UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Dottorato di Ricerca in
"Information and Communication Technologies (ICT)"

Ciclo XXXVIII

Apprendimento Continuo in Condizioni Rumorose e con Composizionalità

Candidata: Monica Millunzi
Relatore (Tutor): Prof. Simone Calderara
Coordinatore del Corso di Dottorato: Prof. Luigi Rovati

Comitato di Revisione:

Prof. Petter N. Kolm	NYU Courant
Prof. Natalia Díaz-Rodríguez	University of Granada

Tesi di dottorato finanziata dall'Unione europea- Next Generation EU, Missione 4, componente 2 "Dalla Ricerca all'Impresa" - Investimento 3.3 "Introduzione di dottorati innovativi che rispondono ai fabbisogni di innovazione delle imprese e promuovono l'assunzione dei ricercatori dalle imprese".

UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

International Doctorate School in
"Information and Communication Technologies (ICT)"

Cycle XXXVIII

Learning under Noise and Compositionality: A Continual Learning Perspective

Candidate: Monica Millunzi
Advisor: Prof. Simone Calderara
Director of the School: Prof. Luigi Rovati

Review Committee:

Prof. Petter N. Kolm — NYU Courant
Prof. Natalia Díaz-Rodríguez — University of Granada

PhD thesis funded by the European Union – NextGenerationEU, Mission 4, Component 2 "From Research to Business" – Investment 3.3 "Introduction of innovative doctoral programmes that meet the innovation needs of enterprises and promote the recruitment of researchers by companies.

"*Give a man a fish and you feed him for a day;*
teach a man to fish and you feed him for a lifetime."

To my parents, who gave me the tools,
and to my brother, who gave me the courage.

To my friends and loved ones, who stood by me,
and to myself, who persevered.

Apprendimento Continuo in Condizioni Rumorose e con Composizionalità

Sommario

Per stare al passo con la natura in continua evoluzione dei dati, i moderni sistemi di intelligenza artificiale richiedono frequenti e costosi riaddestramenti su tutti gli esempi già visti, per evitare il fenomeno del catastrophic forgetting. Questa esigenza ha stimolato un crescente interesse verso il Continual Learning (CL), in cui i modelli apprendono in modo incrementale dai flussi di dati conservando al contempo le conoscenze acquisite. Tuttavia, nonostante i notevoli progressi, permane una fonte di incertezza spesso trascurata: la presenza di etichette rumorose o non affidabili. Le reti neurali profonde devono gran parte del loro successo a grandi dataset puliti, ma in contesti reali e dinamici tali condizioni ideali sono rare. Questo pone una domanda cruciale: come può un sistema continuare ad apprendere efficacemente quando la supervisione stessa è imperfetta?

Per affrontare il problema dell'apprendimento con etichette rumorose in scenari incrementali, proponiamo Alternate Experience Replay (AER), una strategia che alterna fasi di apprendimento e "dimenticanza" del buffer, favorendo la separazione tra campioni puliti e rumorosi. In questo modo, il modello affina progressivamente le proprie rappresentazioni interne limitando la propagazione del rumore. Inoltre, introduciamo Asymmetric Balanced Sampling (ABS), un meccanismo di campionamento che bilancia dinamicamente la conservazione di esempi puliti e complessi durante l'aggiornamento del buffer. La combinazione di questi approcci migliora la robustezza e la stabilità del modello, dimostrando come anche semplici meccanismi di replay di memoria possano ridurre l'impatto della supervisione imperfetta.

La robustezza da sola, tuttavia, non basta. Una seconda sfida riguarda la capacità del modello di comporre e riutilizzare le conoscenze tra compiti diversi, proprietà nota come composizionalità. In questa tesi, mostriamo come rappresentazioni modulari e prospettive di ottimizzazione di secondo ordine possano favorire questa capacità. Introduciamo due paradigmi complementari: Incremental Task Arithmetic (ITA), che ottimizza ciascun modello addestrato su un singolo task individualmente, e Incremental Ensemble Learning (IEL), che ottimizza direttamente la loro composizione. Insieme, questi approcci permettono di combinare componenti apprese progressivamente in sistemi che non solo resistono al forgetting, ma generalizzano per costruzione, adattandosi a nuove e inattese combinazioni di task di apprendimento.

Queste due prospettive, robustezza alla supervisione rumorosa e adattamento composizionale, delineano un modello di Continual Learning resiliente e strutturato. Attraverso analisi empiriche approfondite, mostriamo come i meccanismi intrinseci delle reti neurali possano essere sfruttati per sviluppare modelli incrementali più affidabili e robusti. Questa tesi contribuisce alla ricerca nel campo del Continual Learning, migliorando la robustezza sia in presenza di supervisione rumorosa sia attraverso la composizione modulare della conoscenza appresa. Forniamo una panoramica dello stato dell'arte, approfondimenti metodologici e studi sperimentali accurati su task incrementali complessi, con l'obiettivo di favorire lo sviluppo futuro di sistemi di apprendimento più adattivi e affidabili.

Learning under Noise and Compositionality: A Continual Learning Perspective

Abstract

To keep up with the ever-changing nature of data, modern Artificial Intelligence systems require frequent and costly retraining on all previously seen examples to avoid the phenomenon of catastrophic forgetting. This challenge has fueled the growing interest in Continual Learning (CL), where models learn incrementally from streams of data while retaining prior knowledge. Yet, despite the remarkable progress of CL methods, an often-overlooked source of uncertainty remains: the presence of noisy or unreliable labels. Deep Neural Networks owe much of their success to large, clean datasets, but in dynamic, real-world settings, such ideal conditions are rarely available. This raises a fundamental question — how can a system continue to learn effectively when its supervision is itself imperfect?

To address this, we first revisit the problem of learning under noisy labels in an incremental scenario. We propose Alternate Experience Replay (AER), a strategy that alternates steps of buffer learning and buffer forgetting to encourage the separation of clean and noisy samples in the buffer, allowing the model to progressively refine its internal representations while limiting the propagation of label noise. In addition, we introduce Asymmetric Balanced Sampling (ABS), a complementary sampling mechanism that dynamically promotes the retention of clean and complex examples during the buffer update. By leveraging the interplay between sample selection and memory update, this approach improves robustness and stability across learning stages, showing that even simple replay-based mechanisms can mitigate the impact of imperfect supervision over time.

However, robustness alone is not enough. A second challenge lies in the model's ability to compose and reuse knowledge across tasks, a property known as compositionality. In this thesis, we explore how modular representations and second-order optimization perspectives can enhance this capability within continual learning frameworks. To this end, we introduce two opposed learning paradigms: Incremental Task Arithmetic (ITA), which focuses on optimizing each task model individually, and Incremental Ensemble Learning (IEL), which directly optimizes their composition. Together, these two approaches offer dual perspectives on continual compositionality, combining incrementally learned components into systems that not only resist forgetting but also generalize by construction, adapting flexibly to new and unseen combinations of tasks.

These two perspectives, robustness to noisy supervision and compositional adaptation, outline a vision of continual learning that is both resilient and structured. Through extensive empirical analysis, we show how intrinsic mechanisms within neural networks can be harnessed to achieve more reliable and robust continual learners. With this thesis, we aim to contribute to research in Continual Learning, enhancing robustness both in the presence of noisy supervision and through modular knowledge composition. We provide a comprehensive overview of the state-of-the-art, detailed methodological insights, and thorough experimental studies on challenging incremental tasks, with the ultimate goal of fostering future work toward more adaptive and trustworthy learning systems.

Contents

1 Introduction

This dissertation explores different aspects of learning under incremental conditions, focusing on two directions. The first addresses the challenge of noisy supervision in continual learning, proposing mechanisms that exploit forgetting dynamics to preserve reliable information during sequential training. The second examines model merging from the perspective of incremental learning, studying how modular updates can be derived and combined in large pre-trained architectures through second-order approximations. Although these directions target distinct problems, they both operate within the broader framework of continual learning, characterized by sequential task updates and constrained access to past supervision.

1.1 Overview

Modern machine learning systems increasingly operate in dynamic environments where data distributions evolve, supervision is imperfect, and models must adapt continuously while retaining previously acquired competencies. Despite the success of Deep Neural Network (DNN) across a wide range of domains, their stan-

dard training paradigm assumes access to static datasets with clean labels and unrestricted repeated optimization over the full data, an assumption that rarely holds in realistic deployment scenarios. In applications such as autonomous systems, web-scale perception pipelines, large-scale recommendation, and language-based services, data arrive as streams, annotation quality is variable, and systems are required to integrate new knowledge without compromising earlier abilities, under strict computational and latency constraints that preclude full retraining at each update. These conditions expose fundamental limitations of contemporary deep networks.

The growing relevance of adaptive learning has brought several concrete challenges to the forefront. Maintaining stable performance while acquiring new knowledge remains difficult: networks trained sequentially on distinct tasks tend to overwrite internal representations, rapidly losing performance on earlier ones. Moreover, realistic data streams inevitably contain annotation errors arising from limited human supervision, automatic labeling pipelines, or systematic collection biases. In replay-based systems, such noisy examples are stored and repeatedly reused, progressively contaminating the learning process. Finally, the increasing scale and centrality of pre-trained models call for new frameworks for modularity and compositionality, enabling efficient task-specific modification and model editing without resorting to monolithic, full-parameter updates.

Continual Learning (CL) addresses part of these challenges by enabling models to learn from sequences of tasks while preserving previously acquired knowledge. Among CL strategies, rehearsal-based methods have emerged as particularly effective: they maintain a small memory buffer of past data and interleave replayed samples with current observations, thus approximating standard batch training under strict memory constraints. However, the effectiveness of these approaches depends critically on the quality and representativeness of the buffer. As shown in Chapter 2 “May the Forgetting be With You: ALternate Replay for Learning with Noisy Labels”, even a modest fraction of mislabeled data can significantly impair performance, because replay reinforces incorrect supervision and biases the estimation of the underlying task distributions.

Learning under noisy labels is therefore not a marginal complication, but a

central aspect of continual learning in realistic settings. In Continual Learning (CL) and Continual Learning with Noisy Labels (CLN), both the incoming stream and the replay buffer can contain corrupted annotations, and existing rehearsal pipelines inadvertently propagate noise across tasks. The Alternate Experience Replay (AER) framework introduced in "May the Forgetting Be With You" proposes a decisive shift in perspective: instead of treating forgetting as a pure drawback, it deliberately alternates buffer-learning and buffer-forgetting phases, inducing distinct loss dynamics for clean and noisy samples. By temporarily disabling replay, the model is encouraged to forget examples that are inconsistent with the current representation; noisy or hard-mismatched instances exhibit rapid loss growth, while clean samples remain stable. Periodic restoration of the model state then allows one to exploit this divergence purely for buffer purification, maintaining a persistent gap between clean and noisy samples and enabling more reliable sample selection.

Building upon AER and ABS, the EARL framework (Embracing Amnesic Replay for Learning with Noisy Labels) extends this approach in several directions. It provides a more comprehensive analysis of forgetting dynamics under noise, evaluates robustness under realistic noise conditions (including human and web-scraped labels), and systematically studies the interaction between amnesic replay and sampling strategies. Moreover, EARL is instantiated with modern pre-trained backbones and prompt-based CL baselines, and its effectiveness is demonstrated not only in vision benchmarks but also on natural language understanding ones. This extended study confirms that targeted forgetting can be exploited to increase buffer purity without sacrificing diversity, that the overhead introduced by model reloading and buffer updates is negligible compared to standard training costs, and that the method retains its usefulness even at low or zero noise levels. Overall, EARL consolidates the view that forgetting can be transformed from an undesirable side effect into a controlled mechanism for noise detection and robustness.

In parallel to advances in noise-robust CL, the rapid rise of large pre-trained models has opened a complementary research direction centered on the so-called model compositionality (or model merging). Fine-tuning such models on mul-

tiple tasks produces specialized modules whose parameter updates can, under appropriate conditions, be meaningfully combined. Empirical works on model soups and task arithmetic suggest that simple linear combinations of fine-tuned weights often yield multi-task models with strong transfer properties. However, the principles that guarantee successful composition remain only partially understood and are often studied in linearized regimes. The work in Chapter 4 "A Second-Order Perspective on Model Compositionality and Incremental Learning" addresses this gap by analyzing standard non-linear networks through a second-order Taylor approximation of the loss around the pre-trained weights.

