Pattern Recognition 179 (2026) 113514

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

EARL: Embracing amnesic replay for learning with noisy labels

Monica Millunzi [a,b,*], Lorenzo Bonicelli [a], Angelo Porrello [a], Jacopo Credi [b], Petter N. Kolm [c], Simone Calderara [a]

[a] University of Modena and Reggio Emilia, "Enzo Ferrari" Department of Engineering, Modena, 41125, MO, Italy
[b] Axyon AI, Viale dell'Autodromo, 210, Modena, 41126, MO, Italy
[c] New York University, Courant Institute of Mathematical Sciences, New York, 10012, NY, USA

ARTICLE INFO

Keywords:
Noisy labels
Continual learning
Experience replay
Forgetting

ABSTRACT

Modern Deep Neural Networks struggle to retain knowledge in streaming data environments, often leading to forgetting during incremental training. Most Continual Learning (CL) approaches address this issue by rehearsing past data – stored in a replay buffer – while acquiring new knowledge. However, in practical scenarios, noisy labels can contaminate the replay buffer, undermining performance. This work builds upon the previous "May the Forgetting Be with You", designed to tackle Continual Learning with Noisy Labels (CLN). By leveraging the distinct learning dynamics between correctly and incorrectly labeled examples, the method induces targeted forgetting to identify and filter out noisy labels. We propose EARL, which improves on its predecessor by introducing *i)* a detailed analysis of the learning dynamics occurring in the presence of noise, *ii)* a robust analysis under more realistic noise conditions, *iii)* an evaluation of performance using pre-trained backbones and modern prompt-based CL baselines, *iv)* a detailed study on the influence of different sampling strategies, *v) experiments on Natural Language Processing (NLP) benchmarks*. This work unravels the motivations and findings of the previous research, shedding light on the effectiveness of its components in achieving high performance and minimizing forgetting.

1. Introduction

To keep up with the ever-changing nature of data, modern AI systems require expensive and frequent re-training on *all* previously seen data to avoid the *catastrophic forgetting* [1] phenomenon. This has led to growing interest in Continual Learning (CL). In this field, one of the most promising solutions consists of storing a small amount of data in a memory buffer for later replay [2,3]. Such a strategy is usually referred to as Experience Replay [4,5] and relies on a *balanced* and *representative* set of exemplars to describe the past knowledge. In this respect, mislabeled samples can be particularly harmful to replay-based CL methods, as the noisy samples poisoning the memory buffer may further weaken the model on old tasks. Moreover, given the ever-growing amount of data generated by edge devices, it is impractical to manually label every incoming sample, or even fix semi-automatic annotations with human intervention. Thus, annotation noise is now common in many large-scale scenarios [6–8].

Despite the crucial role of robust and realistic lifelong learning, only a few preliminary works have addressed noisy labels in incremental scenarios [9–12]. These approaches exploit the established memorization effect [13–15] to detect the most reliable samples and primarily exploit these in training. The idea herein is that *the most reliable examples are those prioritized in the early stages of training* (those yielding the lowest loss values). Despite some potential errors, the majority of clean samples can be detected by analyzing the distribution of loss values.

In our study, we expand on previous research in [9] and argue that leveraging the memorization effect encounters limitations in continual learning. Indeed, as the model undergoes continuous fine-tuning, the clean-noisy loss gap decreases as tasks progress [16,17], hampering the effectiveness of the sample detection over time. The issue is often overlooked by current literature, which mainly focuses on *online* CL. Indeed, in this special setting, where only a single training pass is allowed for each task, the model is consistently far from the optimum, thus the memorization effect persists (Fig. 1 – *left*). Nevertheless, we warn against the limitations of such an experimental setup, which cannot fit tasks that demand multiple passes to achieve satisfactory performance [18] (Fig. 1 – *right*), or those characterized by immense amounts of data (*e.g.,* training large language models).

For this reason, this paper investigates the problem of learning with noisy labels from the perspective of offline Continual Learning.

* Corresponding author.
E-mail addresses: monica.millunzi@unimore.it (M. Millunzi), lorenzo.bonicelli@unimore.it (L. Bonicelli), angelo.porrello@unimore.it (A. Porrello), jacopo.credi@axyon.ai (J. Credi), petter.kolm@nyu.edu (P.N. Kolm), simone.calderara@unimore.it (S. Calderara).

https://doi.org/10.1016/j.patcog.2026.113514
Received 3 March 2025; Received in revised form 9 February 2026; Accepted 15 March 2026
Available online 27 March 2026

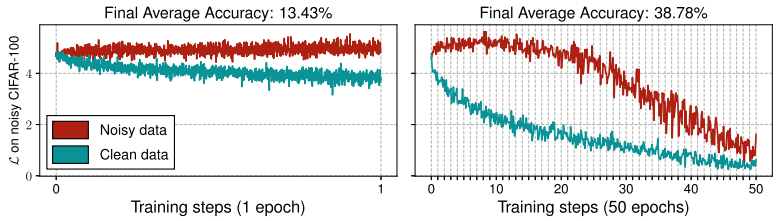

Fig. 1. Accuracy and loss trend for clean (*blue*) and noisy (*red*) samples on CIFAR-100 with 40% symmetric noise. The *left* shows that training for 1 epoch leads to underfitting with 13.4% accuracy. Training for 50 epochs (*right*) improves convergence, accuracy and reduces the loss gap. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Following our initial analysis, we propose a simple yet radical approach to overcome the issue of the vanishing loss gap between clean and noisy samples.

In doing so, we take inspiration from the outstanding works of [17,19], which mathematically prove how **mislabeled examples** exhibit **rapid forgetting**, while complex or rare instances are retained for longer periods (or not forgotten at all). Therefore, our approach intentionally induces forgetting through a strategy we call **Amnesic Replay**. While standard Experience Replay continuously optimizes both new examples from the incoming task and those stored in the memory buffer, Amnesic Replay periodically suspends the optimization on the examples present in the memory buffer. During this period, we observe that the loss associated with noisy examples increases more rapidly than that of clean data, thus providing a quantitative method to identify noisy samples during training. Moreover, such an intuitive strategy is refined through an auxiliary criterion (called *Bi-fold Loss-Aware sampling*) that seeks to retain the most significant samples from past tasks.

We show through several in-depth studies that our proposal – Embracing Amnesic Replay for Learning with Noisy Labels (**EARL**) – significantly improves the stability and performance of CL models while learning with noisy labels. Notably, EARL can be applied to any rehearsal-based technique and varying types of noise, from synthetically generated noise to realistic scenarios where noise arises during the data collection process. Specifically, we evaluate situations where label noise is introduced by human annotators, who may make errors during the annotation process, as well as by automatic annotation process e.g., collected by crawling the web. To mimicking automatic labeling pipelines, we conduct an experiment where the data are automatically annotated through the prediction of the zero-shot CLIP [20]. Finally, unlike the current state-of-the-art, our analysis also covers the continual fine-tuning of pre-trained models, a prominent trend in AI.

In summary, building upon the previous work, our extension includes: ***i)*** an analysis of the memorization effect in online and offline training regimes; ***ii)*** a comparison of noise from different realistic sources; ***iii)*** demonstration that EARL can be applied to a variety of continual learning methods, including those originally designed without buffers; ***iv)*** an analysis of results across different buffer sizes; ***v)*** an evaluation using pre-trained ViT-based models; and ***vi)*** evidence of the effectiveness of both the insertion and sampling strategies through comparisons with other methodologies. ***vii)*** a test on the applicability and effectiveness of EARL also on some NLP benchmarks.

We believe that the contributions above enhance both this approach and our previous work [9], providing a more in-depth analysis, refined methodology, and broader experiments that may further progress the CL community.

2. Related works

Learning with Noisy Labels. Most approaches dealing with learning from a noisy dataset are grounded on the memorization effect [13,15]. Under this assumption, clean samples tend to produce a smaller loss than mislabeled noisy ones during the initial stages of training (small-loss criterion), as they are typically *easier* to train upon. However, multiple epochs are usually necessary to generalize to complex distributions; hence, the reliability of the sample detection deteriorates as the loss gap between clean and noisy samples reduces. To avoid this issue, most existing methods seek to estimate the *noise transition matrix* [21], or detect the noisy samples and either drop them [14,15] or try to correct them [16,22]. Differently, Han et al. [23] builds on top of existing sample selection [14,15] strategies or loss correction [21,24] algorithms with an explicit gradient ascent loss. Instead, we analyze the effect of forgetting from the point of view of CL to take advantage of the ubiquitous forgetting phenomenon.

Sampling strategies. Given their limited capacity, memory buffers require a balanced outlook of all seen classes. For this purpose, many (e.g., Rainbow Memory [25], PuriDivER [11]) update the memory through *reservoir sampling* [26], which guarantees an i.i.d. snapshot of the past data points. However, not every sample comes with the same significance or robustness against forgetting: as highlighted by [25,27] retaining complex samples is crucial. Notably, such a characteristic can be detected by measuring the associated loss value or the model's uncertainty. Other works [28] use a prioritized exemplar selection based on *herding* [29], which aims to produce a buffer population whose feature distribution approximates best the true class distribution of the whole stream.

