Journal on Audio,
Speech, and Music Processing

EMPIRICAL RESEARCH

Open Access

Characterizing continual learning scenarios and strategies for audio analysis

Ruchi Bhatt[1*], Pratibha Kumari[2], Dwarikanath Mahapatra[3], Abdulmotaleb El Saddik[4], Praphul P. Thekkila[1] and Mukesh Saini[1]

Abstract

Audio analysis is used in many real-world applications, but most methods assume a fixed data distribution. In practice, data can shift over time due to changes in distribution or the appearance of new classes, reducing model performance. Traditional deep learning models fail to adapt, making continual learning (CL) crucial. While some studies explore CL for audio, they mainly focus on a class-incremental scenario and largely ignore a domain-incremental scenario. In applications such as anomaly detection and monitoring, domain incremental scenarios are prevalent. To the best of our knowledge, no existing comprehensive benchmark for audio data supports both domain-incremental and class-incremental learning scenarios. To bridge this gap, we have curated a benchmark using the DCASE dataset (2020–2023) that properly considers both scenarios. Previous works have primarily explored anomaly detection approaches where new classes are treated as anomalies. Our work complements these approaches by examining anomaly detection within established classes, which is crucial for audio applications involving rare or unexpected sounds within known classes. We use the proposed dataset to compare regularization-based (EWC, LwF, and SI) and rehearsal-based (GEM, A-GEM, GDumb, Replay, DER++, and Co^{2}L) CL approaches, as well as non-CL approaches (Naive, Cumulative, and Joint training). We evaluate multiple metrics such as accuracy, forward transfer, and backward transfer using two popular backbones: ResNet50 and ViT. The key findings show that Replay outperforms other CL methods, achieving up to 76.53% (domain) and 99.51% (class) accuracy with ViT, and 70.12% (domain) and 96.98% (class) accuracy with ResNet50. The study is very beneficial for researchers and practitioners working in the area of audio analysis for developing adaptive models.

Keywords Machine monitoring, Audio data, Data shift, Continual learning

1 Introduction

Audio plays an important role in machine learning and deep learning applications, contributing to various tasks such as event detection [1], speech recognition [2], audio surveillance [3], sound classification [4], audio signal processing [5], music generation, and environmental sound analysis. Audio-based monitoring offers distinct advantages such as low-cost deployment, non-invasiveness, remote operation, and the ability to capture rich contextual information [6–8]. With advancements in sensor technologies and the widespread availability of audio data, audio-based monitoring has become increasingly prevalent across various domains, including public surveillance, environmental sound monitoring, wildlife

*Correspondence:
Ruchi Bhatt
ruchi.21csz0007@iitrpr.ac.in
[1] Department of Computer Science & Engineering, Indian Institute of Technology Ropar, Rupnagar 140001, Punjab, India
[2] Faculty of Informatics and Data Science, University of Regensburg, Regensburg, Germany
[3] Faculty of Information Technology, Monash University, Melbourne, Australia
[4] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada

conservation, and healthcare. Traditional audio analysis models follow one-time training with all available data and employ the trained model for all future inferences. The assumption of prior availability of all the data to train a deep model is not practical. Data usually comes over time in real-world environments, and the model needs to be updated with the new batch of data [9–11]. However, as deep learning models learn new concepts from newly available data, they tend to forget past learning and therefore perform poorly on past data, leading to a phenomenon known as catastrophic forgetting [12, 13]. To mitigate the effect of catastrophic forgetting, a human-like lifelong learning paradigm, popularly termed as Continual Learning (CL), is gaining major attention recently. CL offers to acquire new skills and accumulate knowledge over time, minimizing the risk of forgetting previously learned experiences. Consequently, several CL strategies have been proposed in various research fields, including object detection [14], medical image analysis [15–18], robotics [19], audio analysis [20–31], etc.

Particularly, in the audio domain, CL becomes pertinent due to the streaming and dynamically evolving nature of sound environments. By continuously updating the deployed model with new audio samples, a CL strategy can be leveraged to recognize and classify new audio events, adapt to new acoustic conditions, and generalize across different audio contexts. CL has been exploited in audio applications including audio analysis/classification [20–24], sound event detection [25, 26], audio captioning [27], audio-visual learning [28, 29], fake audio detection [30], speech recognition [31], etc.

However, efforts to adopt CL strategies in audio monitoring applications, such as anomaly detection, are still less explored. As shown in Table 2, most existing works focus on audio classification, whereas relatively few address anomaly detection. Even in the case of Zhou et al. [32], who use the anomaly dataset from the DCASE2020 task-2 challenge, drift is modeled as a change of machine type, an assumption that does not reflect the gradual and realistic distribution shifts observed in practice. This gap motivates our work.

In this paper, we characterize CL techniques for audio analysis with a case study of audio-based monitoring. For this application, we curated a sequential dataset in "domain-shift" as well as "new-class" situations with different CL scenarios using data from DCASE challenges[1], where we considered naturally occurring drifts such as weather conditions, machine load over time, etc. To the best of our knowledge, this is the first curated audio dataset that enables systematic evaluation of both domain-incremental and class-incremental scenarios for audio anomaly detection within a unified framework. Further, we identify and benchmark popular CL strategies for audio-based monitoring application including EWC [33], SI [34], LwF [35], GEM [36], A-GEM [37], GDumb [38], Replay [39], DER++ [40], and Co^{2}L [41]. We also studied the non-CL strategies, such as Naive, Cumulative, and Joint training. We conduct a thorough analysis of the performance and examine how well catastrophic forgetting is controlled by these CL strategies in different dynamically changing audio environment settings.

The major contributions of this paper are as follows:

- We have given a systematic evaluation framework comparing two widely used CL strategies, regularization-based and rehearsal-based, namely, EWC, LwF, SI, GEM, A-GEM, GDumb, Replay, DER++, and Co^{2}L using multiple metrics such as average accuracy, backward transfer, and forward transfer across two CL scenarios with two backbones: ResNet50 and ViT.
- We propose comprehensive CL datasets for both domain-incremental and class-incremental scenarios using publicly available DCASE challenge data from 2020 to 2023 for audio-based monitoring tasks.

The rest of the paper is structured as follows. Section 2 gives an overview of CL, including popular scenarios, techniques, and related literature in audio analysis. Section 3 presents the frequently benchmarked CL and non-CL baselines. Section 4 includes a detailed description of the curated CL datasets. Section 5 outlines the experimental setup and results. Section 6 includes the discussion, limitations, and future directions, followed by a conclusion in Sect. 7.

2 Continual learning

CL is a rapidly emerging research direction to mitigate catastrophic forgetting and mimic the human way of learning in deep neural networks [42]. It refers to the learning strategy in which a machine continuously accumulates knowledge from the incoming data. Unlike traditional deep learning, where the complete data is accessible for training at the start, in CL, the data comes sequentially, in discrete stages, commonly referred to as "episodes". CL allows the acquisition of new knowledge while retaining the existing knowledge. However, there is a tradeoff between how well a model can retain past knowledge and learn new things. It is called the stability-plasticity dilemma [43]. The plasticity shows the ability of the model to learn new episodes, and the stability refers to maintaining the performance of previously learned

[1] https://dcase.community/.

episodes upon learning new concepts [44]. Efforts to increase either of them hinder the other, and vice versa [45]. The preference for either plasticity or stability also depends on the targeted application.

Various CL strategies have been formulated in the literature to mitigate forgetting and facilitate knowledge accumulation. Figure 1 gives an overview of broader modules of the CL pipeline, including CL scenario, CL data stream, CL strategy, evaluation scheme, and finally, comparable methods. Depending upon what kind of change is expected in data, i.e., new class, new instances, shifted distribution, entirely new episode, etc., there can be various settings, also known as CL scenarios. Once the possible change in data is identified for the application, a dataset stream containing multiple episodes, each having a train and test split, is prepared. Then, a suitable CL strategy from various categories is employed in the deep model to prevent a drop in performance on past episodes upon learning new episodes. Lastly, the effectiveness of a particular CL strategy is analyzed against other CL and non-CL strategies with the help of specifically designed metrics to measure forgetting and knowledge transfer. The subsequent sections discuss the major CL scenarios, strategies, and relevant literature for audio analysis.

2.1 Continual learning scenarios

In CL literature, a "scenario" refers to the specific manner in which new episodes are presented to the model over time. It defines the learning conditions, simulating real-world settings where new information arrives sequentially. In audio-based monitoring, particularly for anomaly detection, two situations are most common: (i) new samples from existing classes, potentially with distribution shifts, and (ii) entirely new intruder/event classes. While CL research has proposed several distinct scenarios across different applications [11], we briefly review them in general; however, as the audio literature has predominantly explored only the two scenarios mentioned above, our dataset design and experiments focus primarily on these. The two major CL scenarios are summarized in Table 1.

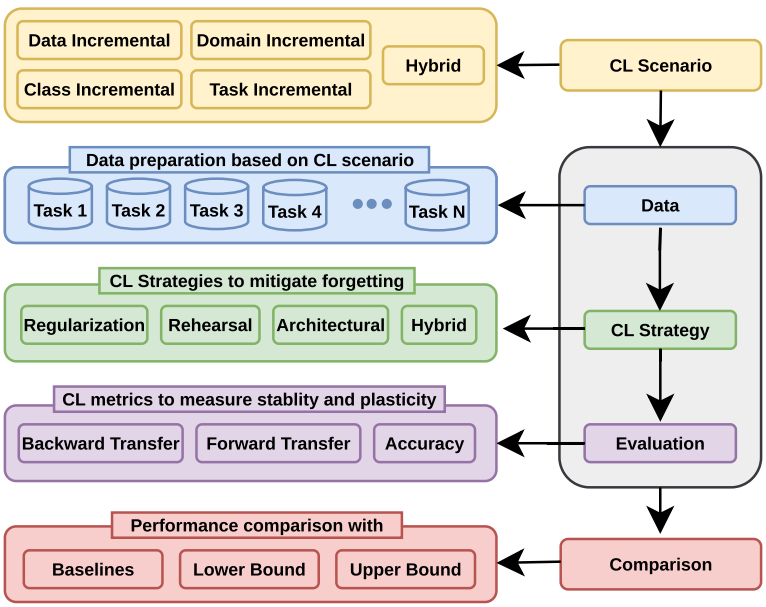

Fig. 1 Overview of a continual learning pipeline. This figure shows the available broader types of CL scenarios, the data preparation for a given CL scenario in subsets called episodes, broad categories of CL strategies based on literature, the well-known metrics used to evaluate the performances in a CL setting, and different aspects of comparative analysis

Table 1 Description of major continual learning scenarios in audio applications

Scenario	Abbreviation	Summary
Domain-incremental	DI	Each episode contains new data distribution with the same classes per episode, episode ID is not provided
Class-incremental	CI	Each episode contains a new set of classes per episode, episode ID is inferred

2.1.1 Domain-incremental scenario

Domain-incremental (DI) scenario deals with situations where the application remains the same, but the context or the information presented changes over time due to the non-stationary environment. Here, the data distribution changes, but the classes remain the same for all subsequent episodes. The DI scenario [46] allows us to continuously update the model with new streams of data from shifted distributions. Examples of various shift sources in monitoring include variations due to weather, seasons, geo-location, surrounding noise levels across the day, etc.

2.1.2 Class-incremental scenario

The class-incremental (CI) scenario [47] involves a single, continuously evolving application across subsequent episodes. In this setting, the model is progressively extended to recognize previously unseen classes as new data streams become available. Typically, the classes in different episodes are disjoint, meaning the knowledge to be acquired is non-overlapping. This lack of shared information makes the CI scenario one of the most challenging among other CL scenarios [40]. An example of the CI scenario in real-world monitoring is to identify cats and dogs in one episode, followed by cows and horses in the next episode [48].

