Towards Macro-AUC Oriented Imbalanced Multi-Label Continual Learning

Yan Zhang, Guoqiang Wu*, Bingzheng Wang, Teng Pang, Haoliang Sun, Yilong Yin*

School of Software, Shandong University
yannzhang9@gmail.com, guoqiangwu90@gmail.com, binzhwang@gmail.com, silencept7@gmail.com,
haolsun@sdu.edu.cn, ylyin@sdu.edu.cn

Abstract

In Continual Learning (CL), while existing work primarily focuses on the multi-class classification task, there has been limited research on Multi-Label Learning (MLL). In practice, MLL datasets are often class-imbalanced, making it inherently challenging, a problem that is even more acute in CL. Due to its sensitivity to imbalance, Macro-AUC is an appropriate and widely used measure in MLL. However, there is no research to optimize Macro-AUC in MLCL specifically. To fill this gap, in this paper, we propose a new memory replay-based method to tackle the imbalance issue for Macro-AUC-oriented MLCL. Specifically, inspired by recent theory work, we propose a new Reweighted Label-Distribution-Aware Margin (RLDAM) loss. Furthermore, to be compatible with the RLDAM loss, a new memory-updating strategy named Weight Retain Updating (WRU) is proposed to maintain the numbers of positive and negative instances of the original dataset in memory. Theoretically, we provide superior generalization analyses of the RLDAM-based algorithm in terms of Macro-AUC, separately in batch MLL and MLCL settings. This is the first work to offer theoretical generalization analyses in MLCL to our knowledge. Finally, a series of experimental results illustrate the effectiveness of our method over several baselines.

Code —
https://github.com/ML-Group-SDU/Macro-AUC-CL

Introduction

Traditional machine learning methods assume an independent and identically distributed (i.i.d.) data pattern. However, in practice, humans continually learn new knowledge and retain previous knowledge. This process, known as Continual Learning (CL) (Ring 1994), aims to adapt to new tasks while mitigating Catastrophic Forgetting (French 1999).

Recently, CL has gained considerable attention (Wang et al. 2023), where the replay-based approaches (Chaudhry et al. 2019; Rebuffi et al. 2017) often show promising performance (Knoblauch, Husain, and Diethe 2020). Most of these works concentrate on multi-class classification, where an instance is associated with a single label. In some real-world scenarios, however, an instance is often associated with multiple labels simultaneously. This raises the study of Multi-Label Learning (MLL) (McCallum 1999). In this paper, our focus lies on *Multi-Label Continual Learning (MLCL)* (Kim, Jeong, and Kim 2020), an inherently more challenging learning task that has not yet been extensively explored. MLCL is increasingly relevant in real-world applications. This setting is common in areas such as healthcare, finance, and autonomous systems, where the ability to handle multiple labels as new labels emerge is crucial (Kassim 2024; Dalle Pezze et al. 2023; Ceccon et al. 2024).

In MLCL (or MLL), there is a natural and inherent challenge - class imbalance, which is a special characteristic of MLL, unlike the imbalance case of multi-class problem. The imbalance issue in MLCL we considered is for each MLL task in the continual setting, which mainly includes two aspects: imbalance within labels, and imbalance between labels. Specifically, imbalance within labels is that for each label, the numbers of negative instances are often higher than the positive instances[1]. In contrast, imbalance between labels is that the number of positive instances of each label is not balanced. For the performance evaluation on imbalanced cases, various measures are developed, e.g., F1 score and mAP, etc. Among them, Macro-AUC (Zhang and Zhou 2013) serves as a suitable and widely-used measure in practice, which is considered in this paper.

Optimizing Macro-AUC in MLL directly can lead to NP-hard problems (Tarekegn, Giacobini, and Michalak 2021), as it is discontinuous and non-convex. So it is common to design a surrogate loss to solve this issue. Meanwhile, it is crucial to address the imbalance problem to optimize Macro-AUC. However, commonly used losses such as cross-entropy loss are ineffective in handling imbalance, leading to relatively low Macro-AUC performance (see Table 1 in Section). To tackle imbalance and maximize Macro-AUC for *multi-label batch learning*, recent theory work (Wu, Li, and Yin 2023) proposes a reweighted univariate (RU) loss with superior formal learning guarantees. Both theory and experiments show that RU loss enjoys the high performance of pairwise loss, without its high computational complexity, and the computational efficiency of univariate losses (e.g., cross-entropy), which suffer from lower

*Corresponding Author

[1]Positive and negative instances refer to samples that either contain or lack the corresponding label.

performance. However, this loss assigns the same margin to each class, which may require further analysis to enhance its performance. In contrast, in *multi-class batch learning*, a label-distribution-aware margin (LDAM) loss, proposed by Cao et al. (2019), assigns a larger margin to the class with a smaller size, resulting in different margins for distinct classes. It has shown promising performance in practice.

Inspired by the above studies, we propose a novel replay-based method that aims to maximize Macro-AUC in MLCL setting. Our method primarily consists of a new loss function and a new memory-updating strategy. Specifically, we propose a new reweighted label-distribution-aware margin (RLDAM) loss to tackle the imbalance, which can inherit the benefits of the RU and LDAM losses. Further, to ensure better compatibility with the RLDAM loss, we propose a new memory-updating approach named Weight Retain Updating (WRU), which is simple but effective. Intuitively, it ensures that the data stored in memory for each class maintains consistency with the original dataset in terms of the number of positive and negative instances (a.k.a. reweighting factors). Theoretically, we provide superior generalization analyses of the RLDAM-based algorithm for Macro-AUC in batch MLL setting, and then extend it to MLCL setting.

Finally, to illustrate the effectiveness of our proposed method, we conduct a series of experiments. Comparisons with other baselines demonstrate the superiority of our approach. Moreover, we have performed ablation studies and experiments to investigate other influencing factors, consistently showing that our method performs well.

Our contribution can be summarized as: (1) To maximize Macro-AUC in MLCL, we propose a novel memory replay-based method, involving a new theory-principled RLDAM loss and a new compatible memory updating strategy called WRU. (2) Theoretically, we analyze the generalization superiority of the RLDAM-based algorithm w.r.t. Macro-AUC in batch MLL and MLCL. To our knowledge, ours is the first theoretical analysis of MLCL. (3) Experimentally, extensive results demonstrate the superiority of our method over several baselines in MLCL.