The second-order formulation yields two key insights. First, it provides an inequality relating the empirical risk of a composed model to the convex combination of the risks of its individual components, under the assumption that all models remain in the "pre-training basin", where higher-order terms are negligible. This shows that compositionality is viable only if each individual model maintains reasonable performance on examples outside its own training distribution. Second, by reinterpreting this requirement, the work frames compositionality as an incremental learning problem: each fine-tuned module must preserve pre-training general capabilities on out-of-distribution data in order for its task vector to remain compatible with others. On this basis, in this work we derive two dual incremental training algorithms: an individual-training scheme, which fine-tunes each module with an additional regularizer anchored to the pre-trained model through a Fisher-based distance, and an ensemble-training scheme, which directly optimizes the composed model while controlling the pairwise alignment of task vectors in the second-order geometry induced by the Hessian or its Fisher surrogate.

These algorithms are evaluated on class-incremental classification settings with varying alignment between pre-training and downstream tasks. The results show that the resulting pools of modules support not only the construction of accurate multi-task models via composition, but also flexible editing operations such as task specialization and selective unlearning. In this sense, the second-order analysis bridges two previously disjoint areas: compositional fine-tuning of pre-trained models and continual learning. The preservation of pre-training knowledge and

out-of-distribution performance, typically studied in CL under the lens of catastrophic forgetting, emerges as a prerequisite for stable model composition; conversely, compositionality offers a new perspective on how incrementally learned modules can be re-used and reconfigured.

Taken together, the three works that underpin this thesis contribute a unified perspective on robust and modular adaptation. The first two introduce replay schemes that intentionally harness forgetting to cope with noisy labels in continual learning, demonstrating that amnesic updates can purify memory buffers and improve long-term performance across synthetic, human, and web noise. The third establishes a theoretical connection between incremental learning and model compositionality in non-linear networks, leading to practical algorithms for training composable modules that can be linearly aggregated into multi-task models without sacrificing pre-training generality.

1.2 Organization

The remainder of the thesis is organized as follows. The present Chapter 1 provides both the motivation and the technical groundwork for the entire dissertation, and the scientific notation used throughout the thesis. It introduces the need for studying learning scenarios where models must adapt progressively rather than being trained in a single *i.i.d*. regime, highlighting the limitations of standard Deep Learning (DL) in non-stationary environments, the effect of noisy labels on replay-based approaches, and the growing relevance of modularity and parameter-efficient adaptation. The material that would traditionally be distributed across background, notation, and related work sections is consolidated in this chapter. Specifically, continual and incremental learning settings are formalized, the notation used throughout the thesis is established, and the principles underlying model composition and task arithmetic are reviewed. This unified organization clarifies how the thesis approaches incremental learning from two complementary perspectives—robustness to noisy supervision and model compositionality—while providing the shared theoretical and methodological context for both research strands.

In Section 1.4, we introduce the core problem settings addressed in this thesis, covering CL and CLN. The section summarizes the main challenges posed by sequential learning, reviews the most established approaches in the CL literature, and describes the benchmarks, evaluation protocols, and baseline methods adopted in the experimental analysis.

To further contextualize the specific challenges arising from imperfect supervision, Section 1.5 provides a dedicated review of Learning with Noisy Labels (LNL), detailing common noise models, representative algorithmic strategies, and their implications in incremental and replay-based settings. Finally, Section 1.6 focuses on task arithmetic and model merging, presenting the fundamental formulations and the most representative methods in this area, thereby establishing a unified reference framework for the compositional and second-order approaches developed in the subsequent chapters.

Chapter 2 [2] forms the beginning of the first research strand and studies the role of forgetting dynamics under noisy supervision. We introduce Alternated Experience Replay (AER) and Asymmetric Balanced Sampling (ABS), a method that leverages controlled forgetting to maintain separability between clean and corrupted samples during continual training [4]. The chapter focuses on the mechanisms underlying noise accumulation and buffer contamination in rehearsal-based continual learning.

Chapter 3 [3] continues this strand by generalizing forgetting-based purification. Through the *Embracing Amnesic Replay for Learning with noisy Labels* (EARL) framework, we examine more realistic noise sources, modern architectures, and multi-modal applications. Together, Chapters 2 and 3 provide a comprehensive investigation of noise-robust continual learning.

In Chapter 4 [5], we explore a complementary direction centered on the incremental adaptation of large pre-trained models through compositional mechanisms. Using a second-order analysis of task vectors, we study when and how modular updates can be combined, and develop two incremental training procedures that support specialization, composition, and structured model editing. This chapter stands on its own while remaining thematically connected through the lens of incremental learning.

Finally, Chapter 5 concludes the dissertation, summarizing the contributions and outlining several directions for future research, including large-scale incremental adaptation, principled handling of unreliable supervision, and refined approaches to building modular DL systems.

1.3 Notation

In this thesis, we consider a parametric model $f_\theta : \mathcal{X} \to \mathbb{R}^C$ with parameters $\theta \in \mathbb{R}^m$, mapping an input $x \in \mathcal{X}$ to class scores over a label set $\mathcal{Y} = \{1, \dots, C\}$. The predicted class is obtained as $\hat{y} = \arg\max_k f_\theta(x)_k$. Unless otherwise specified, we adopt the standard cross-entropy loss $\mathcal{L}(f_\theta(x), y)$ for supervised classification.

In the continual learning setting, the model is exposed to a sequence of T tasks, denoted by $\{\mathcal{T}_1, \dots, \mathcal{T}_T\}$. Each task $\mathcal{T}_t$ is associated with a dataset

$$\mathcal{T}_t = (\mathcal{X}_t, \mathcal{Y}_t) = \{(x_j^{(t)}, y_j^{(t)})\}_{j=1}^{|\mathcal{T}_t|},$$

where samples $(x_j^{(t)}, y_j^{(t)})$ are drawn i.i.d. from a task-specific distribution $p_t(x, y)$. The global training process is non-i.i.d., as the learner observes each task only once and cannot access the data of previous tasks after their completion. In the class-incremental scenario considered in this thesis, tasks have disjoint label sets, so that $\mathcal{Y}_t \cap \mathcal{Y}_{t'} = \emptyset$ for $t \neq t'$ and the model must ultimately discriminate among all classes seen so far.

The overall continual learning objective can be written as the cumulative risk over all tasks

$$\mathcal{L}_{\mathrm{CL}}(\theta) \triangleq \sum_{t=1}^{T} \mathbb{E}_{(x,y)\sim p_t(x,y)} \left[\mathcal{L}(f_\theta(x), y) \right], \tag{1.1}$$

with an ideal solution $\theta^\star = \arg\min_\theta \mathcal{L}_{\mathrm{CL}}(\theta)$, which is unattainable in practice because data from all tasks are not jointly available.

Rehearsal-based methods maintain a finite-capacity memory buffer $\mathcal{M}$ of size B, which stores a subset of past examples. After learning task $\mathcal{T}_t$, the buffer state is denoted by $\mathcal{M}_t \subset \bigcup_{i=1}^{t} \mathcal{T}_i$, with $|\mathcal{M}_t| \leq B$. During training on task t, the model is updated by interleaving current data $(x, y) \sim p_t(x, y)$ with replay sam-

ples $(x_r, y_r) \sim \mathrm{Unif}(\mathcal{M}_{t-1})$. In its simplest form, the rehearsal objective reads

$$\mathcal{L}_{\mathrm{ER}}(\theta) \triangleq \mathbb{E}_{(x,y)\sim p_t}\big[\mathcal{L}(f_\theta(x), y)\big] + \lambda\, \mathbb{E}_{(x_r,y_r)\sim \mathcal{M}_{t-1}}\big[\mathcal{L}(f_\theta(x_r), y_r)\big], \qquad (1.2)$$

where $\lambda \geq 0$ balances stability and plasticity. In this thesis, this formulation is specialized and extended to account for asymmetric replay losses and amnesic updates.

To model learning under noisy labels, we assume that each clean label $y_j^{(t)}$ is corrupted into a noisy label $\tilde{y}_j^{(t)}$ according to a stochastic transition

$$\tilde{y}_j^{(t)} \sim \tilde{p}_t(\tilde{y} \mid y_j^{(t)}),$$

with noise rate $\eta_t = \mathbb{P}(\tilde{y} \neq y)$ that may depend on the task and on the noise mechanism (synthetic, human, web, or machine-generated). The corresponding noisy distribution is $\tilde{p}_t(x, \tilde{y})$ and the CLN objective becomes

$$\mathcal{L}_{\mathrm{CLN}}(\theta) \triangleq \sum_{t=1}^{T} \mathbb{E}_{(x,\tilde{y})\sim \tilde{p}_t}\big[\mathcal{L}(f_\theta(x), \tilde{y})\big]. \qquad (1.3)$$

In replay-based CLN, both the current stream and the buffer may contain noisy pairs $(x, \tilde{y})$, and the buffer state $\mathcal{M}_t$ implicitly mixes clean, hard, and mislabeled samples.

For the compositionality analysis, we consider a pre-trained model with weights θ_0, and a collection of task-specific fine-tuned models $\theta_t = \theta_0 + \tau_t, \qquad t = 1, \ldots, T$, where τ_t denotes the task vector induced by fine-tuning on task t. A composed model is obtained by a convex combination of task vectors,

$$\theta_P = \theta_0 + \sum_{t=1}^{T} w_t \tau_t, \qquad w_t \in [0,1],\ \sum_{t=1}^{T} w_t = 1, \qquad (1.4)$$

giving rise to a compositional predictor $f_P(\cdot) = f(\cdot; \theta_P)$. The second-order approximation of the empirical risk around θ_0 is denoted by $\hat{\ell}_{\mathrm{cur}}(\theta)$ and involves the gradient $\nabla_\theta \ell(\theta_0)$ and the Hessian $H_\ell(\theta_0)$ evaluated at pre-training. Under

the assumption that θ_0 is a local minimum and $H_\ell(\theta_0) \succeq 0$, the Fisher Information Matrix F_{θ_0} provides a tractable diagonal surrogate for $H_\ell(\theta_0)$ and induces a Riemannian metric on the parameter space.

We denote by a_i^t the accuracy on task i after training on task t, and use standard continual learning metrics such as Final Average Accuracy (FAA) and forgetting, defined as functions of $\{a_i^t\}$. When required, we refer to $\mathcal{M}_t^{\text{clean}}$ and $\mathcal{M}_t^{\text{noisy}}$ as the subsets of the buffer that are estimated to be clean or noisy according to the loss-based dynamics induced by the proposed methods.

1.4 Continual Learning Background

Modern DL systems have achieved remarkable success across a wide range of applications, from computer vision and natural language processing to reinforcement learning and generative modeling. However, their standard training paradigm relies on several assumptions that often do not hold in real-world scenarios. In particular, deep networks are typically trained on static datasets with clean labels, under the assumption that data are drawn i.i.d. from a fixed distribution. This setting admits multiple passes over the entire dataset, allowing the model to converge to a solution that minimizes empirical risk. However, many practical applications involve dynamic environments where data distributions evolve over time, supervision is imperfect, and models must continuously adapt while retaining previously acquired capabilities. Examples include autonomous systems that face changing conditions, web-scale perception pipelines that process streams of user-generated content exposed to paraphrasing attacks [1], large-scale recommendation systems that must adapt to changing user preferences, and language-based services that need to integrate new knowledge without compromising earlier abilities.

These constraints expose fundamental limitations of contemporary deep networks, primarily catastrophic forgetting, sensitivity to label noise, and the absence of efficient mechanisms for modular adaptation and model editing.

CATASTROPHIC FORGETTING

Despite their success, Artificial Neural Network (ANN) struggle to remember when learning sequentially. Humans can layer new skills without erasing old ones; neural networks, optimized by backpropagation over shared parameters, tend to overwrite earlier knowledge when exposed to new distributions. The result is catastrophic forgetting—a sharp drop in performance on previous tasks, unlike the gradual decay seen in human memory. First noted in shallow models [114, 136] and later confirmed in deep networks [53], this limitation remains a core obstacle for systems deployed in non-stationary environments.

CONTINUAL LEARNING

Catastrophic forgetting makes iterative deployment expensive: each new data stream risks erasing what the model already knows. CL addresses this by aiming for models that remain plastic enough to absorb new tasks while stable enough to retain past knowledge. Unlike standard *i.i.d*. assumptions, data arrive in sequence, and past samples are typically unavailable. The goal is to maintain performance over time by explicitly managing the stability–plasticity trade-off.

STANDARD SCENARIOS

Forgetting appears in many Machine Learning (ML) tasks, from segmentation [28, 189] and detection [130, 77] to generation [194] and captioning [42], but most CL research is focused on classification because it offers a well-defined comparison ground. Throughout this thesis, CL refers to continual learning for classification unless stated otherwise.