There are other CL scenarios available in the literature, like Instance-incremental (II) (also known as data-incremental), Task-incremental (TI), and Hybrid scenarios. The II scenario allows us to continuously learn from new streams of data from an identical underlying distribution during sequential model training. It does not consider new classes or considerable shifts of data distribution and thus can also be considered as a DI scenario where the data distribution remains unchanged. The TI scenario refers to learning different kinds of tasks over subsequent episodes. The tasks may also be highly uncorrelated. Each task is observed as an episode, and it usually has a disjoint label space. The TI scenario can be considered the CI scenario, where the subsequent episodes are observed as dissimilar tasks, and classes associated with the episodes are disjoint. It can also be considered the DI scenario where the subsequent episodes are observed as similar tasks, and classes associated with them are unchanged. A real-world example of TI learning is the sequential learning of different sports or musical instruments over time. In the real world, it is highly unrealistic that the model will always encounter one type of incremental scenario. It is quite possible that the model can observe new classes as well as domain shifts. The model developed for handling CI situations might underperform for DI situations. Thus, there are attempts in the literature to evaluate the same model on a mix of incremental scenarios called hybrid scenarios [49].

2.2 Continual learning techniques

Existing CL techniques to handle catastrophic forgetting can be broadly classified into three categories [50, 51], viz., rehearsal [36, 37, 39, 52], regularization [33–35, 51], and architectural [53, 54]. In rehearsal-based techniques, a small memory is used to store a few data points from the previous episodes and then replayed along with current data to overcome forgetting. The previous data can be stored and replayed in various forms, including raw samples, deep feature descriptors, and generative models to generate samples or features. In contrast, regularization-based techniques avoid storing past samples and add a regularization term in the loss function or regularize the learning rate to control drastic deviations in learned weights to minimize forgetting. Further, data-focused and prior-focused regularization have been explored. Data-focused approaches use knowledge distillation to transfer past learning to the current model, whereas the prior-focused methods define the importance of the network's parameters, which is then used to penalize larger deviation in important parameters [33]. Architectural techniques mainly aim to keep some network parameters reserved for each episode. Two design choices are explored in this category: fixed network capacity and dynamic network capacity. Lastly, hybrid approaches, i.e., a combination of two or more of the above categories, are also popularly explored in various applications [49].

2.3 Continual learning for audio analysis

Table 2 summarizes various studies that apply CL to audio applications and utilize CL strategies, scenarios, and datasets. For audio scene classification, most works have adopted the CI scenario to handle the dynamic nature of audio data [4, 26, 55–58], using datasets such as ESC10, ESC50, UrbanSound8K, VGGSound, and DCASE TAU19. Only Karam et al. [21] explored the TI scenario for this application. In multi-label audio classification, CI scenarios have been applied to datasets like AudioSet, ESC50, and FSD-MIX-CLIPS [20, 24, 59]. Similarly, multimodal audio-video classification studies [28, 29] have adopted the CI setting with datasets such as VGGSound, AVE-CI, and VS100-CI, while Kim et al. [60] used TI scenarios on AudioSet, VGGSound, MACS, FSD50K, and ClothoV2 datasets. Beyond classification, Berg et al. [27] applied the DI scenario for automated audio captioning using Clotho and AudioCaps. In audio

Table 2 Summary of state-of-the-art continual learning literature for various audio applications

Work	CL strategy	CL scenario	Dataset	Anomaly detection?	Application (shift detail if any)
Wang et al. [4]	Rehearsal	CI	ESC10	No	Audio scene classification
Koh et al. [26]	Architectural	CI	DCASE16, US-SED	No	
Kwon et al. [55]	Rehearsal	CI	EmotionSense, UrbanSound8K	No	
Wang et al. [56]	Regularization	CI	UrbanSound8K, VGGSound, DCASE TAU19	No	
Karam et al. [21]	Rehearsal	TI	ESC50, UrbanSound8K	No	
Mulimani et al. [57]	Regularization	CI	DCASE TUT acoustic scenes 2017, TUT 2016/2017	No	
Sun et al. [58]	Rehearsal	CI	ESC10, ESC50, UrbanSound8K	No	
Wang et al. [20]	Architectural	CI	AudioSet, ESC50	No	Multi-label audio classification
Wang et al. [59]	Architectural	CI	FSD-MIX-CLIPS (1s), FSD-MIX-SED (10s)	No	
Mulimani et al. [24]	Regularization	CI	AudioSet	No	
Mo et al. [28]	Regularization & Rehearsal	CI	VGGSound-Instruments, VGG-Sound-100, VGG-Sound Sources	No	Multi-modal classification (audio,video)
Pian et al. [29]	Rehearsal	CI	AVE-CI, K-S-CI, VS100-CI	No	Multi-modal classification (audio,video)
Kim et al. [60]	Regularization	TI	AudioSet, VGGSound, MACS, FSD50K, ClothoV2, AudioCaps	No	Multi-modal classification (audio,video, text)
Berg et al. [27]	Regularization	DI	Clotho, AudioCaps	No	Automated audio captioning (Audio clips are annotated with 5 captions to each 10–30 sec each of both datasets)
Ma et al. [30]	Regularization	DI	ASVSpoof2019	No	Fake audio detection (Considered 2 subsets of the dataset & selected 4 spoofing attacks with big differences, from each subset separately)
Zhou et al. [32]	Regularization	DI	DCASE 2020 challenge Task-2	Yes	Anomaly sound detection (Considered subsequent machine type as drift)

deepfake detection, Ma et al. [30] modeled DI scenarios in ASVSpoof2019 by treating different spoofing attack types as distribution shifts. For anomalous sound detection, Zhou et al. [32] used a DI setting with the DCASE 2020 Task-2 dataset, addressing machine-type drift for robust monitoring.

From Table 2, it is evident that audio scene classification has been the dominant focus of CL research in the audio domain, whereas anomaly detection, particularly in machine monitoring contexts, has received minimal attention. To date, only Zhou et al. [32] have applied CL to anomaly detection, but their notion of drift was limited to changes in machine type. In contrast, many real-world monitoring scenarios involve more subtle and naturally occurring drifts, such as variations in weather or machine operational conditions over time. Across existing works, CI scenarios that involve new class occurrences are far more common than DI scenarios modeling such distribution shifts, despite the latter being crucial for robust real-world performance. This imbalance is partly due to the lack of suitable public datasets enabling systematic CI and DI evaluation for anomaly detection. To address this gap, we curate the DCASE dataset for each CI and DI scenario, incorporating naturally occurring drifts. This enables systematic evaluation of popular CL methods in both scenarios. Unlike prior studies that focus on either CI or DI, or scene classification, our work evaluates both DI and CI in the context of anomaly detection, providing new insights into continual learning performance under realistic, naturally drifting conditions.

3 Baselines

The CL research community has contributed various CL techniques in the categories mentioned in Sect. 2.2. Their effectiveness is also evaluated against non-CL approaches

that try to mitigate forgetting using retraining or exploiting all the datasets together. However, these approaches do not fit with human-like learning but may perform well at the cost of large storage and computation facilities. For the audio monitoring applications, we have evaluated the following CL and non-CL baselines.

3.1 Continual learning baselines

This paper targets a few major and frequently used CL baselines to conduct a comparative study on the dataset. These baselines are briefly discussed below.

- **Elastic Weight Consolidation:** Elastic Weight Consolidation (EWC) serves as a regularization technique designed to mitigate the issue of forgetting in neural networks. It achieves this by selectively constraining, or partially freezing, the model weights associated with essential components for previously learned episodes. By prioritizing the preservation of crucial knowledge, EWC helps maintain performance on earlier episodes while accommodating the learning of new ones. This method provides a balance, preventing excessive interference with existing knowledge when adapting to novel information [33, 51].
- **Learning without Forgetting:** Learning without Forgetting (LwF) is a regularization method aimed at addressing the challenge of forgetting in neural networks by enforcing stability in the output or predictions. Notably, LwF prioritizes training efficiency, demonstrating faster results compared to the joint training approach. This is a CL approach specifically designed to utilize only new data, assuming the unavailability of past data used for initial network pretraining. This sets LwF apart from other CL methods, which typically focus on leveraging prior knowledge to facilitate the learning of new episodes [35, 51].
- **Synaptic Intelligence:** Synaptic Intelligence (SI) is treated as a variant of EWC. In contrast to EWC, which estimates parameter importance through a diagonal approximation of the Fisher information, SI takes an online approach, dynamically estimating parameter importance based on the ongoing training trajectory. While EWC relies on a fixed approximation, SI adapts continuously during the learning process, providing a more flexible assessment of parameter significance. This distinction underscores the difference in how these methods evaluate and prioritize the importance of parameters in neural network training [34, 51].
- **Gradient Episodic Memory:** Gradient Episodic Memory (GEM) aims to mitigate catastrophic forgetting by utilizing an episodic memory to retain a portion of the current data samples. By adjusting the gradient of each current episode, GEM endeavors to minimize negative backward transfer, thereby preventing the loss of information from previously learned episodes. This approach ensures that the knowledge gained from past episodes positively influences the learning of subsequent episodes, contributing to a more effective and stable learning process [36].
- **Averaged Gradient Episodic Memory:** Averaged Gradient Episodic Memory (A-GEM) tries to alleviate the computational cost caused by GEM. A-GEM can be defined as the more efficient version of GEM. Unlike GEM, which minimizes the loss of each previous episode at each training step by adjusting the gradient of each current episode, A-GEM minimizes the average episodic memory loss over the previous episodes at each training step [37].
- **Greedy sampler and Dumb learner:** in Greedy sampler and Dumb learner (GDumb), the sampler greedily stores part of the data samples as they are encountered while ensuring a balanced representation of classes, and the learner, which is a neural network, trains it dumbly using all the samples available in the memory. Therefore, it is called GDumb. Although GDumb is not meant to handle the CL scenarios, it gives convincing results in comparison to the advanced CL strategies [38].
- **Replay:** the replay strategy can use different types of replay, like, experience replay [39] where the old stored samples from the memory are used for training with the new samples, generative replay [61] where the old data is generated using some generative models and trained with the new encountered data samples, etc.
- **Dark Experience Replay++:** Dark Experience Replay (DER) preserves past knowledge by storing a small replay memory of examples and their logits, ensuring prediction consistency. However, it struggles with distribution shifts as stored logits may become biased. Unlike DER, Dark Experience Replay++ (DER++) [40] improves this by also replaying ground-truth labels, enhancing stability and adaptability of the CL model.
- **Contrastive Continual Learning:** Contrastive Continual Learning (Co^2L) [41] combines contrastive learning and knowledge distillation. The asymmetric contrastive loss helps learn representations, while self-supervised distillation ensures past knowledge is preserved. By decoupling representation and classifier learning, Co^2L maintains stability across episodes while minimizing catastrophic forgetting.

3.2 Non-continual learning baselines

The effectiveness of CL methods is compared against some non-CL approaches that try to mitigate forgetting and achieve high performance using retraining or exploiting all the episode datasets. These approaches do not fit with human brain learning but may perform well at the cost of large amounts of space and computational facilities. These non-CL baselines are discussed below.

- **Joint:** this is a popular approach in the non-CL category, where all the training episodes are jointly trained once and mostly achieve the highest average performance and hence also regarded as an upper bound. However, this method assumes the availability of all the episodes simultaneously, which is a bottleneck.
- **Cumulative:** this non-CL training strategy is another widely adopted approach offering upper-bound performance. Here, the training process is started from scratch each time a new episode is available. It considers the current and all the previous episodes together for retraining. However, previously learned knowledge is wasted due to retraining from scratch each time a new episode is encountered, leading to high computation and storage requirements.
- **Naive:** this non-CL strategy achieves lower-bound performance. Here, a traditional transfer learning mechanism is followed. Whenever a new episode is available, the model adapts itself to this new data without following any mechanism to control forgetting past data.

