Rehearsal with Auxiliary-Informed Sampling for Audio Deepfake Detection

Falih Gozi Febrinanto[1,2], *Kristen Moore*[2], *Chandra Thapa*[2], *Jiangang Ma*[1], *Vidya Saikrishna*[1], *Feng Xia*[3]

[1]Institute of Innovation, Science and Sustainability, Federation University Australia, Australia
[2]CSIRO's Data61, Australia
[3]School of Computing Technologies, RMIT University, Australia

{f.febrinanto, j.ma, v.saikrishna}@federation.edu.au, {kristen.moore, chandra.thapa}@data61.csiro.au, f.xia@ieee.org

Abstract

The performance of existing audio deepfake detection frameworks degrades when confronted with new deepfake attacks. Rehearsal-based continual learning (CL), which updates models using a limited set of old data samples, helps preserve prior knowledge while incorporating new information. However, existing rehearsal techniques don't effectively capture the diversity of audio characteristics, introducing bias and increasing the risk of forgetting. To address this challenge, we propose **R**ehearsal with **A**uxiliary-**I**nformed **S**ampling (RAIS), a rehearsal-based CL approach for audio deepfake detection. RAIS employs a label generation network to produce auxiliary labels, guiding diverse sample selection for the memory buffer. Extensive experiments show RAIS outperforms state-of-the-art methods, achieving an average Equal Error Rate (EER) of 1.953% across five experiences. The code is available at: https://github.com/falihgoz/RAIS.

Index Terms: audio deepfake detection, rehearsal-based continual learning, sample selection

1. Introduction

Deep learning has shown promising performance in audio deepfake detection [1, 2, 3, 4]. However, as audio deepfake generation evolves, relying on past data without adaptation leads to performance degradation over time [5, 6]. Fine-tuning on new data risks catastrophic forgetting [7, 8, 9], where the model forgets previously acquired knowledge as it adapts to new information. Alternatively, retraining the model from scratch is computationally expensive and discards prior learning [8, 10].

Continual learning (CL) enables models to retain knowledge while integrating new data and has been applied in computer vision, robotics, graph data, and natural language processing (NLP) [11, 10]. In audio deepfake detection, CL methods such as DFWF [12], RWM [5], and RAWM [6] rely on regularization but assume no access to past data [13, 14]. However, updating the model without access to prior datasets can introduce a bias toward newly observed data [15].

Rehearsal-based CL overcomes this by storing and replaying past samples, with methods like Experience Replay (ER) [16] and ER with Asymmetric Cross-Entropy (ER-ACE) [17] proving effective in reducing forgetting [18]. The challenge lies in selecting representative samples within a fixed-size memory buffer. While random sampling lacks balance, strategies such as feature centroid distance [17, 19], class mean [20], gradient-based [21, 22], and class-balanced selection [23] rely only on primary labels (fake/real). However, audio contains diverse paralinguistic features, and ignoring them can result in a less diverse sample selection, ultimately limiting CL performance.

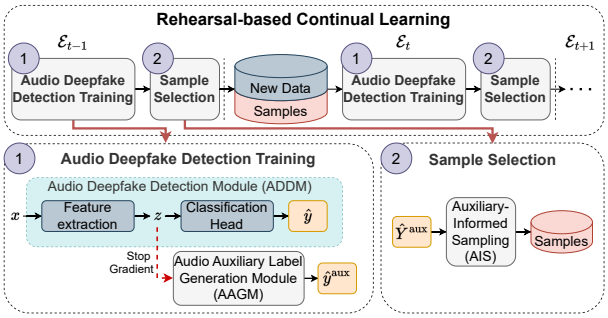

Figure 1: *The Proposed Framework of RAIS*

In this work, we propose **Rehearsal with Auxiliary-Informed Sampling (RAIS)**, which enhances stored sample quality by incorporating auxiliary labels to capture diverse audio characteristics. Since these auxiliary labels are latent, we introduce an audio auxiliary generation module that infers them via masked prediction. These labels then guide sample selection, ensuring a balanced representation of informative and diverse samples from past experiences. Our main contributions are as follows:

- We propose **RAIS**, a rehearsal-based CL approach for audio deepfake detection that improves sample diversity.
- We develop an audio auxiliary generation module that infers auxiliary labels via masked prediction, eliminating manual labeling.
- We introduce an auxiliary label-informed sampling strategy that leverages the generated labels to select diverse and informative samples.
- Extensive experiments show RAIS outperforms state-of-the-art CL methods, achieving the lowest average EER.