To make results comparable, the community converged on shared settings, benchmarks, and metrics. The authors in [159, 160] introduced a taxonomy where a supervised classification dataset is split into sequential tasks; data from past tasks cannot be revisited, but the learner is notified when a task boundary occurs. Three canonical scenarios differ in how data are partitioned and how inference is conducted:

- **Domain-Incremental Learning (Domain-IL)**: tasks share the same label space but come from different input distributions. At test time, the model must handle inputs from any domain without task labels. Datasets are often crafted by domain-specific transformations or collected across multiple domains (*e.g.* DomainNet [127]).
- **Task-Incremental Learning (Task-IL)**: each task introduces a disjoint set of classes, and the task identity is provided at inference. Conditioning on the task simplifies knowledge separation and typically yields strong performance [46, 9], while reducing classifier bias from imbalanced exposure [181].
- **Class-Incremental Learning (Class-IL)**: tasks are split as in Task-IL, but no task label is available at test time. The model must jointly infer the class among all previously seen classes, making this the most challenging and widely used setup [46].

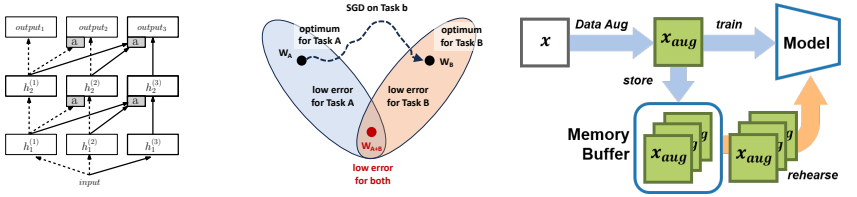
These standard scenarios provide a common yardstick yet leave gaps with real-world incremental applications [8, 41], motivating newer, more realistic variants. Since this thesis emphasizes the Class-IL setting, other scenarios are discussed only when relevant to the experiments.

1.4.1 STATE OF THE ART

The surge of interest in continual learning has produced a steady stream of work addressing catastrophic forgetting from multiple perspectives. Existing approaches are commonly grouped into three broad families, *i.e.* architecture-oriented, regularization-based, and rehearsal-centric methods, each relying on different principles to mitigate interference across tasks [46, 41]. This section summarizes the approaches most frequently cited and adopted as baselines throughout this thesis.

Architectural Methods

Architectural approaches allocate separate subsets of parameters to different tasks, typically through modular designs or multi-head architectures. By isolating task-



(a) (b) (c)

Figure 1.1: Topologies of continual learning methods: (a) architectural, (b) regularization-based, and (c) rehearsal-based approaches.

specific components, interference between tasks is reduced and previously acquired knowledge can be preserved more effectively. However, this comes at the cost of parameter growth as the number of tasks increases, as well as the need for task identity at test time. As a result, these methods are generally better suited to the task-incremental setting than to the class-incremental focus of this work.

A representative example is Progressive Neural Networks (PNN) [142], where each task is assigned a dedicated backbone and lateral connections are introduced to enable knowledge transfer across tasks. While forgetting is naturally avoided, memory usage scales linearly with the number of tasks.

Regularization Methods

Regularization-based approaches retain a single model while constraining parameter updates to preserve directions that are important for previously learned tasks. These methods are typically lightweight and scalable, though their effectiveness may degrade as the number and diversity of tasks increase.

Elastic Weight Consolidation (EWC) [81] penalizes changes to parameters deemed important for past tasks, as estimated through the Fisher Information Matrix. To reduce computational overhead, faster variants such as Online EWC [146] approximate the Fisher information. Learning without Forgetting (LwF) [97] instead relies on knowledge distillation, encouraging the current model to match the outputs of the previous one while learning new data. In class-incremental settings with disjoint label spaces, this strategy can be less effective, motivating extensions such as LwF-MC [137], which modifies the loss to better handle class-

incremental scenarios.

Rehearsal Methods

Rehearsal-based strategies stabilize learning by storing a subset of past examples—or their synthetic counterparts—and replaying them alongside incoming data. Methods in this family differ in how the memory buffer is populated, how replay is performed, and whether raw samples or compressed representations are stored. In some applications, privacy or storage constraints limit the feasibility of retaining raw data. The most widely used strategy for populating the memory buffer is *reservoir sampling* [164], an algorithm designed to maintain a representative random sample of items from a data stream of unknown or arbitrarily large size in a single pass, guaranteeing that each observed item has an equal probability of being retained. This property makes it particularly well-suited for continual learning settings, where the full data distribution is never available at once. Building on this foundation, Experience Replay (ER) [136, 141] maintains a small buffer (often populated via reservoir sampling) and jointly trains on buffered and current samples, serving as a simple yet strong baseline. iCaRL [137] extends this paradigm by combining exemplar rehearsal with knowledge distillation and a nearest-mean-of-exemplars classifier constructed from the retained buffer. GDumb [132] takes a more radical stance, accumulating a buffer throughout the data stream but deferring all learning to evaluation time, training the model from scratch solely on the buffer contents, an approach that yields surprisingly strong performance when the buffer is sufficiently large.

More recent methods refine replay mechanisms through representation alignment, prototype modeling, or auxiliary losses. Dark Experience Replay++ (DER++) [25] stores logits as soft targets, while DER++ combines this strategy with standard replay for improved robustness. COPE [41] maintains slowly evolving class prototypes in a shared embedding space, enabling smoother transitions across tasks.

Additional variants explore architectural decoupling or buffer refinement. ER-ACE [27] modifies standard replay with an asymmetric loss to balance old and new classes. X-DER [20] revises stored samples to inject new evidence about past data, improving adaptation on challenging benchmarks.

Pretrain-Exploiting Methods

A more recent line of work leverages large pretrained models as a starting point for continual learning. Since these backbones already encode broad and transferable representations, such approaches often outperform methods trained from scratch. Many classic continual learning algorithms can also be applied on top of pretrained models for fair comparison. Recent work has largely focused on Vision Transformers and parameter-efficient fine-tuning schemes, particularly prompt-based adaptation strategies, which are discussed in more detail later in the thesis.

Learning to Prompt (L2P) [173] maintains a pool of prompts that are dynamically selected based on the input, with or without replay buffers. DualPrompt [172] introduces both general and task-specific prompts to balance transfer and specialization without rehearsal. CODA-Prompt [149] assigns prompts per task and combines them through attention mechanisms, updating only the active prompt set. SLCA [196] slows representation learning and realigns classifiers post hoc to counter progressive overfitting. CO2L [29] leverages contrastive pretraining on clean data and preserves transferable representations through self-supervised distillation.

Other methods address multimodal or zero-shot continual learning: AttriCLIP [168] freezes the vision–language model and uses attribute-based prompts; PromptFusion [33] separates stability and adaptation with dual prompt-tuning branches; ZSCL [203] preserves zero-shot performance via feature distillation and parameter averaging; MoE-AD [191] adopts mixture-of-experts adapters with task routing and OOD detection to maintain generalization.

1.4.2 BENCHMARKS AND EVALUATION METRICS

The empirical evaluation of CL methods critically depends on both the choice of benchmarks and the metrics used to quantify learning dynamics over time. Unlike standard supervised learning, CL requires models to acquire new knowledge sequentially while retaining performance on previously learned tasks. As a consequence, evaluation protocols must explicitly account for task ordering, interference between tasks, and long-term knowledge retention. This chapter presents

the most commonly adopted benchmarks and evaluation metrics in the CL literature, with a particular focus on class-incremental image classification settings.

Most CL benchmarks for classification are constructed by partitioning a multi-class dataset into a sequence of T disjoint tasks, each containing a subset of classes. The model is trained sequentially on each task without access to data from previous tasks and is evaluated on all tasks seen so far, enabling a systematic analysis of catastrophic forgetting and knowledge transfer.

Natural Image Benchmarks

Seq. MNIST [86] is one of the earliest benchmarks adopted in continual learning, typically split into multiple tasks by grouping digit classes. Despite its simplicity and low visual complexity, it remains useful for rapid prototyping and controlled studies of forgetting mechanisms.

Seq. SVHN [121] provides a more challenging alternative by introducing real-world visual variability through street-view house numbers. In continual learning settings, it is commonly partitioned into class-incremental tasks and used to study robustness under mild distributional complexity.

Seq. CIFAR-10 and Seq. CIFAR-100 [83] are among the most widely adopted benchmarks in CL. Both datasets are commonly split into sequences of class-incremental tasks; in particular, Seq. CIFAR-100 enables the construction of long task sequences with few samples per class, making it well suited to stress-test scalability, memory management, and forgetting behavior.

Seq. miniImageNet [163] is frequently employed to evaluate continual learning methods on higher-resolution images and more diverse visual content. Its use is especially common when assessing deep architectures and representation reuse across tasks.

Seq. ImageNet-R [63] further increases evaluation difficulty by introducing significant domain shifts through artistic and non-photorealistic renditions of ImageNet classes, enabling the analysis of robustness under distributional changes in continual learning scenarios.

Fine-Grained Benchmarks

Seq. CUB-200-2011 [165] is a fine-grained bird classification dataset characterized by high inter-class similarity and subtle visual differences. In continual learning, it is commonly used to evaluate feature reuse and robustness to forgetting in challenging discrimination regimes.

Seq. Cars-196 [82] presents a similar level of difficulty in the automotive domain, with visually similar categories differentiated by fine-grained attributes. Experiments on both fine-grained benchmarks are typically conducted using ImageNet-pretrained backbones to ensure stable optimization and fair comparisons.

Remote Sensing and Medical Benchmarks

Seq. EuroSAT [62] is a remote sensing dataset for land use and land cover classification, commonly adapted to class-incremental continual learning to evaluate performance under varying spatial patterns and acquisition conditions.

Seq. RESISC45 [35] provides a larger and more diverse remote sensing benchmark, enabling the study of continual learning under increased scene variability and class diversity.

Seq. ISIC [37] is widely used in the medical imaging domain for skin lesion classification. In continual learning settings, it allows the evaluation of learning dynamics under limited data availability, class imbalance, and high annotation uncertainty.

Seq. ChestX-ray datasets [170] are employed to assess continual learning methods on large-scale medical data streams, where weak supervision and label noise are prevalent and retraining from scratch is often impractical.

Evaluation Metrics

For a principled evaluation, we consider access to a test set for each of the T tasks. After the model finishes learning about the task t, we evaluate its *test* performance on all T tasks. Let $a_t^{t'}$ denote the test performance on task t of the model, after it has been trained on all the tasks up to task t'. Using this notation, several metrics

have been proposed to characterize both final performance and learning dynamics.

Final Average Accuracy

The most commonly reported metric is the *FAA*, which measures the average performance across all tasks after completing the full training sequence:

$$\text{FAA} \triangleq \frac{1}{T}\sum_{t=1}^{T} a_t^T. \tag{1.5}$$

FAA provides a compact summary of the final model quality, but does not capture how performance evolves throughout training.

Backward and Forward Transfer

Backward Transfer (BWT) quantifies the influence of learning new tasks on previously learned ones:

$$\text{BWT} \triangleq \frac{1}{T-1}\sum_{t=1}^{T-1} \left(a_t^T - a_t^t\right). \tag{1.6}$$

In particular, it measures the influence of learning subsequent tasks on previously learned ones $t < T$. Positive BWT occurs when learning $t' > t$ improves performance on t, while negative BWT occurs when it reduces performance on t. Negative BWT values indicate catastrophic forgetting, while positive values suggest beneficial backward knowledge transfer.

Forward Transfer (FWT) measures how prior knowledge affects performance on future tasks before training on them:

$$\text{FWT} \triangleq \frac{1}{T-1}\sum_{t=2}^{T} \left(a_t^{t-1} - b_t\right), \tag{1.7}$$

where a_t^{t-1} is indeed the performance on task t after learning tasks up to $t-1$ and b_t denotes a baseline performance on task t, typically obtained from a randomly initialized model.

Forgetting Metrics

A direct measure of forgetting is obtained by comparing the maximum performance achieved on a task with its final performance. The *Final Average Forgetting (FF)* is defined as:

$$\mathrm{FF} \triangleq \frac{1}{T-1} \sum_{i=1}^{T-1} \left(\max_{t \in \{i,\dots,T\}} a_i^t - a_i^T \right). \tag{1.8}$$

FF explicitly captures the degree of knowledge loss induced by subsequent training. Normalized and adjusted variants of FF have also been proposed to improve comparability across tasks and datasets, especially when peak accuracies differ substantially.