4 Datasets

To evaluate CL techniques for audio-based monitoring, it is essential to have sequential datasets that capture real-world variability. As shown in Table 2, no suitable dataset exists for the DI scenario, except for the dataset considered by Zhou et al. [32]. However, their study is limited to a small subset (DCASE 2020 Challenge Task 2) of the broader DCASE repository. Moreover, the dataset used contains audio samples from a single domain, and the authors treat different machine types as domain shifts. In contrast, our focus is on real-world causes of distributional shifts in machine audio, such as changes in operational conditions (e.g., operating speed, viscosity, heating temperature, varying load, signal-to-noise ratio, and environmental noise), which naturally introduce domain variability. To the best of our knowledge, no benchmark datasets currently exist for audio monitoring that explicitly capture such naturally occurring domain shifts. Nevertheless, the DCASE repository provides a large and well-curated collection of datasets for abnormality detection in machine sounds, which closely aligns with audio-based monitoring applications. This makes DCASE a promising resource for evaluating CL techniques in this domain. The datasets include diverse machine sounds under normal and abnormal conditions, with distribution shifts arising from varying machine loads and environmental factors. Building on this, we curate benchmark datasets for CL that represent different learning scenarios, leveraging publicly available audio datasets from the DCASE community.

DCASE challenge offers datasets for various audio analysis-related applications every year. Among various tasks, the task-2 of DCASE is for audio anomaly detection and has been held continuously since the year 2020. We have utilized data from the following DCASE challenge tasks from years 2020 to 2023: (a) DCASE2020 task-2: "Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring", (b) DCASE2021 task-2: "Unsupervised Anomalous Sound Detection for Machine Condition Monitoring under Domain Shifted Conditions", (c) DCASE2022 task-2: "Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques", and (d) DCASE2023 task-2: "First-Shot Unsupervised Anomalous Sound Detection for Machine Condition Monitoring". Each of the tasks offers audio clips from 6 to 7 distinct machine types in normal and abnormal classes. The dataset in these challenges is given in three segments: development, evaluation, and additional training. Since the evaluation segment does not include labels and the additional training set contains only normal samples, we rely exclusively on the development segment. Within this segment, DCASE further provides 'training data' and 'test data'. For our study, we use only the 'test data' portion, as it contains a balanced number of normal and abnormal samples. This subset is split in a 70:30 ratio for curating training and testing episodes. Each audio clip is 10-s long, from which spectrogram representations are computed. Below, we describe how these datasets are harnessed to construct episode sequences for various CL experiments. Specifically, we focus on CI and DI scenarios, as they represent the most common forms of evolving data and are frequently explored in the literature (Table 2).

4.1 Dataset for DI scenario

Here, we consider development datasets provided for task 2 of the DCASE2021, DCASE2022, and DCASE2023 challenges, as they have the same machine type but have domain shift conditions due to weather and variable machine load. The development dataset has a source domain and a target domain with a shifted data distribution, which would serve as the DI condition for our case. Further, the test part of the development segment only

offers balanced normal and abnormal classes, and hence, we consider only the test part, not the training part of the development segment. These datasets have normal and abnormal audio clips for each of the seven machine types, including Fan, Gearbox, Pump, Slider, ToyCar, ToyTrain, and Valve. The end goal here is abnormal sound detection in a sequence of data that may contain a distribution shift over time. We curate 6 episodes by utilizing source and target domain data from DCASE2021, DCASE2022, and DCASE2023. The first and second episodes represent the source and the target domains of DCASE2021, having 4306 (2153 normal and 2153 abnormal) and 4218 (2109 normal and 2109 abnormal) samples, respectively. Then, the third and fourth episodes are curated from the source and target domains of DCASE2022, offering 2100 (1050 normal and 1050 abnormal) and 2100 (1050 normal and 1050 abnormal) samples, respectively. Lastly, the fifth and sixth episodes are curated from the source and target domains of the DCASE2023 challenge, which have 700 (350 normal and 350 abnormal) and 700 (350 normal and 350 abnormal) samples, respectively. A pictorial representation is provided in Fig. 2 for better visualization of the sequence of episodes under the DI scenario.

4.2 Dataset for CI scenario

Contrary to the DI scenario datasets discussed in the earlier section, here, the end goal is machine-type classification. In the CI scenario, there needs to be the inclusion of novel classification classes over time. Hence, we opt to classify machine types as the DCASE datasets offer a large number of machine types, which can then be sequentially learned by the CL model. To focus only on machine-type classification, either normal or abnormal data should be considered. Since the normal class data is available in large amounts, we consider the normal class samples of different machine sounds here. Specifically, we curate six episodes by utilizing the training part (it has only normal samples) of the development segment in the task-2 of DCASE2020 and the source domain of the additional training segment in the task-2 of DCASE2023 challenges. The choice of these datasets is such that they have disjoint machine classes. Specifically, we consider ToyCar and ToyConveyor machine types from DCASE2020 in the first episode, Valve and Fan machine types in the second episode, and then Pump and Slider machine types in the third episode. Then, from the DCASE2023, two machine types, viz., Vacuum and ToyTank, are considered in the fourth episode, ToyNscale and ToyDrone are considered in the fifth episode, and finally, the last episode has the remaining classes from DCASE2023, including Bandsaw, Grinder, and Shaker. Inside an episode, the number of samples for each machine type is considered in a balanced amount, i.e., the machine types have 3000, 2804, 990, 990, and 990 samples in the first, third, fourth, fifth, and sixth episodes, respectively. Except for the second episode, in which the two machine types, namely

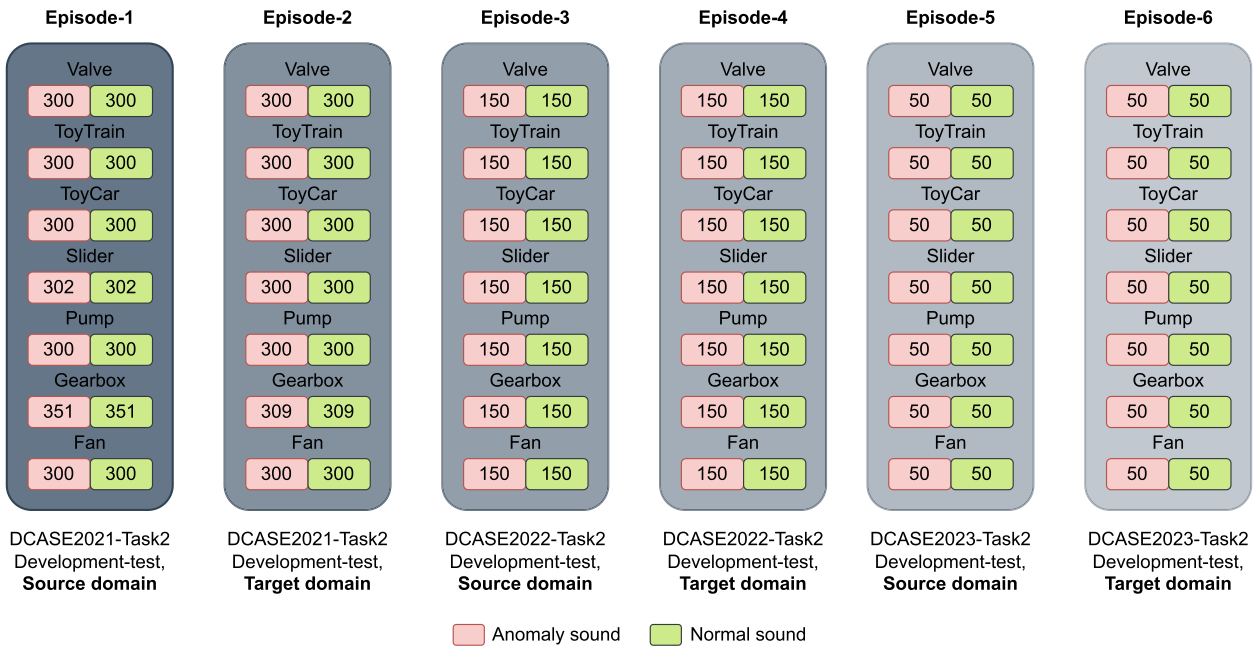

Fig. 2 Benchmarking dataset for DI scenario. Here, the CL model aims to distinguish between normal and abnormal machine sounds in domain shift conditions

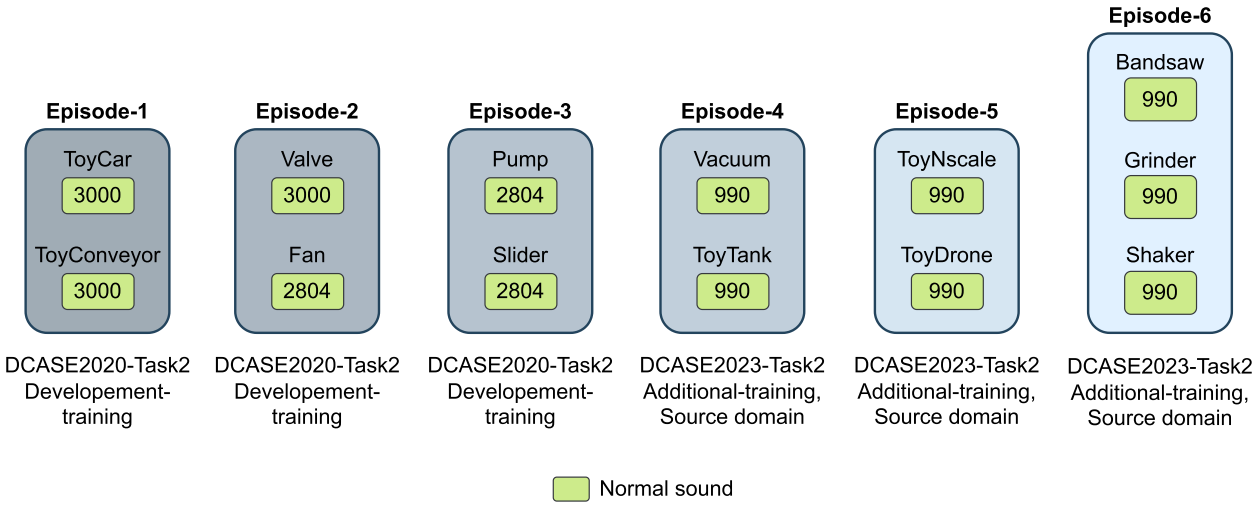

Fig. 3 Benchmarking dataset for CI scenario. Here, the classification model aims to distinguish among machine types in a class incremental fashion. Here, only the normal machine sounds of machine types are considered due to their larger availability

Table 3 Details of curated continual learning datasets for domain and class incremental scenarios. Here, T1, T2, . . . T6 denote the six episodes

Scenario	Application	Episodes	# train clips	# test clips
Domain-incremental	Anomalous sound detection	T1 (DCASE2021 source), T2 (DCASE2021 target), T3 (DCASE2022 source), T4 (DCASE2022 target), T5 (DCASE2023 source), T6 (DCASE2023 target)	3098, 3036, 1512, 1512, 504, 504	862, 844, 420, 420, 140, 140
Class-incremental	Machine sound identification	T1 (ToyCar, ToyConveyor), T2 (Valve, Fan), T3 (Pump, slider), T4 (Vacuum, ToyTank), T5 (ToyNscale, ToyDrone), T6 (Bandsaw, Grinder, Shaker)	4320, 4178, 4037, 1425, 1425, 2138	1200, 2361, 3483, 3879, 4275, 4869

Valve and Fan, have 3000 and 2804 samples, respectively. A visual representation of the CI scenario dataset is depicted in Fig. 3. For both the DI and CI scenarios, Table 3 summarizes the target applications, episode-wise data, and the number of training and testing clips per episode.

5 Experiments

This section outlines the experimental setup, evaluation of baseline CL models, and optimal hyperparameter selection. It defines the evaluation metrics used to assess performance and presents a detailed comparison of results, highlighting the effectiveness of different approaches.