Preliminaries

Notations. Let a regular, boldfaced lower and upper letter denote a scaler (e.g., a), vector (e.g., $\mathbf{a}$), and matrix (e.g., $\mathbf{A}$), respectively. a_i denotes its i-th element. For a matrix $\mathbf{A}$, denote its i-th row, j-th column, and (i, j)-th element as $\mathbf{a}_i, \mathbf{a}^j$, and a_{ij}, respectively. Let a blackboard bold letter denote a set (e.g., $\mathbb{A}$). $|\mathbb{A}|$ denotes its cardinality. $[n]$ denotes the set $\{1, ..., n\}$. $[\![\cdot]\!]$ denotes the indicator function, i.e., it returns 1 if the condition holds and 0 otherwise.

Problem Setup

Multi-Label Learning (MLL). Given a MLL training dataset $\mathbb{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ i.i.d. drawn from P with a sample size n, where P is a distribution over $\mathcal{X} \times \{0, 1\}^K$ and K is the size of label set $\mathbb{K}$. $\mathbf{x}_i$ and $\mathbf{y}_i$ denote the input and corresponding output. Note that $\mathbf{y}_i \in \{0, 1\}^K$ is a multi-hot label vector. For any $k \in \mathbb{K}$, $y_{ik} = 1$ (or 0) denotes that the k-th label is relevant (or irrelevant). The objective of MLL is to learn a hypothesis or predictor $f = (f_1, ..., f_K) : \mathcal{X} \to \mathbb{R}^K$ from a hypothesis space $\mathcal{F} := \{f\}$. In batch MLL, the learner knows the label size K in advance.

Multi-Label Continual Learning (MLCL). In MLCL, the learner continuously encounters a sequence of tasks $\{\mathcal{T}^1, ..., \mathcal{T}^T\}$, where T denotes the length of the task sequence, and each task $\mathcal{T}^t$ ($t \in [T]$) is an MLL task.

Notably, when a task identifier (e.g., t) appears in the superscript, it does not mean the power of t but denotes it is for the t-th task, and this definition holds for all symbols. Moreover, we place $+$ or $-$ in the superscript position to indicate that the symbol is about the positive or negative instances of a particular label (e.g. n_k^{t+} is the sample size of the positive instances for the class k in the t-th task).

Specifically, at each time step t, the learner encounters a new task $\mathcal{T}^t$ with a dataset $\mathbb{D}^t = \{(\mathbf{x}_i^t, \mathbf{y}_i^t)\}_{i=1}^{n^t}$, which is i.i.d. drawn from a MLL distribution P^t over $\mathcal{X} \times \{0, 1\}^{K^t}$, where K^t is the label size for the current task $\mathcal{T}^t$. The goal of MLCL is to learn a hypothesis $f^t = (f_1^t, ..., f_{\widetilde{K}^t}^t) : \mathcal{X} \to \mathbb{R}^{\widetilde{K}^t}$ over all encountered tasks from a hypothesis space $\mathcal{F}^t := \{f^t\}$, where $\widetilde{K}^t = \sum_{i=1}^t K^i$. Notably, the learning objective reflects that the hypothesis applies to both the new task and all previously encountered tasks.

Multi-Label Class Incremental Learning (MLCIL). Among MLCL, here we focus on one practical and challenging case - MLCIL where the prediction layer of the learner is a *single-head* setup, and the task ID is not accessible during the learning process. Similarly to MLL, MLCIL assumes that the continual learner knows the total number of classes $K = \sum_{t \in [T]} K^t$ in advance. However, unlike batch MLL, the learner of MLCIL learns only task-known classes in each task and does not consider task-agnostic classes. Besides the above description, we also make the following assumptions.

Assumption 1 (For MLCIL). *(1) Each task has distinct classes:* $\forall t \neq t', \mathbb{K}^t \cap \mathbb{K}^{t'} = \emptyset$.
(2) Within each task, there exists at least one sample associated with more than one label (multi-label setup) [2]*:* $\forall t \in [T], \exists (\mathbf{x}_i^t, \mathbf{y}_i^t) \in \mathbb{D}^t, \sum_j y_{ij}^t > 1$.
(3) Samples may appear repeatedly across different tasks (same as Dong et al. (2023)): $\exists t \neq t'$, $\mathbb{X}^t \cap \mathbb{X}^{t'} \neq \emptyset$.

Assumption (2) and (3) reflects the differences between MLCIL and multi-class CL. In multi-class CL settings, an instance exhibits only one label and task-disjoint assumption often holds, where both are opposite to MLCIL.

Replay-Based CL. We consider the replay-based CL framework, specifically rehearsal, because it is simple and effective. It maintains a memory buffer $\mathcal{M}$ for selected samples of all previous tasks. The data of each task $\mathcal{T}^t$ saved in memory is denoted as $\mathbb{M}^t$, which is a sampled subset from $\mathbb{D}^t$.

Evaluation Measure

Various measures are often used in MLCIL for a comprehensive evaluation, e.g., F1 score, and mAP. Concerning the

[2]This assumption is reasonable as most practical MLL problems hold. While it is theoretically possible to sample datasets where the number of labels per sample is less than two in a probabilistic fashion, this case is extremely rare in practice.

(label-wise) class imbalance, Macro-AUC (Zhang and Zhou 2013) is an appropriate and widely used measure in practice but has not yet been considered in MLCIL, which we focus on. Macro-AUC macro-averages the AUC measure across all class labels. Given a dataset $\mathbb{D}$ and a predictor $f \in \mathcal{F}$, Macro-AUC is defined as follows:

$$\frac{1}{K}\sum_{k=1}^{K}\frac{1}{\left|\mathbb{D}_k^+\right|\left|\mathbb{D}_k^-\right|}\sum_{(p,q)\in\mathbb{D}_k^+\times\mathbb{D}_k^-}\llbracket f_k\left(\mathbf{x}_p\right)>f_k\left(\mathbf{x}_q\right)\rrbracket,$$

where for each class k, $\mathbb{D}_k^+$ and $\mathbb{D}_k^-$ denote the index subset of positive and negative instances, respectively, and $f_k(\mathbf{x})$ is the logit output of input $\mathbf{x}$. The objective for maximizing Macro-AUC is: $\frac{1}{K}\sum_{k=1}^{K}\frac{1}{|\mathbb{D}_k^+||\mathbb{D}_k^-|}\sum_{(p,q)\in\mathbb{D}_k^+\times\mathbb{D}_k^-}\mathcal{L}_{0/1}(\mathbf{x}_p,\mathbf{x}_q,f_k)$, where the $0/1$ loss is $\mathcal{L}_{0/1}(\mathbf{x}^+,\mathbf{x}^-,f_k)=\llbracket f_k(\mathbf{x}^+)\leq f_k(\mathbf{x}^-)\rrbracket$. The associated expected risk w.r.t. the $0/1$ loss is

$$R^{0/1}(f)=\frac{1}{K}\sum_{k=1}^{K}\underset{\mathbf{x}_p\sim P_k^+,\mathbf{x}_q\sim P_k^-}{\mathbb{E}}\left[\mathcal{L}_{0/1}(\mathbf{x}_p,\mathbf{x}_q,f_k)\right], \quad (1)$$

where $P_k^+=P(\mathbf{x}|y_k=1), P_k^-=P(\mathbf{x}|y_k=0)$. When measuring the test performance, we use an overall Macro-AUC, i.e., the mean of Macro-AUC on all tasks. Besides, we use Forgetting (Chaudhry et al. 2018) to evaluate the memory ability of the continual learning technique.