Continual Learning with Noisy Labels. Most of the research on CNL [10–12] has focused on rehearsal strategies for *online* (single epoch) CL. In this setup, conventional replay strategies fall short, as the buffer becomes contaminated with mislabeled samples. To address this challenge, PuriDivER [11] introduces a sampling strategy that seeks a balance between *purity* and *diversity* for buffer samples. Methods like SPR [10] and CNLL [12] use multiple buffers to gradually isolate clean samples. Although these methods can produce a purified memory buffer by exploiting the small-loss criterion, they are restricted to the online setting, where the effects of forgetting and underfitting are blended together [18]. Indeed, to avoid underfitting and remain competitive, they require extensive fine-tuning on the attained buffer after the end of each task. In complex scenarios with a limited amount of storage per task, the effectiveness of such a strategy is limited [30,31]. CLTR [32] introduces a time-varying regularization strategy, leveraging the observation that networks memorize clean samples before noisy ones; the method dynamically regulates updates for noisy *versus* clean samples. CO2L [33] leverages contrastive pretraining on clean data and preserves transferable representations through self-supervise d distillation, making it a good baseline to test in CLN. More recent approaches include NLOCL [34], which relies on an online buffer separation and a semi-supervised fine-tuning on labeled and unlabeled samples. In addition, CSReL [35], which proposes a reducible loss (ReL), *i.e.,* a forward-pass metric that estimates the marginal performance gain induced by adding a candidate sample to the coreset (memory buffer). By selecting samples that maximize estimated loss reduction relative to a holdout model, CSReL favors representative and informative data while suppressing noisy ones. RACE [36] is a replay-free method that uses a pretrained Vision Transformer to obtain robust feature representations. It combines

a confidence-weighted objective with an unsupervised clustering-based label correction stage that relabels samples via majority voting.

Theoretical foundations of Forgetting Dynamics. Our methodology is grounded on the principle that learning dynamics differ between clean and noisy samples, with noisy samples undergoing forgetting at earlier stages of training. While this intuition is not originally ours, we provide a brief overview of the findings in [17], as they constitute a key foundation for justifying the rationale behind our work.

In Section 5 of their work [17], the authors formalize a two-stage, over-parameterized linear model to demonstrate that the so-called second-split forgetting effectively filters out label noise first.

- **First-split learning**: Training a linear model on the linearly separable split S_A (with noisy labels) for T epochs until 100% accuracy results in weights $\mathbf{w}_A(T)$ that are close to the max-margin separator $\hat{\mathbf{w}}_A$ of S_A:

  $$\mathbf{w}_A(T) = \hat{\mathbf{w}}_A \log T + \rho_A(T),$$

  where $\rho_A(T)$ is a small residual. This reflects the implicit max-margin dynamics of gradient descent [37], which steer the model toward the hard-margin SVM solution while correctly classifying all (clean and noisy) training points.
- **Second-split forgetting**: In the second stage, the model is initialized with $w_B(0) = w_A(T)$ and further trained on a clean set S_B for $T' = f(T)$ epochs. Since S_B contains only correctly labeled examples, its influence gradually reorients the decision boundary toward the max-margin separator of the clean distribution. As a consequence, mislabeled examples from S_A become inconsistent with the new decision boundary, and their predictions are eventually flipped. In contrast, correctly labeled (including rare) examples remain compatible with the updated separator and are retained.

Briefly, **Theorem 2** of [17] (Intermediate-Time Forgetting) provides a high-probability guarantee that the following holds:
(i) Noisy samples from S_A are forgotten (their predictions flip to the correct label)
(ii) Clean and rare examples from S_A are retained (their predictions remain correct). The exact probability and the time T' depends on various factors, including class separability, model overparameterization, and the data's signal-to-noise ratio. Nonetheless, we computed empirical intermediate-forgetting timings for different architectures and refer the reader to Appendix B for such a detailed analysis.

3. Method

3.1. Problem setting

Following previous efforts in [9], we focus on Class-Incremental Continual Learning (ClassIL), where data comes as a sequence of tasks $t \in \{0, \dots, T-1\}$, each denoted as a separate classification dataset $D_t = \{\mathbf{X_t}, \mathbf{Y_t}\}$. During each task, data is assumed to be drawn from an i.i.d. distribution, but $D_i \cap D_j = \emptyset$ for all tasks $i \neq j$. In this scenario, an ideal model $f_\theta(\mathbf{x})$ should learn to classify all observed classes $\bigcup_{t=0}^{T-1} \mathbf{Y_t}$. Let $\mathcal{L}$ be a classification loss (*e.g.*, cross-entropy), then the aim is to solve:

$$\theta^* = argmin_\theta \mathbb{E}_t \Big[\mathbb{E}_{(\mathbf{x},y)\sim D_t} \Big[\mathcal{L}(f_\theta(\mathbf{x}), y) \Big] \Big], \tag{1}$$

Such an objective is not directly viable in CL, as we do not have access to $D_{<t}$. On the other hand, simply optimizing on the current task t results in biased predictions and forgetting of all previously acquired knowledge. To solve this issue, rehearsal methods leverage a small memory buffer $\mathcal{M}$ to store and replay part of the incoming samples. Formally, the generalized learning objective for replay-based methods is:

$$\theta^* = argmin_\theta \mathbb{E}_{(\mathbf{x},y)\sim D_t} \Big[\mathcal{L}(f_\theta(\mathbf{x}), y) \Big] + \mathcal{L}_R, \tag{2}$$

where $\mathcal{L}_R$ is an auxiliary *replay regularization* term. Although its form may vary depending on the method employed, we consider the basic Experience Replay [4,5]:

$$\mathcal{L}_R = \mathbb{E}_{(\mathbf{x}_r,y_r)\sim \mathcal{M}} \Big[\mathcal{L}(f_\theta(\mathbf{x}_r), y_r) \Big]. \tag{3}$$

Most replay methods based on Eq. (3) fill the memory buffer during training using a sampling strategy termed *reservoir* [26], which ensures that examples from subsequent tasks have an equal probability of being stored. Hence, while training, the memory buffer could contain examples from both current and past tasks.

Continual Learning under Noisy Labels. We aim herein to address the limitations of replay approaches in noisy scenarios, where the incoming task D_t includes a dataset contaminated with **erroneous annotations** $\tilde{y}_i \sim \tilde{Y}_t$. In this context, the memory buffer is prone to containing mislabeled examples, which can further degrade performance in subsequent tasks. Therefore, our main objective is to maintain a buffer with as few noisy samples as possible.

Due to the memorization effect of DNNs [13], to avoid retaining noisy examples, one can leverage the **small-loss criterion** [14,16,22,38], a well-established method for distinguishing between clean and noisy samples. Initially proposed in the context of standard offline learning, this criterion is based on the empirical observation that neural networks tend to learn easy, clean examples first. Thus, they can be identified by looking at their loss values $\mathcal{L}(f_\theta(\mathbf{x}), \tilde{y})$ after few epochs of training: if the loss is "low", the samples are likely labelled correctly.

We hence investigate the application of the small-loss criterion in a **continual setup**, considering a simple baseline based on vanilla Experience replay and *reservoir* sampling. To do so, we initially leverage a *synthetic* noisy scenario (for more realistic settings, including data crawled from the web, refer to Section 4). In detail, we inject annotation noise into Split CIFAR-10 [18] by randomly switching the labels of some examples, amounting to 40% of the training set (*noise rate*). Afterwards, we focus on the second task and report the per-sample loss in Fig. 2, for both examples from the current task (*Stream*, top) and the memory buffer (*Buffer*, bottom). Regarding the first, the **difference** (green curve) in loss value between noisy and clean examples peaks after the initial epochs and then deteriorates, corroborating the findings of previous works [16,17]. In contrast, for examples from the memory buffer, the loss difference stays near zero, making the small-loss criterion ineffective for distinguishing noisy/clean. This is due to the examples being overly optimized for rehearsal-based regularization, causing the model to quickly fit and memorize them [39].

3.2. Amnesic replay

Intuition. Motivated by the deterioration of the small-loss criterion for buffer datapoints, in the following we present a sampling approach that unearths noisy datapoints stored into the buffer. In essence, the idea is to periodically pause the optimization of the replay regularization term during training and then observe how the loss of each example in the memory buffer responds to this change. Intuitively, we expect that the loss of mislabeled samples learned by overfitting [16], will exhibit noticeable changes after several updates to the model's parameters [17,19,39], while correctly labeled samples will show negligible variations.