5.1 Experimental setup

In all approaches, we use a pre-trained ResNet50 [62] and a vision transformer (ViT) [63] as base models, both initialized with ImageNet weights. We use Adam as an optimizer. In order to select the best hyperparameters for the classification model, multiple experiments are performed with the cumulative approach, as it gives the upper bound. Specifically, we search best values for epochs in {10, 15,... 50}, batch sizes in {4, 8,... 32}, and learning rates in $\{1e-04, \ldots 1e-05\}$. Best performance is achieved with a batch size of 8, a learning rate of $1e-04$, and epochs of 30 for the CI scenario. For the DI scenario, the best performance is achieved with a batch size of 8, a learning rate of $1e-03$, and epochs of 30. To facilitate a common evaluation scheme, the selected hyperparameters are used in all the approaches. Further, the hyperparameter values for each CL strategy, e.g., the regularization factor (λ), memory size ($\mathcal{B}$), etc., are selected from a range of values. The values reported below correspond to the best-performing settings identified through this search. For the DI experiment, we

Table 4 Summary of hyperparameter search spaces in continual learning methods

Hyperparameter	Method	Search space	Selection criterion and sources
λ	EWC, SI	{0.5, 1, 2, 3}	Best trade-off between stability and adaptability; higher λ improves stability but may reduce plasticity [33, 34, 64]
α	LwF	{1, 2, 3}	Balance knowledge retention and adaptation; higher α increases stability but can hinder adaptation [35, 64]
T	LwF	{1, 2, 3}	Smooth soft targets for better transfer; moderate T found most effective in experiments [35, 64]

set $\lambda = 2$ in EWC, $\{\alpha = 1, T = 1\}$ in LwF, and $\lambda = 0.5$ in SI. In case of the CI experiment, $\lambda = 2$ is set in EWC, $\{\alpha = 2, T = 2\}$ kept for LwF, and $\lambda = 2$ for SI. Further, for both DI and CI, the memory sizes ($\mathcal{B}$) are kept as 2000 samples for the GDumb and Replay approaches. Also, the patterns per episode are kept as 200 samples for the GEM and A-GEM approaches. In case of DER++, we kept $\{\alpha = 0.5, \beta = 0.5\}$ for both the scenarios, while for Co^{2}L we kept $T = 0.5$. A detailed discussion of experimental results across the hyperparameter ranges is provided in Sect. 6, while Sect. 5.3 reports the results obtained with the best-performing configuration for each CL method. The range of hyperparameter is summarized in Table 4.

5.2 Evaluation metrics

Here, we describe the evaluation procedure and metrics in CL with given T episodes. Each of the training sessions is a classification problem; therefore, we compute accuracy on the testing data of each episode. After completing the training session of t^{th} episode using training data Tr_t, we compute accuracy values on testing datasets $\{Te_1, Te_2, \ldots Te_T\}$ yielding T accuracy values. Thus, after completing all the training sessions on a stream of T episodes, we get a train-test matrix $R^{T\times T}$, where $R_{t,j}$ represents accuracy on the j^{th} episode after completing the training on the t^{th} episode. An example of a train-test matrix with 5 episodes is shown in Fig. 4. The train-test matrix evaluates stability and plasticity in CL, where rows represent training stages, and columns indicate testing phases. Plasticity is reflected in high accuracy on newly trained episodes ($R_{t,t}$), while stability is assessed by retaining performances of previous episodes ($R_{t,j}$ for $j < t$). Red regions indicate testing on untrained (unseen) episodes, blue regions show retained knowledge, and yellow regions highlight the capability to learn new data. This matrix helps assess how well a model adapts to new episodes while preserving past knowledge.

Apart from average accuracy across episodes, other important evaluation measures in CL include backward and forward transfer, which can be computed from this R matrix. Below, we describe metrics for forward transfer, backward transfer, and average model performance.

Backward transfer (BWT): BWT is a forgetting measure that quantifies the amount of influence of the current episode on the previously learned episodes [36]. It measures the improvement in the performance of an episode after learning new episodes. Thus, a positive value of BWT signifies that learning the current episode t has caused improvement in the performance of an old episode $k < t$. On the other hand, a negative value of BWT indicates that learning the current episode t has caused degradation in the performance of an old episode. Thus, a CL strategy with greater BWT values is superior to one with a smaller BWT. Blue and yellow cell values in Fig. 4 are utilized to compute BWT using the equation [65] below.

$$BWT = \frac{2}{T(T-1)} \sum_{i=2}^{T} \sum_{j=1}^{i-1} (R_{i,j} - R_{j,j}) \tag{1}$$

	Te_1	Te_2	Te_3	Te_4	Te_5
Tr_1	$R_{1,1}$	$R_{1,2}$	$R_{1,3}$	$R_{1,4}$	$R_{1,5}$
Tr_2	$R_{2,1}$	$R_{2,2}$	$R_{2,3}$	$R_{2,4}$	$R_{2,5}$
Tr_3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$	$R_{3,4}$	$R_{3,5}$
Tr_4	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$	$R_{4,5}$
Tr_5	$R_{5,1}$	$R_{5,2}$	$R_{5,3}$	$R_{5,4}$	$R_{5,5}$

Fig. 4 A representation of the train-test matrix achieved by sequentially training with 5 episodes

Forward transfer (*FWT*)**:** *FWT* signifies the amount of positive influence of learning an episode on future episodes. It also shows the zero-shot learning capabilities of the CL model. The higher the value of *FWT*, the better the generalization capability of the model. Red cell values in Fig. 4 are utilized to compute the *FWT* [65] measure with the following equation.

$$FWT = \frac{2}{T(T-1)} \sum_{t<j}^{T} R_{t,j} \tag{2}$$

Accuracy: given the train-test matrix $R^{T\times T}$, we compute average accuracy, *ACC* [36] across episodes after completing the final episode T as below:

$$ACC = \frac{1}{T} \sum_{j=1}^{T} R_{T,j} \tag{3}$$

Díaz-Rodríguez et al. [65] define a more rigorous measure of average accuracy by computing *ACC* after completing each episode, termed as *A*. It considers model performances at every training session t and thus better represents the dynamic aspects of CL.

$$A = \frac{2}{T(T+1)} \sum_{i\geq j}^{T} R_{i,j} \tag{4}$$

5.3 Results

This section presents the performance evaluation results for DI and CI scenarios, comparing different approaches based on key metrics. It includes the performance of CL strategies using the best-performing hyperparameter selections.

5.3.1 Performance analysis in domain-incremental scenario

For the DI scenario, we compute average accuracy over all episodes after completing a training session and report the results via Fig. 5. We can see that the joint training approach gives the upper bound of performance. CL approaches perform similarly. Further, the DER++ strategy and Co^{2}L strategy give the lowest performance compared to others.

A detailed comparison of the best performance achieved by various approaches is provided in Table 5. Specifically, for each approach, we report forgetting measures by *BWT*, forward transfer by *FWT*, incremental learning capabilities by *A*, and final average accuracy by *ACC*. Here, we can also see that joint and cumulative approaches in the non-CL category tend to outperform in all performance metrics as compared to CL approaches. However, the assumption of the availability of all the episodes together and retraining from scratch whenever a new episode arrives limits their applicability in real life. While comparing the CL approaches, we can see that LwF better handles forgetting, achieving the best *BWT* of −1.23, unlike EWC and SI. However, other metrics (*A* and *ACC*) are lower compared to those achieved by EWC and SI. Unlike [66], EWC shows better *A* (67.67) as compared to LwF (68.87), but usually LwF performs better. This shows that the CL strategy depends on the data and task, so our results complement earlier DI studies rather than contradict them. Remarkably, replay outperforms other CL approaches, including LwF, by achieving (*FWT*, *A*, and *ACC*) as {68.98, 70.12, 71.48}. Further, we can see that GDumb, with the same memory size as Replay, performs poorly. GEM and A-GEM approaches are highly compute-intensive compared to Replay and achieve lower performance than Replay. For the two newer methods, DER++ and Co^{2}L, performance in this DI setting is notably weaker. DER++ attains an *ACC* of only 50.02 and *FWT* of 49.66, indicating limited ability to transfer knowledge between domains. This is likely because it relies heavily on distillation and auxiliary loss functions without storing many past examples, which makes it less effective in preserving fine-grained domain-specific features. Co^{2}L performs somewhat better than DER++

Table 5 Performance comparison for DI scenario. First/second best performance in CL categories indicated in red/brown, respectively. Bold: Upper bound [36, 65]

	Method	Parameters	*BWT* [65]	*FWT* [65]	*A*[65]	*ACC* [36]
CL	EWC	$\lambda = 2$	-5.22	65.09	67.67	71.93
	LwF	$\alpha = 1, T = 1$	-1.23	65.85	66.87	68.34
	SI	$\lambda = 0.5$	-9.44	63.92	67.31	69.18
	GDumb	$\mathcal{B}$=2000	-2.25	61.55	61.60	61.42
	Replay	$\mathcal{B}$=2000	-2.22	68.98	70.12	71.48
	GEM	$\mathcal{B}$=200*6	-7.40	64.03	66.70	68.43
	A-GEM	$\mathcal{B}$=200*6	-6.83	64.91	67.28	67.32
	DER++	$\alpha = 0.5, \beta = 0.5$	-2.35	49.66	50.00	50.02
	Co^{2}L	$T = 0.5$	-3.91	56.29	57.32	56.92
Non-CL	Naive	-	-9.44	63.91	67.31	69.18
	Cumulative	-	**-0.17**	**71.23**	**72.15**	75.40
	Joint	-	–	–	–	**76.81**

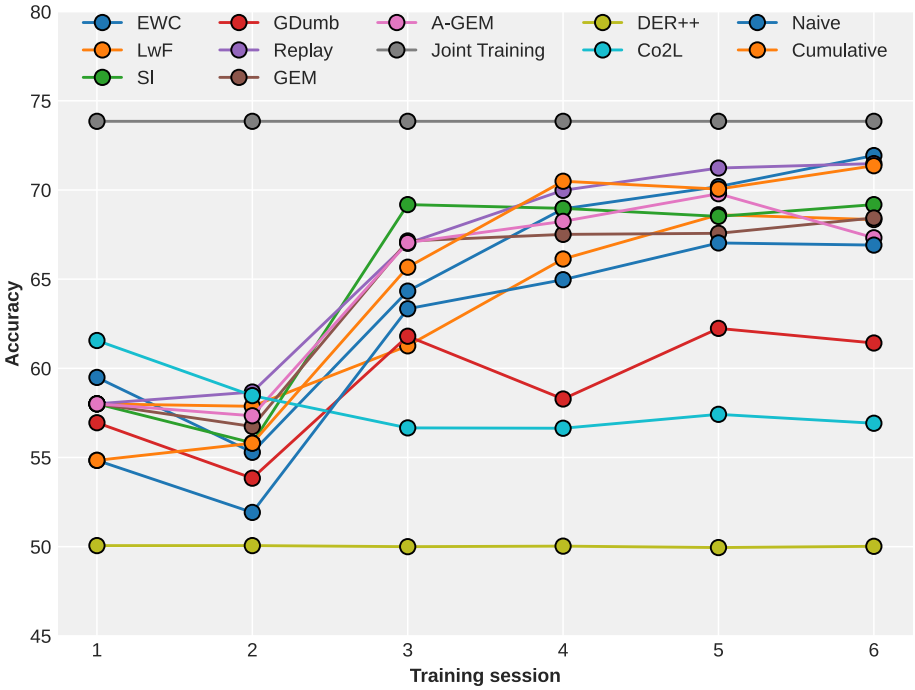

Fig. 5 Comparison of different approaches in terms of average accuracy over all episodes after each training session in DI scenario [36, 65]

($ACC = 56.92$, $FWT = 56.29$) due to its use of contrastive learning, but still falls well short of Replay. Without sufficient exemplar replay, it loses important details over multiple domain shifts, even though its representation learning helps to some extent.

Overall, these observations suggest that CL approaches may be picked based on application requirements. For example, a system demanding a solution with low memory will favor memory-free approaches such as EWC, LwF, or SI. Otherwise, Replay remains the most reliable choice in DI audio anomaly detection, offering the best overall balance of knowledge retention and adaptability.

5.3.2 Performance analysis in class-incremental scenario

Figure 6 shows the performance of the different approaches in terms of average accuracy on only seen classes after learning each episode in the CI scenario. It is expected that over the session, performance would drop as more and more classes are added, making learning difficult. However, Replay can maintain the performance at a steady level over the training session. We can see that it outperforms all other CL approaches, even surpassing non-CL strategies that are supposed to give an upper-bound performance, such as cumulative and joint.