2. Methodology

In the CL setting, training is performed across a sequence of experiences $\mathcal{E}_i$. During the initial experience $\mathcal{E}_0$, the memory buffer $\mathcal{M}$ is empty, so training relies solely on the dataset $\mathcal{D}_0$. For subsequent experiences ($i > 0$), $\mathcal{D}_i$ is combined with samples from $\mathcal{M}$ to train the model while mitigating forgetting of previously learned knowledge.

RAIS, shown in Figure 1, enhances continual learning for audio deepfake detection through two key modules: the Audio Deepfake Detection Module (ADDM) for classification and the Audio Auxiliary Label Generation Module (AAGM) for generating auxiliary labels to guide sample selection. The auxiliary labels help mitigate catastrophic forgetting by ensuring diverse, informative samples are retained.

2.1. Audio Deepfake Detection Training

RAIS employs two key modules: ADDM, which classifies audio as fake or bona fide, and AAGM, which generates auxiliary labels to improve sample selection. These two modules are jointly trained to enhance learning and knowledge retention.

2.1.1. Audio Deepfake Detection Module (ADDM)

ADDM consists of two components: a feature extractor g that encodes input audio into a latent representation and a classification head c that maps the latent representation to logits. Given an input audio signal $x \in \mathbb{R}^T$, the ADDM processes it as $\text{SoftMax}(c(g(x)))$, where $g(x)$ produces a latent representation $z \in \mathbb{R}^D$, and c maps z to logits. These logits are then converted into a probability distribution $p = (p_0, p_1)$ using the SoftMax function, where p_0 represents the probability of fake audio ($y = 0$) and p_1 represents the probability of bona fide audio ($y = 1$). The final classification decision is $\hat{y} = \arg\max(p)$. The ADDM is optimized using the cross-entropy loss $\mathcal{L}_{\text{ADDM}}$ which encourages the model to correctly classify audio as either fake or bona fide.

2.1.2. Audio Auxiliary Label Generation Module (AAGM)

While primary labels (fake/bona fide) provide essential supervision, they fail to capture the rich paralinguistic characteristics inherent in audio. To address this, we introduce AAGM, which automatically generates auxiliary labels to guide learning. AAGM is inspired by meta-auxiliary learning [24], but differs in key ways:

1. Decoupled Optimization: Unlike conventional meta-auxiliary learning, which jointly optimizes both the main task and label generation, AAGM prevents *conflicting gradients* [25] that could degrade primary task performance. This is achieved using a **stop-gradient** to isolate AAGM updates.
2. Independent Training: Instead of a multi-task objective, we train AAGM separately using a masked prediction objective [26], ensuring the auxiliary network remains independent of ADDM.

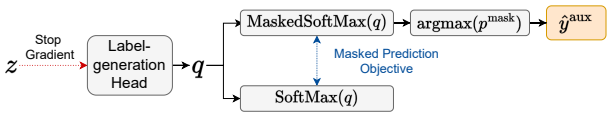

Figure 2: *Audio Auxiliary Label Generation Module (AAGM)*

Figure 2 illustrates AAGM's design. It begins with a detached latent representation z from the feature extractor. The label-generation head h produces a vector $q = h(z) \in \mathbb{R}^K$, where K is the number of possible auxiliary labels. For each input, AAGM categorizes it into one of these K labels. AAGM then generates two multi-class probability vectors through separate branches:

1. Masked branch (Auxiliary-Specific SoftMax): To enforce distinct auxiliary labels for fake and bona fide samples, we apply MaskedSoftMax [24]. The first $K/2$ labels are assigned to fake audio, while the remaining $K/2$ labels correspond to bona fide, ensuring category-specific labels. For each sample, a binary mask vector M is constructed based on its ground-truth label y. Specifically, if the sample is fake ($y = 0$), the mask activates only the first $K/2$ positions, setting the remaining positions to zero. If the sample is bona fide ($y = 1$), the mask activates only the last $K/2$ positions, leaving the others at zero. This mask ensures that each sample is only assigned to valid auxiliary labels within its respective class. The masked probability vector p^{mask} is then computed as:

$$p^{\text{mask}} = \frac{\exp(q) \odot M}{\sum(\exp(q) \odot M)}. \tag{1}$$

2. Unmasked branch (Standard SoftMax): The second branch calculates a standard *SoftMax* probability vector p^{unmask}.

AAGM is trained with a combination of two losses: (i) MSE Loss: which aligns the masked and unmasked probability vectors, and (ii) Diversity Loss (KL Divergence): which prevents trivial solutions by ensuring a uniform distribution of generated labels:

$$\mathcal{L}_{\text{AAGM}} = \underbrace{\|p^{\text{mask}} - p^{\text{unmask}}\|^2}_{\text{MSE Loss}} + \underbrace{\text{KL}\left(\bar{p}^{\text{mask}} \,\|\, \frac{1}{K}\mathbf{1}\right)}_{\text{Diversity Loss}}, \tag{2}$$

where $\bar{p}^{\text{mask}}$ is the average of masked probabilities over the batch, and $\frac{1}{K}\mathbf{1}$ is the uniform distribution over K auxiliary classes. The final auxiliary label is determined as $\hat{y}^{\text{aux}} = \arg\max(p^{\text{mask}})$.

Overall Objective. The final training loss is $\mathcal{L} = \mathcal{L}_{\text{ADDM}} + \mathcal{L}_{\text{AAGM}}$ with stop-gradient decoupling, ensuring that AAGM does not interfere with the primary audio deepfake detection ADDM task training.

2.2. Sample Selection via Auxiliary-Informed Sampling

Traditional sampling methods rely solely on primary labels (fake or bona fide), which can lead to poor sample diversity. We propose Auxiliary-Informed Sampling (AIS), a novel strategy that leverages auxiliary labels to ensure diversity and informativeness in selected samples.

AIS maintains a memory buffer $\mathcal{M}$ composed of segments $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_i\}$, where each $\mathcal{G}_k$ corresponds to a past experience $\mathcal{E}_k$. For each new experience $\mathcal{E}_i$, the allocation size is set as $L = \frac{|\mathcal{M}|}{i+1}$. After training on $\mathcal{E}_i$, L samples are selected and added to $\mathcal{M}$. Since the memory buffer has a fixed size, each past memory segment $\mathcal{G}_k$ is updated to retain only the top L most representative samples, ensuring that past experiences remain well-represented.

The AIS strategy first partitions the dataset into two primary categories: fake samples, $\mathcal{D}_{i,\hat{y}_{\text{aux}} \in [0, \frac{K}{2}-1]}$, and bona fide samples, $\mathcal{D}_{i,\hat{y}_{\text{aux}} \in [\frac{K}{2}, K-1]}$. Within each category, samples are further divided into groups based on their auxiliary labels. For example, in the bona fide category, the groups are indexed from $K/2$ to $K-1$. Each group is then sorted in descending order based on an importance score s, defined as:

$$s = \frac{1}{2}\left(p_{\hat{y}} + p^{\text{mask}}_{\hat{y}^{\text{aux}}}\right), \tag{3}$$

where $p_{\hat{y}}$ is the classification confidence from ADMM, and $p^{\text{mask}}_{\hat{y}_{\text{aux}}}$ is the confidence score from AAGM. The scoring mechanism prioritizes samples with higher overall confidence, ensuring that the most reliable and informative samples are retained.

To maintain class balance, AIS introduces a ratio r, selecting $L \times r$ fake samples and $L \times (1-r)$ bona fide samples. AIS then performs stratified selection using a round-robin approach across the auxiliary label groups within each category, ensuring each auxiliary label is represented within its category in $\mathcal{M}$. If a group runs out of samples, the process continues with the remaining groups until the required number of samples is reached. Finally, the selected samples from both categories are merged and re-sorted in descending order by the importance