In formal terms, we extend the objective in Eq. (3) by introducing a binary variable $\delta = \{0, 1\}$, which we use to selectively enable and disable the replay regularization: $\mathcal{L}_R = [\delta = 1]\mathbb{E}_{(\mathbf{x}_r,y_r)\sim\mathcal{M}}\mathcal{L}(f_\theta(\mathbf{x}_r), \tilde{y}_r)$

By doing this, when $\delta = 0$ we are allowing f_θ to change while disregarding the samples from the buffer, thus *encouraging* their forgetting. We depict the effect of this strategy in Fig. 2, where δ takes the value 0 for one epoch and 1 for the next, alternating every epoch. As can be seen, alternating regularization and induced forgetting **exacerbates the loss difference**, with a remarkable impact not only on buffer data (bottom) but also on examples from the current task (top). Note that this discrep-

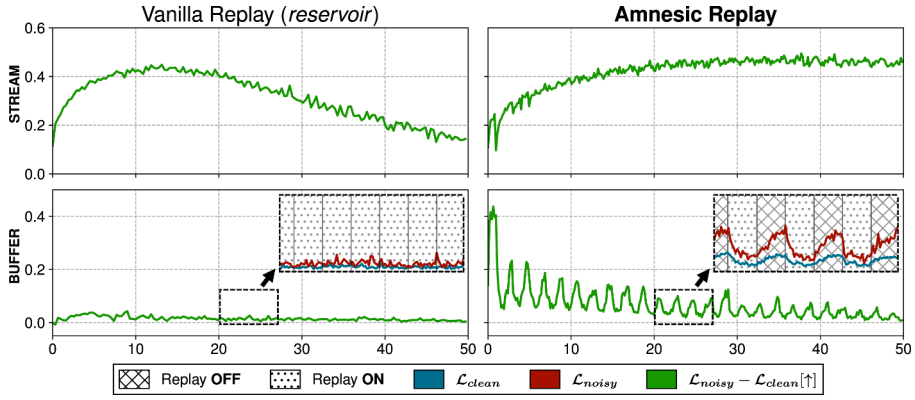

Fig. 2. Trend of the loss difference between clean and noisy data among training epochs, for Vanilla Replay (*reservoir*) and Amnesic Replay *(ours)*, with forgetting induced every other epoch.

ancy is maintained in the long term and also when replay regularization is re-enabled ($\delta = 1$).

Based on the analysis above, we propose a new strategy named **Amnesic Replay**, which encourages the separation of clean and noisy losses by inducing the model *temporary amnesia* (forgetting) on the buffer samples. To take advantage of the increased separation brought by forgetting, the sampling strategy of Amnesic Replay updates the memory buffer (*i.e.*, applying insertion and deletion operations) **only** when the replay regularization is paused ($\delta = 0$).

After pausing regularization, the model experiences severe forgetting, and repeated iterations may worsen performance. Therefore, we reload the model to its previous state during each transition from $\delta = 0 \rightarrow \delta = 1$.

3.3. Embracing amnesic replay for LNL

Here, we present the proposed sampling algorithm, called **E**mbracing **A**mnesic **R**eplay for **L**earning with noisy labels (EARL). It builds on the concept of Amnesic Replay introduced in the previous section and offers a method to: *i)* insert new examples into the memory buffer (*insertion*); and *ii)* remove old examples to make space (*deletion*).

Insertion. As shown before, Amnesic Replay allows for an effective application of the small-loss criterion in an incremental scenario. We leverage its effects by ranking current batch examples based on their loss value and follow the insertion procedure in [9] that, considering α as a specified percentage of samples from $|\mathcal{B}|$, it avoids using the α samples with the highest loss during the insertion phase.

Deletion. As shown in Fig. 2 (*right*), Amnesic Replay has a positive impact (*i.e.*, high loss difference) not only on buffer data (bottom) but also on examples from the current task (top). Unlike vanilla replay (Fig. 2, *left*), indeed, the loss difference remains significant as training progresses. Since the insertion technique employs the small-loss criterion, we can assert that most of the stored samples from the current task are likely to be **clean** by the end of the task. Therefore, in later tasks, instead of their *purity* (*i.e.,* the rate of clean examples, which is already high), we should prioritize their *diversity* [11], *i.e.,* the extent of intra-class variation captured by these examples.

It is with this intuition in mind that we devise a **bi-fold removal policy**, which applies different strategies depending on whether the example is from past tasks or the current task. Specifically, to select which examples should be replaced:

- **Case a)**. If the example is from the current task, we remain uncertain about the correctness of its label. Therefore, we continue to use the small-loss criterion, assigning a higher removal probability to examples with higher loss.
- **Case b)**. If the example is from the old tasks, based on both Amnesic Replay and the insertion policy, we **trust** its label. Indeed, if the example were mislabeled, it would have already been discarded under the clause a). Therefore, for these examples, we reverse the small-loss criterion, preferring to retain those with **higher loss** (associated with higher diversity, see below).

For clause Case b), the approach operates on the assumption that higher loss is linked to greater diversity in the data distribution. Intuitively, high loss values correspond to examples near the decision boundary between two or more classes [25,27]; therefore, these examples feature visual patterns that are heterogeneous and peculiar of distinct (but similar) classes, rendering them more varied than those lying on the mode of the data distribution. It is worth noting that high-loss examples have also been found beneficial in standard continual learning scenarios, as demonstrated by the authors of [27]. They showed that considering examples with high loss provides a simple yet effective criterion for modelling their importance during replay.

Formally, for each $\mathbf{x} \in \mathcal{M}$, we use the *score* function $s(\mathbf{x})$ in Eq. (4) to select the examples to be replaced. A summary of the entire procedure is available in the supplementary material. In that section, we provide additional details, such as how to achieve a balanced representation of samples from both past and current tasks (for instance, we re-normalize the scores based on the number of samples from either group stored in the memory buffer).

$$s(\mathbf{x}) = \begin{cases} \mathcal{L}(\mathbf{x}, \tilde{y}), & \text{if } (\mathbf{x}, \tilde{y}) \sim D_t \\ -\mathcal{L}(\mathbf{x}, \tilde{y}), & \text{if } (\mathbf{x}, \tilde{y}) \sim D_{<t} \end{cases}. \tag{4}$$

Notably, our method needs **only one single hyperparameter** (α), which streamlines its applicability across diverse scenarios.

4. Experiments

For the reasons discussed above, we follow the established offline ClassIL scenario [18,25,28,30], where multiple training epochs are allowed per task. We also use a task-aware approach to allow comparison with other works.

4.1. Setting

Datasets. To comply with the current CNL literature, we start our evaluation on convolutional models with no pre-training (Section 4.3). We test on the **CIFAR-100** dataset [40] with *synthetic uniform noise, where each label is randomly flipped to another class. We also consider* ***CIFAR-100N*** *[41], a human-annotated version of the dataset with instance-dependent noise, collected via Amazon Mechanical Turk, reflecting realistic annotation errors by non-experts. We also include* a version of CIFAR-100 annotated by a foundation model, *i.e.,* CLIP [20] in a zero-shot setup, which we name **CIFAR-100C** and simulates *automatic labeling pipelines*.

Table 1
Comparison of Final Average Accuracy and Final Forgetting (FAA [↑] ± std (FF) [↑]) of traditional CL and CLN methods for buffer size $\mathcal{M} = 500$. EARL consistently provides a performance boost, regardless of the source of noise.

Benchmark	CIFAR-100	CIFAR-100N	CIFAR-100C	ANIMAL-10N
noise source	*synthetic*	*human-annotation*	*machine-annotation*	*web-scraped*
noise rate	*40%*	*40.20%*	*35.31%*	*08.00%*
Multitask	38.46 ± 0.92 (-)	47.72 ± 0.22 (-)	55.20 ± 0.89 (-)	57.35 ± 0.77 (-)
Finetune	07.55 ± 0.14 (71.51)	8.66 ± 0.06 (79.56)	8.73 ± 0.03 (80.99)	13.73 ± 0.05 (78.52)
*i*DivideMix [22]	10.88 ± 0.60 (20.71)	16.28 ± 0.62 (25.23)	18.52 ± 0.81 (25.47)	32.59 ± 0.35 (26.79)
PuriDivER [11]	08.16 ± 0.43 (67.30)	10.06 ± 0.32 (77.53)	11.05 ± 0.67 (78.33)	13.69 ± 0.60 (73.54)
CLTR [32]	8.40 ± 0.36 (64.21)	10.74 ± 0.42 (69.13)	14.30 ± 0.50 (68.95)	16.01 ± 0.55 (67.47)
CO^2L [33]	16.51 ± 0.71 (45.92)	18.32 ± 0.89 (49.28)	16.84 ± 0.77 (41.10)	26.92 ± 0.64 (29.15)
DER++ [18]	13.80 ± 0.28 (50.22)	23.45 ± 0.37 (53.67)	30.05 ± 1.20 (48.40)	30.29 ± 0.27 (41.26)
w. **EARL**	26.37 ± 0.58 (41.08)	30.13 ± 1.21 (36.57)	33.23 ± 0.98 (33.33)	31.80 ± 0.31 (33.43)
ER-ACE [31]	12.64 ± 0.04 (42.14)	25.48 ± 0.59 (38.57)	30.46 ± 0.28 (33.54)	31.85 ± 1.07 (27.46)
w. **EARL**	**27.94 ± 0.16** (30.24)	**30.69 ± 0.56** (28.78)	**33.23 ± 0.19** (26.97)	**34.22 ± 0.87** (18.35)

In addition, to cover most sources of noise, we include **ANIMAL-10N** [8] and **FOOD-101N** [42], two datasets containing *web-scraped* data with naturally noisy labels originating from surrounding metadata or captions, a common scenario in large-scale web-based data collection.