We report a detailed comparison of all approaches in terms of *BWT*, *FWT*, *A*, and *ACC* in Table 6. We can observe that regularization-based approaches (EWC, LwF, SI) perform similarly to the naive approach, showing limited ability to retain knowledge when new classes are introduced. We observe LwF, EWC, and SI show *A* as 20.28, 10.79, and 10.13, respectively, which is similar to [28]. While comparing newer methods, DER++ performs slightly better than regularization-based approaches but fails to match the performance of other rehearsal-based strategies. It achieves an *ACC* of only 8.33 and *FWT* of 13.34, indicating severe forgetting and limited capability to transfer knowledge between tasks. This is likely because DER++ relies mainly on distillation and auxiliary loss functions without storing enough past examples, which is inadequate for large class shifts. Co^2L also performs poorly in this CI scenario, with *ACC* dropping to 2.42 despite achieving a slightly higher *FWT* (15.98) than DER++. Its contrastive learning framework improves representation learning to some extent, but cannot preserve past class knowledge without substantial replay. In contrast, Replay achieves the best overall results with (*BWT*, *FWT*, *A*, *ACC*) of $\{-1.04, 96.97, 96.98, 95.74\}$, significantly outperforming other rehearsal-based methods such as GDumb (77.18%), GEM (42.19%), and A-GEM (64.82%) in average accuracy. GDumb, although simpler and less carefully designed, still ranks second among CL methods, surpassing more complex approaches like GEM and A-GEM.

From Tables 5 and 6, we conclude that the rehearsal-based strategies perform well in both DI and CI scenarios, while regularization approaches show inadequate

Table 6 Performance Comparison for CI scenario. First/second best performance in CL categories indicated in red/brown, respectively. Bold: Upper bound

	Method	Parameters	*BWT* [65]	*FWT* [65]	*A*[65]	*ACC* [36]
CL	EWC	$\lambda = 2$	-54.60	0.61	10.79	2.07
	LwF	$\alpha = 2, T = 2$	-45.72	12.35	20.28	2.96
	SI	$\lambda = 2$	-54.77	0	10.13	2.03
	GDumb	$\mathcal{B}$=2000	-17.58	73.39	77.18	53.15
	Replay	$\mathcal{B}$=2000	-1.04	96.97	96.98	95.74
	GEM	$\mathcal{B}$=200*6	-39.87	34.34	42.19	33.01
	A-GEM	$\mathcal{B}$=200*6	-22.61	62.11	64.82	50.93
	DER++	$\alpha = 0.5, \beta = 0.5$	-35.24	13.34	20.50	8.33
	Co^{2}L	$T = 0.5$	-37.61	15.98	21.26	2.42
Non-CL	Naive	-	-54.96	0.55	10.81	2.03
	Cumulative	-	-4.93	86.18	85.92	88.22
	Joint	-	–	–	–	90.04

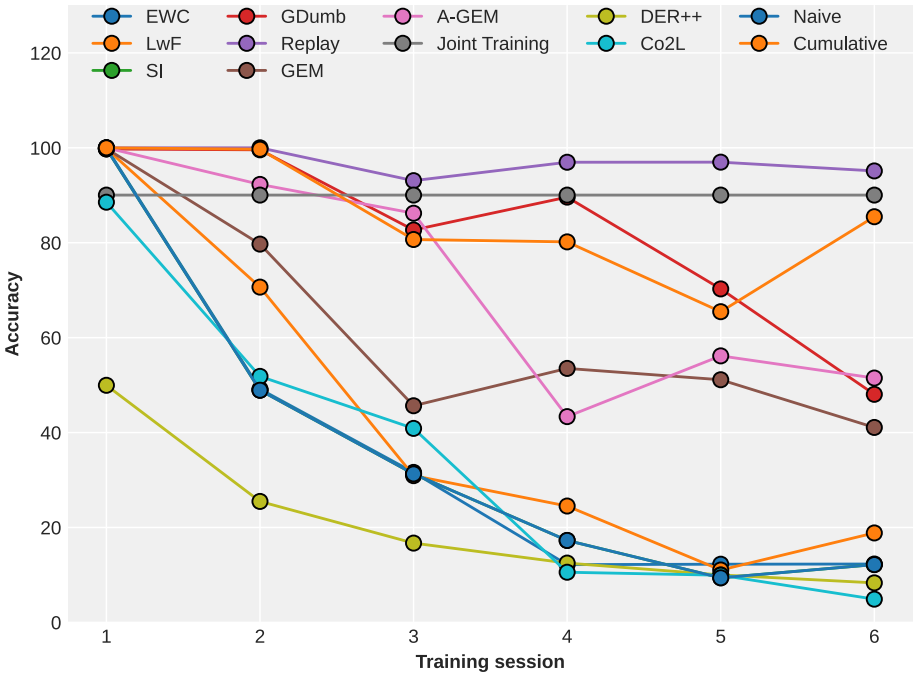

Fig. 6 Comparison of different approaches in terms of average accuracy after each training session in the CI scenario

performance in the case of CI scenarios. This is attributed to the fact that the CI scenario is the most challenging scenario [40] as the new episode consists of novel classes. Hence, it is very challenging to learn new episodes along with remembering the past just with the regularization step in the loss function. Some samples from past classes are important to rejuvenate the information from past classes in the current model. Further, we observe that Tables 5 and 6 show negative *BWT* in all approaches, indicating that none of the approaches can completely mitigate forgetting and suffer from stability issues. However, the considerable *FWT* highlights better forward transfer or plasticity.

5.4 Older vs. newer continual learning strategy

The results in Tables 5 and 6 show clear differences between older and newer CL methods. Among the older methods, Replay gives the best balance between remembering past knowledge and learning new data in both DI and CI scenarios. This is because it stores and reuses actual examples from previous tasks, which helps the model adapt without forgetting too much. GEM and A-GEM, although also using stored samples, require more computation and show lower performance, possibly because their gradient projection step limits their flexibility in learning new patterns.

For the newer methods, DER++ and Co^{2}L do not perform well in our setting. In the DI scenario, DER++

reaches an *ACC* of only 50.02, while Co^{2}L achieves 56.92, both much lower than Replay. This is likely because DER++ depends heavily on distillation and extra loss terms but stores very few past examples, and Co^{2}L relies on contrastive learning without strong replay, which makes it harder to preserve fine details from past tasks. The difference becomes even larger in the CI scenario, where new classes are introduced. Here, DER++ records an *ACC* of only 8.33 and Co^{2}L drops to 2.42, while Replay achieves 95.74. These results suggest that, although distillation and contrastive learning can improve feature representation, they are not enough to prevent forgetting in audio tasks without enough stored examples.

Overall, Replay-based methods remain the most effective for audio anomaly detection when the data changes over time. However, in cases where memory is very limited, simpler regularization-based methods like EWC or LwF may be used, though they will perform worse in more challenging CI scenarios.

6 Discussion

This section contains a detailed discussion of the performance of different approaches in DI and CI scenarios. Different approaches are evaluated with varying ranges of hyperparameters to analyze their effectiveness in mitigating catastrophic forgetting and facilitating knowledge transfer, shown in Table 4.

6.1 Analysis on domain-incremental scenario

From Table 7, we observe that different CL strategies perform equally well in the DI scenario for both ResNet50 and ViT. For ResNet50, among the regularization approaches (EWC, LwF, and SI), EWC benefits from increasing the regularization factor λ, improving *BWT* from -7.85 to -5.22 and *ACC* from 66.99 to 71.93, but it still struggles compared to memory-based methods. SI, which works similarly to EWC, does not show much improvement with higher λ. On higher λ, forgetting decreases (stability increases) in EWC and SI, while forward transfer (plasticity) increases in EWC but decreases slightly in SI. This difference comes from how the two methods work internally. EWC uses the Fisher Information Matrix (FIM) to find which parameters are most important for past tasks [33]. It then applies stronger constraints to these important parameters, while allowing other weights to change more freely. Increasing λ makes these constraints stronger, which reduces forgetting (better *BWT*) but still allows some learning of new tasks, leading to the moderate increase in *FWT* and *A*. SI, on the other hand, calculates parameter importance by tracking their changes during training using an online path integral [34]. However, it applies the constraints more evenly across all weight parameters compared to EWC. With a higher λ, this uniform restriction can make it harder to adapt to new tasks, which explains the drop in *FWT* and *A* even though forgetting is reduced. Similar stability–plasticity trade-offs have been noted in other CL reviews [67]. LwF shows that increasing temperature (T) improves performance, with *ACC* rising from 68.34 to 71.02 when T increases from 1 to 2 for $\alpha = 1$. However, when the weighting factor α is set to higher values, such as $\alpha = 3$, performance drops significantly, with *ACC* reaching only 58.72 at $T=1$. This suggests that placing too much emphasis on old episodes (higher α) can hinder adaptation to new episodes. Memory-based approaches outperform regularization methods by directly storing past knowledge. GDumb performs poorly, showing that naive storage is insufficient. Replay achieves strong performance, with *ACC* rising from 68.00 to 71.48 as memory increases from 50 to 2000, highlighting the benefit of larger buffers. Moreover, Replay executes faster than GEM and A-GEM, even if the memory increases. In addition, GEM and A-GEM achieve strong results. A-GEM, particularly with larger memory sizes (400*6), achieves the highest *ACC* of 70.96. At the same time, GEM at (400*6) reaches 69.96, showing that gradient-based memory can be nearly as effective as storing full samples while reducing memory overhead. However, GEM takes much longer to execute than A-GEM, even when using the same memory, making A-GEM a more efficient choice in terms of both performance and execution time. We observe that the performance increases with the increase in memory size [37]. Overall, memory-based methods outperform regularization-based ones, highlighting the importance of explicit knowledge retention in mitigating forgetting.

From Table 7, we also observe that when using Vision Transformer (ViT) as the base model for the DI scenario, CL approaches show some improvements over their performance with ResNet50. EWC benefits from higher values of λ, with its best performance reaching an *ACC* of 73.26, which is better than its ResNet50 counterpart. Similarly, LwF achieves better results with moderate values of α and T, with its highest *ACC* improving to 74.44. However, SI continues to struggle with forgetting, with only minor gains compared to ResNet50. Further, Replay achieves the highest *ACC* of 77.82 with a memory size of 2000, showing a clear improvement over its ResNet50 performance. GEM and A-GEM also deliver strong results, with A-GEM reaching 74.25 at 300*6 memory, benefiting from better gradient alignment. Although GDumb sees slight improvements, its naive memory usage remains a limitation. Overall, ViT enhances learning stability across all approaches, but memory-based approaches continue to be the most effective in reducing forgetting.