Table 1: *Dataset statistics across different experiences in the CL setting.*

$\mathcal{E}$	**ASVSpoof 2019 LA** $\mathcal{E}_0$ (English)			**VCC 2020** $\mathcal{E}_1$ (Multi-language)			**InTheWild** $\mathcal{E}_2$ (English)			**CFAD** $\mathcal{E}_3$ (Chinese)			**OpenAI-LJSpeech** $\mathcal{E}_4$ (English)		
	Fake	Bona fide	Total	Fake	Bona fide	Total	Fake	Bona fide	Total	Fake	Bona fide	Total	Fake	Bona fide	Total
Train	22,800	2,580	25,380	2,920	805	3,725	5,908	9,981	15,889	25,600	12,800	38,400	6,550	6,550	13,100
Dev	22,296	2,548	24,844	1,460	402	1,862	2,954	4,991	7,945	9,600	4,800	14,400	3,275	3,275	6,550
Eval	63,882	7,355	71,237	1,460	403	1,863	2,954	4,991	7,945	42,000	21,000	63,000	3,275	3,275	6,550

score s, forming a new memory segment $\mathcal{G}_i$. This segment is then integrated into the memory buffer $\mathcal{M}$, ensuring past experiences remain well-represented.

3. Experiments

3.1. Datasets and Experimental Settings

Datasets. We evaluate our method and the baseline methods in a CL setting for audio deepfake detection across five experiences. The initial experience, $\mathcal{E}_0$, uses ASVspoof 2019 LA [27] with its original splits (training, development, and evaluation). Experiences $\mathcal{E}_1$ and $\mathcal{E}_2$ consist of the VCC 2020 [27] and InTheWild [28] datasets, each split into 25% development, 25% evaluation, and 50% training. Experience $\mathcal{E}_3$, is the CFAD [29] dataset using its provided splits (training, development, and evaluation, combining seen and unseen tests). To incorporate more advanced speech generation tools, the final experience is generated with the OpenAI TTS API[1] using scripts from LJSpeech[2]. Each transcript was synthesized with a random OpenAI TTS voice (*alloy*, *echo*, *fable*, *onyx*, *nova*, *shimmer*) and model type (*tts-1* or *tts-1-hd*). Fake samples were generated by OpenAI TTS, while bona fide samples were from LJSpeech; both were split with the same proportions as VCC 2020 and InTheWild. Dataset statistics are provided in Table 1.
Experimental Settings. Audio clips were standardized to 4 seconds. We used Wav2Vec2 [30] (wav2vec2-xls-r-300m) as the front-end to convert raw audio into a 2D matrix, then passed it to AASIST [31, 4] for feature extraction. The model was trained with a dropout rate of 0.1, a batch size of 64, and the Adam optimizer with a learning rate of 0.00001. Training was conducted for 10 epochs with early stopping. The classification head $c(\cdot)$ and the label-generation head $h(\cdot)$ consist of two linear layers with hidden dimensions of 80 and 32. The auxiliary label size was set to $K = 90$, with parameter sensitivity analysis provided in Section 3.3. The fake/bona fide ratio was set to $r = 0.8$. For evaluation, we used the equal error rate (EER) and the average EER across all experiences. Each baseline was run three times with different seeds, reporting the mean and standard deviation.

3.2. Experimental Results

Baselines. We compare our method against several baselines. The naive baselines include "Trained on $\mathcal{E}_0$," where the model is trained only on the first dataset and evaluated on all experiences; "Trained on all," where all datasets are merged for the best possible performance; and "Fine-tune," where the model is updated on each new experience without strategies to mitigate forgetting. For regularization-based continual learning methods with no memory buffer, we include EWC [32], LwF [33], and OWM [34], as well as 2 audio deepfake-specific methods: RAWM [6] and RWM [5]. Additionally, we evaluate rehearsal-based strategies with buffer sizes of 256 and 512, including ER [16] and ER-ACE [17], where we use different sample selection strategies, including reservoir [17], herding [20], and class-balanced [23] for ER and ER-ACE, and additionally MIR [22] for ER only. Lastly, we excluded the gradient-based sampling method GSS [21] as it led to *out-of-memory* issues, making it impractical for high-dimensional audio data.
Results and Analysis. Table 2 presents the results, showing that RAIS with a 512-sample buffer achieves an average EER of 1.953%, closely matching the best possible performance of the Train-on-all approach. Overall, CL strategies demonstrate competitive performance relative to Train-on-all while being significantly more efficient, eliminating the need for retraining from scratch with each new experience. In contrast, naive methods such as training exclusively on the initial experience $\mathcal{E}_0$ quickly become obsolete as new experiences emerge, while fine-tuning alone fails to retain knowledge from earlier tasks. Replay-based strategies consistently outperform regularization-based approaches (e.g., EWC, LwF, RAWM, and RWM), which lack memory buffers and cannot leverage past data. However, these methods are not directly comparable, as rehearsal-based methods, regularization-based CL retains knowledge by applying constraints to model updates instead of revisiting past data. Compared to other rehearsal strategies such as Experience Replay (ER) and ER with Asymmetric Cross-Entropy (ER-ACE), RAIS further improves performance by introducing an additional network branch that generates diverse auxiliary labels, leading to better sample selection and a more representative memory buffer based on meaningful and diverse audio characteristics.