For experiments involving a pre-trained backbone, we wish to evaluate both the resilience to noise and the plasticity of the models. Therefore, we primarily focus on **ISIC** [43,44] and **EuroSAT-RGB** [45,46], as these two datasets hold low domain similarity [47] w.r.t. the pre-train (ImageNet [48]).

We define sequential CL tasks for each dataset following the ClassIL setting. Namely, for Food-101N, ANIMAL-10N, and EuroSAT-RGB we split the classes into 5 tasks. We split ISIC into 3 tasks and CIFAR-100 into 10 tasks.

To also evaluate a Continual NLP task, we include a subset of the **GLUE benchmark** [49], a widely used collection of language understanding tasks. Specifically, we consider six sentence- and sentence-pair classification tasks-MNLI, SICK, RTE, SciTail, QNLI, and SNLI-which are encountered sequentially in this order. Together, they form a single dataset of 6 tasks.

Backbones. We employ a Vision Transformer (ViT) [50] for ISIC, EuroSAT-RGB, and Food-101N, and a ResNet18 [51] for CIFAR-100 and ANIMAL-10N.

Metrics. All results are presented in terms of Final Average Accuracy (FAA) and Final Average Forgetting (FF) and averaged across 3 runs, computed at the end of the last training task. We refer the reader to the supplementary material for further details.

4.2. Baseline methods

CL-based methods. Since our work stands out for being the first investigating noisy labels in an offline CL setting, we assess EARL's effectiveness by applying it to a selection of both pre-trained and initialized from scratch architectures. For the former, we consider the ViT-B/16 architecture to allow comparison against prompt-based approaches. In particular, we consider **L2P** [52] and **CODA-Prompt** [53], as they represent the most widely adopted methods for rehearsal-free learning. Moreover, we also consider **SLCA** [54], as it stands out for achieving higher performance w.r.t. prompting in most scenarios. For rehearsal-based methods, we employ **ER-ACE** [31] and **DER++** [18] due to their simplicity and effectiveness. Unless otherwise noted, L2P, CODA-Prompt, and SLCA do not make use of a memory buffer. However, since we find that they fall short in the presence of label noise or domain dissimilarity w.r.t. the pre-train, we will also equip them with a small memory buffer based on ER-ACE.

CLN method. For a thorough comparison, we include the currently available CLN-based method, adapted for a multi-epoch scenario. In particular, we compare against **PuriDivER** [11],, **SPR** [10], and **CNLL** [12]. Since the last two methods use multiple memory buffers, we use the same overall memory budget for a fair comparison and test on a smaller dataset. Details and results of these two methods in the multi-epoch scenario in the Appendix. We adapted the **CLTR [32] regularization to our incremental task scenario by applying it to both stream and buffer samples. Additionally, we leverage clean pretraining distillation for our noisy tasks through CO^2L [33]**. We include an additional baseline that applies the regularization of DivideMix [22] on samples from all seen tasks using a reservoir memory buffer, which we name ***i*DivideMix**. We select DivideMix as a compelling representative baseline for LNL methods because it consistently outperforms similar noise-robust learning methods and sample-selection approaches across several benchmarks [11,41].

Finally, we provide an upper bound (**Multitask**, *i.e.,* training on all tasks jointly) and a lower bound (**Finetune**, *i.e.,* training with no measures against forgetting or label noise).

4.3. Results

Not pre-trained backbones. We analyze the benefits brought by EARL on popular rehearsal baselines by computing the Final Average Accuracy (FAA) these exhibit on different datasets, before and after applying EARL to them (Tables 1 and 2). We can see that our proposal improves the performance of all base methods on both synthetic (CIFAR-100) and real (CIFAR-100N, CIFAR-100C, ANIMAL-10N) noisy benchmarks. In the table, we also report the Final Average Forgetting (FF) for each experiment. To delve into more detail, the average gain in FAA points across tasks is 14.99 on CIFAR-100, 4.76 on CIFAR-100N and 2.25 on CIFAR-100C. It's worth noting that EARL remains effective even at lower noise levels, demonstrating an average improvement of 4.22 points on ANIMAL-10N. This modest increase can be ascribed to the limited ratio of noisy data in this dataset. Finally, in all scenarios, we demonstrate to surpass the LNL and CNL competitors by far.

Pre-trained backbones. We aim to examine pre-trained models in noisy environments, with Table 3 showing the Final Average Accuracy on two datasets with injected noise.

We highlight the advantages of integrating a buffer into pre-trained CL prompt-tuning methods, particularly in noisy environments. This becomes evident when comparing the results in Table 3 for the three methods (CODA-Prompt, L2P, SLCA) with ($\mathcal{M} = 500$) and without buffer ($\mathcal{M} = 0$). Furthermore, using the buffer, we can boost the performance of all models with EARL. Specifically, we achieve an average increase of accuracy of 7.73 and 20.21 for experiments conducted respectively on ISIC and EuroSAT-RGB.

Table 2
Comparison of Final Average Accuracy and Final Forgetting (FAA [↑] ± std (FF) [↑]) of traditional CL and CLN methods for a fixed buffer size $\mathcal{M} = 2000$. EARL consistently provides a performance boost, regardless of the source of noise.

Benchmark	CIFAR-100	CIFAR-100N	CIFAR-100C	ANIMAL-10N
noise source	*synthetic*	*human-annotation*	*machine-annotation*	*web-scraped*
noise rate	*40%*	*40.20%*	*35.31%*	*08.00%*
Multitask	38.46 ± 0.92 (-)	47.72 ± 0.22 (-)	55.20 ± 0.89 (-)	57.35 ± 0.77 (-)
Finetune	07.55 ± 0.14 (71.51)	8.66 ± 0.06 (79.56)	8.73 ± 0.03 (80.99)	13.73 ± 0.05 (78.52)
*i*DivideMix [22]	20.09 ± 1.13 (13.58)	28.37 ± 1.20 (13.99)	32.26 ± 1.29 (12.66)	36.08 ± 1.44 (19.90)
PuriDivER [11]	17.46 ± 0.79 (64.21)	12.25 ± 0.91 (75.63)	14.20 ± 0.58 (73.82)	18.36 ± 0.96 (66.94)
CLTR [32]	10.42 ± 1.08 (74.60)	17.13 ± 0.87 (62.27)	22.36 ± 0.84 (58.20)	25.17 ± 0.67 (55.29)
CO^2L [33]	24.07 ± 0.68 (41.15)	30.44 ± 0.74 (34.45)	34.14 ± 0.62 (36.04)	31.21 ± 0.71 (33.09)
DER++ [18]	21.68 ± 0.67 (44.53)	35.05 ± 0.96 (40.19)	40.77 ± 0.88 (34.16)	32.41 ± 0.72 (32.50)
w. **EARL**	39.26 ± 1.14 (27.98)	**38.75 ± 1.08** (27.58)	**41.78 ± 0.77** (25.39)	37.66 ± 1.65 (26.41)
ER-ACE [31]	22.20 ± 0.72 (34.55)	34.73 ± 0.73 (30.67)	39.30 ± 0.37 (27.88)	37.29 ± 0.30 (24.67)
w. **EARL**	**40.35 ± 0.25** (21.12)	38.51 ± 0.83 (22.06)	41.00 ± 0.13 (20.89)	**38.66 ± 0.88** (11.15)

Table 3
Results in terms of FAA [↑] and (FF) [↓] on benchmarks using a pre-trained ViT.

	Benchmark *noise*	EuroSAT *40% symm*	ISIC *40% symm*
	no noise	96.88 (–)	78.25 (–)
	Multitask	93.17 (–)	50.60 (–)
	Finetune	18.73 (89.53)	29.93 (79.37)
$\mathcal{M} = 0$	CODA-Prompt [53]	60.78 (16.61)	41.97 (4.09)
	L2P [52]	48.37 (02.78)	32.87 (03.49)
	SLCA [54]	35.87 (77.72)	31.85 (70.89)
$\mathcal{M} = 500$	CODA-Prompt [53]	62.97 (18.72)	43.37 (23.77)
	w. **EARL**	87.95 (06.88)	52.99 (22.34)
	L2P [52]	77.72 (12.07)	49.03 (09.26)
	w. **EARL**	80.34 (08.67)	51.88 (12.09)
	SLCA [54]	56.26 (11.49)	41.01 (30.49)
	w. **EARL**	92.28 (03.68)	54.18 (17.05)
	ER-ACE [31]	76.39 (18.10)	52.16 (20.67)
	w. **EARL**	93.62 (02.48)	56.00 (21.47)

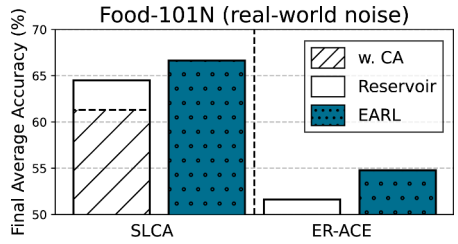

Fig. 3. Final average accuracy (FAA) [↑] of EARL when applied to SLCA and ER-ACE to learn on the Food-101N dataset.