Table 7 Performance comparison of different approaches with different hyperparameters with ResNet50 and ViT backbones in DI scenario. $\mathcal{B}$ refers to memory size. 'Cum.' corresponds to the cumulative learning method. 'NA' indicates the experiments we were unable to run due to inapplicability (e.g., Joint) or intractable training time (e.g., GEM on memory size of 300 and 400)

		ResNet50				ViT			
Method	**Parameters**	***BWT***	***FWT***	***A***	***ACC***	***BWT***	***FWT***	***A***	***ACC***
EWC	$\lambda = 0.5$	−7.85	64.51	67.03	66.99	−7.01	69.83	**72.29**	**73.26**
	$\lambda = 1$	−6.65	63.86	66.79	68.46	−7.32	69.10	71.51	72.98
	$\lambda = 2$	**−5.22**	**65.09**	**67.67**	**71.93**	**−4.80**	**70.23**	71.96	71.47
LwF	$\alpha = 1, T = 1$	−1.23	**65.85**	**66.87**	68.34	−1.22	71.60	72.48	73.67
	$\alpha = 1, T = 2$	2.50	65.82	66.65	**71.02**	−5.13	71.42	73.22	74.03
	$\alpha = 1, T = 3$	−0.68	65.13	66.74	70.77	−6.06	69.80	72.20	73.87
	$\alpha = 2, T = 1$	−1.59	62.65	62.65	63.10	**1.38**	71.51	71.69	**74.44**
	$\alpha = 2, T = 2$	**2.92**	63.76	64.26	68.23	−2.18	72.09	**73.33**	74.36
	$\alpha = 2, T = 3$	0.73	65.03	66.02	69.81	−5.41	71.05	73.03	74.01
	$\alpha = 3, T = 1$	−2.87	60.03	60.97	58.72	0.80	68.76	69.01	71.60
	$\alpha = 3, T = 2$	2.26	61.48	62.01	65.65	−0.50	**72.11**	72.89	74.16
	$\alpha = 3, T = 3$	1.14	63.52	64.62	68.85	−1.02	71.73	72.83	74.42
SI	$\lambda = 0.5$	−9.44	**63.92**	**67.31**	**69.18**	−6.32	69.49	71.57	71.99
	$\lambda = 1$	**−7.86**	63.44	66.50	68.20	**−5.60**	69.30	71.69	72.70
	$\lambda = 2$	**−7.86**	63.44	66.50	68.20	−6.33	**70.00**	**72.33**	**72.92**
GDumb	$\mathcal{B}$=2000	−2.25	**61.55**	**61.60**	**61.42**	−1.84	**63.90**	**65.55**	**66.38**
	$\mathcal{B}$=1500	**−0.64**	60.21	60.14	58.69	−1.11	63.35	63.94	64.49
	$\mathcal{B}$=1000	−2.36	57.44	57.82	56.62	**−0.61**	59.42	59.81	63.37
Replay	$\mathcal{B}$=2000	**−2.22**	**68.98**	**70.12**	**71.48**	0.03	**75.88**	**76.53**	**77.82**
	$\mathcal{B}$=1500	−2.40	68.71	69.74	70.80	**0.15**	74.91	75.13	76.27
	$\mathcal{B}$=1000	−3.65	67.69	69.36	70.31	0.14	74.94	75.46	77.44
	$\mathcal{B}$=500	−3.88	66.69	68.49	70.61	−1.67	72.82	73.64	74.91
	$\mathcal{B}$=100	−7.17	64.37	67.02	68.61	−4.06	71.42	72.99	74.14
	$\mathcal{B}$=50	−6.67	64.06	66.77	68.00	−4.38	70.90	72.55	73.81
GEM	$\mathcal{B}$=10*6	−6.79	63.97	66.68	68.46	−4.55	71.32	**73.06**	73.35
	$\mathcal{B}$=50*6	−5.12	63.93	66.25	69.88	**−1.50**	71.49	72.58	**74.82**
	$\mathcal{B}$=200*6	−7.40	**64.03**	**66.70**	68.43	−2.36	**72.00**	72.96	74.21
	$\mathcal{B}$=300*6	**−1.06**	63.80	65.41	69.63	NA	NA	NA	NA
	$\mathcal{B}$=400*6	−1.78	63.71	65.46	**69.96**	NA	NA	NA	NA
A-GEM	$\mathcal{B}$=10*6	−3.72	64.68	66.70	68.99	−5.95	68.82	70.93	71.15
	$\mathcal{B}$=50*6	−2.66	65.13	67.07	69.12	−5.25	70.54	72.49	74.04
	$\mathcal{B}$=200*6	−6.83	64.91	**67.28**	67.32	−6.31	70.39	72.64	73.04
	$\mathcal{B}$=300*6	**−0.15**	**65.50**	66.81	70.47	**−5.20**	70.74	72.79	**74.25**
	$\mathcal{B}$=400*6	−1.44	64.91	66.76	**70.96**	−5.56	**71.24**	**73.20**	73.18
Naive	–	−9.44	63.91	67.31	69.18	−6.61	69.91	72.21	72.65
Cum.	–	−0.17	71.23	72.15	75.40	2.96	77.77	77.88	81.01
Joint	–	–	–	–	76.81	–	–	–	81.50

6.2 Analysis on class-incremental scenario

In the CI scenario, as shown in Table 8 for ResNet50, we observe that EWC, evaluated with different λ values, shows high forgetting (BWT −54.74 to −54.60) and minimal forward transfer (FWT near zero). Similarly, SI struggles with retention, exhibiting BWT between −54.96 and

Table 8 Performance Comparison of different approaches with different hyperparameters with ResNet50 and ViT backbones in CI scenario. $\mathcal{B}$ refers to memory size. 'Cum.' corresponds to the cumulative learning method. 'NA' indicates the experiments we were unable to run due to inapplicability (e.g., Joint) or intractable training time (e.g., GEM on memory size of 300 and 400)

		ResNet50				**ViT**			
Method	**Parameters**	***BWT***	***FWT***	***A***	***ACC***	***BWT***	***FWT***	***A***	***ACC***
EWC	$\lambda = 1$	−54.74	0.27	10.44	2.03	−55.75	1.41	**12.00**	2.03
	$\lambda = 2$	**−54.60**	**0.61**	**10.79**	**2.07**	**−55.03**	1.37	11.77	2.03
	$\lambda = 3$	−54.61	0.06	10.17	2.03	−55.29	**1.42**	11.89	2.03
LwF	$\alpha = 1, T = 1$	−48.68	13.62	21.91	14.31	−53.99	6.67	17.26	2.45
	$\alpha = 1, T = 2$	−30.69	22.20	26.73	28.29	**−52.15**	**10.27**	**20.85**	2.71
	$\alpha = 1, T = 3$	−40.85	13.12	20.38	22.80	−53.75	5.63	16.34	**6.38**
	$\alpha = 2, T = 1$	−18.99	36.75	37.75	36.99	−55.33	3.33	13.96	2.31
	$\alpha = 2, T = 2$	−45.72	12.35	20.28	2.96	−55.28	3.36	14.11	2.04
	$\alpha = 2, T = 3$	−37.07	16.52	22.79	28.35	−54.07	5.29	16.64	3.41
	$\alpha = 3, T = 1$	**−13.96**	**40.17**	**39.93**	37.46	−55.37	2.76	13.45	2.97
	$\alpha = 3, T = 2$	−27.60	27.46	30.96	**38.34**	−55.13	3.61	14.33	2.17
	$\alpha = 3, T = 3$	−50.42	9.63	18.39	11.39	−54.95	4.27	15.08	2.77
SI	$\lambda = 1$	−54.96	**0.55**	**10.81**	2.03	−53.23	**12.19**	**22.99**	2.03
	$\lambda = 2$	**−54.77**	0	10.13	2.03	−54.65	9.30	20.30	2.25
	$\lambda = 3$	−54.96	**0.55**	**10.81**	2.03	**−53.07**	8.17	19.01	**3.66**
GDumb	$\mathcal{B}$=2000	−17.58	73.39	**77.18**	**53.15**	−0.76	97.82	97.77	96.91
	$\mathcal{B}$=1500	−18.06	71.08	72.92	52.35	−0.58	98.36	98.31	97.39
	$\mathcal{B}$=1000	**−12.95**	**73.59**	74.14	52.13	**−0.43**	**98.80**	**98.76**	**97.91**
Replay	$\mathcal{B}$=2000	**−1.04**	**96.97**	**96.98**	**95.74**	**−0.13**	**99.52**	**99.51**	**99.39**
	$\mathcal{B}$=1500	−1.92	94.75	94.86	93.58	−0.36	99.19	99.20	98.88
	$\mathcal{B}$=1000	−1.88	94.50	94.58	93.53	−0.53	98.90	98.93	98.47
	$\mathcal{B}$=500	−2.87	94.01	94.30	92.11	−0.44	98.74	98.76	98.41
	$\mathcal{B}$=100	−1.08	86.66	85.66	82.99	−3.87	93.00	93.59	91.72
	$\mathcal{B}$=50	−5.24	78.95	78.77	75.79	−19.89	72.94	76.02	52.13
GEM	$\mathcal{B}$=10*6	−7.91	63.74	63.43	61.69	−9.22	83.83	85.63	82.94
	$\mathcal{B}$=50*6	**−5.47**	**66.76**	**65.97**	**64.69**	−1.49	95.27	95.41	95.61
	$\mathcal{B}$=200*6	−39.87	34.34	42.19	33.01	**−0.54**	**97.93**	**97.96**	**98.20**
	$\mathcal{B}$=300*6	−44.82	33.28	41.87	39.00	NA	NA	NA	NA
	$\mathcal{B}$=400*6	−53.92	21.39	31.76	11.75	NA	NA	NA	NA
A-GEM	$\mathcal{B}$=10*6	−28.50	27.90	33.08	45.59	−48.94	34.29	43.02	21.94
	$\mathcal{B}$=50*6	−26.20	31.70	36.49	50.31	−32.85	54.30	60.38	57.82
	$\mathcal{B}$=200*6	**−22.61**	**62.11**	**64.82**	50.93	−39.60	46.89	53.92	44.51
	$\mathcal{B}$=300*6	−26.50	59.73	63.34	39.73	**−31.29**	**55.07**	**60.93**	**74.80**
	$\mathcal{B}$=400*6	−23.30	60.43	63.29	**54.45**	−32.25	54.86	60.67	56.58
Naive	–	−54.96	0.55	10.81	2.03	−54.81	1.13	11.51	2.03
Cum.	–	−4.93	86.18	85.92	88.22	0.38	96.69	96.71	99.63
Joint	–	–	–	–	**90.04**	–	–	–	**100**

−54.77, highlighting its limitations in mitigating forgetting. In contrast, LwF shows significantly better BWT and FWT values across various hyperparameter settings. With higher T, BWT deteriorates, while FWT fluctuates rather than consistently improving. However, a higher α generally enhances performance. Particularly, LwF

($\alpha = 3, T = 1$) achieves the best results with a BWT of -13.96 and FWT of 40.17, indicating a forward knowledge transfer. Among rehearsal-based approaches, Replay ($\mathcal{B}=2000$) achieves the best results with minimal forgetting (BWT -1.04) and high forward transfer (FWT: 96.97). Even with less memory, Replay $\mathcal{B}=1000$) maintains high performance (BWT -1.88, FWT 94.50). While GEM $\mathcal{B}=50$*6) balances forgetting and transfer (BWT -5.47, FWT 66.76), increasing memory does not always improve performance, as seen with GEM $\mathcal{B}=400$*6), which suffers a significant drop in FWT. A-GEM follows a similar trend, with A-GEM $\mathcal{B}=200$*6) performing best (BWT -22.61, FWT: 62.11), though it still experiences notable forgetting compared to Replay. Additionally, in terms of execution time, Replay takes much less time to execute as compared to GEM/A-GEM as the computation of the gradient is eliminated. Also, as compared to A-GEM, GEM takes more time as it needs to solve a complex optimization step every time the model updates, which takes a lot of extra time. A-GEM makes this process much simpler by using only one projection with random memory samples, so it runs much faster. This means that even with the same memory size, A-GEM is more efficient than GEM in both speed and performance. GDumb ($\mathcal{B}=1000$) performs better than regularization-based methods, achieving BWT -12.95 and FWT: 73.59.

Also, from Table 8, we observe that ViT generally improves performance across various methods as Compared to ResNet50 similar to DI. EWC and SI still suffer from significant forgetting, with BWT values around -55, making them less effective for CI scenarios. This is so because each episode introduces entirely new classes that require high flexibility to learn. However, both EWC and SI impose strong constraints on parameter updates to protect old knowledge, which limits the model's adaptability. As a result, while forgetting is reduced, the model struggles to acquire the new classes effectively, leading to poor overall performance in a CI scenario. LwF performs slightly better than EWC and SI, especially with higher α values, improving forward transfer while still experiencing some forgetting. Among the rehearsal-based approaches, GEM and A-GEM show moderate performance. GEM $\mathcal{B}=200$*6) achieves a low BWT of -0.54 and a high FWT of 97.93, but increasing memory beyond this does not consistently improve results. A-GEM, on the other hand, struggles with high forgetting, with A-GEM $\mathcal{B}=10$*6) showing a BWT of -48.94, indicating severe forgetting. However, A-GEM $\mathcal{B}=300$*6) performs best among its variants, with a BWT of -31.29 and FWT of 55.07, still lagging behind Replay. Replay consistently outperforms all others, achieving retention and forward knowledge transfer. Even with lower memory, it maintains strong performance. Replay $\mathcal{B}=1000$ vs. $\mathcal{B}=2000$ shows only a slight drop in BWT (-0.53 vs. -0.13), FWT (98.90 vs. 99.52), A (98.93 vs. 99.51), and ACC (98.47 vs. 99.39). Even at $\mathcal{B}=100$, it still performs well (BWT $= -3.87$, FWT $= 93.00$, A $= 93.59$, ACC $= 91.72$). This shows that Replay remains effective even with reduced memory, with performance not dropping drastically as memory size decreases.