Discussion

Several metrics have been proposed to characterize performance in Continual Learning, as no single measure can fully capture the underlying learning dynamics. In particular, complementary metrics such as Forgetting (FF) and Backward Transfer (BWT) are useful to analyze stability–plasticity trade-offs beyond final performance. Nonetheless, in this thesis we primarily rely on FAA as the main evaluation metric, in line with standard practice in the literature, and report Forgetting (FF) as a secondary measure to quantify knowledge degradation over time. Results are averaged across multiple task orderings and random seeds to ensure robustness.

1.5 Learning with Noisy Labels

Forgetting represents one of the central challenges in incremental training, significantly hindering the ability of modern AI systems to continuously assimilate new information in streaming data settings. Within the Continual Learning (CL) paradigm, most existing methods mitigate catastrophic forgetting through the replay of a limited memory buffer containing samples from past tasks. While effective under ideal conditions, these replay-based strategies become fragile in realistic scenarios, where data annotations are often affected by noise due to time constraints and limited human supervision. This issue motivates the study of Continual Learning under Noisy labels (CLN), a setting in which conventional buffer management and sample selection mechanisms may inadvertently reinforce corrupted information. In this chapter, we introduce Alternate Experience Replay (AER), a novel approach that explicitly exploits forgetting as a signal to disentangle clean, complex, and noisy samples within the memory buffer. The underlying intuition is that samples that poorly conform to the previously learned data distribution—either due to intrinsic difficulty or incorrect labeling—are more likely to be forgotten over time. Leveraging this observation, AER promotes a structured buffer organization that facilitates selective retention. To further enhance the effectiveness of this separation, we propose Asymmetric Balanced Sampling, a tailored sample selection strategy that emphasizes label purity for the current task while preserving informative and representative samples from earlier tasks. Through extensive empirical evaluations, we show that the proposed framework improves both predictive performance and buffer quality when compared to existing loss-based purification approaches, highlighting the advantages of exploiting forgetting dynamics in CLN scenarios.

LNL addresses supervised learning scenarios in which the observed annotations are corrupted versions of the unknown ground-truth labels. This setting naturally arises in large-scale data collection pipelines involving non-expert annotators, weak supervision, or web-scraped data [49, 151]. Despite their strong generalization capabilities, deep neural networks (DNNs) are particularly vulnerable to label noise due to their high capacity, which allows them to eventually fit cor-

1

rupted labels and degrade test performance [195]. This behavior has motivated extensive research on robust training strategies that mitigate memorization while preserving useful learning signals.

1.5.1 PROBLEM FORMULATION AND TAXONOMY OF LABEL NOISE

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote a clean dataset sampled from an unknown distribution $p(\mathbf{x}, y)$, where $y \in \{1, \dots, C\}$. Under label noise, the learner instead observes $\tilde{\mathcal{D}} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^N$, where $\tilde{y}$ is a corrupted version of y. Training typically proceeds via empirical risk minimization (ERM), which is known to be non-robust under label noise [120], as gradients computed from corrupted labels introduce systematic bias in parameter updates.

A common abstraction models label noise through a class-conditional transition matrix T, where $T_{ij} = \Pr(\tilde{y} = j \mid y = i)$ [126]. Under this instance-independent assumption, the corruption process is conditionally independent of the input features once the true label is given. Within this framework, two synthetic regimes are widely adopted in experimental evaluation. Symmetric (or uniform) noise assumes that any label may flip uniformly at random to any other class, i.e., $T_{ii} = 1-\tau$ and $T_{ij} = \tau/(c-1)$ for $i \neq j$, with global noise rate $\tau \in [0, 1]$. Conversely, asymmetric (or label-dependent) noise reflects more realistic, semantically structured confusions: mistakes are concentrated within particular class pairs, e.g., "cat" $\rightarrow$ "dog" is far more likely than "cat" $\rightarrow$ "car" [151]. A special case is pair-flip noise, where each class i can only be corrupted into a single target class j with probability τ, while remaining correct with probability $1 - \tau$.

Although this class-conditional setting enables theoretical tractability and facilitates strategies such as transition-matrix estimation or correction, it overlooks the variability in how errors occur in real annotation pipelines. More realistic scenarios are captured by instance-dependent noise, where the corruption probability additionally depends on the input features [184]. In this setting, mislabeling is governed by functions $\rho_{ij}(x) = \Pr(\tilde{y} = j \mid y = i, x)$, making visually ambiguous, low-quality, or atypical samples more prone to corruption than prototypi-

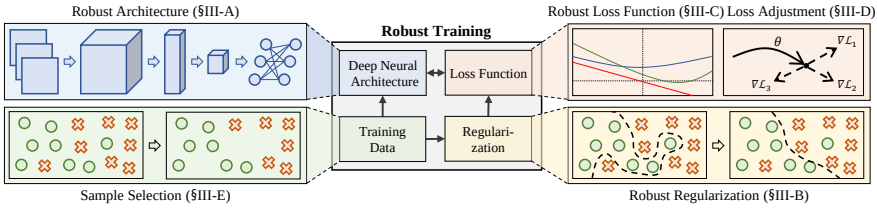


Figure 1.2: Figure borrowed from [151] illustrating the main methods for LNL.

cal ones. As a result, the effective transition behavior becomes input-specific and cannot be expressed by a single global matrix. This significantly complicates both modeling and learning: the noise structure is latent, data-dependent, and often entangled with the model's representation, while errors may concentrate on the hardest and most informative examples. Consequently, methods designed under the class-conditional assumption tend to degrade under instance-dependent corruption, motivating the development of approaches that identify unreliable samples, separate aleatoric uncertainty from label noise, or jointly recover the underlying clean structure during training.

1.5.2 FAMILIES OF ROBUST LEARNING METHODS

Existing approaches to LNL can be broadly grouped into five methodological families, depending on how robustness is incorporated into the learning process [151]. At a high level, these families intervene on different components of the standard supervised learning pipeline: the network architecture, the regularization scheme, the loss function, the per-sample loss contributions, and the effective training set. While conceptually distinct, they are often combined in practice (e.g., robust losses with strong data augmentation, or sample selection with semi-supervised learning), and several recent methods can be viewed as hybrids that lie at the intersection of these categories [57].

Robust Architecture

Robust-architecture methods explicitly model the label corruption process by augmenting the classifier with additional components, such as noise adaptation

layers that map clean label predictions to noisy label distributions [155, 52]. Under the common assumption of class-conditional noise, the adaptation layer implements a parametric surrogate of the transition matrix T, which is typically learned jointly with the base network and removed at test time. This family includes variants with different parameterizations and regularization strategies (e.g., fixed-size linear layers, EM-based estimation, or structured constraints) and can be extended to partially instance-dependent noise by conditioning the transition parameters on features [184]. Beyond simple adaptation layers, dedicated architectures have been proposed to better capture complex noise patterns, for example by introducing auxiliary networks that explicitly predict noise types or by constraining the transition structure via human priors or graph-based modules [56, 190]. These methods aim to decouple representation learning from noise modeling and provide a principled mechanism to correct predictions, but they typically rely on strong assumptions about the form of the corruption process, can be sensitive to transition-matrix misspecification, and may be hard to transfer across architectures and datasets.

Robust Regularization

Robust regularization strategies constrain the optimization process to prevent overfitting to noisy supervision. Both explicit regularizers and implicit mechanisms—such as early stopping and data augmentation—have been shown to reduce memorization of corrupted labels [97]. Explicit regularization approaches modify the training objective or updates, for example via weight decay, dropout, adversarial training, gradient clipping, or bilevel schemes that regularize parameters using a small clean validation set [73, 117]. Other works treat label noise as an additional source of uncertainty and regularize confusion across annotators or pre-train on large external corpora before fine-tuning on noisy data [156, 64]. Implicit regularization exploits the stochasticity of SGD, data augmentation, and label smoothing to bias learning toward simpler decision boundaries that fit clean patterns before fitting noise [129, 198]. Techniques such as mixup or related interpolation-based augmentations can be interpreted as enforcing local linearity in feature and label space, thereby smoothing out isolated mislabeled points. In practice, these

regularization mechanisms are rarely used in isolation: they are usually combined with other robustness techniques (e.g., robust losses or sample selection), and their effectiveness depends on architecture, optimization hyperparameters, and the severity and structure of the noise.

Robust Loss Functions

An alternative line of work focuses on replacing cross-entropy with loss functions that are inherently more robust to label noise. Examples include mean absolute error and its variants, generalized cross-entropy, symmetric cross-entropy, bi-tempered losses, and curriculum-inspired surrogates of the 0–1 loss [51, 201, 171, 10, 105]. Many of these constructions are motivated by risk-consistency analyses showing that, under suitable conditions on the noise process and the loss (e.g., symmetry conditions or boundedness), minimizing the noisy risk recovers the Bayes-optimal classifier for the clean distribution [51]. In particular, symmetric or bounded losses limit the influence of individual high-loss samples, thereby preventing mislabeled points from dominating the gradients; generalized or bi-tempered cross-entropy interpolates between the optimization-friendly behavior of standard cross-entropy and the noise-tolerance of MAE. While attractive due to their simplicity and compatibility with standard training pipelines, robust losses may exhibit sensitivity to the noise type, often require careful tuning of temperature or mixing hyperparameters, and can slow optimization or hurt performance on clean data if not properly calibrated.

Loss Adjustment and Label Refurbishment

Loss adjustment methods modify per-sample contributions based on estimates of label reliability, with the goal of making the optimization dynamics themselves robust to noise. Loss correction techniques explicitly correct the loss using an estimated transition matrix [126], either by re-mapping logits through a forward correction layer or by correcting the loss values via an inverse mapping (backward correction). Their robustness critically depends on the quality of the estimated transition matrix, which may require anchor points, held-out clean

data, or strong identifiability assumptions [64, 182]. Reweighting approaches down-weight likely noisy samples based on confidence or loss statistics [169, 30], or learn a parametric weighting function via meta-learning on a small clean validation set [139, 148]. Label refurbishment methods update training targets using model predictions, either through soft-label interpolation or confidence-based relabeling [138, 11]. Recent variants refine the basic bootstrapping idea using temporally smoothed predictions, mixture models over the loss distribution, or explicit identification of refurbishable examples that the model predicts consistently over time [106, 150, 34]. Meta-learning approaches further generalize this idea by treating the loss-adjustment rule itself (weights, corrected labels, or both) as a learnable component optimized to minimize an auxiliary objective on clean data [148, 202]. Overall, loss adjustment enables full exploitation of the training set but may accumulate error when correction or weighting is inaccurate, especially at high noise rates or under complex instance-dependent corruptions.

Sample Selection and Hybrid Methods

Sample selection methods aim to identify reliable subsets of the training data and optimize the model primarily on these samples, thereby avoiding explicit correction of potentially incorrect labels. They are typically motivated by the memorization dynamics of deep networks: DNNs tend to fit easy, correctly labeled examples first and only later memorize hard or mislabeled instances [12]. This leads to the widely used "small-loss" heuristic, where low-loss examples in each mini-batch are treated as clean and retained for training, while high-loss examples are discarded or deferred [58, 74]. Multi-network approaches such as co-teaching and its variants leverage agreement or disagreement between peers to mitigate confirmation bias, letting one network select small-loss samples for the other [58, 192]. However, aggressive selection may discard informative data and can fail when loss distributions of clean and noisy samples overlap strongly. To address this, iterative or multi-round procedures refine the clean set over time, and hybrid approaches combine selection with semi-supervised learning: selected examples are treated as labeled, and the remaining ones as unlabeled, enabling the model to exploit all data through consistency or pseudo-labeling objectives [150,

90, 122]. In practice, state-of-the-art robust training pipelines often sit in this hybrid regime, integrating sample selection, semi-supervised learning, and additional regularization or loss-adjustment mechanisms to balance exploration of noisy data with exploitation of high-confidence supervision.

1.5.3 EXPERIMENTAL PROTOCOLS AND COMMON PRACTICE

Evaluation protocols for LNL typically rely on benchmark datasets with synthetically injected noise or on real-world datasets with naturally corrupted labels [151]. Standard practice includes reporting performance on clean test sets, varying noise rates and types, averaging results across multiple runs, and carefully controlling training dynamics through warm-up phases and early stopping.