Method

In this section, we propose our method to maximize Macro-AUC for MLCIL. Firstly, we propose a new reweighted label-distribution-aware margin (RLDAM) loss to tackle the multi-label imbalance by combining the reweighted loss and LDAM loss. Regarding the loss, a new memory updating strategy named Weight Retain Updating (WRU) is proposed for replay-based MLCIL.

Reweighted Label-Distribution-Aware Margin Loss

For a clear presentation, we first propose a new RLDAM loss in the batch MLL scenario and then discuss it in the continual learning scenario.

Batch Learning Scenario In the batch learning scenario, to maximize Macro-AUC, recent theory work (Wu, Li, and Yin 2023) proposed a Reweighted Univariate (RU) loss with good properties as $\mathcal{L}_{\mathtt{RU}}(\mathbf{x}^+,\mathbf{x}^-,f_k)=\ell(f_k(\mathbf{x}^+))+\ell(-f_k(\mathbf{x}^-))$, where $\mathbf{x}^+$ (or $\mathbf{x}^-$) denotes a positive (or negative) instance for the label k, and the base loss $\ell(\cdot)$ can be any *margin*-based loss (e.g., hinge loss or logistic loss), for binary classification. The empirical risk w.r.t. $\mathcal{L}_{\mathtt{RU}}$ is $\widehat{R}_{\mathbb{D}}^{\mathtt{RU}}(f)=\frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{n}\left(\frac{[[y_{ik}=1]]}{|\mathbb{D}_k^+|}\ell(f_k(\mathbf{x}_i))+\frac{[[y_{ik}\neq1]]}{|\mathbb{D}_k^-|}\ell(-f_k(\mathbf{x}_i))\right)$ where $1/|\mathbb{D}_k^+|$ and $1/|\mathbb{D}_k^-|$ are reweighted factors. Intriguingly, the theory work (Wu, Li, and Yin 2023) shows that the learning algorithm with the RU loss offers a better learning guarantee than the one with the original univariate loss. However, for the base loss of RU considered in Wu, Li, and Yin (2023), the margin for each class is the same, which might need further analysis for improvement.

On the other hand, to tackle the imbalance issue, a label-distribution-aware margin (LDAM) loss is proposed by Cao et al. (2019), where their theory suggests that it can improve generalization by assigning different margins for distinct classes. In that work, for a loss function in binary classification, such as hinge loss $\mathcal{L}_{\mathtt{HG}}(\mathbf{x},\mathbf{y},f)=\max(0,\Delta-yf(\mathbf{x}))$, the optimal trade-off of two margins is derived as $\Delta_1=\frac{\lambda}{n_1^{1/4}}$ and $\Delta_2=\frac{\lambda}{n_2^{1/4}}$ with a constant λ for distinct classes, where n_1 and n_2 are the number of samples for two classes. Then it extends the margins to multi-class hinge loss as:

$$\mathcal{L}_{\mathtt{LDAM-HG}}(\mathbf{x},\mathbf{y},f)=\max(\max_{j\neq k}\{f_j\}-f_k+\Delta_k,0), \quad (2)$$

where $\forall k\in\mathbb{K}$, $\Delta_k=\frac{\lambda}{n_k^{1/4}}$ and n_k denotes the number of instances for class k. In the following of our work, we take inspiration from the optimal trade-off between two classes and its multi-class extension.

Our new RLDAM loss. To combine the best of the above two losses, we integrate LDAM loss into RU loss to further improve generalization for the first time. Applying LDAM to the base loss of RU loss, we obtain a new Reweighted Label-Distribution-Aware Margin Loss (RLDAM) as

$$\begin{aligned}\mathcal{L}_{\mathtt{RM}}\left(\mathbf{x}^+,\mathbf{x}^-,f_k\right)=&\ell\left(f_k\left(\mathbf{x}^+\right)-\Delta_k^+\right)+\\&\ell\left(-f_k\left(\mathbf{x}^-\right)-\Delta_k^-\right),\end{aligned} \quad (3)$$

where Δ_k^+ and Δ_k^- are the margin of positive and negative instances of class k. Note that different classes exhibit different margins. [3]

Next, we will analyze the RLDAM loss in the continual learning setting. Before moving to the continual learning scenario, we first consider the scenario where multiple tasks are learned jointly in the batch learning mode. And the empirical risk of each class k in a task $\mathcal{T}^t$ is

$$\begin{aligned}\widehat{R}_{\mathbb{D}_k^t}^{\mathtt{RM}}(f)=&\frac{1}{\left|\mathbb{D}_k^{t+}\right|\left|\mathbb{D}_k^{t-}\right|}\sum_{(p,q)\in\mathbb{D}_k^{t+}\times\mathbb{D}_k^{t-}}\mathcal{L}_{\mathtt{RM}}\left(\mathbf{x}_p,\mathbf{x}_q,f_k\right)\\=&\sum_{i=1}^{n^t}\Bigg([[y_{ik}=1]]\frac{1}{|\mathbb{D}_k^{t+}|}\ell\left(f_k\left(\mathbf{x}_i\right)-\Delta_k^{t+}\right)+\\&[[y_{ik}\neq1]]\frac{1}{|\mathbb{D}_k^{t-}|}\ell\left(-f_k(\mathbf{x}_i)-\Delta_k^{t-}\right)\Bigg),\end{aligned} \quad (4)$$

where $\mathbb{D}_k^t$ is the subset of the dataset for class k in the t-th task $\mathcal{T}^t$. Then, the empirical risk of each task $\mathcal{T}^t$ and overall risk for T tasks are:

$$\widehat{R}_{\mathbb{D}^t}^{\mathtt{RM}}(f)=\frac{1}{|\mathbb{K}^t|}\sum_{k\in\mathbb{K}^t}\widehat{R}_{\mathbb{D}_k^t}^{\mathtt{RM}}(f), \quad (5)$$

$$\widehat{R}_{\mathcal{T}}^{\mathtt{RM}}(f)=\frac{1}{T}\left(\widehat{R}_{\mathbb{D}^T}^{\mathtt{RM}}(f)+\sum_{i=1}^{T-1}\widehat{R}_{\mathbb{D}^i}^{\mathtt{RM}}(f)\right). \quad (6)$$

[3] We will derive the optimal value of Δ_k^+ and Δ_k^- and prove that minimizing the RLADM loss can have a better learning guarantee w.r.t. Macro-AUC than that of the RU loss in Section .