6.3 Effect of memory size on performance

As established from Sects. 6.1 and 6.2, Replay emerges as the most effective CL technique among all other explored CL techniques, demonstrating consistent performance and efficiency across different memory sizes. So, here we only focus on the Replay approach to study the effect of memory size on performance metrics. Figures 7 and 8 show {BWT, FWT, A, and ACC} by ResNet50 and ViT backbones for different memory sizes in DI and CI scenarios, respectively. We can observe that it yields larger

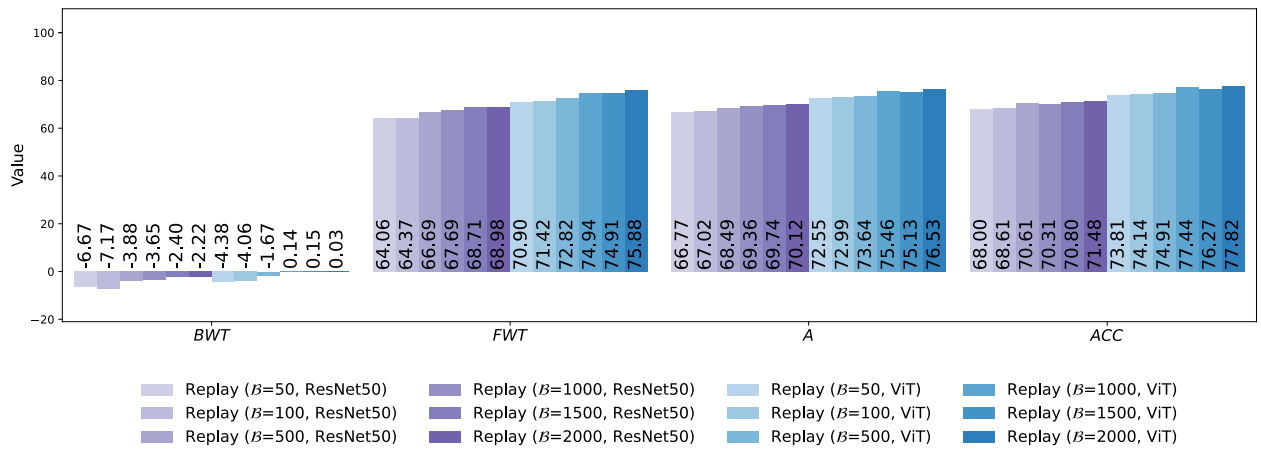

Fig. 7 Memory size ($\mathcal{B}$) vs. performance (A) with ResNet50 and ViT backbones in DI scenario

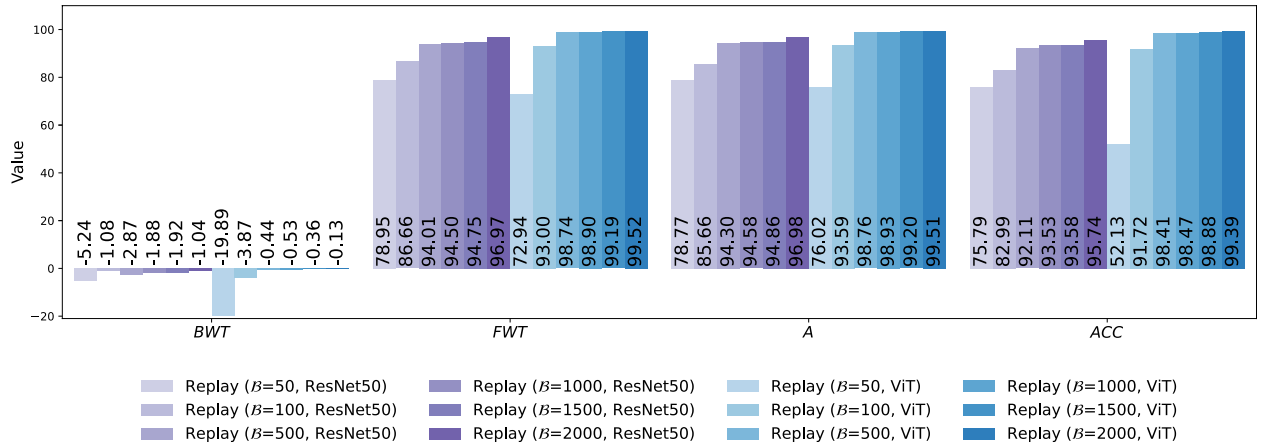

Fig. 8 Memory size ($\mathcal{B}$) vs. performance (A) with ResNet50 and ViT backbones in CI scenario

performance gains across all four metrics as memory size increases, with the most notable improvements occurring when the memory grows from very small sizes (e.g., 50–100 samples) to moderate sizes (500–1000 samples). Beyond this point, performance saturates, indicating that relatively moderate memory sizes are sufficient to achieve strong results without requiring excessively large memory. Figures 7 and 8 show that increasing memory size improves performance (A) for both ResNet50 and ViT, but the increase slows as memory grows. For ResNet50 in the DI scenario, A increases by less than 1% from 50 to 100 (66.77 to 67.02), about 1% from 500 to 1000 (68.49 to 69.36), and around 1% again from 1000 to 2000 (69.36 to 70.41), though memory usage becomes very high. In the CI scenario, A improves by about 7% from 50 to 100 (78.77 to 85.66), less than 1% from 500 to 1000 (94.30 to 94.58), and about 1% from 1000 to 2000 (94.58 to 95.48). Similarly, for ViT in the DI scenario, accuracy increases by about 2% from 50 to 100 (72.41 to 74.32), less than 1% from 500 to 1000 (75.44 to 75.89), and about 1% from 1000 to 2000 (75.89 to 76.53). In the CI scenario, accuracy rises by about 6% from 50 to 100 (93.63 to 99.51), but further increase is very small, with less than 1% improvement from 500 to 1000 (99.38 to 99.51) and no meaningful change beyond 1000. This highlights a clear trade-off in performance vs. storage requirements: while larger buffers can yield slightly higher accuracy, they also require greater storage capacity. In real-world audio monitoring, where memory resources may be constrained, such as edge devices, moderately sized buffers often provide the best balance between efficiency and performance. Furthermore, we observe that Replay with a stronger backbone, such as ViT, outperforms ResNet50, suggesting that the use of more powerful architectures can further amplify the benefits of the Replay technique in CL.

6.4 Results on different orderings of episodes

Based on the performance of the Replay strategy, we studied it further on different training strategies on DI. We are mainly trying to study the replay strategy further by changing the original dataset episode sequence to study the impact of the variation of a dataset on performance. The dataset is arranged in two different sequences to examine the impact of episode order on learning. The first sequence S1 (1, 2, 5, 6, 3, 4) corresponds to episodes from different years (2021, 2023, and 2022), analyzing how skipping between years affects learning retention and adaptation. The second sequence, S2 (1, 3, 5, 2, 4, 6), structures episodes such that data from the source domain is presented first, followed by corresponding target-domain data within each year. From Table 9, we observe that S2 results in better FWT and A for both architectures (ResNet50 and ViT), suggesting improved knowledge transfer across episodes. ViT achieves higher ACC and FWT compared to ResNet50, making it more effective in absolute performance. However, ResNet50 demonstrates more stable BWT, indicating less forgetting of previous episodes. These findings emphasize that while ViT excels in accuracy, ResNet50 may be a more stable choice for handling sequential learning, as it has stable BWT. This further supports the practical relevance of Replay in real-world applications, ensuring that audio

Table 9 Performance comparison of the Replay strategy with different backbones and sequences in DI scenario

Sequence	Backbone	*BWT* [65]	*FWT* [65]	*A* [65]	*ACC* [36]
S1	ViT	4.33	75.27	74.18	77.29
	ResNet50	3.77	67.12	66.77	72.16
S2	ViT	1.58	76.99	76.85	77.71
	ResNet50	1.20	70.02	70.20	71.30

monitoring systems can continuously adapt to new sound environments while maintaining past knowledge.

6.5 Limitation and future directions

CL in audio-based applications such as machine monitoring or farm surveillance faces several unexplored real-world challenges. These challenges include domain shifts caused by weather, seasonal changes, and dynamic background environments, which are not explored in this work. In this study, we focus on DI and CI scenarios, as these are the most studied (as discussed in Sect. 2.1.2) and do not capture multimodal scenarios (e.g., audio-video, audio-text), hybrid continual learning scenarios that combine new class and domain-shift challenges, or more complex conditions such as blurred domain boundaries where transitions occur gradually. In this work, we have curated the DCASE challenge dataset, which provided the DI and CI settings for the evaluations, given its well-structured normal/abnormal sound samples, annotations, consistent recording conditions, and natural domain variety, unlike other datasets like UrbanSound8K or ESC-50, which have only class variability and therefore can be used for only the CI scenario.

In future work, we can explore practical aspects such as robustness to background noise, handling noisy or imbalanced data, concept drift, hardware variability, and cross-domain transferability, which are crucial for real-world deployment. Extending evaluations to additional datasets for different scenarios would also enhance generalizability. For this, useful dataset criteria include: (i) coverage of both normal and abnormal sounds, enabling the construction of realistic DI and CI scenarios, (ii) offers consistent annotation and controlled recording conditions across multiple domains, and (iii) its domain structure aligns with our research goal of studying continual learning under gradual, naturally occurring drifts, and (iv) sufficient size for sequential task construction, which are supported by the DCASE dataset. One of the possible ways to simulate a hybrid scenario is to interleave the episodes from DI and CI scenarios created in Figs. 2 and 3. Thus, a sequence of 12 episodes with alternating domains and new class situations will serve as a hybrid scenario. Our goal is to extend this framework to multimodal datasets, hybrid continual learning settings, and the outlined practical challenges, thereby improving both robustness and generalizability for real-world monitoring applications.

7 Conclusion

We investigate CL approaches on DI and CI scenarios with 6 sequential episodes curated from a large range of DCASE datasets for machine sound monitoring. With the extensive comparative analysis performed, we observe that the performance of rehearsal-based strategies outperforms regularization-based CL approaches in the CI scenario, whereas both categories of approaches perform seemingly well in the DI scenario. We achieve overall accuracy (A) for the Replay approach as 76.53% (DI) and 99.51% (CI) with ViT and 70.12% (DI) and 96.98% (CI) accuracy with ResNet50. The results also indicate that performance generally decreases with smaller memory sizes, yet several strategies, particularly Replay, maintain competitive accuracy even with significantly reduced memory. This demonstrates that effective continual learning is achievable without large storage requirements, making these methods suitable for resource-constrained scenarios such as edge-device deployment. Overall, our analysis highlights the importance of selecting appropriate continual learning strategies suited to the needs and challenges of audio data, balancing accuracy with memory usage to ensure robust and practical performance in real-world applications.

Acknowledgements
Authors acknowledge ANNAM.AI, an AI-CoE of the Ministry of Education, Govt. of India at the Indian Institute of Technology Ropar, for resources and support to execute this work.

Code availability
Not applicable.

Authors' contributions
RB was the main contributor to the implementation of the experiment and the writing of the manuscript, PK analyzed the experimental results, PK and MS carefully proofread the manuscript, PPT helped in executing additional CL strategies as asked in the revision of the manuscript, and DM and AES provided valuable insights and suggestions for improving the paper's content. All authors read and approved the final manuscript.

Funding
This work is supported by the grant received from DST, Govt. of India, for the Technology Innovation Hub at the IIT Ropar in the framework of the National Mission on Interdisciplinary Cyber-Physical Systems.

