.035	32.409	0.897
DFWF	4.324	6.275	6.994	24.697	1.332
RWM (Ours)	**3.665**	**2.247**	**5.616**	**15.993**	**0.861**

(b)

Table 1: The EER(%) of our method compared with various methods across different tasks. All experiments are trained using the training set in order to $\mathrm{S} \rightarrow \mathrm{T_k}$ (e.g., $\mathrm{S} \rightarrow \mathrm{T_1} \rightarrow \mathrm{T_2}$) and are evaluated using the evaluation set of each dataset.

Comparison With Other Methods In this study, we evaluated the performance of our proposed method for audio deepfake detection in both two-dataset and three-dataset continual learning scenarios and compared it with several other methods. The results showed that our method achieved the best detection performance compared to other methods in both scenarios, even in the presence of significant acoustic environment differences (Table 1b). In the three-dataset continual learning scenario, our method still achieved the best performance on both old and new datasets. These results suggest that our method is effective and robust for audio deepfake detection in various continual learning scenarios with different levels of acoustic environment differences.

Comparing to Others With Limited Training Samples To verify the sensitivity of our method to the amount of training data for new tasks in continual learning, we conducted experiments with different numbers of training data for new tasks and compared our method with others, as shown in Table 2. The results show that our method performs better than other continual learning methods on new

Method	10		100		1000	
	S	**T_2**	**S**	**T_2**	**S**	**T_2**
Baseline	0.258	91.473	0.258	91.473	0.258	91.473
Replay-All	0.897	15.326	1.203	4.198	2.715	2.162
Finetune	8.223	19.385	16.058	6.503	16.437	4.999
EWC	7.301	18.599	7.666	8.977	6.148	7.576
OWM	7.021	19.684	8.229	8.177	7.860	4.364
LwF	8.019	19.673	5.750	5.950	4.037	6.391
DFWF	6.894	19.992	4.246	9.879	5.129	8.864
RWM (Ours)	**2.463**	**17.252**	**1.507**	**5.305**	**7.670**	**3.921**

Table 2: The EER(%) of limited samples experiments. All experiments are first trained using the training set of S and then trained on 1000, 100, and 10 samples of the training set of T_2 respectively. All experiments are evaluated using the evaluation set on S and T_2.

Method	**S → T_1**		**S → T_1 → T_2**		
	S	**T_1**	**S**	**T_1**	**T_2**
Baseline	0.258	24.532	0.258	24.532	91.473
RWM(Ours)	**0.438**	**0.212**	**2.896**	**7.693**	**1.161**
–LRR	2.448	0.540	4.756	10.132	3.647
–WM	7.324	0.510	4.636	28.765	2.543

(a)

Method	**S → T_2**		**S → T_2 → T_1**		
	S	**T_2**	**S**	**T_2**	**T_1**
Baseline	0.258	91.473	0.258	91.473	24.532
RWM(Ours)	**3.665**	**2.247**	**5.616**	**15.993**	**0.861**
–LRR	8.130	5.065	6.675	26.619	1.042
–WM	20.976	4.978	13.362	35.368	0.876

(b)

Table 3: The EER(%) on evaluation sets of the ablation studies. All experiments are trained using the training set in order to S → T_k (e.g., S → T_1 → T_2) and are evaluated using the evaluation set of each dataset.

tasks with less training data, and generally has better performance in mitigating forgetting compared to other methods. However, when the number of training data decreases from 100 to 10, the ability of our method to mitigate forgetting decreases. This is because our method requires data to allow the model to learn the appropriate gradient modification direction. If the amount of training data is too small, the model may not learn the optimal modification direction, resulting in poorer performance on the old dataset.

Ablation Studies for Our Method We also conducted ablation experiments similar to image recognition, as shown in Table 3. From the results, we can observe that continual learning on the new dataset without the LRR and WM can cause the model to disrupt previously learned knowledge, resulting in an increase in error rate on the old dataset, particularly evident in different acoustic environments of the new and old datasets as shown in Table 3a, 3b. The results demonstrate that the gradient direction modification mechanism has a positive effect on overcoming forgetting in most experimental settings. However, this mechanism also reduces the learning performance on new tasks. Additionally, we observed that the self-learning mechanism of gradient-modified radian introduced in our method has a positive impact on the performance of both overcoming forgetting and acquiring new knowledge in all experimental settings. Furthermore, it significantly alleviates the recognition performance loss caused by the introduction of the gradient direction modification on new tasks.