Continual Learning Scenario For a general CL scenario, at each time step t, the empirical risk $\widehat{R}^{\mathtt{RM}}_{\mathcal{T}^t}$ originally is equal to $\widehat{R}^{\mathtt{RM}}_{\mathbb{D}^t}(f)$ in Eq. (6) as only the data in $\mathcal{T}^t$ can be accessed. Here we focus on the memory replay-based framework, which explicitly stores a subset $\mathbb{M}^i$ of $\mathbb{D}^i$ for each previous task $\mathcal{T}^i$ $(i \in [t-1])$. The (adjusted) empirical risk can be written as $\widehat{R}^{\mathtt{RM}}_{\mathcal{T}^t}(f) = \frac{1}{t}\left(\widehat{R}^{\mathtt{RM}}_{\mathbb{D}^t}(f) + \sum_{i=1}^{t-1} \widehat{R}^{\mathtt{RM}}_{\mathbb{M}^i}(f)\right)$.

Note that we should make the risk on subset $\mathbb{M}^i$ serving as an (approximate) unbiased estimator of the risk on whole dataset $\mathbb{D}^i$, i.e., $\mathop{\mathbb{E}}_{\mathbb{M}^i}\left[\widehat{R}^{\mathtt{RM}}_{\mathbb{M}^i}(f)\right] \approx \widehat{R}^{\mathtt{RM}}_{\mathbb{D}^i}(f)$. Since past $\mathbb{D}^i$ is inaccessible in CL, we instead minimize the empirical risk on $\mathbb{M}^i$, which can also reduce the risk on $\mathbb{D}^i$ with high probability. We will discuss it in detail in the subsequent section.

Weight Retain Updating for Memory

To achieve the goal of $\mathop{\mathbb{E}}_{\mathbb{M}^i}\left[\widehat{R}^{\mathtt{RM}}_{\mathbb{M}^i}(f)\right] \approx \widehat{R}^{\mathtt{RM}}_{\mathbb{D}^i}(f)$, we should maintain consistency in the $|\mathbb{D}_k^+|, |\mathbb{D}_k^-|$ of Eq. (4), which is the critical reweighting factor, between $\mathbb{M}_k$ and $\mathbb{D}_k$ for each class $k \in \mathbb{K}^t$. Note that the widely-used strategies for updating memory buffer in CL, such as Reservoir Sampling (RS) (Vitter 1985), do not consider this, resulting in varying degrees of negative impact on the RLDAM loss of data stored in memory. Therefore, we propose a new Weight Retain Updating (WRU) strategy to address this issue.

Specifically, after learning a task, for each class $k \in \mathbb{K}^t$, we first calculate the $|\mathbb{D}_k^+|$ and $|\mathbb{D}_k^-|$, respectively. Then, we design a greedy algorithm to select $|\mathbb{M}^t|$ samples from $\mathbb{D}^t$. The goal of the greedy algorithm is to minimize the discrepancy in the ratio of positive and negative instances between $\mathbb{D}^t$ and $\mathbb{M}^t$. The selection principle is outlined as $s^* = \arg\min_s \sum_{k\in\mathbb{K}^t} |\mathtt{Rat}(\mathbb{D}^t, k) - \mathtt{Rat}(\mathbb{M}^t \cup (\mathbf{x}_s, \mathbf{y}_s), k)|$, where s^* is the index of our best choice in each selection step, and the function $\mathtt{Rat}(\mathbb{D}, k) = |\mathbb{D}_k^+|/|\mathbb{D}_k^-|, \mathbb{D}_k^+ \subset \mathbb{X}, \mathbb{D}_k^- \subset \mathbb{X}$ takes a dataset and a class index as input, and outputs the ratio. We repeatedly perform this selection for storage until the memory is full.

The above procedure will give an approximate ratio, however, given that the cost of storing a few constants is much lower than storing samples, we rather explicitly store $|\mathbb{D}_k^+|$ and $|\mathbb{D}_k^-|$ into memory along with $|\mathbb{M}^t|$ samples selected using our designed algorithm from $\mathbb{D}^t$. Thus we can maintain the original constants of positive and negative instances between $\mathbb{D}^t$ for its corresponding subset $\mathbb{M}^t$.

Further, to fully utilize the memory space, we follow Rebuffi et al. (2017) to store M/t samples for each task after learning $\mathcal{T}^t$. Due to the fixed memory size, we remove some past samples before storing new ones, where we just remove samples without changing the stored $|\mathbb{D}_k^+|$ and $|\mathbb{D}_k^-|$.

Overall, the training procedure of our continual learning method is summarized in Algorithm 1.

Theoretical Analyses

Here, we first analyze the generalization bound of the RLDAM-based algorithm in the batch MLL, and then give the generalization of our algorithm with RLDAM loss and WRU in MLCL. Notably, for batch MLL, technically, compared with the work (Wu, Li, and Yin 2023) we mainly follow, we introduce a new definition of fractional Rademacher complexity for the hypothesis space with its upper bound, and a new contraction inequality (see Appendix B for details). Moreover, to the best of our knowledge, this is the first theoretical work on MLCL.