1.5.4 CONTINUAL LEARNING UNDER NOISE

Most Continual Learning (CL) formulations assume that the supervision provided by the data stream is reliable. In realistic deployments, however, labels may be corrupted by weak supervision, annotation shortcuts, or automatic harvesting, and the resulting errors are not i.i.d. over time: the stream can drift, tasks can overlap (blurry boundaries), and the label noise can interact with the rehearsal mechanism itself. In this setting, replay may amplify noise because mislabeled samples can be repeatedly revisited, while online updates performed on contaminated batches can rapidly destabilize previously acquired decision boundaries. Recent works that explicitly combine CL with noisy-label learning therefore focus on two coupled goals: (i) controlling catastrophic forgetting through memory-based or regularization-based mechanisms, and (ii) preventing the memory and the training signal from being dominated by corrupted labels through purification, robust objectives, or semi-supervised learning (SSL).

A representative replay-based approach is *Self-Purified Replay* (SPR) by [80]. The core observation is that standard supervised replay can be brittle when labels are unreliable, since the model receives contradictory gradients from mislabeled exemplars stored in memory. SPR addresses this by decoupling representation

consolidation from noisy supervision using a self-supervised replay component (*Self-Replay*), which leverages Self-Supervised Learning (SSL) signals on buffered (and recent) samples to mitigate forgetting without relying exclusively on the provided labels [80]. A second component (*Self-Centered filter*) aims to maintain a purified episodic memory by filtering samples using a centrality-based criterion constructed via stochastic graph ensembles, with the intent of removing atypical or suspicious points that are more likely to be mislabeled [80]. Conceptually, SPR treats buffer *purity* as a first-class requirement in CL: rather than assuming stored exemplars are trustworthy, it actively curates memory content and reduces the effect of noisy labels through self-supervised consolidation.

CNLL by [78] follows a related intuition—purify the stream before replay-like reuse—but emphasizes a lightweight pipeline suited to continual noisy-label scenarios. CNLL proposes a simple purification step to cleanse the incoming online data, and subsequently performs fine-tuning in a semi-supervised fashion so that all samples can still contribute: purified samples are treated as labeled, while the remaining data can be exploited through SSL objectives that do not require trusting their labels [78]. The resulting design reflects a practical trade-off that appears repeatedly in noisy-label learning: aggressively discarding suspected noisy samples can improve purity but reduces data coverage, whereas SSL-style utilization of uncertain samples can preserve representation learning signal while limiting the harm of incorrect supervision [78]. Within a CL perspective, CNLL can be read as an online stream-processing mechanism that continuously separates more reliable supervision from uncertain data and uses different learning signals accordingly, rather than applying a single supervised objective to all samples.

A complementary line is developed in *Online Continual Learning on a Contaminated Data Stream with Blurry Task Boundaries* [15]. This work explicitly considers a task-free, online regime where task identities are unavailable and distributions change gradually, which makes classical task-incremental assumptions less applicable [15]. The authors argue that, under label noise, episodic memory must satisfy two competing properties: *purity* (avoiding corrupted supervision) and *diversity* (covering the evolving data distribution so that replay remains representative). To balance these objectives, they propose a unified strategy that com-

bines label-noise-aware diverse sampling for memory management with robust learning based on semi-supervised learning to exploit both reliable and unreliable samples [15]. The emphasis on purity–diversity balance is particularly relevant to CL, because memory is simultaneously the mechanism used to prevent forgetting and a potential vector through which noise is repeatedly re-injected; the method therefore couples memory selection with a robust training rule rather than treating them as independent components [15].

Finally, CLTR (*Continual Learning Time-varying Regularization*) frames robustness to noisy labels through a regularization perspective that explicitly borrows from CL principles [95]. Instead of relying primarily on rehearsal, CLTR views LNL as a process that benefits from time-dependent control over parameter updates, motivated by the idea that early training may capture more reliable structure while later stages are more prone to fitting noise [95]. In this sense, CLTR uses a time-varying regularization mechanism to constrain learning dynamics so as to reduce sensitivity to corrupted labels, aiming to better control the direction and magnitude of updates when noise becomes dominant [95]. This perspective is aligned with a broader trend: when label reliability changes over time (either because the stream changes or because the model transitions from fitting simple patterns to memorizing noise), temporal scheduling of robustness constraints can be an effective alternative to purely loss-based filtering.

Overall, these works illustrate three recurring design patterns for CL under noisy supervision: (i) *purified replay*, where the buffer is actively curated and replay is supported by self-supervised signals to reduce dependence on noisy labels [80]; (ii) *stream purification with SSL utilization*, where reliable samples drive supervised updates and uncertain samples contribute through semi-supervised objectives [78, 15]; and (iii) *time-dependent robust constraints*, where regularization schedules inspired by CL are used to prevent late-stage memorization of noise [95]. These patterns motivate studying memory purity, selection dynamics, and training signals jointly, since in continual settings the mechanism used to preserve past knowledge can also amplify annotation noise if not explicitly controlled.

1.6 Model Merging and Task Arithmetic

Pre-trained models are widely adopted as backbones and are often modified after training to adapt to new tasks, correct undesired behaviors, or incorporate new information. Beyond standard fine-tuning, a growing line of work studies how models can be edited or combined directly in weight space. In this context, task arithmetic and model merging approaches represent a simple yet effective paradigm, where task-specific changes are encoded as weight differences with respect to a common pre-trained model and then manipulated through linear operations. These methods enable model composition, behavior removal, and knowledge transfer across tasks, often without additional training, and have recently gained attention as a practical alternative to more traditional adaptation strategies.

Task arithmetic

Standard and Parameter-Efficient (PEFT) fine-tuning have been shown to support addition/subtraction of task vectors. However, while the evidence in [200, 70] is primarily empirical, we derive theoretical insights about the pre-conditions for task arithmetic, emphasizing the importance of staying close to the pre-training basin. In this respect, our derivations ground previous findings regarding the efficacy of low learning rates [70, 124]. Remarkably, staying within the pre-training basin has also been proved beneficial in [144] for ensemble learning. The conditions for compositionality are also studied by [124] on *linearized* networks (Eq. 4.7). Albeit considering this work inspirational, we see the pros of task arithmetic in the non-linear regime. Firstly, non-linear models surpass their linearized counterparts in single-task accuracy, making them more attractive. To reduce the gap, linearization-aware fine-tuning has to be used [101], contrarily to our approach that is compatible with the prevalent fine-tuning techniques. Secondly, linearized inference requires the demanding Jacobian-vector product (three times slower than a forward pass).

Ensemble learning

While original model soups [179] combine multiple weights fine-tuned on the same dataset, we herein managed to unlock *running model soups* in cross-dataset incremental settings, with possible returns in terms of forward transfer [102]. The optimization of the whole deep ensemble is also discussed in [72] for standard ensembles, *i.e*. averaging the outputs of different models. Their derivation regards the decomposition of the ensemble loss into the strength of the individual learners and their diversity. However, [72] use this result to elucidate the shortcomings of jointly trained deep ensembles, whereas we leverage it to provide effective regularization for model soups in incremental scenarios. Several works [91, 92, 145] build an ensemble through a cumulative mean of intermediate checkpoints sampled along the training trajectory, with benefits in terms of generalization and preservation of zero-shot pre-training capabilities [203]. Differently, [76] maintain a population of models trained with varying configurations (*e.g*. data augmentations). They also gradually push each weight toward the population average, thus encouraging alignment across individual models.

Incremental Learning

In addition to the formulations discussed in Section 1.4, namely regularization-based, replay-based, and architectural approaches, which represent the most established paradigms in the literature, there is a growing body of work that explores methods based on the allocation of new modules [108, 6]. Among the latter, SEED [143] manages an ensemble of expert networks learned incrementally. SEED stores separate models and combines their outputs. addition, a recent trend capitalizes on prompt-tuning [172, 174, 149], devising a pool of learnable prompts. Notably, the extent to which these models support compositionality is investigated in [128]. [22] propose À-la-carte Prompt Tuning (APT), an attention mechanism that enables the creation of bespoke models by composing arbitrary prompts.

NTK-based Incremental learning.

The authors of Tangent Model Composition (TMC) [101] build on the foundational work of [124] to address incremental learning. They enforce task arithmetic across subsequent tasks through linearization-aware fine-tuning, which entails a first-order Taylor approximation of the output function around θ_0. In this context, each task of TMC is effectively equivalent to training a kernel predictor using the **Neural Tangent Kernel (NTK)** [71], defined as $k_{\text{NTK}}(x, x') = \nabla_\theta f(x; \theta_0)^\top \nabla_\theta f(x'; \theta_0)$. Notably, other recent works [100] have adopted the NTK framework to tackle incremental learning: *e.g.* TKIL [183] exploit the NTK formulation to align representations for current and past tasks.

2

May the Forgetting be With You: Alternate Replay for Learning with Noisy Labels

In this chapter, we address the problem of Continual Learning (CL) under noisy conditions, with a particular focus on rehearsal-based strategies. Among other strategies, one prominent one is to interleave examples from the current and old tasks (rehearsal). To do so, a small selection of past data is retained in a memory buffer [161, 32], as in Experience Replay (ER) [136, 141]. The underlying idea is straightforward: by interleaving past samples with new data, the model is constrained to preserve previously learned decision boundaries. However, when memory capacity is necessarily small, the effectiveness of replay strictly depends on the buffer content: naive retention strategies may store redundant or unrepresentative examples and, more critically, enforce noisy supervision. Several works have highlighted how low-capacity buffers amplify overfitting [162, 19], while more recent studies emphasize the detrimental impact of noisy annotations in CL [80, 15]. Indeed, noisy labels are an inescapable characteristic of continual settings, where data must be annotated on-the-fly under restricted tem-

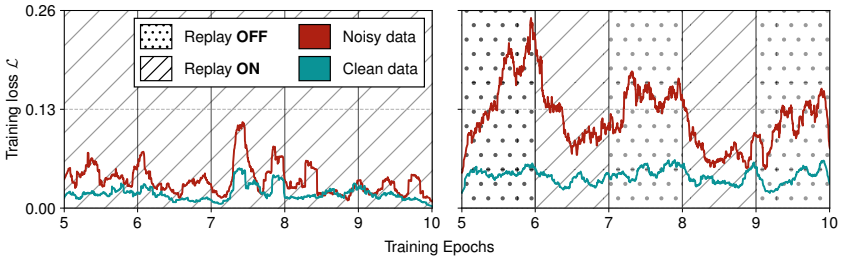


Figure 2.1: Training loss of clean and noisy during the second task of Seq. CIFAR-10 with 40% noise. The loss is computed on examples from the first task stored in the memory buffer. Standard replay makes the two indistinguishable (*left*) but alternating epochs of replay and forgetting maintain a significant loss separation (*right*).

poral constraints [185, 88, 94].

To cope with buffer contamination, existing approaches attempt to purify memory content by identifying clean samples, typically relying on the small-loss criterion and the memorization effect [12, 59, 74]. However, while effective in offline learning, such criteria become fragile in incremental settings: since learning does not restart from scratch but builds upon previously learned representations, adaptation is faster and the loss separation between clean and noisy samples quickly collapses [195, 12].

To address this limitation, we adopt a fundamentally different perspective and deliberately leverage forgetting as a signal rather than a phenomenon to suppress. Building upon the findings of [157, 107], which show that mislabeled examples are forgotten more rapidly than clean or informative ones, we exploit forgetting dynamics to restore loss separability within the memory buffer. Empirically, we observe that alternately disabling replay induces a sharp increase in loss for noisy samples, while clean examples remain stable, a gap that persists even when replay is reactivated due to faster re-adaptation on reliable data [12, 74, 175].

To address this limitation, we adopt a fundamentally different perspective: rather than treating forgetting solely as a problem to be mitigated, we deliberately leverage it as a signal to detect noisy or unreliable samples in the data stream. While prior approaches focus on suppressing forgetting, our method exploits it as a mechanism for identifying data that hinders stable learning. We build upon

the work of [157, 107], which theoretically demonstrates that mislabeled examples are quickly forgotten, whereas complex or rare instances tend to be retained for longer periods or may not be forgotten at all.

The insight driving this Chapter is that *forgetting must be exploited strategically* to restore the separability necessary for reliable noise discrimination.