Remarkably, with the use of EARL, we surpass the Multitask case in certain scenarios, *i.e.,* our upper bound on the noisy dataset. Therefore, we also provide the upper bound of the conventional scenario, involving the multitask model trained on the same datasets without any noise.

In Fig. 3 we also evaluate two models (pre-trained and not) on Food-101N. Here, we find an increase in accuracy for both SLCA and ER-ACE. Consistent with the results in Table 3, the advantages of employing a buffer *vs.* not using one are evident (left bar plot). Furthermore, EARL is beneficial for both models.

Foundation models as erroneous annotators. Due to the high costs of human annotation, an emerging trend involves the use of pseudo-labels generated by vision-language models with high zero-shot performance [55]. However, these models are not infallible, and incorrect pseudo-labels introduce challenging label noise.

We study this label noise by using CLIP with ViT-B/32 to re-annotate the CIFAR-100 training dataset, simulating annotation from an *external automatic source unavailable for training*. The noise rate thus corresponds to CLIP's error rate (35.31%). We apply all methods from Section 4.3 and present results in Tables 1 and 2 (3rd column). As can be seen, EARL continues to provide a notable performance gain (2.65% on average) even under this peculiar form of noise.

Natural Language Understanding. Since noisy labels may also occur in the robust natural language understanding (NLU) field, we provide a small study testing the effectiveness of our method on a subset of the General Language Understanding Evaluation (GLUE) benchmark [49], a widely used collection of natural language understanding tasks including question answering, sentiment analysis, and textual entailment. The six sentence- and sentence-pair classification tasks considered are encountered sequentially, forming a single dataset of 6 tasks. Multitask and Finetuning serve as baselines. For continual learning, we store 5000 examples in a buffer, which provides sufficient coverage across all tasks while remaining memory-efficient. Noise is synthetically introduced by randomly flipping labels. As shown in Table 4, the regularization provided by EARL effectively cleans the buffer, yielding consistent benefits in terms of Final Average Accuracy in both Class-IL and Task-IL scenarios.

5. Model analysis

Question i) How do **sampling strategies** affect EARL's overall performance? *Question ii)* How do sampling strategies influence overall **buffer purity** and diversity? *Question iii)* How sensitive is the model to α and to other selection strategies? *Question iv)* Does EARL remain effective under **low or no noise**?

Comparing against different Sampling Techniques. To assess the validity of our sampling strategy, we here compare it against some state-of-the-art sampling techniques. In particular, we conducted experiments using the following:

- PuriDivER [11]: seeks to balance purity and diversity in the replay buffer. This is achieved through a score function that considers both the likelihood of a sample being correctly labeled (purity) and its representational uniqueness (diversity). From the best of our knowledge, this is the current state-of-the-art method that has been proposed to address the problem of noisy labels in online continual learning, which is the closest to our scenario.

Table 4
Comparison of final average accuracy for a text benchmark in both Class-IL and Task-IL scenarios, with buffer purity reported when a buffer is used.

M = 5000	*Class-IL*	*Task-IL*	*Class-IL*	*Task-IL*	*Class-IL*	*Task-IL*	*Class-IL*	*Task-IL*
GLUE	**Multitask**		**Finetune**		**ER-ACE**		***w.* EARL**	
Noise 20%	74.20	80.06	14.88	58.12	64.44	76.70	69.55	79.66
Buff. Purity	✘		✘		0.83		0.98	
Noise 40%	65.89	68.96	14.06	53.78	54.85	65.67	62.95	73.30
Buff. Purity	✘		✘		0.64		0.90	

Table 5
Comparison against state-of-the-art sampling strategies across different sources of noise. Final Average Accuracy on CIFAR and Food-101N; buffer size = 500.

Dataset	CIFAR–100				Food–101N
Noise	sym 20%	sym 40%	sym 60%	Instance-based	Web-based
PuriDivER	12.04	8.38	5.29	08.03	**29.85**
Rainbow	13.53	07.79	04.30	19.28	55.83
Herding	16.92	10.23	05.55	20.52	56.80
Bi-Fold (ours)	26.69	25.17	19.17	24.05	57.78

Table 6
Bi-fold *vs.* Herding and PuriDivER's sampling strategies, with both Vanilla and Amnesic Replay.

CIFAR-100N	$\mathcal{M} = 500$		$\mathcal{M} = 2000$	
Sampling Strategy	*Vanilla*	*Amnesic*	*Vanilla*	*Amnesic*
Puridiver	19.50	23.46	24.15	30.84
Herding	**30.11**	30.75	35.98	37.56
Bi-fold	29.80	**31.27**	**38.00**	**38.80**

- Herding [28,29]: a widely-adopted sampling strategy that focuses on selecting samples that most closely represent the current model's learned features for each class. It does this by selecting samples that minimize the distance to the class means in feature space.
- Rainbow Memory [25]: this strategy selects, for each class, samples that are diverse in the feature space by considering the model's uncertainty under different augmentations of the data.

For this experiment, we start from the same underlying model (ER-ACE) and evaluate against two datasets: CIFAR-100 and Food-101N. For what concerns the first, we apply both synthetic noise (20%, 40%, and 60% symmetric noise) and real-world noise obtained by human erroneous annotations (instance-level noise). For the second dataset, the labels are noisy by design, as they are collected from the web. We use a buffer size of 500 samples for all methods. The results in terms of accuracy are summarized in Table 5 and show that our sampling strategy outperforms all other methods across all noise levels and datasets, achieving the highest accuracy. In particular, we find that PuriDivER performs poorly in the presence of more realistic noise, such as the instance-level noise of CIFAR-100 and the web-collected labels of Food-101N. On the other hand, Herding and Rainbow Memory experience a sharp drop in performance as the noise increases, while our method remains robust across all noise levels.

Additionally, to assess the impact of the rehearsal process on different sampling approaches, in Table 6 (right part) we compare three strategies on the human-annotated CIFAR-100N: *Herding, which generates a representative set of samples from the stream data distribution, Puridiver and our Bi-Fold Loss-Aware Sampling, both based on reservoir and aiming for a balance between purity and diversity, but through different sampling scores.* As expected, the impact of a larger buffer size is always beneficial regardless of both the replay method and the sampling strategy involved. Plus, combining Amnesic Replay with our sampling strategy outperforms all other sampling strategies. Notably, in the scenario with a small buffer size (Table 6 with $\mathcal{M} = 500$), Herding appears to gain benefits from the use of Amnesic Replay. However, standard replay does not fully exploit the potential of our sampling strategy.

Buffer Composition. In addition to the performance analysis previously presented, we acknowledge that our primary objective in such a rehearsal-based scenario is to prevent performance degradation through the maintenance of a high-quality buffer. To this end, we present a comprehensive examination of the purity characterizing our buffer throughout the training process (*i.e., across multiple tasks.)*

The Purity is computed as the percentage of clean samples inside the memory buffer at the end of each incremental task. Fig. 4 illustrates the percentage of clean samples in the buffer for each method across different noise levels. For this experiment, we consider only the CIFAR-100 dataset, since we need both the real label and the noisy label to compute the percentage of clean samples. As depicted, our strategy maintains the highest percentage of clean samples in the buffer, which is crucial for effective learning in the presence of noise. Notably, while other sampling strategies tends to drop the percentage of clean samples significantly as noise increases, our method remains robust, showing only a slight decrease in the percentage of clean samples even at high noise levels.

Sample Selection. To evaluate the effectiveness of our proposed sample selection strategy, we conduct a comparative analysis against a well-established baseline from the literature [11,22]. Specifically, we evaluate our α-threshold insertion method (Section 3.3) against a Gaussian Mixture Model (GMM) approach that partitions samples into clean and noisy subsets. We evaluate both methods using two key metrics: Final Average Accuracy (FAA) and buffer Purity, measured as the average percentage of clean samples retained in the buffer at the end of each task. Table 7 compares our α-insertion strategy with the GMM baseline.

To facilitate the interpretation of the table results, recall that α denotes the fraction of highest-loss samples in each batch that are discarded before inserting data into the buffer. When $\alpha = 0$ (i.e., all samples are inserted), the buffer's purity converges to approximately $(1 - \text{noise}\%)$. In this case, no mechanism is applied to mitigate noise, and thus the noise distribution in the buffer closely reflects that of the original dataset. By contrast, our insertion strategy based on the threshold α consistently achieves higher accuracy than the GMM-based selection, even for relatively low thresholds in the range of 25%–50%. and can reach *higher purity* levels than GMM-based sample selection.

On the Influence of Lower Noise Rates and Systematic Mislabeling. When injecting synthetic noise into datasets, our choice of the error rate for each dataset is guided by referencing the closest noise rates found in similar real noisy datasets. (*e.g.,* CIFAR-100N) [41,56]. Therefore, for the majority of our experiments, we maintain a fixed noise rate of 40% on synthetically noised datasets.