3.3. Ablation Studies and Parameter Sensitivity

Ablation Studies. We performed ablation studies using a 512-sample buffer, removing key components one at a time. Three configurations were evaluated. First, RAIS without the Audio Auxiliary Label Generation Module (AAGM), denoted (-) AAGM, which relies solely on main task labels and their confidence scores for sample selection. Second, RAIS without Auxiliary-Informed Selection (AIS), denoted (-) AIS, where sample selection is based only on the highest score (Equation 3) without diversifying across auxiliary label groups. Third, RAIS without Diversity Loss, denoted as (-) DL, removes the diversity loss in $\mathcal{L}_{\text{AAGM}}$. Table 3 shows that the complete RAIS consistently outperforms its ablated variants.

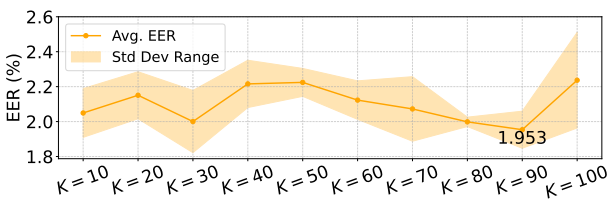

Figure 3: *Average EER (%) across different experiences for varying auxiliary label size K.*

Parameter Sensitivity. We assessed the sensitivity of the aux-

[1]https://platform.openai.com/docs/guides/text-to-speech
[2]https://keithito.com/LJ-Speech-Dataset/

Table 2: *EER (%) comparison of CL methods trained sequentially ($\mathcal{E}_0 \rightarrow \mathcal{E}_1 \rightarrow \mathcal{E}_2 \rightarrow \mathcal{E}_3 \rightarrow \mathcal{E}_4$) and evaluated on all test sets after the final stage. Lower values indicate better performance (↓). **Bold** marks the best result, underline denotes the second-best, and Blue highlights non-CL methods.*