2.1 Problem Setting

We define the Continual Learning framework as the process of learning from a sequential series of T tasks. During each task $t \in \{0, 1, \dots, T-1\}$, input samples $\mathbf{X_t}$ and their annotations $\mathbf{Y_t}$ are drawn from an i.i.d. distribution $\mathcal{D}_t$. We follow the well-established class-incremental scenario [161, 46, 25] in which $\mathbf{Y}_{t-1} \cap \mathbf{Y}_t = \emptyset$ and at task t the learner f_θ is required to distinguish between all observed classes. In this setting, we must simultaneously address the challenges posed by both noisy labels $\tilde{\mathbf{Y}}_t$ and the problem of forgetting. Therefore, for a given instance $\mathbf{x}_i \in \mathcal{D}_t$, we indicate with $\tilde{y}_i \sim \tilde{Y}_i$ the labels corrupted with annotation noise and with $\Pr(\tilde{y}_i \neq y_i)$ the respective *noise rate*. Ideally, we wish to minimize:

$$\theta^* = \arg\min_\theta \mathbb{E}_t \left[\mathop{\mathbb{E}}_{\mathcal{B} \sim \mathcal{D}_t} \left[\mathcal{L}(f_\theta(\mathbf{x}), \tilde{y}) \right] \right], \tag{2.1}$$

where $\mathcal{L}$ is the cross-entropy loss and $\mathcal{B} = (\mathbf{x}, \tilde{y})$. As in CL the objective above is inaccessible, we leverage a fixed-size buffer $\mathcal{M}$ to store and replay part of the incoming samples. As a result, the generalized objective for rehearsal CL can be defined as:

$$\theta^* = \arg\min_\theta \mathop{\mathbb{E}}_{\mathcal{B} \sim \mathcal{D}_t} \left[\mathcal{L}(f_\theta(\mathbf{x}), \tilde{y}) \right] + \mathcal{L}_R, \tag{2.2}$$

where the *replay regularization* term $\mathcal{L}_R$ depends on the choice of the replay-based method. Although our approach can be equally applied to advanced choices of $\mathcal{L}_R$ [25, 19, 65, 21] (see Section 2.6), in this work we build upon the simplest strategy and leverage Experience Replay [136, 141]:

$$\mathcal{L}_R = \mathop{\mathbb{E}}_{(\mathbf{x}_r, y_r) \sim \mathcal{M}} \left[\mathcal{L}(f_\theta(\mathbf{x}_r), \tilde{y}_r) \right]. \tag{2.3}$$

As the objective in Eq. 2.3 could result in bias accumulation toward the current task [7], we adopt the asymmetric cross-entropy loss introduced in [27].

2.2 Alternate Experience Replay (AER)

As mentioned, our main focus is on constructing a memory set $\mathcal{M}$ that is as clean and representative as possible. Since this objective involves distinguishing between noisy and clean examples when populating the memory set, our methodology seeks to maintain a significant gap between the losses of clean and noisy samples. To illustrate such a phenomenon, we depict the loss trend of clean and noisy samples in a memory buffer produced by a rehearsal baseline (ER-ACE [27]). In particular, Figure 2.1 (left) shows the loss value sampled during standard training; differently, in Figure 2.1 (right) we alternatively switch replay regularization on and off at each epoch. As can be seen, stopping replay has a distinct impact: while the loss value of clean samples remains low, it hugely increases for mislabeled ones. We remark that this gap holds even when replay regularization turns on, as the model easily adapts to clean samples and hence learns them faster [12, 74, 175]. This effect is exacerbated in the popular offline (*i.e.* multi-epoch) CL setting [137, 181, 25], where we might be forced to trade-off convergence on the current task to avoid overfitting the mislabeled samples [195, 12].

To counteract the vanishing effect of the small-loss criterion and encourage the separation between the losses of noisy and clean samples, our novel methodology named **Alternate Experience Replay** (**AER**) induces forgetting of buffer datapoints. We refer the reader to Algorithm 1 for a summary of the overall procedure. Specifically, we divide the training epochs for the current task into two categories: **buffer learning** and **buffer forgetting** epochs. The training process involves alternating between these two modes of learning.

- **Buffer learning.** In this regime, we train the model with standard replay (line 6) as in Eq. 2.2. Importantly, we do not modify the samples stored in the memory buffer $\mathcal{M}$ (no insertion or removal operations are performed).
- **Buffer forgetting.** In this case (line 8), we omit regularization on the

Algorithm 1 Overall procedure of AER with ABS

Input: stream data $\mathcal{D}_t$, buffer $\mathcal{M}$, training epochs E

for epoch in $1, \ldots, E$ **do**
  $\theta_{\text{CHK}} \leftarrow \theta_{\text{epoch}}$ ▷ save current parameters of $f_\theta(\cdot)$
  for batch $\mathcal{B} \sim \mathcal{D}_t$ **do**
    $p \leftarrow \text{normalize}(\{s(x); \forall (x, \tilde{y}) \in \mathcal{M}\})$ ▷ compute asymmetric scores (*selection*)
    if epoch in E_{on} **then**
      train on $\mathcal{B} \cup (\mathbf{x}, \tilde{y}) \sim \mathcal{M}$ ▷ buffer learning
    else
      train on $\mathcal{B} \sim \mathcal{D}_t$ ▷ buffer forgetting
      $\mathcal{R} \leftarrow \text{reservoir}(\{(\mathbf{x}, \tilde{y}) \in \mathcal{B} : \mathcal{L}(\mathbf{x}, \tilde{y}) < \mathcal{L}_\alpha\})$ ▷ sample insertion
      $\mathcal{M}[z \sim p] \leftarrow \mathcal{R}$ ▷ replace data sampled with p with stream data $\mathcal{R}$
  if epoch in E_{off} **then**
    $\theta_{\text{epoch}} \leftarrow \theta_{\text{CHK}}$ ▷ restore previous model checkpoint

memory buffer and focus the training exclusively on data from $\mathcal{D}_t$. By halting regularization and causing the subsequent forgetting of buffer datapoints, the loss of noisy examples is likely to increase more rapidly than that of clean ones [157, 107]. This, in turn, makes the small-loss criterion reliable once again (see Figure 2.1, right). On top of that, we update $\mathcal{M}$ (line 9) through a loss-based selection strategy during these epochs (see Section 2.3).

This way, at the end of each buffer forgetting epoch, we get a cleaner version of the memory buffer. However, cycling between buffer learning and forgetting could result in the buffer being under-optimized, as it is effectively exploited only during the former epochs. We avoid this through *model checkpointing*: specifically, at the start of each forgetting epoch, we save the parameters of the model f_θ (line 12) and restore them at the end of the same epoch (line 2). While this option results in the model being optimized for only half of the epochs, we prove in Section 2.5 that the trade-off significantly enhances the final accuracy of the model.

2.3 Asymmetric Balanced Sampling (ABS)

In this section, we outline the sampling strategy used to *insert* and *delete* examples into and from the memory buffer during each buffer forgetting epoch.

2.3.1 SAMPLE INSERTION

Given a batch of data $\mathcal{B}$ from the current task, the first step is to determine which examples should be included in the buffer. To encourage the inclusion of clean examples, we exploit the memorization effect and employ a simple criterion that involves applying a threshold to the loss value. Formally, let α denote the percentage of samples within the current batch that we intend to exclude from the insertion procedure, we compute:

$$\mathcal{R} = \{(\mathbf{x}, \tilde{y}) \in \mathcal{B} : \mathcal{L}(\mathbf{x}, \tilde{y}) < \mathcal{L}_\alpha\} \tag{2.4}$$

where $\mathcal{L}_\alpha$ is the loss value at the α-th percentile of the loss distribution over $\mathcal{B}$. For our experiments, we set α to 75, thus discarding the 75% of samples with the highest loss and treating the remaining 25% as candidates to be inserted in the buffer (lines 9-10 of Algorithm 1).

2.3.2 SAMPLE SELECTION

We approach the selection process by sampling from a probability distribution $p(\mathbf{x})$ defined over all exemplars $\forall \mathbf{x} \in \mathcal{M}$ in the buffer. To model such a distribution, as carried out by most methods [26, 14, 15], we leverage the score $s(x) = \mathcal{L}(x, y) \geq 0$ given by the loss function. It is noted that a valid distribution can be then obtained by normalizing these scores, such that $p(x) = \frac{s(x)}{Z}$ where $Z = \sum_{x \in \mathcal{M}} s(x)$. We refer to this strategy as **Loss-Aware Symmetric Sampling** (**LASS**). In LASS, examples with higher loss are more likely to be replaced, leading to a memory buffer that maintains greater **purity**.

However, we argue that a replacement criterion based on loss value like LASS could be detrimental in terms of **diversity**, as it discourages the retention of complex yet clean samples into the memory buffer. Intuitively, high loss values correspond to examples near the decision boundary between two or more classes [9, 26, 14]; 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 [26]. They showed that considering examples with high loss provides a simple yet effective criterion for modelling their importance during replay.

In light of this, we propose a novel replacement strategy called **Asymmetric Balanced Sampling** (**ABS**, see Figure 2.2) that looks for a compromise between two contrasting objectives. Namely, it aims to ensure the inclusion of both *ii)* high-loss (i.e., complex) samples from the past and *i)* small-loss (i.e., clean) samples from the present. To do so, it builds upon an **asymmetric score** (line 4) that is different depending on whether a given example in the memory buffer belongs

to a past task or the current one. 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).

Formally, for each $\mathbf{x} \in \mathcal{M}$, we use the score $s(x)$ in Eq. 2.5, *i.e.* is equal to the loss $\mathcal{L}(\mathbf{x}, \tilde{y})$ if the example comes from the current stream of data $\mathcal{D}_t$ (as in LASS). Conversely, to encourage diversity, the criterion is **reversed** for examples from past tasks $\mathcal{D}_{<t}$, with the score being equal to $-\mathcal{L}(\mathbf{x}, \tilde{y})$.

$$s(\mathbf{x}) = \begin{cases} \mathcal{L}(\mathbf{x}, \tilde{y}), & \text{if } (\mathbf{x}, \tilde{y}) \sim \mathcal{D}_t \\ -\mathcal{L}(\mathbf{x}, \tilde{y}), & \text{if } (\mathbf{x}, \tilde{y}) \sim \mathcal{D}_{<t} \end{cases} \tag{2.5}$$

By taking this approach, we accept that a few mislabeled examples from past tasks might remain in the memory buffer. However, given the joint effect of the insertion policy and the LASS-like side of the replacement criterion – both of which tend to favor purity among examples from the current task – we can be cautiously optimistic that the examples from past tasks are correctly annotated (see Section 2.6 for an empirical analysis) and derive that high-loss items from earlier tasks are more likely to be informative outliers rather than mislabeled noisy samples. Finally, to achieve a balanced representation of both current and previous tasks in the memory buffer, we decide whether to replace a sample from the current or previous task based on their relative sizes. Considering $|\mathcal{M}_{cur}|$ as the number of samples from the current task in $\mathcal{M}$, we define a Bernoulli distribution with probability $\frac{|\mathcal{M}_{cur}|}{|\mathcal{M}|}$, where a success corresponds to sampling from $\mathcal{M}_{cur}$.

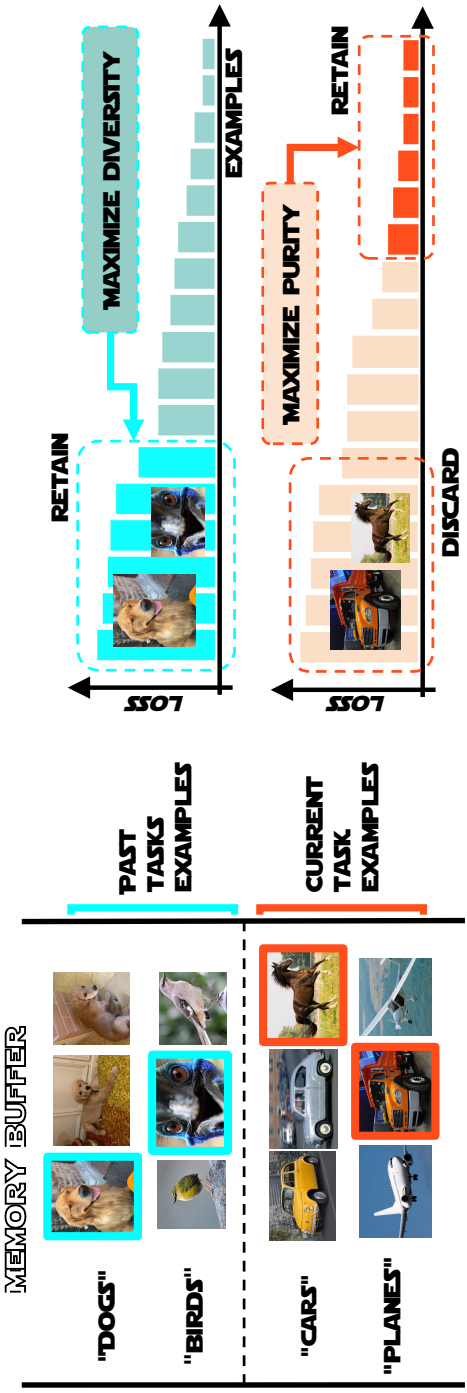


Figure 2.2: Asymmetric Balanced Sampling (ABS). Past examples are chosen to retain the most complex ones, while the criterion is reversed for the current task to maximize purity.