For the RLDAM-based algorithm in batch MLL. Firstly, we introduce the quantity of label-wise class imbalance (Wu, Li, and Yin 2023) as $\tau_k = \min\{|\mathbb{D}_k^+|, |\mathbb{D}_k^-|\}/n, \forall k \in [K]$. In continual learning setting, we denote τ_k^i as the imbalance of the i-th task. For simplicity, here we consider the general kernel-based hypothesis class, formally written as:

$$\mathcal{F} = \left\{\mathbf{x} \mapsto \mathbf{W}^\top \Phi(\mathbf{x}) : \mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_K)^\top, \|\mathbf{w}_k\|_{\mathbb{H}} \leq \Lambda\right\}, \quad (7)$$

where $\kappa : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is a positive definite symmetric kernel and its induced reproducing kernel Hilbert space (RKHS) is $\mathbb{H}$, and $\Phi : \mathcal{X} \to \mathbb{H}$ is a feature mapping of κ. Note that although we utilize deep neural networks (DNNs) in experiments, it can still provide valuable insights because recent theory (Jacot, Gabriel, and Hongler 2018) has established the connection between over-parameterized DNNs and Neural Tangent Kernel (NTK)-based methods. More specifically, Huang et al. (2020); Tirer, Bruna, and Giryes (2022) analyses the ResNet, which is used in our work, from the NTK perspective. Then, we introduce the following mild assumption for subsequent analyses.

Assumption 2. *(1) The training data $\mathbb{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ is i.i.d. sampled from the distribution P, where $\exists\, r > 0$, it satisfies $\kappa(\mathbf{x}, \mathbf{x}) \leq r^2$ for all $\mathbf{x} \in \mathcal{X}$.*
(2) The hypothesis class is defined in Eq. (7).

Algorithm 1: Replay-based Continual Learning Procedure.

Input: Tasks $\mathcal{T}$, Task length $T (T > 1)$, Memory $\mathcal{M}$, Memory size M;
Parameter: Learning rate η, Batch size B, Epochs n_e, The model f;
Output: Learned parameter Θ of f;

for $t \in (1, T)$ **do**
  Get dataset $\mathbb{D}^t$ from $\mathcal{T}^t$;
  if $t = 1$ **then**
    Perform Batch Learning on the $\mathbb{D}^t$;
  else // Continual learning procedure;
    Get $\{\mathbb{M}^1, ..., \mathbb{M}^{t-1}\}$ from $\mathcal{M}$;
    // Training iteration
    for $i \in (1, n_e)$ **do**
      for each batch $\mathbb{B}^t \in \mathbb{D}^t$ $(|\mathbb{B}^t| = B)$ **do**
        Sample a batch $\mathbb{B}^{\mathcal{M}}$ with size of B from $\cup_{i\in(1,t-1)}\mathbb{M}^i$;
        Update model parameters Θ according to Eq. (19) (Appendix E.2) with $\mathbb{B}^t, \mathbb{B}^{\mathcal{M}}, \eta$;
  // Training ending;
  Updating the memory according to Sec. ;
return Θ;

---

(3) The base loss $\ell(z)$ is the hinge loss and bounded by B, and $\forall k \in \mathbb{K}$, $\ell(z-\Delta_k^+)$ and $\ell(z-\Delta_k^-)$ is ρ_k^+- and ρ_k^--Lipschitz continuous with $\rho_k^+ = 1/\Delta_k^+, \rho_k^- = 1/\Delta_k^-$. [4]

Besides, the expected risk w.r.t. RLDAM loss can be defined as $R^{\texttt{RM}}(f) = \mathop{\mathbb{E}}_{\mathbb{D}}\left[\widehat{R}^{\texttt{RM}}_{\mathbb{D}}(f)\right]$ and we can get $R^{0/1}(f) \le R^{\texttt{RM}}(f)$. Then, we can obtain the learning guarantee of the RLDAM-based algorithm w.r.t. the Macro-AUC measure, informally as follows (see Appendix A for formal details):

$$R^{0/1}(f) \le \widehat{R}^{\texttt{RM}}_{\mathbb{D}}(f) + O\left(\frac{1}{\sqrt{n}}\left(\frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{\tau_k}}(\rho_k^+ + \rho_k^-)\right)\right).$$

This bound indicates that the batch algorithm with $\mathcal{L}_{\texttt{RM}}$ has an imbalance-margin-aware learning guarantee of $O\left(\left(\frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{\tau_k}}(\rho_k^+ + \rho_k^-)\right)\right)$ w.r.t. Macro-AUC. Compared with the one of the $\mathcal{L}_{\texttt{RU}}$-based algorithm (Theorem 4 in Wu, Li, and Yin (2023)), with identical $\rho_k^+ = \rho_k^-$ for all classes with one base loss, ours with $\mathcal{L}_{\texttt{RM}}$ can assign class-aware ρ_k^+ and ρ_k^- for distinct classes with two different base losses for each task. As $\rho_k^+ = 1/\Delta_k^+$, we can adjust the margins Δ_k^+ and Δ_k^+ to make $\rho_k^+ + \rho_k^-$ smaller under the constant constraint of $\Delta_k^+ + \Delta_k^-$, leading to better guarantees.

Choice of Optimal Margins. Then, how to choose the optimal margins is critical. Here we follow the idea in Cao et al. (2019), which proves that the optimal margins for binary classification are $\Delta^+ = \frac{\lambda}{(n_1)^{1/4}}$ and $\Delta^- = \frac{\lambda}{(n_2)^{1/4}}$, and extend it to multi-class classification with $\Delta_k = \frac{\lambda}{(n_k)^{1/4}}$ for each class k. For our margin choices, it becomes more complicated, including two main steps.

Firstly, in a two-label MLL task, we can derive the optimal margins for each class as $\widetilde{\Delta}_1 = \frac{\lambda}{|\mathbb{D}_1^+|^{1/4}}, \widetilde{\Delta}_2 = \frac{\lambda}{|\mathbb{D}_2^+|^{1/4}}$ (see Proposition 1 in Appendix B.3). Similarly to Cao et al. (2019), we extend the results to an MLL task with more than two labels, getting $\widetilde{\Delta}_k = \frac{\lambda}{|\mathbb{D}_k^+|^{1/4}}$. This implies that distinct classes in a multi-label dataset should be given different margins associated with the number of positive instances.

Secondly, for only one-label MLL task, indeed, it can be seen as a binary classification from positive and negative instances. Similarly to the above analysis, we analyze the optimal margins, obtaining $\Delta_k^+ = \Delta_k^-$ (see Proposition 1. (2) in Appendix B.3). Intuitively, this result is somewhat surprising because it assigns equal margins to positive and negative instances for one label. This may be puzzling as it treats positive and negative instances equally. However, we reweight the loss for positive and negative instances as the second equation in Eq. (4), implying that the positive and negative instances have been "balanced". Hence, it makes sense to assign equal margins to both positive and negative instances.