Data availability
The data used in this work is generated from the DCASE challenge dataset. The curated dataset will be publicly available after the publication of the manuscript.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
All authors agreed to submit the manuscript for publication in the journal.

Competing interests
The authors declare that they have no competing interests.

Received: 1 September 2024 Accepted: 3 March 2026
Published online: 07 May 2026

References

1. P. Choudhary, P. Kumari, N. Goel, M. Saini, in *International Conference on Computer Vision and Image Processing*, Low-Intensity Human Activity Recognition Framework Using Audio Data in an Outdoor Environment (Springer, 2022), pp. 646–658

2. A.B. Nassif, I. Shahin, I. Attili, M. Azzeh, K. Shaalan, Speech recognition using deep neural networks: a systematic review. IEEE Access **7**, 19143–19165 (2019)
3. M. Crocco, M. Cristani, A. Trucco, V. Murino, Audio surveillance: a systematic review. ACM Computing Surveys (CSUR) **48**(4), 1–46 (2016)
4. Z. Wang, C. Subakan, E. Tzinis, P. Smaragdis, L. Charlin, in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Continual learning of new sound classes using generative replay (IEEE, 2019), pp. 308–312
5. H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.Y. Chang, T. Sainath, Deep learning for audio signal processing. IEEE J. Sel. Top. Signal Process. **13**(2), 206–219 (2019)
6. P. Choudhary, N. Goel, M. Saini, A fingerprinting based audio-seismic systems for human target localization in an outdoor environment using regression. IEEE Sens. J. **22**(8), 7944–7960 (2022)
7. P. Choudhary, P. Kumari, N. Goel, M. Saini, An audio-seismic fusion framework for human activity recognition in an outdoor environment. IEEE Sens. J. **22**(23), 22817–22827 (2022)
8. R. Bhatt, S. Singh, P. Choudhary, M. Saini, in *2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, An Experimental Study of the Concept Drift Challenge in Farm Intrusion Detection using Audio (IEEE, 2022), pp. 1–8
9. P. Kumari, M. Saini, Anomaly detection in audio with concept drift using dynamic huffman coding. IEEE Sens. J. **22**(17), 17126–17138 (2022)
10. P. Kumari, P. Choudhary, V. Kujur, P.K. Atrey, M. Saini, Concept drift challenge in multimedia anomaly detection: a case study with facial datasets. Signal Process. Image Commun. **123**, 117100 (2024)
11. P. Kumari, J. Chauhan, A. Bozorgpour, B. Huang, R. Azad, D. Merhof, Continual learning in medical image analysis: a comprehensive review of recent advancements and future prospects. Med. Image Anal. (2025). https://doi.org/10.1016/j.media.2025.103730
12. C. Shao, Y. Feng, Overcoming catastrophic forgetting beyond continual learning: Balanced training for neural machine translation (2022). arXiv preprint arXiv:2203.03910
13. R.M. French, Catastrophic forgetting in connectionist networks. Trends Cogn. Sci. **3**(4), 128–135 (1999)
14. A.G. Menezes, G. de Moura, C. Alves, A.C. de Carvalho, Continual object detection: a review of definitions, strategies, and challenges. Neural Netw. (2023). https://doi.org/10.1016/j.neunet.2023.01.041
15. M.M. Derakhshani, I. Najdenkoska, T. van Sonsbeek, X. Zhen, D. Mahapatra, M. Worring, C.G. Snoek, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Lifelonger: A benchmark for continual disease classification (Springer, 2022), pp. 314–324
16. P. Kumari, D. Reisenbüchler, L. Luttner, N.S. Schaadt, F. Feuerhake, D. Merhof, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Continual Domain Incremental Learning for Privacy-Aware Digital Pathology (Springer, 2024), pp. 34–44
17. P. Kumari, A. Bozorgpour, D. Reisenbüchler, E. Jost, M. Crysandt, C. Matek, D. Merhof, Domain-incremental white blood cell classification with privacy-aware continual learning. Sci. Rep. **15**(1), 25468 (2025)
18. Y. Sadegheih, P. Kumari, D. Merhof, Modality-independent brain lesion segmentation with privacy-aware continual learning (2025). arXiv preprint arXiv:2503.20326
19. T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, N. Díaz-Rodríguez, Continual learning for robotics: definition, framework, learning strategies, opportunities and challenges. Inf. Fusion **58**, 52–68 (2020)
20. Y. Wang, N.J. Bryan, M. Cartwright, J.P. Bello, J. Salamon, in *ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Few-shot continual learning for audio classification (IEEE, 2021), pp. 321–325
21. S. Karam, S.J. Ruan, Q.M. ul Haq, Task incremental learning with static memory for audio classification without catastrophic interference. IEEE Consum. Electron. Mag. **11**(5), 101–108 (2022)
22. N. Muthuchamy Selvaraj, X. Guo, A. Kong, B. Shen, A. Kot, Adapter incremental continual learning of efficient audio spectrogram transformers (2023). arXiv e-prints pp. arXiv-2302
23. X. Zhang, J. Yi, C. Wang, C. Zhang, S. Zeng, J. Tao, What to remember: Self-adaptive continual learning for audio deepfake detection (2023). arXiv preprint arXiv:2312.09651
24. M. Mulimani, A. Mesaros, Class-incremental learning for multi-label audio classification (2024). arXiv preprint arXiv:2401.04447
25. Y. Wang, J. Salamon, N.J. Bryan, J.P. Bello, in *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Few-shot sound event detection (IEEE, 2020), pp. 81–85
26. E. Koh, F. Saki, Y. Guo, C.Y. Hung, E. Visser, in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, Incremental learning algorithm for sound event detection (IEEE, 2020), pp. 1–6
27. J. Berg, K. Drossos, Continual learning for automated audio captioning using the learning without forgetting approach (2021). arXiv preprint arXiv:2107.08028
28. S. Mo, W. Pian, Y. Tian, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Class-Incremental Grouping Network for Continual Audio-Visual Learning (2023), pp. 7788–7798
29. W. Pian, S. Mo, Y. Guo, Y. Tian, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Audio-visual class-incremental learning (2023), pp. 7799–7811
30. H. Ma, J. Yi, J. Tao, Y. Bai, Z. Tian, C. Wang, Continual learning for fake audio detection (2021). arXiv preprint arXiv:2104.07286
31. S. Sadhu, H. Hermansky, in *Interspeech*, Continual Learning in Automatic Speech Recognition (2020), pp. 1246–1250
32. H. Zhou, K. Wang, H. Li, W. Yang, J. Yao, Y. Chai, Anomaly sound detection of industrial devices by using teacher-student incremental continual learning. Meas. Sci. Technol. **35**(5), 056107 (2024)
33. 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. Proc. Natl. Acad. Sci. U. S. A. **114**(13), 3521–3526 (2017)
34. F. Zenke, B. Poole, S. Ganguli, in *International conference on machine learning*, Continual learning through synaptic intelligence (PMLR, 2017), pp. 3987–3995
35. Z. Li, D. Hoiem, Learning without forgetting. IEEE Trans. Pattern Anal. Mach. Intell. **40**(12), 2935–2947 (2017)
36. D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning. Adv. Neural Inf. Process. Syst. **30** (2017)
37. A. Chaudhry, M. Ranzato, M. Rohrbach, M. Elhoseiny, Efficient lifelong learning with a-gem (2018). arXiv preprint arXiv:1812.00420
38. A. Prabhu, P.H. Torr, P.K. Dokania, in *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Gdumb: A simple approach that questions our progress in continual learning (Springer, 2020), pp. 524–540
39. D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, G. Wayne, Experience replay for continual learning. Adv. Neural Inf. Process. Syst. **32** (2019)
40. P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline. Adv. Neural. Inf. Process. Syst. **33**, 15920–15930 (2020)
41. H. Cha, J. Lee, J. Shin, in *Proceedings of the IEEE/CVF International conference on computer vision*, Co2l: Contrastive continual learning (2021), pp. 9516–9525
42. M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: defying forgetting in classification tasks. IEEE Trans. Pattern Anal. Mach. Intell. **44**(7), 3366–3385 (2021)
43. M. Mermillod, A. Bugaiska, P. Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects (2013)
44. S. Kim, L. Noci, A. Orvieto, T. Hofmann, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Achieving a Better Stability-Plasticity Trade-Off via Auxiliary Networks in Continual Learning (2023), pp. 11930–11939
45. L. Wang, X. Zhang, H. Su, J. Zhu, A comprehensive survey of continual learning: theory, method and application. IEEE Trans. Pattern. Anal. Mach. Intell. (2024). https://doi.org/10.1109/TPAMI.2024.3367329
46. M.J. Mirza, M. Masana, H. Possegger, H. Bischof, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, An efficient domain-incremental learning approach to drive in all weather conditions (2022), pp. 3001–3011
47. X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, Y. Gong, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Few-shot class-incremental learning (2020), pp. 12183–12192
48. G.M. van de Ven, T. Tuytelaars, A.S. Tolias, Three types of incremental learning. Nat. Mach. Intell. **4**(12), 1185–1197 (2022)

49. C. Chen, W. Min, X. Li, S. Jiang, Hybrid incremental learning of new data and new classes for hand-held object recognition. J. Vis. Commun. Image Represent. **58**, 138–148 (2019)
50. H. Li, P. Barnaghi, S. Enshaeifar, F. Ganz, Continual learning using bayesian neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(9), 4243–4252 (2020)
51. D. Maltoni, V. Lomonaco, Continuous learning in single-incremental-task scenarios. Neural Netw **116**, 56–73 (2019)
52. S.A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, icarl: Incremental classifier and representation learning (2017), pp. 2001–2010
53. A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks (2016). arXiv preprint arXiv:1606.04671
54. J. Yoon, E. Yang, J. Lee, S.J. Hwang, Lifelong learning with dynamically expandable networks (2017). arXiv preprint arXiv:1708.01547
55. Y.D. Kwon, J. Chauhan, C. Mascolo, Fasticarl: Fast incremental classifier and representation learning with efficient budget allocation in audio sensing applications (2021). arXiv preprint arXiv:2106.07268
56. Z. Wang, C. Subakan, X. Jiang, J. Wu, E. Tzinis, M. Ravanelli, P. Smaragdis, Learning representations for new sound classes with continual self-supervised learning. IEEE Signal Process. Lett. **29**, 2607–2611 (2022)
57. M. Mulimani, A. Mesaros, Incremental learning of acoustic scenes and sound events (2023). arXiv preprint arXiv:2302.14815
58. Y. Sun, X. Chang, G. Xu, Y. Wang, in *2023 International Conference on New Trends in Computational Intelligence (NTCI)*, Environmental Sound Classification Based on Continual Learning, vol. 1 (IEEE, 2023), pp. 155–159
59. Y. Wang, N.J. Bryan, J. Salamon, M. Cartwright, J.P. Bello, in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Who calls the shots? rethinking few-shot learning for audio (IEEE, 2021), pp. 36–40
60. G. Kim, H.H. Wu, L. Bondi, B. Liu, in *ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Multi-Modal Continual Pre-Training For Audio Encoders (IEEE, 2024), pp. 691–695
61. H. Shin, J.K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay. Adv. Neural Inf. Process. Syst. **30** (2017)
62. K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Deep residual learning for image recognition (2016), pp. 770–778
63. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale (2020). arXiv preprint arXiv:2010.11929
64. J. Armstrong, D.A. Clifton, in *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Continual learning of longitudinal health records (IEEE, 2022), pp. 01–06
65. N. Díaz-Rodríguez, V. Lomonaco, D. Filliat, D. Maltoni, Don't forget, there is more than forgetting: new metrics for continual learning (2018). arXiv preprint arXiv:1810.13166
66. Q. Wang, X. Song, Y. He, J. Han, C. Ding, X. Gao, Y. Gong, in *Proceedings of the Computer Vision and Pattern Recognition Conference*, Boosting Domain Incremental Learning: Selecting the Optimal Parameters is All You Need (2025), pp. 4839–4849
67. M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: defying forgetting in classification tasks. IEEE Trans. Pattern Anal. Mach. Intell. **44**(7), 3366–3385 (2022)

Publisher's Note