This approach ensures that the likelihood of replacing a sample is proportional to the current task's sample size within the entire memory buffer.

2.3.3 MORE DETAILS ON THE ABS PROCEDURE

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. 2.7, 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 2.3.2. On the other hand, based on our buffer insertion policy combined with AER – as mentioned in clause **Case b)** of Section 2.3.2 –, and by looking at the

results of Section 2.6 and our previous work [118], 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. 2.6):

$$\begin{aligned} p_{curr}(\mathbf{x}) &= \frac{s(\mathbf{x})}{z_{curr}} \quad \text{with } z_{curr} = \sum\nolimits_{\mathbf{x}\in\mathcal{M}_{curr}} s(\mathbf{x}) \\ p_{past}(\mathbf{x}) &= \frac{s(\mathbf{x})}{z_{past}} \quad \text{with } z_{past} = \sum\nolimits_{\mathbf{x}\in\mathcal{M}_{past}} s(\mathbf{x}) \end{aligned} \tag{2.6}$$

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

$$p(\mathbf{x}) = \varphi p_{curr}(\mathbf{x}) + (1-\varphi)p_{past}(\mathbf{x}). \tag{2.7}$$

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.

2.4 Buffer Consolidation

To further reduce label noise, the buffer $\mathcal{M}$ is consolidated at the end of each task by applying a semi-supervised refinement inspired by MixMatch [17]. While AER and ABS together balance sample purity and complexity preservation, loss-based filtering alone may discard informative yet hard examples, especially under high noise rates. Consolidation selectively leverages confident buffered samples as labeled supervision, whereas uncertain ones are treated as unlabeled. Let $u(x)$ denote the uncertainty of a buffered example: low-uncertainty samples contribute strongly with their (possibly noisy) labels, while uncertain ones rely more on model-driven guidance. Specifically, pseudo-labels are obtained by averag-

ing predictions over stochastic augmentations, and then combined with $\tilde{y}$ in an uncertainty-weighted manner:

$$\hat{y} = \frac{u(x)\tilde{y}}{1+u(x)} + \frac{1}{1+u(x)} \cdot \frac{1}{\eta} \sum_{i=1}^{\eta} f_\theta(\mathcal{A}_i(x)). \tag{2.8}$$

This dual correction mechanism mitigates brittle decision boundaries in presence of severe noise, enabling robust supervision extraction from partially corrupted buffers without discarding valuable information.

2.5 Experiments

Datasets and noise settings. We conduct experiments on five distinct datasets and various levels of noise. Specifically, we use the **Seq. CIFAR-100** dataset [83], which contains 32×32 images from 100 categories, split into 10 tasks, and the **Seq. NTU RGB+D** [147] dataset for 3D skeleton-based human action recognition, featuring 60 classes divided into 6 tasks. On these datasets, we inject two types of synthetic noise commonly employed in literature [89, 74, 59]: *symmetric* and *asymmetric* noise. In the first scenario, we replace the ground-truth label with probability $r \in [0, 1]$ determined by the designated noise rate. The asymmetric or class-dependent noise setting, instead, is an approximation of real-world corruption patterns, altering labels within the same superclass as in [126, 201]. To further address real-world label noise, we evaluate our method on **Seq. Food-101N** [88] (5 tasks), composed of images gathered from the web, thus containing *instance-level* annotation noise. Additionally, ResNet18 [60] is used for Seq. CIFAR-100 with 50 epochs per task, ResNet34 [60] for Food-101N with 20 epochs, and EfficientGCN-B0 [152] for Seq. NTU-60 with 30 epochs.

Benchmarking. In line with notable CL works [137, 65, 181, 14, 20, 116, 48], we adhere to a class-incremental and **multi-epoch** setting, in which samples can be experienced multiple times within the respective task. The results are presented in terms of FAA, computed at the end of the last task. All results are averaged across 5 runs.

Table 2.1: Final Average Accuracy (FAA) [↑] on multiple datasets and noise rates.
† Additional baselines adapted to the multi-epoch scenario.

Benchmark	**Seq. CIFAR-100**					**Seq. NTU-60**	
	symm			*asymm*		*symm*	
Noise rate	20	40	60	20	40	20	40
Joint	54.77±0.61	38.46±0.92	23.36±1.09	56.70±0.57	42.61±0.92	68.26±0.69	63.02±0.88
Finetune	08.65±0.13	07.55±0.14	06.15±0.17	07.78±0.14	05.73±0.09	14.30±0.51	11.73±1.07
ER [164]	25.14±0.28	14.64±0.23	8.92±0.23	29.42±0.39	18.91±0.86	29.95±2.16	16.02±0.27
+ *CoTeaching* [59]	25.79±0.61	14.46±0.49	8.92±0.30	32.18±2.55	20.76±2.44	43.87±0.78	30.71±1.86
+ *DivideMix* [89]	33.31±0.27	22.91±0.43	13.58±1.0.2	36.98±0.78	26.10±1.10	40.92±0.97	32.07±1.73
GDumb [132]	16.96±0.61	11.31±0.45	7.62±0.28	17.25±0.28	11.75±0.06	11.34±0.21	6.86±0.86
+ *CoTeaching* [59]	17.02±0.50	13.17±0.31	8.17±0.99	17.07±0.54	12.05±0.62	12.37±2.04	8.82±0.51
+ *DivideMix* [89]	19.26±0.97	15.67±0.97	10.51±0.32	18.80±1.55	13.29±0.29	15.96±1.16	7.49±1.11
PuriDivER [15]	27.53±0.53	24.36±0.40	17.81±0.43	25.46±1.44	18.84±0.64	39.33±1.59	38.86±0.79
PuriDivER.ME†	41.25±0.63	37.61±0.85	27.18±0.76	41.65±0.49	30.22±0.74	43.10±1.11	38.07±1.06
OURs	44.34±0.48	38.64±0.57	26.34±0.85	41.24±0.40	29.26±0.91	47.71±0.89	43.11±2.12
w. consolidation	**46.11**±1.46	**40.27**±0.40	**34.81**±1.63	**43.67**±0.73	**32.64**±0.48	**48.73**±1.20	**45.19**±0.05

We compare against PuriDivER [15], the current state-of-the-art selection strategy for CLN, as well as common rehearsal CL baselines. For the latter, we follow [15] and apply both CoTeaching [59] and DivideMix [89] to consolidate the buffer of ER [141, 136] and GDumb [132]. Since current CLN methods are designed for the online setting (*i.e.* a single training epoch is allowed), a direct comparison would be problematic: based on Section 2.2, we hence refine PuriDivER by suspending memory updates after the first epoch, naming such method as **PuriDivER.ME**. We also compare with SPR [80] and CNLL [78], adapted for offline CLN and with the same overall memory budget for fairness. Given the huge computational demands of SPR and CNLL, evaluating them on complex datasets like CIFAR-100 and NTU proved impractical: hence, we employ the smaller Seq. CIFAR-10 dataset (5 tasks).

Finally, the upper bound is attained by training jointly on all tasks (*Joint*), while the lower bound is attained by training without any countermeasure to forgetting or noise (*Finetune*).

Table 2.2: Comparison with SPR and CNLL. ‡training iterations spread across epochs.

Seq. CIFAR-10 – 40% *symm*			
Buffer size (total)		2500	*unlimited*
CNLL	*1 epoch*	38.14	57.26
CNLL	*50 epochs*	35.46	43.43
OURs	*50 epochs*	67.10	76.83
Buffer size (total)		1000	
SPR‡	*25 epochs*	26.34	
OURs	*25 epochs*	63.65	

Table 2.3: Ablation study for each component of our proposal – 60 % symmetric noise.

Seq. CIFAR-100 – 60% *symm*					
ER	w. ACE	α	AER	ABS	**FAA**
✓	✓				11.65
✓	✓	✓			19.97
✓	✓	✓	✓		24.19
✓	✓	✓		✓	21.68
✓	✓	✓	✓	✓	**26.34**
✓		✓	✓	✓	22.02

2.5.1 COMPARISON WITH STATE-OF-THE-ART

The results of our main evaluation are presented in Section 2.5. To streamline the discussion, we first compare our approach with traditional continual learning baselines, followed by an analysis of methods designed for continual learning under noisy labels (*e.g.* PuriDivER).

Comparison with rehearsal baselines. As outlined by Section 2.5, the approaches relying solely on buffer consolidation – such as ER and GDumb – are poorly effective, especially as noise levels rise. Regarding GDumb, its training phase is limited to the content of the memory buffer, preventing it from utilizing the data variety available throughout the task. This limitation is also evident from the comparison with standard ER, which consistently outperforms GDumb when noise levels are low. These outcomes highlight the benefits of performing multiple training iterations. However, this advantage turns into a double-edged sword as the stream becomes noisier, leading to a significant drop in performance.

Comparison with CNL methods. Firstly, we highlight the substantial improvement achieved by our adapted PuriDivER.ME, which outperforms PuriDivER by an average of 8.36%. Both versions perform buffer consolidation [15] at the end of each CL task; however, PuriDivER relies on a model trained over multiple epochs, which leads to the degradation of the small-loss criterion, an issue

Table 2.4: Performances [↑] of our method and main competitor on a real-world noisy dataset.

Benchmark	**Food-101N**
Joint	39.91±1.05
PuriDivER.ME	28.62±0.85
OURs	29.86±1.18
w. buffer fit.	**34.79**±0.64

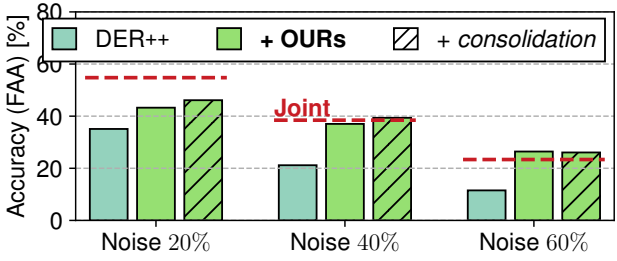


Figure 2.3: FAA ([↑]) of DER++ with our method and buffer fitting.

outlined in Section 2.2. Moreover, both PuriDivER.ME and ER + DivideMix are consistently surpassed by our proposal. In particular, we measure an average 1.50% gain over the best competitor's performance without any buffer consolidation, suggesting that our proposal improves the purity and diversity of samples in the buffer. However, as the sample selection is not perfect, applying an additional buffer consolidation technique tends to be more effective in more complex noise scenarios, with an average improvement of 4.71%.

We conduct additional comparisons with CNLL and SPR (Table 2.2) and PuriDivER.ME (Table 2.4). For the latter, we adopt the more realistic Food-101N dataset (*i.e.*, images collected from the web and automatically labeled). Even in these scenarios, our approach remains superior, both with and without buffer consolidation. We remark that these considerable gains come with a remarkable speed-up in terms of both time and resources used (see supplementary materials), making it more suitable for a multi-epoch incremental scenario.

Table 2.5: Comparison of Computational Cost [↓] of different methods when trained on CIFAR-10 with 40% noise.

Computational Cost	**PuriDivER**	**PuriDivER.ME**	**OURs**
Total Time (*hours*)	5h15m	2h50m	**1h30m**
Epoch Time (*seconds*)	76.90s	**15.69s**	18.56s
Task Time (*minutes*)	73m50s	19m39s	**16m33s**
Memory Used (*GB*)	7.77	7.50	**6.75**

Additional details on SPR and CNLL

In the main paper we provide a comparison between our proposal and SPR and CNLL. Nevertheless, as these methods were originally designed for the single-epoch setting, we had to design specific adjustments to make them viable for our scenario.

SPR initially stores samples in a *delayed buffer* – then splits into clean and noisy sets, with the former stored in a separate long-term buffer – and then optimizes the model for approximately 7,000 training iterations through a self-supervised (SSL) objective. This implies that SPR involves approximately 448× iterations than standard training*, making it unfeasible for our scenario due to time constraints. Indeed, while on CIFAR-10 our method takes around 16 minutes to complete 1 task (Table 2.5), SPR would require over 119 hours. We thus opt to distribute the training iterations of SPR across 25 epochs (see Table 2.2). Finally, as SPR employs two distinct memory buffers, we set the buffer size to 1000 for a fair comparison.