Buffer Size	Method	Sampling	$\mathcal{E}_0$	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_3$	$\mathcal{E}_4$	Avg EER.
-	Trained on $\mathcal{E}_0$	-	7.836 ± 5.820	2.481 ± 0.993	8.041 ± 0.133	10.170 ± 1.195	20.193 ± 11.098	9.744 ± 3.260
-	Trained on all	-	0.553 ± 0.126	0.000 ± 0.000	0.294 ± 0.023	8.013 ± 0.203	0.000 ± 0.000	1.772 ± 0.030
-	Fine-tune	-	1.446 ± 0.432	6.865 ± 4.155	4.855 ± 1.387	9.802 ± 1.127	0.000 ± 0.000	4.594 ± 1.071
-	EWC	-	1.894 ± 0.605	7.858 ± 4.228	6.044 ± 3.486	11.241 ± 2.215	$\mathbf{0.000 \pm 0.000}$	5.408 ± 1.950
-	LwF	-	1.523 ± 0.501	2.647 ± 1.249	$\mathbf{0.521} \pm 0.223$	10.052 ± 0.731	0.275 ± 0.153	3.004 ± 0.552
-	OWM	-	1.550 ± 0.419	4.301 ± 2.940	6.438 ± 2.002	11.343 ± 2.247	$\mathbf{0.000 \pm 0.000}$	4.726 ± 0.376
-	RAWM	-	1.627 ± 1.206	5.376 ± 6.090	4.234 ± 1.898	10.451 ± 1.582	$\mathbf{0.000 \pm 0.000}$	4.338 ± 2.138
-	RWM	-	1.428 ± 0.987	4.549 ± 3.152	4.147 ± 0.626	11.211 ± 0.186	$\mathbf{0.000 \pm 0.000}$	4.267 ± 0.779
256	ER	MIR	0.970 ± 0.450	3.060 ± 1.249	1.743 ± 0.106	9.029 ± 0.998	0.020 ± 0.018	2.964 ± 0.412
256	ER	Class-balanced	1.215 ± 0.169	1.406 ± 0.517	1.122 ± 0.459	8.151 ± 0.172	0.010 ± 0.018	2.381 ± 0.180
256	ER	Reservoir	$\mathbf{0.657 \pm 0.172}$	1.241 ± 0.248	1.837 ± 0.076	8.171 ± 0.215	$\mathbf{0.000 \pm 0.000}$	2.381 ± 0.033
256	ER	Herding	2.479 ± 1.641	0.331 ± 0.143	1.282 ± 0.278	8.351 ± 0.060	$\mathbf{0.000 \pm 0.000}$	2.489 ± 0.292
256	ER-ACE	Class-balanced	0.879 ± 0.348	1.241 ± 0.430	1.563 ± 0.386	8.197 ± 0.110	0.010 ± 0.018	2.378 ± 0.131
256	ER-ACE	Reservoir	1.169 ± 0.575	1.654 ± 0.379	2.017 ± 0.418	8.119 ± 0.765	0.020 ± 0.035	2.596 ± 0.126
256	ER-ACE	Herding	1.106 ± 0.417	1.489 ± 1.082	1.710 ± 0.313	$\mathbf{7.930 \pm 0.357}$	0.010 ± 0.018	2.449 ± 0.240
256	**RAIS**	**AIS**	0.847 ± 0.143	0.248 ± 0.248	2.070 ± 0.182	7.971 ± 0.349	$\mathbf{0.000 \pm 0.000}$	2.230 ± 0.095
512	ER	MIR	1.541 ± 0.594	2.564 ± 1.433	1.316 ± 0.281	8.492 ± 0.454	0.010 ± 0.018	2.785 ± 0.343
512	ER	Class-balanced	0.979 ± 0.756	0.910 ± 0.379	0.928 ± 0.292	8.365 ± 0.138	$\mathbf{0.000 \pm 0.000}$	2.236 ± 0.259
512	ER	Reservoir	1.260 ± 0.265	0.744 ± 0.430	1.002 ± 0.122	8.108 ± 0.307	0.010 ± 0.018	2.225 ± 0.094
512	ER	Herding	1.187 ± 0.532	0.331 ± 0.143	0.922 ± 0.209	8.043 ± 0.203	$\mathbf{0.000 \pm 0.000}$	2.097 ± 0.067
512	ER-ACE	Class-balanced	1.296 ± 0.263	2.647 ± 0.758	0.848 ± 0.117	8.179 ± 0.200	0.010 ± 0.018	2.596 ± 0.198
512	ER-ACE	Reservoir	0.934 ± 0.188	2.233 ± 0.657	1.289 ± 0.189	8.348 ± 0.232	0.010 ± 0.018	2.563 ± 0.222
512	ER-ACE	Herding	0.748 ± 0.286	1.820 ± 0.798	1.423 ± 0.434	8.014 ± 0.053	0.010 ± 0.018	2.403 ± 0.182
512	**RAIS**	**AIS**	0.666 ± 0.187	$\mathbf{0.083 \pm 0.143}$	0.821 ± 0.151	8.197 ± 0.184	$\mathbf{0.000 \pm 0.000}$	$\mathbf{1.953 \pm 0.106}$

Table 3: *EER (%) comparisons of RAIS and its ablated variants, where lower is better (↓). **Bold** denotes the best result.*

Method	$\mathcal{E}_0$	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_3$	$\mathcal{E}_4$
(-) AAGM	3.89 ± 2.35	1.90 ± 1.65	2.61 ± 1.99	8.74 ± 0.09	0.01 ± 0.02
(-) AIS	3.56 ± 1.72	3.08 ± 3.37	3.01 ± 1.51	8.81 ± 0.57	0.01 ± 0.01
(-) DL	1.78 ± 0.65	0.45 ± 0.32	2.28 ± 0.61	$\mathbf{8.16 \pm 0.14}$	0.01 ± 0.01
RAIS	$\mathbf{0.67 \pm 0.19}$	$\mathbf{0.08 \pm 0.14}$	$\mathbf{0.82 \pm 0.15}$	8.20 ± 0.18	$\mathbf{0.00 \pm 0.00}$