Finally, we can get $\Delta_k^+ = \Delta_k^- = \frac{\lambda}{|\mathbb{D}_k^+|^{1/4}}$ with a constant λ as a hyper-parameter to be tuned in experiments.

For RLDAM-based algorithm with WRU in MLCL. Based on the above theory result in batch MLL, we can obtain the following learning guarantee of the RLDAM-based algorithm with WRU in MLCL, where the techniques mainly follow recent theory work (Shi and Wang 2023) in domain-incremental continual learning and the work (Mansour, Mohri, and Rostamizadeh 2009) in domain adaptation.

Theorem 1 (**Learning guarantee of RLDAM-based algorithm with WRU in MLCL, full proof in Appendix C**). *Suppose* Assumption 1 and 2 *hold. Let n^i and $\tilde{n}^i$ denote the number of samples in task $\mathcal{T}^i$ and from previous tasks in the memory buffer. According to class-incremental setup, let $f^{t,i}$ denote the outputs of function f for a specific task $\mathcal{T}^i$ when learning task $\mathcal{T}^t$. Let F^{t-1} be the model learned after $t-1$ tasks, and $F^{t-1,i}$ denote the outputs of model F^{t-1} for a specific task $\mathcal{T}^i$. Assume constants $\alpha^i + \beta^i + \gamma^i = 1$. Then, with probability at least $1-\delta$, we have*

$$\begin{aligned}
\sum_{i=1}^{t} R_{\mathbb{D}^i}(f^{t,i}) \le &\left\{\sum_{i=1}^{t-1}\gamma^i \widehat{R}_{\mathbb{D}^i}(f^{t,i}) + \widehat{R}_{\mathbb{D}^t}(f^{t,t})\right\} \\
&+ \left\{\sum_{i=1}^{t-1}\gamma^i\epsilon^i + \epsilon^t + 2\sum_{i=1}^{t-1}\beta^i(\epsilon^i + \epsilon^t)\right\} \\
&+ \left\{\sum_{i=1}^{t-1}\gamma^i\xi^i + \xi^t + B\sum_{i=1}^{t-1}\beta^i(\xi^i + \xi^t)\right\} \\
&+ \sum_{i=1}^{t-1}(\alpha^i + \beta^i) R_{\mathbb{D}^i}(F^{t-1,i}) \\
&+ \left\{\sum_{i=1}^{t-1}\alpha^i\psi^i + \sum_{i=1}^{t-1}(\beta^i)\psi^t\right\} + \sum_{i=1}^{t-1}\beta^i \texttt{disc}(\mathbb{D}^i, \mathbb{D}^t),
\end{aligned}$$

where $n^i = \tilde{n}^i$ when $i < t$, and

$$\begin{aligned}
\epsilon^i &= \frac{4\Lambda r^i}{\sqrt{n^i}}\left(\frac{1}{|\mathbb{K}_i|}\sum_{k\in\mathbb{K}_i}\sqrt{\frac{1}{\tau_k^i}}(\rho_k^{i+} + \rho_k^{i-})\right), \\
\xi^i &= 6B\sum_{i=1}^{t-1}\gamma^i\sqrt{\frac{\log(\frac{6}{\delta})}{2n^i}}\left(\sqrt{\frac{1}{|\mathbb{K}_i|}\sum_{k\in\mathbb{K}_i}\frac{1}{\tau_k^i}}\right), \\
\psi^i &= R_{\mathbb{D}^i}(f^{t,i}, F^{t-1,i}).
\end{aligned}$$

Remark. *From the above bound of right terms, we can get valuable insights into the continual learning process, demonstrating the effectiveness of our proposed RLDAM loss and WRU.* ***(1)*** *The second term that involves all ϵ is the bound of model complexity which contains the ρ_k^+, ρ_k^- for all tasks. Similar to the batch learning scenario, we can select better margins to make the bound tighter,* ***which indicates the effectiveness of our proposed RLDAM loss****.* ***(2)*** *In both the second and third terms there are $n^i \cdot \tau_k^i$ in denominator. Recall the definition of τ_k^i, we can get $n^i \cdot \tau_k^i = \min\{\mathbb{D}_k^{i+}, \mathbb{D}_k^{i-}\}$. With our proposed WRU, we can guarantee that $n^i \cdot \tau_k^i$ for class in memory is* ***stable and unbiased*** *relative to multi-task joint training.* ***(3)*** *The fourth term is the* ***expected risk*** *of F^{t-1} on all previous tasks, which reflects how well the model learned in the previous step affects the performance of the current model (**forward transfer**).* ***(4)*** *The fifth term measures* ***the gap between the current and***

[4] Note that, the widely-used hinge and logistic loss are both 1-Lipschitz continuous and we analyze the hinge loss here for clarity. Similar analyses can be obtained for other margin-based losses.

previous model** on each task, which suggests that model updates that deviate too much from the previous in each step of learning will result in a larger bound, i.e., forgetting. The replay approach we used can be viewed as a regularization that restricts the degree of variation in the model. By applying WRU, this item can be reduced.* ***(5) *The last term is a measure of the* ***discrepancy distance*** *(defined in Definition 3, Appendix C.1) between the current task and all past tasks. If $\mathbb{D}^i = \mathbb{D}^t$, this term will be zero, thus reducing forgetting.*

Related Work

Continual Learning. Continual learning methods are mainly divided into three branches (De Lange et al. 2021): replay-based approaches (Chaudhry et al. 2019; Rebuffi et al. 2017), regularization-based approaches (Zenke, Poole, and Ganguli 2017; Farajtabar et al. 2020), and architecture-based approaches (Fernando et al. 2017; Mallya and Lazebnik 2018). Here we focus on replay-based approaches.

The rehearsal (De Lange et al. 2021) (or experience replay (Chaudhry et al. 2019; Wang et al. 2023)) approach is a branch of replay-based methods, which explicitly retrain on a subset of stored samples while training on new tasks. Concerning rehearsal memory, many approaches have been explored to exploit how to design and utilize it. Considering different tasks, several memory-updating strategies are proposed (Chaudhry et al. 2019; Riemer et al. 2018; Rebuffi et al. 2017). Among them, Reservoir Sampling (Vitter 1985) is a sampling method commonly used to design memory updating strategies (Riemer et al. 2018; Chaudhry et al. 2019).