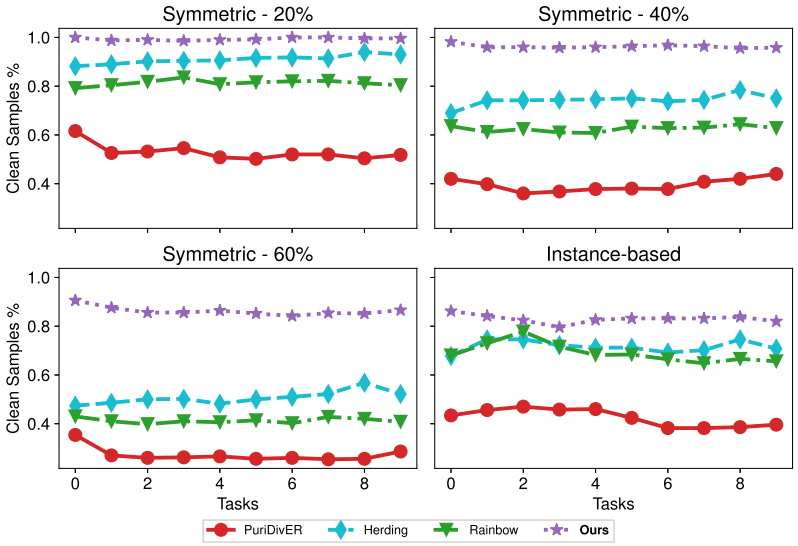

Fig. 4. Purity levels achieved through different sampling strategies. Trend of buffer clean percentages across incremental training tasks on CIFAR100.

Table 7
Ablation study on the sample selection strategy - *insertion phase*.

ER-ACE on CIFAR-100 - $\mathcal{M} = 2000$												
Noise/α	$\alpha = 0\%$		$\alpha = 25\%$		$\alpha = 50\%$		$\alpha = 75\%$		$\alpha = 90\%$		GMM	
	FAA	Purity	FAA	Purity	FAA	Purity	FAA	Purity	FAA	Purity	FAA	Purity
Sym 40%	27.01	0.65	31.99	0.75	36.92	0.86	40.35	0.98	39.40	0.98	32.39	0.73
Sym 60%	15.30	0.44	17.35	0.49	20.74	0.61	26.34	0.87	30.67	0.90	18.61	0.52

Table 8
FAA on CIFAR-100 with different noise rates and in absence of noise ($|\mathcal{M}| = 2000$).

Method		No Noise	Symmetric Noise					Systematic Noise	
CIFAR-100		0%	5%	10%	20%	40%	60%	40%	*avg.*
$\mathcal{M} = 2000$	*i*DivideMix	38.68	39.13	33.80	29.21	20.09	14.2	22.04	28.16
	PuriDivER	33.30	30.74	28.43	22.43	17.46	9.48	17.94	22.54
	ER-ACE	50.06	45.77	42.42	31.14	22.20	11.65	20.88	32.02
	+ **EARL**	49.81	46.73	46.58	44.34	40.35	26.34	30.32	**40.64**

However, to ensure a comprehensive evaluation and demonstrate that the effectiveness of our method extends beyond specific noise scenarios, we compute the Final Average Accuracy for several important CL and CLN baselines from the main manuscript under six additional noise scenarios. Regarding symmetric uniform noise, we evaluate some low-noise scenarios (*i.e., 5%, 10% noise rate) to understand whether there exists a threshold below which EARL loses its effectiveness. Furthermore, we assess whether EARL is detrimental in the absence of noise (0%) and we investigate another type of noise not included in the main table. We call the former Systematic Mislabeleling Noise. This occurs when mislabeling happens with a certain percentage but among semantically similar classes, reflecting realistic error patterns that may arise in practical annotation scenarios where human annotators are more likely to confuse visually or conceptually related categories.* We present the result for such an evaluation in Table 8. We note that, in the absence of noise, EARL leads to only a marginal change in performance, as the model faces no disruptive noise to correct. Plus, such marginal change may be partly due to EARL effectively halving training epochs. However, even with just 5% or 10% label noise, EARL delivers substantial improvements. Unsurprisingly, the performance of each method drops as noise raises. Moreover, we see that the behaviour of the various methods do not vary with changes in noise levels, and our method consistently outperforms the others even in more complex noisy scenarios, *e.g.*, 60% and systematic mislabeling noise.

6. Conclusions

We propose a revised version of our previous work "May the Forgetting be With You", a methodology to deal with the problem of Noisy Label learning in Continual Learning. We start by observing that forgetting does not impact all samples equally and find that alternating epochs of learning and forgetting pushes the noisy-clean loss gap apart for both stream and buffer data. We introduce **Amnesic Replay** to leverage such a phenomenon and ensure separation between clean, complex, and noisy samples. We also propose **Bi-Fold Loss-Aware Sampling**, which enhances the purity of the attained buffer without sacrificing important stored samples.

Our analysis validates our previous work and demonstrates its effectiveness across backbones trained from scratch and pre-trained, under seven datasets with varying similarity to the pre-training and four distinct noise scenarios.

Code Availability Statement. All results discussed in this study are reproducible, with the full code available on the following GitHub repository: https://github.com/monnieka/EARL.

CRediT authorship contribution statement

Monica Millunzi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investiga-

tion, Data curation, Conceptualization; **Lorenzo Bonicelli:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Angelo Porrello:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization; **Jacopo Credi:** Writing – review & editing, Validation, Supervision, Resources, Funding acquisition; **Petter N. Kolm:** Writing – review & editing, Supervision, Formal analysis; **Simone Calderara:** Writing – review & editing, Visualization, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Data availability

The code to reproduce the result of this work can be found in the Code Availability Statement section.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Monica Millunzi reports financial support was provided by AxyonAI SRL. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by MUR/DM352 and Axyon AI SRL .

Appendix A. Overview of Bi-fold loss-aware sampling

We want to delve into detail about the sampling objective proposed in Section 3 where for each $\mathbf{x} \in \mathcal{M}$ we define a *score* function $s(\mathbf{x})$:

$$s(\mathbf{x}) = \begin{cases} \mathcal{L}(\mathbf{x}, \tilde{y}), & \text{if } (\mathbf{x}, \tilde{y}) \sim \mathcal{D}_t \\ -\mathcal{L}(\mathbf{x}, \tilde{y}), & \text{if } (\mathbf{x}, \tilde{y}) \sim \mathcal{D}_{<t} \end{cases} \tag{A.1}$$

When determining which instances to exclude from the buffer during its update, we adopt different sampling strategies for either past or current task samples, using $s(\mathbf{x})$ to prioritize the release of those with **higher score**. In detail, such a process involves two separate phases:

1. **Achieving a balanced buffer.** To ensure a balance between current and past tasks in terms of the number of samples inside the buffer, we compute the ratio of such samples in the buffer, respectively $r_{curr} = \frac{\mathcal{M}_{curr}}{\mathcal{M}}$ and $r_{past} = \frac{\mathcal{M}_{past}}{\mathcal{M}}$, where $r_{curr} + r_{past} = 1$. We can define the quantity $q = r_{curr}$ (thus $1 - q = r_{past}$) and later use this as the probability of replacing a sample from respectively the current task (q) or past tasks ($1 - q$):
   - If the buffer contains a lot of samples from the current task, q will be high, so we are more likely to pick – for replacement – samples from the current task.
   - If the buffer contains few samples from the current task q will be low (and $1 - q$ will be high), so we're more likely to replace samples from the past tasks.

   Formally, this corresponds to sampling from a binomial distribution ϕ with probability q to determine whether to replace a sample from the present or the past, see Eq. (A.3), ensuring a balance between the two groups.
2. **Prioritizing replacement of high-score samples.** During the ongoing optimization process, the buffer might still contain erroneous labels for samples belonging to the current task. Among these samples, we want to *discard* the ones most likely to be noisy (high-loss) – **Case a)** of Section 3.3. On the other hand, based on our buffer insertion policy combined with Amnesic Replay – as mentioned in clause – **Case b)** of Section 3.3 –, and by looking at the results of Section 5 and our previous work [9], we can assume that at the end of each task we are able to clean the buffer for current samples thoroughly. Therefore, on the subsequent tasks, among these clean samples coming from the old completed tasks, we want to *retain* the most complex inside the buffer (high-loss). We thus define the following ***normalized probabilities*** (Eq. (A.2)):

$$\begin{aligned} p_{curr}(\mathbf{x}) &= \frac{s(\mathbf{x})}{z_{curr}} \quad \text{with } z_{curr} = \textstyle\sum_{\mathbf{x} \in \mathcal{M}_{curr}} s(\mathbf{x}) \\ p_{past}(\mathbf{x}) &= \frac{s(\mathbf{x})}{z_{past}} \quad \text{with } z_{past} = \textstyle\sum_{\mathbf{x} \in \mathcal{M}_{past}} s(\mathbf{x}) \end{aligned} \tag{A.2}$$

Overall, when updating the buffer, we sample elements to be replaced from the following distribution:

$$p(\mathbf{x}) = \phi p_{curr}(\mathbf{x}) + (1 - \phi) p_{past}(\mathbf{x}). \tag{A.3}$$

In summary, if phase 1. determines that we need to replace a sample from the current task, the sampling will prioritize replacing items with low-loss, guided by p_{curr}. Conversely, if a sample from a past task needs replacement, it will be sampled with probability p_{past}, thus prioritizing the release of high-loss samples.