Table 4: *Forgetting Rate Comparisons on $\mathcal{E}_1$. CL methods are trained sequentially ($\mathcal{E}_1 \rightarrow \mathcal{E}_2$). Lower values indicate better performance (↓). **Bold** denotes the best result, and Blue denotes non-CL methods.*

Buffer Size	Method	Sampling	$\mathcal{E}_1$	$\mathcal{E}_2$	F_1
-	Trained on $\mathcal{E}_1$	-	0.00 ± 0.00	10.75 ± 3.88	-
-	Trained on all	-	0.08 ± 0.14	0.36 ± 0.17	-
-	Fine-tune	-	6.12 ± 1.52	0.31 ± 0.02	+ (6.12 ± 1.52)
512	ER	Herding	0.33 ± 0.38	0.34 ± 0.08	+ (0.33 ± 0.38)
512	**RAIS**	**AIS**	0.17 ± 0.14	0.29 ± 0.07	+ $\mathbf{(0.17 \pm 0.14)}$

iliary label size K, which determines the number of possible labels predicted by AAGM. We analyzed K values of 10, 20, 30, 40, 50, 60, 70, 90, and 100. As shown in Figure 3, $K = 90$ achieved the best performance with the lowest average EER.

3.4. Discussion

Forgetting in Audio Deepfake Detection. Surprisingly, Table 2 shows no forgetting of $\mathcal{E}_0$ knowledge when comparing "Trained on $\mathcal{E}_0$" with "Fine-Tune." This may be due to the presence of unseen deepfake attacks in the ASVspoof 2019 LA ($\mathcal{E}_0$) evaluation set, which were not part of the training data (similarly for $\mathcal{E}_3$). However, some of these unseen deepfake generators—or generators with similar fingerprints—may appear in the training sets of later experiences. Fine-tuning on new experiences can therefore, in some cases, improve performance by exposing the model to previously unseen patterns. This is reflected in the EER drop for $\mathcal{E}_0$ from 7.836% (trained solely on $\mathcal{E}_0$) to 1.446% after sequential fine-tuning.

To directly assess forgetting, we designed an experiment where the evaluation set only includes attack types that are present in the training set, ensuring similar generator fingerprints in the train, development, and test splits. For this, we used datasets from $\mathcal{E}_1$ (VCC 2020) and $\mathcal{E}_2$ (InTheWild). Table 4 reports the forgetting rate F_1 for $\mathcal{E}_1$, defined as the performance drop after sequentially learning $\mathcal{E}_1 \rightarrow \mathcal{E}_2$ compared to training on $\mathcal{E}_1$ alone. Notably, RAIS consistently achieves the lowest forgetting rate (0.17%), outperforming the second-best baseline ER with Herding, demonstrating superior mitigation of forgetting in CL scenarios.

Limitations and Future Directions. RAIS maintains CL performance with a 512-sample memory buffer ($\sim$ 50 MB), offering high efficiency compared to full dataset storage (>50 GB). Future work should explore scalability for larger CL scenarios. Secondly, while RAIS prioritizes fake samples to reduce privacy risks, storing genuine audio may still pose concerns. Investigating privacy-preserving techniques such as differential privacy could help mitigate these risks. Additionally, the interpretability of auxiliary labels remains an open question. Future research should examine what these labels capture, how they evolve over time, and whether they provide deeper insights into deepfake detection.

4. Conclusion

We introduced Rehearsal with Auxiliary-informed Sampling (RAIS), a CL approach for audio deepfake detection. RAIS improves sample diversity in the memory buffer by automatically generating auxiliary labels, capturing diverse audio characteristics without the need for manual labeling. These labels guide sample selection, ensuring a balanced representation of audio features. Extensive experiments show that RAIS outperforms state-of-the-art CL and experience replay methods across five experiences, achieving the lowest average EER.

5. References

[1] Z. Wu, R. K. Das, J. Yang, and H. Li, "Light convolutional neural network with feature genuinization for detection of synthetic speech attacks," in *INTERSPEECH*, 2020.

[2] X. Liu, M. Liu, L. Wang, K. A. Lee, H. Zhang, and J. Dang, "Leveraging positional-related local-global dependency for synthetic speech detection," in *ICASSP*, 2023.

[3] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *ICASSP*, 2022.

[4] H. Tak, M. Todisco, X. Wang, J.-w. Jung, J. Yamagishi, and N. Evans, "Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation," in *Proc. The Speaker and Language Recognition Workshop (Odyssey)*, 2022.