Theoretical study on CL is challenging even in the multi-class continual learning community. Several studies (Knoblauch, Husain, and Diethe 2020; Evron et al. 2023; Lin et al. 2023) on continual learning demonstrate the challenges involved in conducting theoretical analysis for continual learning. However, we strive to theoretically derive a generalization bound for MLCL on replay-based framework following the proof in Shi and Wang (2023), and verify the effectiveness of our proposed method.

Multi-Label Continual Learning. PRS (Kim, Jeong, and Kim 2020) is an earlier work to tackle the MLL problem under CL settings, in which a Partition Reservoir Sampling (PRS) is proposed to maintain a balanced knowledge of both classes. Liang and Li (2022) proposes optimizing class distribution in memory (OCDM) for MLCL. OCDM formulates the memory update mechanism as an optimization problem to minimize the distribution distance between the whole dataset and memory data and then updates the memory by solving this problem. Dong et al. (2023) proposes a knowledge restore and transfer (KRT) framework for Multi-Label Class-Incremental Learning, which includes a dynamic pseudo-label (DPL) module and an incremental cross-attention (ICA) module.

Experiments

In this section, we conduct experiments to illustrate the effectiveness of our method, which is summarized as follows: (1) We conduct comparison experiments with other baselines to illustrate the superiority of our method. (2) The memory size is influential to the replay-based approaches. We illustrate that our method consistently outperforms ER and is less sensitive to memory sizes. (3) Ablation studies show the effect of each component proposed in our method.

Please see Appendix E for more details and results.

	Baseline	C-VOC	C-COCO	C-NUS
Macro-AUC	FT	53.62	69.46	66.91
	FT-RLDAM	70.30	70.07	67.89
	ER (Chaudhry et al. 2019)	79.17	72.68	69.31
	ER-RS (Chaudhry et al. 2019)	79.18	73.07	70.14
	PRS (Kim, Jeong, and Kim 2020)	76.96	70.83	69.64
	OCDM (Liang and Li 2022)	79.98	74.79	73.23
	KRT (Dong et al. 2023)	79.22	74.30	75.23
	Ours	**88.69**	**77.93**	**79.77**
Forgetting	FT	45.63	26.13	26.58
	FT-RLDAM	23.24	30.15	30.30
	ER (Chaudhry et al. 2019)	3.44	17.56	8.64
	ER-RS (Chaudhry et al. 2019)	2.31	15.45	7.85
	PRS (Kim, Jeong, and Kim 2020)	0.95	**4.93**	8.83
	OCDM (Liang and Li 2022)	**-0.19**	7.01	**0.71**
	KRT (Dong et al. 2023)	3.93	1.50	5.60
	Ours	2.05	8.22	8.72

Table 1: Comparison results with other baselines on three benchmarks. The boldfaced items denote the best results.

Experimental Setup

Benchmarks. Following previous research in Multi-Label Continual Learning (Kim, Jeong, and Kim 2020; Liang and Li 2022; Dong et al. 2023), we utilize three commonly used multi-label classification datasets: PASCAL VOC (Everingham et al. 2015), MSCOCO (Lin et al. 2014) and NUS-WIDE (Chua et al. 2009). These datasets are then transformed into their continual versions as C-PASCAL-VOC, C-MSCOCO, and C-NUS-WIDE. More details are described in Appendix E.1. For simplicity and space-saving, we use C-VOC, C-COCO, and C-NUS to represent three benchmarks.

Baselines. We compare our method with several baselines, including plain ER (with a random sample strategy), ER-RS (Chaudhry et al. 2019), PRS (Kim, Jeong, and Kim 2020), OCDM (Liang and Li 2022) and KRT (Dong et al. 2023). ER and RS are commonly used in CL. PRS, OCDM and KRT are recent MLCL methods. Finetune performs sequential training among tasks without any CL techniques.

Comparison Results

We start by illustrating the imbalance statistics of an example of three tasks in C-MSCOCO, as shown in Fig.1, Appendix E. Task 1 exhibits varying sample sizes, with one class having around six thousand samples, while some others have only a few hundred. A similar phenomenon is observed in Task 2 and Task 3.

Regarding this kind of imbalance problem, we conduct experiments to demonstrate the effectiveness of our proposed method. Table 1 shows that our method outperforms other baselines on all benchmarks. While our focus is on

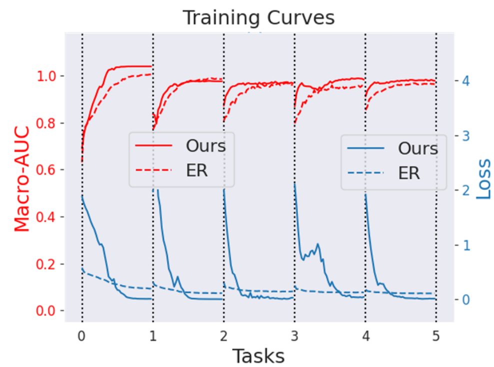

Figure 1: The comparison of training curves between our method and ER on C-PASCAL-VOC.

Figure 2: The comparison of overall test performances (test on all tasks after learning each single task) between our method and ER on C-PASCAL-VOC.

maximizing Macro-AUC using RLDAM loss and WRU, the forgetting metric is not our primary concern. However, our method still achieves agreeable forgetting performance. Comparing Finetune (FT) and Finetune with RLDAM loss (FT-RLDAM), we observe a significant improvement in Macro-AUC with RLDAM loss. This highlights our method's focus on enhancing overall and per-task performance. Moreover, with RLDAM loss, Finetune performs comparably to ER on C-MSCOCO and C-NUS-WIDE.

Fig.1 presents the training curves on C-PASCAL-VOC. Our method consistently outperforms ER in Macro-AUC, except for the first task, and converges faster in terms of training loss. Fig.2 demonstrates similar trends in test performances, with our method consistently achieving higher overall Macro-AUC compared to ER.

Ablation Studies

We conduct ablation experiments to validate the effect of each component that we propose. From the results shown in Table 2, we can observe that the plainest ER just using BCE loss without any component we proposed underperforms others. When applying reweighted loss or margin loss, the Macro-AUC gains huge improvements. Furthermore, the combination of the two losses improves the performance further. The WRU retains the ratio of positive and negative instances, which tackles the imbalance in the memory and hence improves the overall Macro-AUC.

Reweighted loss	Margin loss	WRU	C-VOC	C-COCO	C-NUS
×	×	×	79.17	72.68	69.31
√	×	×	83.04	74.86	76.81
√	×	√	84.92	75.36	77.60
×	√	×	78.80	73.95	76.03
√	√	×	86.29	77.61	78.59
√	√	√	**88.69**	**77.93**	**79.77**

Table 2: Ablation studies for each proposed component of our method. The √ denotes "with", and the × denotes "without". The first row without any component is actually ER with BCE loss. Note that "×" does not signify the absence of any replay strategy but rather the usage of a plain ER.