Appendix B. Analysis of forgetting timing

In Section 2 we introduced a short overview of forgetting dynamics, from [17]. To quantify the aforementioned forgetting epoch T' and formalize what "rapid forgetting" means across different architectures, we indeed leverage the **second-split forgetting** metric. Following [17]'s protocol, we split the training set into two partitions (S_A, S_B): the model is first trained on S_A, then forgetting statistics are computed during fine-tuning on the second split S_B. An example is said to undergo a forgetting event when the accuracy on that example decreases between two consecutive updates. By collecting forgetting events, we can track the epoch after which an original training example from S_A is no longer classified correctly (forgotten) as the network is fine-tuned on a randomly held-out partition of the dataset S_B. We conduct our experiments on CIFAR-100 (50 epochs per split), following the aforementioned protocol, and extract our statistics of interest. We evaluate two architectures: ResNet-18 (trained from scratch) with a learning rate of 0.03, and ViT-B/16 (pretrained) with a learning rate of 0.01. We introduced two types of noise in the dataset: synthetic uniform noise (40%), and automatic annotation noise, using CLIP zero-shot to assign labels.

From the results in Fig. B.5 we can state that earlier forgetting of mislabeled examples occurs consistently across architectures and noise types. In particular, random noisy samples (red bars) are forgotten very rapidly (~16th epoch), while a more complex source of noise makes the forgetting time T' higher (~39th epoch). For completeness, we also report over each correspondent bar the loss values for both clean and noisy examples at the epoch in which noisy examples undergo forgetting.

These results show two key insights: first, the loss values provide a clear signal to distinguish between clean and noisy samples, which forms the basis of our sample selection mechanism. Second, the variable timing of natural forgetting justifies our approach of actively inducing forgetting events.

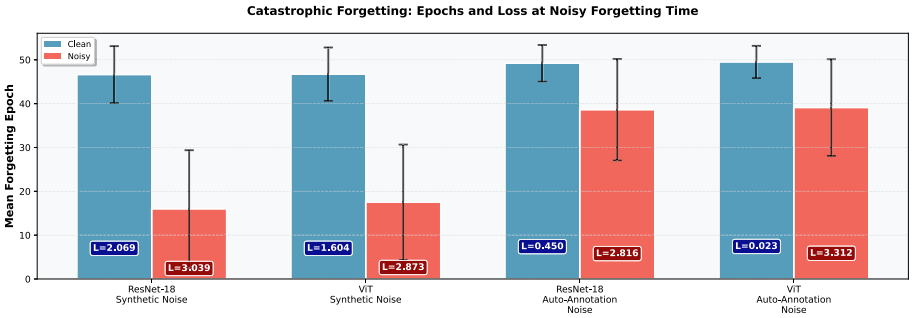

Fig. B.5. Forgetting time comparison across different backbones and noise types.

Appendix C. On the applicability of current CNL methods in offline ClassIL

Table C.9
Final Average Accuracy (FAA) [↑] (and runtime difference w.r.t. EARL [↓]) of current CNL methods. **: Training iterations spread across epochs;†: Impractical runtime; ⋄: Buffer size constraints.

5 task CIFAR-10 – 40% *symm* noise – 50 epochs per task				
Memory budget	500	1000	2500	*unlimited*
EARL	**58.68** (×1)	**62.04** (×1)	**67.10** (×1)	**76.83** (×1)
SPR** [10]	⋄	25.97 (×25.0)	†	†
CNLL [12]	⋄	⋄	35.46 (×4.6)	43.43 (×2.3)

In Section 1 we have shown the advantage of the offline CL scenario, which allows us to fit even complex distributions and extends the scope of CNL to a broader community. To better depict the improvements brought by EARL, here we compare against the current CNL literature. Since these methods were originally designed for a single epoch scenario, we include some refinements to better adapt them for training on multi-epochs. In particular, considering SPR, we distribute its 7000 training iterations per batch across the task (otherwise it would take ×448 more iterations than standard training). For experiments in the *unlimited* buffer scenario of CNLL, we freeze memory after the first epoch to prevent it from growing indefinitely.

As shown in Table C.9, EARL does not require buffer fitting, as it can take advantage of the multiple training epochs. While the performance gain is remarkable, it may be expected since all methods except EARL are not designed for offline CL. EARL's runtime remains mostly unchanged going from 500 to 1000 buffer size (around 1 h and a half to complete). Our proposal shows significant improvements in terms of efficiency, remaining viable in all scenarios.

Appendix D. Experiments

To evaluate our proposal we build on Mammoth (`PyTorch`). All methods share epochs and batch size, chosen via Multitask performance. Competitors tune the learning rate; EARL fixes it at 0.03.

Augmentation We apply random crops and horizontal flips to all datasets; PuriDivER uses AutoAugment [57]as in the original paper.

Training We fix batch size (32 for image datasets, 16 for GLUE). Image tasks: 50 epochs on CIFAR-100/ANIMAL-10N, 20 on Food-101N, 30 on ISIC, 5 on EuroSAT, using SGD with constant *lr*. GLUE: MNLI/SNLI/QNLI for 3 epochs, RTE/SICK/SciTail for 20, with AdamW ($3 * 10^{-4} lr$), 0.01 wt decay.

Datasets

We empirically validate our method on seven different classification benchmarks. In each experiment, samples from the main dataset are split into disjoint sets based on their class and organized into tasks, following the ClassIL setup.

CIFAR-100 [40]. This dataset has $60,000$ 32×32 colour images, in particular it has 100 classes with 600 images each. We organize classes in 10 tasks, each containing 5 classes from the same superclass. We corrupt the dataset at hand to obtain a uniform noise configuration, by randomly flipping 40% of the labels.

CIFAR-100N [41]. This is a variant of CIFAR-100 equipped with human-annotated real-world noisy labels collected from Amazon Mechanical Turk. The authors demonstrate that real-world noisy labels follow an instance-dependent pattern of noise and estimate the noise rate of the fine labels to be approximately 40.21%.

CIFAR-100C We re-annotate the CIFAR-100 training set using CLIP - with ViT-B/32 as the image encoder. The noise rate in this experiment matches the error rate of CLIP (35.31%) since the corrupted dataset labels corresponds to the zero-shot prediction of CLIP. Notice that in such a scenario automatic annotations are available, but the source of annotation is not accessible for training purposes.

Food-101N [42]. The dataset is intended for learning to address label noise with minimal human supervision. It contains about $310,009$ images of food dishes organized into 101 noisy classes, which we split into 5 tasks. The estimated noise rate is 20%.

ANIMAL-10N [8]. This dataset contains 64×64 RGB images of 5 pairs of animals which look very similar: (cat, lynx), (jaguar, cheetah), (wolf, coyote), (chimpanzee, orangutan), and (hamster, Guinea pig). The images are crawled from online search engines and then classified by 15 recruited participants. The final dataset has $50,000$ training images and $5,000$ for the test set. We split the dataset in 5 tasks. The noise rate of the dataset is approximately 8%.

ISIC [43,44]. We use the 2018 ISIC Challenge dataset, namely the one created for the *disease classification task*. From the original dataset we remove the most frequent class *melanocytic nevus*, and keep the other 6 classes. The obtained dataset consists of about 3310 instances. Then, we split the dataset in 3 tasks composed of 2 classes each.

EuroSAT [45,46]. The dataset is composed of Sentinel-2 satellite pictures to address the difficulty of classifying land use and land cover. It consists of 10 classes, for a total of $27,000$ labeled and geo-referenced images. We use the RGB version and split the 10 classes into 5 incremental learning tasks.

GLUE [49]. Consists of 6 incremental sentence or sentence-pair classification tasks. Tasks have 2–3 classes: MNLI/SNLI/SICK (entailment, neutral, contradiction);
QNLI/RTE/SciTail (entailment, not-entailment/neutral). We inject synthetic noise.

Appendix E. About Gaussian mixture models as sample selectors

A common technique for handling noisy labels [11,22] uses a Gaussian Mixture Model (GMM) to separate clean and noisy data. In Section 5, we adopt this in place of our sample selection strategy, to evaluate the buffer's Accuracy and Purity (Table 7). Using the Expectation-Maximization algorithm, we fit a GMM to the training loss of all examples to estimate the probability $p_G(\cdot)$ of an example belonging to a category. For a noisy example $(x_i, \tilde{y}_i)$, the label purity is given by the posterior probability of GMMs: $p_G\big(g|\ell(x_i, \tilde{y}_i; \theta)\big)$, where g represents the Gaussian component for clean samples. We thus obtain the clean set $\mathcal{C}$ and noisy set $\mathcal{N}$ as: $\mathcal{C} := \big\{(x_i, \tilde{y}_i) \in \mathcal{M} : p_G\big(g|\ell(x_i, \tilde{y}_i; \theta)\big) \geq \lambda\big\}$, and $\mathcal{N} := \big\{(x_i, \tilde{y}_i) \in \mathcal{M} : p_G\big(g|\ell(x_i, \tilde{y}_i; \theta)\big) < \lambda\big\}$. A sample is deemed clean if its posterior probability exceeds a threshold λ.