[5] X. Zhang, J. Yi, C. Wang, C. Y. Zhang, S. Zeng, and J. Tao, "What to remember: Self-adaptive continual learning for audio deepfake detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 569–19 577.

[6] X. Zhang, J. Yi, J. Tao, C. Wang, and C. Y. Zhang, "Do you remember? overcoming catastrophic forgetting for fake audio detection," in *International Conference on Machine Learning*. PMLR, 2023, pp. 41 819–41 831.

[7] Z. Wang, E. Yang, L. Shen, and H. Huang, "A comprehensive survey of forgetting in deep learning beyond continual learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[8] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[9] D. Salvi, V. Negroni, L. Bondi, P. Bestagini, and S. Tubaro, "Freeze and learn: Continual learning with selective freezing for speech deepfake detection," in *ICASSP*, 2025.

[10] F. G. Febrinanto, F. Xia, K. Moore, C. Thapa, and C. Aggarwal, "Graph lifelong learning: A survey," *IEEE Computational Intelligence Magazine*, vol. 18, no. 1, pp. 32–51, 2023.

[11] Y.-C. Yu, C.-P. Huang, J.-J. Chen, K.-P. Chang, Y.-H. Lai, F.-E. Yang, and Y.-C. F. Wang, "Select and distill: Selective dual-teacher knowledge transfer for continual learning on vision-language models," in *European Conference on Computer Vision*. Springer, 2025, pp. 219–236.

[12] H. Ma, J. Yi, J. Tao, Y. Bai, Z. Tian, and C. Wang, "Continual learning for fake audio detection," in *INTERSPEECH*, 2021.

[13] Y. Chen, J. Yi, C. Fan, J. Tao, Y. Ren, S. Zeng, C. Y. Zhang, X. Yan, H. Gu, J. Xue *et al.*, "Region-based optimization in continual learning for audio deepfake detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 22, 2025, pp. 23 651–23 659.

[14] X. Chen, W. Lu, R. Zhang, J. Xu, X. Lu, L. Zhang, and J. Wei, "Continual unsupervised domain adaptation for audio deepfake detection," in *ICASSP*, 2025.

[15] A. Krutsylo, "The inter-batch diversity of samples in experience replay for continual learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 395–23 396.

[16] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[17] L. Caccia, R. Aljundi, N. Asadi, T. Tuytelaars, J. Pineau, and E. Belilovsky, "New insights on reducing abrupt representation change in online continual learning," in *International Conference on Learning Representations*, 2022.

[18] L. Hanmo, D. Shimin, L. Haoyang, L. Shuangyin, C. Lei, and Z. Xiaofang, "Effective data selection and replay for unsupervised continual learning," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 1449–1463.

[19] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[20] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[21] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[22] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," *Advances in neural information processing systems*, vol. 32, 2019.

[23] A. Chrysakis and M.-F. Moens, "Online continual learning from imbalanced data," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1952–1961.

[24] S. Liu, A. Davison, and E. Johns, "Self-supervised generalisation with meta auxiliary learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[25] C. Ding, Z. Lu, S. Wang, R. Cheng, and V. N. Boddeti, "Mitigating task interference in multi-task learning via explicit task routing with non-learnable primitives," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7756–7765.

[26] A. Baevski, A. Babu, W.-N. Hsu, and M. Auli, "Efficient self-supervised learning with contextualized target representations for vision, speech and language," in *International Conference on Machine Learning*. PMLR, 2023, pp. 1416–1429.

[27] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," in *INTERSPEECH*, 2019.

[28] N. M. Müller, P. Czempin, F. Dieckmann, A. Froghyar, and K. Böttinger, "Does audio deepfake detection generalize?" in *INTERSPEECH*, 2022.

[29] H. Ma, J. Yi, C. Wang, X. Yan, J. Tao, T. Wang, S. Wang, and R. Fu, "Cfad: A chinese dataset for fake audio detection," *Speech Communication*, vol. 164, p. 103122, 2024.

[30] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *NeurIPS*, 2020.

[31] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *ICASSP*, 2022.

[32] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[33] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[34] G. Zeng, Y. Chen, B. Cui, and S. Yu, "Continual learning of context-dependent processing in neural networks," *Nature Machine Intelligence*, vol. 1, no. 8, pp. 364–372, 2019.