	Memory Size	C-VOC	C-COCO	C-NUS
ER	200	67.36	63.36	63.94
	500	74.31	62.68	64.95
	1000	75.75	63.97	67.25
	1500	78.77	62.53	69.06
	2000	79.17	69.47	69.31
Ours	200	**78.88**	**75.78**	**76.08**
	500	**81.86**	**76.03**	**76.52**
	1000	**85.70**	**77.15**	**78.74**
	1500	**87.89**	**77.55**	**78.48**
	2000	**88.49**	**77.93**	**79.77**

Table 3: The Macro-AUC of our method compared with ER under different memory sizes.

Effect of Memory Size

We use the data replay approach as our CL technique. Consequently, we illustrate in this subsection that our proposed method performs well with different memory size setups. Specifically, we set five memory sizes: [200, 500, 1000, 1500, 2000]. The results are shown in Table 3, from which we can see that our method consistently outperforms ER and is less sensitive to changes in memory size. That indicates our method is effective regardless of memory size.

Conclusion and Discussion

In this paper, we focus on maximizing Macro-AUC for Multi-Label Continual Learning to tackle the imbalance problem, where the replay-based approach is considered. To maximize Macro-AUC, we propose an RLDAM loss inspired by recent theoretical principled reweighted univariate loss and LDAM loss. Furthermore, we provide theoretical analyses of the generalization bound, supporting the superiority of our method. Then, based on the RLDAM loss and data replay framework, we propose the Weight Retain Updating (WRU) storing samples to maintain the numbers of positive and negative instances of each class into memory. Finally, comprehensive experiments illustrate the effectiveness of our proposed method.

Ethical Statement

Please see Appendix F for limitations and impacts in detail. Our work can contribute to the MLCL community, while without any significant societal impacts.

Acknowledgments

This work was supported by the NSFC (Nos. U23A20389, 62206159, 62176139), the Natural Science Foundation of Shandong Province (Nos. ZR2022QF117, ZR2021ZD15, ZR2024MF101), the Fundamental Research Funds of Shandong University, the Fundamental Research Funds for the Central Universities. G. Wu and H. Sun were also sponsored by the TaiShan Scholars Program (Nos. tsqn202306051, tsqn202312026).

References

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

Ceccon, M.; Pezze, D. D.; Fabris, A.; and Susto, G. A. 2024. Multi-Label Continual Learning for the Medical Domain: A Novel Benchmark. *arXiv preprint arXiv:2404.06859*.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, 532–547.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P.; Torr, P.; and Ranzato, M. 2019. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, 1–9.

Dalle Pezze, D.; Deronjic, D.; Masiero, C.; Tosato, D.; Beghi, A.; and Susto, G. A. 2023. A multi-label continual learning framework to scale deep learning approaches for packaging equipment monitoring. *Engineering Applications of Artificial Intelligence*, 124: 106610.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3366–3385.

Dong, S.; Luo, H.; He, Y.; Wei, X.; Cheng, J.; and Gong, Y. 2023. Knowledge Restore and Transfer for Multi-Label Class-Incremental Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18711–18720.

Everingham, M.; Eslami, S. A.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111: 98–136.

Evron, I.; Moroshko, E.; Buzaglo, G.; Khriesh, M.; Marjieh, B.; Srebro, N.; and Soudry, D. 2023. Continual Learning in Linear Classification on Separable Data. *arXiv preprint arXiv:2306.03534*.

Farajtabar, M.; Azizan, N.; Mott, A.; and Li, A. 2020. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, 3762–3773. PMLR.

Fernando, C.; Banarse, D.; Blundell, C.; Zwols, Y.; Ha, D.; Rusu, A. A.; Pritzel, A.; and Wierstra, D. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*.

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4): 128–135.

Huang, K.; Wang, Y.; Tao, M.; and Zhao, T. 2020. Why Do Deep Residual Networks Generalize Better than Deep Feedforward Networks?—A Neural Tangent Kernel Perspective. *Advances in neural information processing systems*, 33: 2698–2709.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Kassim, M. A. 2024. *Multi-label Lifelong Machine Learning using Deep Generative Replay*. Ph.D. thesis, Université d'Ottawa— University of Ottawa.

Kim, C. D.; Jeong, J.; and Kim, G. 2020. Imbalanced continual learning with partitioning reservoir sampling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, 411–428. Springer.

Knoblauch, J.; Husain, H.; and Diethe, T. 2020. Optimal continual learning has perfect memory and is np-hard. In *International Conference on Machine Learning*, 5327–5337. PMLR.

Liang, Y.-S.; and Li, W.-J. 2022. Optimizing Class Distribution in Memory for Multi-Label Online Continual Learning. *arXiv preprint arXiv:2209.11469*.

Lin, S.; Ju, P.; Liang, Y.; and Shroff, N. 2023. Theory on Forgetting and Generalization of Continual Learning. *arXiv preprint arXiv:2302.05836*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Mallya, A.; and Lazebnik, S. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 7765–7773.

Mansour, Y.; Mohri, M.; and Rostamizadeh, A. 2009. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*.

McCallum, A. K. 1999. Multi-label text classification with a mixture model trained by EM. In *AAAI 99 workshop on Text Learning*. Citeseer.

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

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*.

Ring, M. B. 1994. *Continual learning in reinforcement environments*. The University of Texas at Austin.

Shi, H.; and Wang, H. 2023. A Unified Approach to Domain Incremental Learning with Memory: Theory and Algorithm. *Advances in Neural Information Processing Systems*, 36.

Tarekegn, A. N.; Giacobini, M.; and Michalak, K. 2021. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118: 107965.

Tirer, T.; Bruna, J.; and Giryes, R. 2022. Kernel-based smoothness analysis of residual networks. In *Mathematical and Scientific Machine Learning*, 921–954. PMLR.

Vitter, J. S. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1): 37–57.

Wang, L.; Zhang, X.; Su, H.; and Zhu, J. 2023. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*.

Wu, G.; Li, C.; and Yin, Y. 2023. Towards Understanding Generalization of Macro-AUC in Multi-label Learning. *International Conference on Machine Learning*.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, 3987–3995. PMLR.

Zhang, M.-L.; and Zhou, Z.-H. 2013. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8): 1819–1837.