References

[1] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: the sequential learning problem, Psychology of learning and motivation, Elsevier, 24, 1989, pp. 109–165.
[2] G.M. van de Ven, T. Tuytelaars, A.S. Tolias, Three types of incremental learning, Nat. Mach. Intell. 4 (12) (2022) 1185–1197.
[3] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P.K. Dokania, P.H.S. Torr, M. Ranzato, On tiny episodic memories in continual learning, in: International Conference on Machine Learning Workshop, 2019.
[4] R. Ratcliff, Connectionist models of recognition memory: constraints imposed by learning and forgetting functions, Psychol. Rev. 97 (2) (1990) 285.
[5] A. Robins, Catastrophic forgetting, rehearsal and pseudorehearsal, Conn. Sci. 7 (2) (1995) 123–146.
[6] T. Xiao, T. Xia, Y. Yang, C. Huang, X. Wang, Learning from massive noisy labeled data for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.
[7] L. Bossard, M. Guillaumin, L. Van Gool, Food-101–mining discriminative components with random forests, in: Proceedings of the European Conference on Computer Vision, 2014.
[8] H. Song, M. Kim, J.-G. Lee, SELFIE: Refurbishing unclean samples for robust deep learning, in: International Conference on Machine Learning, 2019.
[9] M. Millunzi, L. Bonicelli, A. Porrello, J. Credi, P.N. Kolm, S. Calderara, May the forgetting be with you: alternate replay for learning with noisy labels, in: British Machine Vision Conference, 2024.
[10] C.D. Kim, J. Jeong, S. Moon, G. Kim, Continual learning on noisy data streams via self-purified replay, in: IEEE International Conference on Computer Vision, 2021.
[11] J. Bang, H. Koh, S. Park, H. Song, J.-W. Ha, J. Choi, Online continual learning on a contaminated data stream with blurry task boundaries, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022.
[12] N. Karim, U. Khalid, A. Esmaeili, N. Rahnavard, CNLL: a semi-supervised approach for continual noisy label learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022.
[13] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M.S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al., A closer look at memorization in deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 233–242.
[14] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, M. Sugiyama, Co-teaching: robust training of deep neural networks with extremely noisy labels, in: Advances in Neural Information Processing Systems, 2018.
[15] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, L. Fei-Fei, MentorNet: learning data-driven curriculum for very deep neural networks on corrupted labels, in: International Conference on Machine Learning, 2018.
[16] E. Arazo, D. Ortego, P. Albert, N. O'Connor, K. McGuinness, Unsupervised label noise modeling and loss correction, in: International Conference on Machine Learning, 2019.
[17] P. Maini, S. Garg, Z. Lipton, J.Z. Kolter, Characterizing datapoints via second-split forgetting, Adv. Neural Inf. Process. Syst. 35 (2022) 30044–30057.
[18] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, in: Advances in Neural Information Processing Systems, 2020.
[19] M. Toneva, A. Sordoni, R.T.d. Combes, A. Trischler, Y. Bengio, G.J. Gordon, An empirical study of example forgetting during deep neural network learning, in: International Conference on Learning Representations Workshop, 2019.
[20] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, 2021.
[21] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, L. Qu, Making deep neural networks robust to label noise: a loss correction approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[22] J. Li, R. Socher, S.C.H. Hoi, DivideMix: learning with noisy labels as semi-supervised learning, in: International Conference on Learning Representations Workshop, 2020.
[23] B. Han, G. Niu, X. Yu, Q. Yao, M. Xu, I. Tsang, M. Sugiyama, SIGUA: forgetting may make learning with noisy labels more robust, in: International Conference on Machine Learning, 2020.
[24] N. Natarajan, I.S. Dhillon, P.K. Ravikumar, A. Tewari, Learning with noisy labels, Adv. Neural Inf. Process. Syst. 26 (2013) .
[25] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, J. Choi, Rainbow memory: continual learning with a memory of diverse samples, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021.
[26] J.S. Vitter, Random sampling with a reservoir, ACM Trans. Math. Software 11 (1) (1985) 37–57.
[27] P. Buzzega, M. Boschini, A. Porrello, S. Calderara, Rethinking experience replay: a bag of tricks for continual learning, in: International Conference on Pattern Recognition, 2020.
[28] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, iCaRL: incremental classifier and representation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[29] M. Welling, Herding dynamical weights to learn, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 1121–1128.
[30] M. Boschini, L. Bonicelli, P. Buzzega, A. Porrello, S. Calderara, Class-incremental continual learning into the extended DER-verse, IEEE Trans. Pattern Anal. Mach. Intell. 45 (5) (2022) 5497–5512.
[31] L. Caccia, R. Aljundi, N. Asadi, T. Tuytelaars, J. Pineau, E. Belilovsky, New insights on reducing abrupt representation change in online continual learning, in: International Conference on Learning Representations Workshop, 2022.
[32] Y. Li, Z. Guo, L. Wang, CLTR: continual learning time-varying regularization for robust classification of noisy label images, Pattern Recognit. 171 (2026) 112137.
[33] H. Cha, J. Lee, J. Shin, Co2l: contrastive continual learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9516–9525.
[34] K. Cheng, Y. Ma, G. Wang, L. Zong, X. Liu, NLOCL: noise-labeled online continual learning, Electronics 13 (13) (2024) 2560.
[35] R. Tong, Y. Liu, J.Q. Shi, D. Gong, Coreset selection via reducible loss in continual learning, in: The Thirteenth International Conference on Learning Representations, 2025.
[36] X. Yang, G. Lai, D. Meng, X. Cao, X. Yang, RACE: robust adaptive and clustering elimination for noisy labels in continual learning, Knowl. Based Syst. 324 (2025) 113783 113783.
[37] D. Soudry, E. Hoffer, M.S. Nacson, S. Gunasekar, N. Srebro, The implicit bias of gradient descent on separable data, J. Mach. Learn. Res. 19 (70) (2018) 1–57.
[38] H. Wei, L. Feng, X. Chen, B. An, Combating noisy labels by agreement: a joint training method with Co-regularization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020.
[39] L. Bonicelli, M. Boschini, A. Porrello, C. Spampinato, S. Calderara, On the effectiveness of lipschitz-driven rehearsal in continual learning, in: Advances in Neural Information Processing Systems, 2022.
[40] A. Krizhevsky, et al., Learning Multiple Layers of Features from Tiny Images, Technical Report, Citeseer, 2009.
[41] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, Y. Liu, Learning with noisy labels revisited: a study using real-world human annotations, in: International Conference on Learning Representations Workshop, 2022.
[42] K.-H. Lee, X. He, L. Zhang, L. Yang, CleanNet: transfer learning for scalable image classifier training with label noise, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5447–5456.
[43] N. Codella, V. Rotemberg, P. Tschandl, M.E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, et al., Skin lesion analysis toward melanoma detection 2018: a challenge hosted by the international skin imaging collaboration (isic), arXiv preprint arXiv:1902.03368 (2019).
[44] P. Tschandl, C. Rosendahl, H. Kittler, The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, Sci. Data 5 (1) (2018) 1–9.
[45] P. Helber, B. Bischke, A. Dengel, D. Borth, Introducing euroSAT: a novel dataset and deep learning benchmark for land use and land cover classification, in: IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2018, pp. 204–207.
[46] P. Helber, B. Bischke, A. Dengel, D. Borth, EuroSAT: a novel dataset and deep learning benchmark for land use and land cover classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12 (7) (2019) 2217–2226.
[47] J. Oh, S. Kim, N. Ho, J.-H. Kim, H. Song, S.-Y. Yun, Understanding cross-domain few-shot learning based on domain similarity and few-shot difficulty, Adv. Neural Inf. Process. Syst. 35 (2022) 2622–2636.
[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[49] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S.R. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019, OpenReview.net, 2019.
[50] 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, in: ICLR, 2021.
[51] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[52] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, T. Pfister, Learning to prompt for continual learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 139–149.
[53] J.S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, Z. Kira, CODA-prompt: continual decomposed attention-based prompting for rehearsal-free continual learning, in: Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition, 2023, pp. 11909–11919.
[54] G. Zhang, L. Wang, G. Kang, L. Chen, Y. Wei, SLCA: slow learner with classifier alignment for continual learning on a pre-trained model, in: IEEE International Conference on Computer Vision, 2023, pp. 19148–19158.
[55] C.-E. Wu, Y. Tian, H. Yu, H. Wang, P. Morgado, Y.H. Hu, L. Yang, Why is prompt tuning for vision-language models robust to noisy labels?, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2023.
[56] W. Li, L. Wang, W. Li, E. Agustsson, J. Berent, A. Gupta, R. Sukthankar, L.V. Gool, WebVision challenge: visual learning and understanding with web data, arXiv preprint arXiv:1705.05640 (2017).
[57] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q.V. Le, AutoAugment: learning augmentation strategies from data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 113–123.