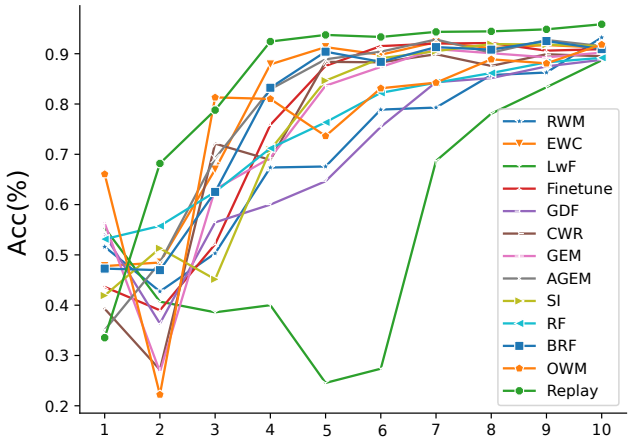


Figure 3: The performance of different continual learning methods after training 5 experiences of CLEAR. All methods are trained using the training sets of $\mathbf{Exp}_1$ to $\mathbf{Exp}_5$ in sequence. The accuracy of all methods in all experiences has been added to the supplemental material.

Image Recognition

Dataset We use the CLEAR benchmark to evaluate the performance of our method for image recognition. CLEAR is a classical continual learning benchmark that is based on the natural temporal evolution of visual concepts of Internet images. Task-based sequential learning is adopted with a sequence of 10-way classification tasks by splitting the temporal stream into 10 buckets, each consisting of a labeled subset for training and evaluation. A small labeled subset ($\mathbf{Exp}_1$, $\mathbf{Exp}_2$, $\mathbf{Exp}_3$, ... $\mathbf{Exp}_{10}$) consisting of 11 temporally dynamic categories with 300 labeled images per category, which includes illustrative categories such as computer, cosplay, etc., as well as a background category. In continual learning, only the current task data is available at each timestamp, except for the replay-based algorithm. The train and evaluation datasets of each labeled subset are generated by using the classic 70/30% train-test split as Table 6 in our supplementary material.

Experimental Setup In the image recognition experiment, all continual learning methods are conducted using an upstream-downstream framework. The upstream component utilized the default pre-trained ResNet 50 (He et al. 2016) of torchvision as a feature extractor, which will be frozen during the continual learning process, producing 2048-dimensional features. The downstream classifier was a linear layer with input and output dimensions of 2048 and

Continual Learning Methods	Accuracy on each experience									
	$\mathbf{Exp_1}$	$\mathbf{Exp_2}$	$\mathbf{Exp_3}$	$\mathbf{Exp_4}$	$\mathbf{Exp_5}$	$\mathbf{Exp_6}$	$\mathbf{Exp_7}$	$\mathbf{Exp_8}$	$\mathbf{Exp_9}$	$\mathbf{Exp_{10}}$
Replay-All	**94.85**	**94.65**	**94.75**	**94.65**	**95.86**	**95.35**	**95.15**	**94.65**	**95.76**	**96.16**
Finetune	87.68	90.00	91.11	91.82	90.40	89.90	90.30	90.61	90.61	93.33
EWC	84.04	84.95	85.86	87.07	85.66	85.56	86.97	86.16	85.76	87.78
LwF	88.59	88.89	87.27	90.51	87.68	87.78	87.47	87.47	88.79	88.48
GDF	91.11	91.62	88.38	91.01	88.79	89.19	90.20	87.68	90.10	90.30
CWR	90.71	91.72	90.71	91.52	89.49	90.91	91.62	90.71	91.82	93.74
GEM	88.38	89.70	90.81	91.41	90.20	89.29	90.91	89.60	90.71	93.03
AGEM	92.32	91.41	92.02	93.43	91.52	92.32	92.22	91.52	92.83	94.75
SI	91.31	92.02	91.41	93.74	91.52	91.72	92.63	91.11	92.22	95.05
BRF	89.29	88.99	88.18	88.48	89.19	89.19	90.10	88.38	89.19	90.00
RF	88.38	90.30	90.00	91.62	90.91	90.61	91.41	91.41	91.11	93.33
OWM	91.62	92.12	91.82	93.64	91.72	92.42	92.22	92.32	92.42	95.05
RWM (Ours, $r_s=3$)	92.15	**94.12**	92.34	93.48	**94.91**	93.01	92.78	93.52	93.76	92.70
RWM (Ours, $r_s=4$)	**93.64**	93.64	**92.53**	**93.84**	93.23	**93.13**	**92.93**	**93.03**	**94.14**	**95.25**
RWM (Ours, $r_s=5$)	93.35	92.95	93.01	92.75	92.62	92.45	93.17	92.64	93.96	93.68

Table 4: The accuracy(%) of the model after training on all CLEAR experiences. All results are (re)produced by us and averaged over 7 runs with standard deviations. The full details of all methods have been described in supplementary material.

Ablation study	Accuracy on each experience									
	$\mathbf{Exp_1}$	$\mathbf{Exp_2}$	$\mathbf{Exp_3}$	$\mathbf{Exp_4}$	$\mathbf{Exp_5}$	$\mathbf{Exp_6}$	$\mathbf{Exp_7}$	$\mathbf{Exp_8}$	$\mathbf{Exp_9}$	$\mathbf{Exp_{10}}$
RWM (Ours)	**93.64**	**93.64**	**92.53**	**93.84**	**93.23**	**93.13**	**92.93**	**93.03**	**94.14**	**95.25**
–LRR	91.62	92.12	91.82	93.64	91.72	92.42	92.22	92.32	92.42	95.05
–WM	87.68	90.00	91.11	91.82	90.40	89.90	90.30	90.61	90.61	93.33

Table 5: The ablation study of our method. All results are the accuracy(%) on the CLEAR experiences.

11, respectively. The experiment used a batch size of 512 and an initial learning rate of 1, which decayed by a factor of 0.1 after 60 epochs. We employed the SGD optimizer with a momentum of 0.9. The α in Eq 1 is 0.1 and the norm in Eq 10 is $\boldsymbol{L}^2$ norm. The details of all continual learning methods have been described in supplementary material.

Comparison With Other Methods In this experiment, we compared RWM with several other continual learning methods. As shown in this table, the performance of our method was second only to Replay-All after training on all experiences, which is considered the upper bound of continual learning performance. However, from Fig 3, it can be seen that our method had lower accuracy than most of the other continual learning methods before the experience 8. This is because our method requires the model to learn the direction of gradient modification. Therefore, the model not only needs to learn to discriminate input data, but also needs to learn the modified direction for different sample data on different tasks, which is the major limitation of RWM. The results also demonstrate the influence of varying r_s on the outcomes. Thus, determining the optimal r_s stands as a crucial avenue for our forthcoming research works.

Ablation Study for Our Method we also conducted an ablation study to evaluate the efficacy of our proposed method. The findings, presented in Table 5, demonstrate that both the self-learned gradient-modified radian and the gradient direction modification positively impact recognition performance. Notably, our observations reveal that, in most cases, the gain in recognition performance resulting from the gradient direction modification exceeds that achieved through the self-learning of the modification radian mechanism. This observation suggests the potential for exploring more refined strategies for learning modification radian in future endeavors. Moreover, they underscore the significance of the self-learned gradient-modified radian and the gradient direction modification in attaining superior recognition performance in the context of continual learning.

Conclusion

This paper proposes an effective continual learning algorithm, Radian Weight Modification (RWM), designed to enhance the adaptability and resilience of audio deepfake detection models in the face of emerging and diverse attack types. The core principle of RWM revolves around the insightful categorization of classes into two distinct groups based on feature distribution similarities. This strategic partitioning enables the algorithm to dynamically adjust gradient modification directions, effectively balancing the acquisition of new knowledge and the preservation of previously learned information across tasks. The experimental results showcased the remarkable effectiveness of RWM in comparison to mainstream continual learning methods for audio deepfake detection, signifying its robustness in addressing the challenges posed by new deepfake attack types. In addition, RWM also demonstrates successful extension to diverse machine learning domains, notably image recognition.

Acknowledgments

This work is supported by the Scientific and Technological Innovation Important Plan of China (No. 2021ZD0201502), the National Natural Science Foundation of China (NSFC) (No. 62322120, No. 62306316, No.61831022, No.U21B2010, No.62101553, No.61971419, No.62006223, No. 62206278).

References

Baevski, A.; Zhou, Y.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ben-Israel, A.; and Greville, T. N. 2003. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media.

Bengio, Y.; and LeCun, Y. 2007. Scaling Learning Algorithms Towards AI. In *Large Scale Kernel Machines*. MIT Press.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2019. Efficient Lifelong Learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Conneau, A.; Baevski, A.; Collobert, R.; Mohamed, A.; and Auli, M. 2020. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

Haykin, S. S. 2002. *Adaptive filter theory*. Pearson Education India.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kinnunen, T.; Sahidullah, M.; Delgado, H.; Todisco, M.; Evans, N. W. D.; Yamagishi, J.; and Lee, K. 2017. The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection. In Lacerda, F., ed., *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2–6. ISCA.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.

Lin, Z.; Shi, J.; Pathak, D.; and Ramanan, D. 2021. The CLEAR Benchmark: Continual LEArning on Real-World Imagery. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Lv, Z.; Zhang, S.; Tang, K.; and Hu, P. 2022. Fake Audio Detection Based On Unsupervised Pretraining Models. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 9231–9235. IEEE.

Ma, H.; Yi, J.; Tao, J.; Bai, Y.; Tian, Z.; and Wang, C. 2021. Continual Learning for Fake Audio Detection. In Hermansky, H.; Cernocký, H.; Burget, L.; Lamel, L.; Scharenborg, O.; and Motlícek, P., eds., *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, 886–890. ISCA.

Martín-Doñas, J. M.; and Álvarez, A. 2022. The Vicomtech Audio Deepfake Detection System Based on Wav2vec2 for the 2022 ADD Challenge. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 9241–9245. IEEE.

Müller, N. M.; Czempin, P.; Dieckmann, F.; Froghyar, A.; and Böttinger, K. 2022. Does Audio Deepfake Detection Generalize? *arXiv preprint arXiv:2203.16263*.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113: 54–71.

Prabhu, A.; Torr, P. H. S.; and Dokania, P. K. 2020. GDumb: A Simple Approach that Questions Our Progress in Continual Learning. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J., eds., *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12347 of *Lecture Notes in Computer Science*, 524–540. Springer.

Sahidullah, M.; Kinnunen, T.; and Hanilçi, C. 2015. A comparison of features for synthetic speech detection. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, 2087–2091. ISCA.

Shah, S.; Palmieri, F.; and Datum, M. 1992. Optimal filtering algorithms for fast learning in feedforward neural networks. *Neural networks*, 5(5): 779–787.

Tak, H.; Todisco, M.; Wang, X.; Jung, J.-w.; Yamagishi, J.; and Evans, N. 2022. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. *arXiv preprint arXiv:2202.12233*.

Todisco, M.; Wang, X.; Vestman, V.; Sahidullah, M.; Delgado, H.; Nautsch, A.; Yamagishi, J.; Evans, N.; Kinnunen, T.; and Lee, K. A. 2019. ASVspoof 2019: Future horizons in spoofed and fake audio detection. *arXiv preprint arXiv:1904.05441*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Veaux, C.; Yamagishi, J.; MacDonald, K.; et al. 2017. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*.

Wang, T.; Fu, R.; Yi, J.; Tao, J.; Wen, Z.; Qiang, C.; and Wang, S. 2021. Prosody and Voice Factorization for Few-Shot Speaker Adaptation in the Challenge M2voc 2021.

ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 8603–8607.

Wang, X.; and Yamagishi, J. 2021. Investigating self-supervised front ends for speech spoofing countermeasures. *arXiv preprint arXiv:2111.07725*.

Wang, Y.; Stanton, D.; Zhang, Y.; Skerry-Ryan, R. J.; Battenberg, E.; Shor, J.; Xiao, Y.; Ren, F.; Jia, Y.; and Saurous, R. A. 2018. Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis. In *International Conference on Machine Learning*.

Wu, Z.; Kinnunen, T.; Evans, N.; Yamagishi, J.; Hanilçi, C.; Sahidullah, M.; and Sizov, A. 2015. ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *Sixteenth annual conference of the international speech communication association*.

Yamagishi, J.; Wang, X.; Todisco, M.; Sahidullah, M.; Patino, J.; Nautsch, A.; Liu, X.; Lee, K. A.; Kinnunen, T.; Evans, N.; et al. 2021. ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection. *arXiv preprint arXiv:2109.00537*.

Yi, J.; Fu, R.; Tao, J.; Nie, S.; Ma, H.; Wang, C.; Wang, T.; Tian, Z.; Bai, Y.; Fan, C.; et al. 2022. Add 2022: the first audio deep synthesis detection challenge. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 9216–9220. IEEE.

Yi, J.; Tao, J.; Fu, R.; Yan, X.; Wang, C.; Wang, T.; Zhang, C. Y.; Zhang, X.; Zhao, Y.; Ren, Y.; Xu, L.; Zhou, J.; Gu, H.; Wen, Z.; Liang, S.; Lian, Z.; Nie, S.; and Li, H. 2023. ADD 2023: the Second Audio Deepfake Detection Challenge. *CoRR*, abs/2305.13774.

Zeng, G.; Chen, Y.; Cui, B.; and Yu, S. 2019. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8): 364–372.

Zhang, Y.; Jiang, F.; and Duan, Z. 2021. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 28: 937–941.

Zhang, Y.; Zhu, G.; Jiang, F.; and Duan, Z. 2021. An empirical study on channel effects for synthetic voice spoofing countermeasure systems. *arXiv preprint arXiv:2104.01320*.