CNLL uses variable-length buffers to store confident clean and noisy samples, which implies a CL setting with unrestricted memory across tasks. To ensure fairness in comparison, we adhere to the well-established memory-budgeted CL [32, 161, 25] setting. Thus, for CNLL we allocate a total memory budget of 2,500 exemplars across all 5 buffers specified by the original method.

As we move from the single to multi-epoch setting, we find a reduced effective-

*assuming a buffer size of 500, batch size of 32, and 10,000 samples

ness of the regularization of CNLL; such a result is in line with our hypothesis of Section 2.2: as more epochs are allowed to learn the current task, sample selection based on the small-loss criterion fails to distinguish clean and noisy samples. Moreover, we find that such an outcome is maintained even in an unrestricted setting, where the memory budget is not a concern.

Finally, the performance gap w.r.t. our proposal is even more pronounced for SPR‡, where our method attains significantly higher accuracy in considerably less time; indeed, our reduced version of SPR requires around $109\times$ more time than our proposal, in line with our estimation.

2.5.2 TRAINING DETAILS

To evaluate our proposal we build upon the open-source codebase provided by `Mammoth` [25, 27, 20], a CL framework based on `PyTorch`.

On the choice of datasets and noise

We empirically validate our method on four different classification benchmarks as mentioned in the main paper. For experiments on CIFAR-10/100 [83] and NTU-60 [147], we corrupt the labels of the datasets at hand to obtain different noise configurations, which we then keep fixed for each of the experiments for fairness of results comparison across multiple methods.

In the process of injecting symmetric noise, we replace the ground-truth label with probability $r \in [0, 1]$ determined by the designated noise rate. The asymmetric or *class-dependent* noise setting is an approximation of real-world corruption patterns, which alters labels within the same superclass. For example, in the CIFAR-100 dataset, each image comes with a "fine" label (specific class) and a "coarse" label (superclass). Here, label transitions are parameterized by r such that the wrong class and true class have probability r and $1 - r$, respectively. This results in sample ambiguity occurring only between similar classes, as it would in a realistic scenario.

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. We

obtain the following versions of the datasets.

Seq. CIFAR-10 The original dataset contains 50, 000 train and 10, 000 test low-resolution color images in 10 different classes. During training the model encounters 2 classes per task, namely ("airplane", "car"), ("bird", "cat"), ("deer", "dog"), ("frog", "horse"), ("ship", "truck").

Seq. CIFAR-100 This original dataset is like the CIFAR-10, except it has 100 classes with 600 images each. Images are grouped into 20 superclasses, thus each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs). Following this categorization, we organize classes in 10 tasks, each containing 5 classes from the same superclass.

Seq. NTU-60 It comprises 60 action classes with 56,880 video samples, including 3D skeletal data (25 body joints per frame), all captured simultaneously using three Kinect V2 cameras. We here split the dataset into 6 tasks of 10 classes each.

Seq. Food-101N The dataset is composed of 101 web-crawled food images, split into 5 tasks (the first 4 containing 20 classes and the latter containing the remaining 21). Each class is relatively balanced, with an average of around 523 images per class and a standard deviation of around 11 (totaling 52867 images resized to 224×224). The dataset contains instance-level noise, thus simulating a real-world scenario.

Notice that since some labels are incorrect, real class distribution for each task might vary. Details on the noisy labels injected on Seq. CIFAR-10/100, Seq. NTU-60 are released with the code.

Architecture We use ResNet [60] family as a backbone for all the methods involved in our evaluation. ResNet18 is used for CIFAR-10/100 and ResNet34 is used for Food-101N, as in [15]. All the experiments do not feature pretraining.

Augmentation We apply random crops and horizontal flips to both stream and buffer examples, for each dataset at hand. For the implementations of PuriDivER, we use AutoAugment [39] as in the original paper [15].

Training We deliberately hold batch size out of the hyperparameter space and keep it fixed to 32 for both stream and buffer examples. For each task, we train for 50 epochs for CIFAR-10/100, and 20 for Food-101N.

Buffer consolidation with MixMatch At the end of each task, we finetune

the model on the buffer examples only, for 255 epochs. During this stage, we use SGD with Warm Restart (SGDR) through Cosine Annealing and a batch size of 64. For the purpose of label co-refinement, we set the number of different augmentations η of Eq. 2.10 to perform on the samples in the *uncertain* set to 3.

Hyperparameters We choose to use different buffer sizes relying on the dataset length. For experiments conducted on CIFAR-10 and CIFAR-100, the buffer size is set to 500 and 2000, respectively. We set the buffer size to 500 for experiments on NTU. Finally, we use a buffer size of 2000 for Food-101.

Table 2.6: Hyperparameters used for CIFAR-10 under symmetric label noise.

CIFAR-10	**20%**	**40%**	**60%**
Joint / SGD	$lr = 0.03$	$lr = 0.03$	$lr = 0.03$
OURs	$lr = 0.03$	$lr = 0.03$	$lr = 0.03$
+ consolidation	$lr = 0.03, lr_c = 0.1, \lambda_u = 0.01$	$lr = 0.03, lr_c = 0.1, \lambda_u = 0.01$	$lr = 0.03, lr_c = 0.1, \lambda_u = 0.01$
ER	$lr = 0.1, lr_b = 0.05$	$lr = 0.1, lr_b = 0.05$	$lr = 0.1, lr_b = 0.1$
+ CoTeaching	$lr = 0.1, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$	$lr = 0.1, lr_b = 0.05$
+ DivideMix	$lr = 0.1, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$	$lr = 0.1, lr_b = 0.05$
PuriDivER	$lr = 0.001, lr_b = 0.05, \alpha = 0.1$	$lr = 0.001, lr_b = 0.05, \alpha = 0.1$	$lr = 0.001, lr_b = 0.05, \alpha = 0.1$
PuriDivER.ME	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$
GDumb	$lr_b = 0.1$	$lr_b = 0.03$	$lr_b = 0.03$
+ CoTeaching	$lr_b = 0.01$	$lr_b = 0.03$	$lr_b = 0.03$
+ DivideMix	$lr_b = 0.03$	$lr_b = 0.03$	$lr_b = 0.01$

Table 2.7: Hyperparameters used for CIFAR-100 under symmetric label noise.

CIFAR-100	**20%**	**40%**	**60%**
Joint/SGD	$lr = 0.03$	$lr = 0.03$	$lr = 0.03$
OURs	$lr = 0.03$	$lr = 0.03$	$lr = 0.03$
+ consolidation	$lr = 0.03, lr_c = 0.05, \lambda_u = 0.01$	$lr = 0.03, lr_c = 0.1, \lambda_u = 0.1$	$lr = 0.03, lr_c = 0.1, \lambda_u = 0.1$
DividERMix	$lr = 0.03$	—	—
ER	$lr = 0.03, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$
+ CoTeaching	$lr = 0.03, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$
+ DivideMix	$lr = 0.1, lr_b = 0.01$	$lr = 0.1, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$
PuriDivER	$lr = 0.001, lr_b = 0.05, \alpha = 0.1$	$lr = 0.001, lr_b = 0.05, \alpha = 0.1$	$lr = 0.001, lr_b = 0.05, \alpha = 0.1$
PuriDivER.ME	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$
GDumb	$lr_b = 0.05$	$lr_b = 0.05$	$lr_b = 0.05$
+ CoTeaching	$lr_b = 0.05$	$lr_b = 0.05$	$lr_b = 0.05$
+ DivideMix	$lr_b = 0.05$	$lr_b = 0.05$	$lr_b = 0.05$

We select the other hyperparameters by performing a grid search and using the FAA as the selection criterion for the best parameters. In Tables 2.6 to 2.9 we report the best values for each model, categorized by dataset and noise type.

Table 2.8: Hyperparameters used for CIFAR-100 under asymmetric label noise.

CIFAR-100	**20%**	**40%**
Joint/SGD	$lr = 0.03$	$lr = 0.03$
OURs	$lr = 0.03$	$lr = 0.03$
+ buffer fit	$lr = 0.03, lr_b = 0.05$	—
+ consolidation	$lr = 0.03, lr_c = 0.05, \lambda_u = 0.005$	$lr = 0.1, lr_c = 0.05, \lambda_u = 0.1$
ER	$lr = 0.03, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$
+ CoTeaching	$lr = 0.03, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$
+ DivideMix	$lr = 0.03, lr_b = 0.01$	$lr = 0.1, lr_b = 0.01$
PuriDivER	$lr = 0.001, lr_b = 0.05, \alpha = 0.1$	$lr = 0.001, lr_b = 0.05, \alpha = 0.1$
PuriDivER.ME	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$
GDumb	$lr_b = 0.05$	$lr_b = 0.05$
+ CoTeaching	$lr_b = 0.05$	$lr_b = 0.05$
+ DivideMix	$lr_b = 0.1$	$lr_b = 0.1$

Table 2.9: Hyperparameters used for NTU RGB-D under symmetric label noise.

NTU RGB-D	**20%**	**40%**
Joint	$lr = 0.1$	$lr = 0.1$
SGD	$lr = 0.1$	$lr = 0.03$
OURs	$lr = 0.1$	$lr = 0.1$
+ consolidation	$lr = 0.1, lr_b = 0.1, \lambda_r = 0.01$	$lr = 0.1, lr_b = 0.1, \lambda_r = 0.01$
DividERMix	$lr = 0.03$	—
ER	$lr = 0.03, lr_b = 0.05$	$lr = 0.03, lr_b = 0.05$
+ CoTeaching	$lr = 0.1, lr_b = 0.05$	$lr = 0.1, lr_b = 0.05$
+ DivideMix	$lr = 0.1, lr_b = 0.05$	$lr = 0.1, lr_b = 0.05$
PuriDivER	$lr = 0.3, lr_b = 0.05, \alpha = 0.1$	$lr = 0.3, lr_b = 0.05, \alpha = 0.1$
PuriDivER.ME	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$	$lr = 0.03, lr_b = 0.05, \alpha = 0.1$
GDumb	$lr_b = 0.03$	$lr_b = 0.1$
+ CoTeaching	$lr_b = 0.1$	$lr_b = 0.1$
+ DivideMix	$lr_b = 0.3$	$lr_b = 0.03$

2.5.3 RESULTS IN TERMS OF FINAL FORGETTING

We repeat each of the experiments five times. We report in Section 2.5 of the main paper the Final Average Accuracy for all the experiments, with standard error values.

We also provide the final forgetting measure in Eq. 2.9 for all methods of the main comparison in Table 2.10.

$$FF \triangleq \frac{1}{T-1}\sum_{j=0}^{T-2} f_j, \text{s.t.} f_j = \max_{t\in 0,\dots,T-2} a_j^t - a_j^{T-1} \tag{2.9}$$

Table 2.10: Final Forgetting (FF) [↓] of CNL methods on our selection of bencharks. † Additional baselines created by adapting existing loss-based and CL approaches to the multi-epoch scenario.

Benchmark	**Seq. CIFAR-100**					**Seq. NTU-60**	
	symm			*asymm*		*symm*	
Noise rate	20	40	60	20	40	20	40
Joint	✗	✗	✗	✗	✗	✗	✗
Finetune	81.52	71.51	57.96	73.91	54.71	85.09	73.46
Reservoir	55.88	55.79	44.90	43.48	35.22	54.53	61.33
+ *CoTeaching*	55.20	54.95	36.07	54.77	26.03	34.30	29.03
+ *DivideMix*	22.33	26.73	20.94	23.45	16.73	18.45	18.70
PuriDivER	20.52	18.21	14.77	22.51	17.26	41.29	34.25
PuriDivER.ME†	24.34	25.06	26.83	25.40	21.82	25.76	18.41
OURs	22.89	21.26	22.13	21.19	16.90	12.94	14.05
w. consolidation	19.03	11.67	12.02	20.15	9.28	8.54	0.29

where a_j^t is the accuracy of the model on the j^{th} task after training on t tasks. These additional results depict a **lower** degree of **forgetting** of our proposal w.r.t. the baselines.

When paired with Section 2.5 of the manuscript, such evidence shows higher overall effectiveness in learning from a noisy source of data, allowing more stable convergence on the current task and lower losses due to forgetting.

2.6 Model analysis

2.6.1 ABLATIVE STUDIES

We herein aim to investigate the impact of each component. Starting from the base rehearsal method used in our research, *i.e.* ER-ACE [27], we gradually introduce our two main contributions, AER and ABS, one at a time. As seen from the results in Table 2.3, each additional feature produces an increase in performance on Seq. CIFAR-100. For an in-depth analysis of the effects of the asymmetric cross-entropy loss function (ACE), we compare against the standard cross-

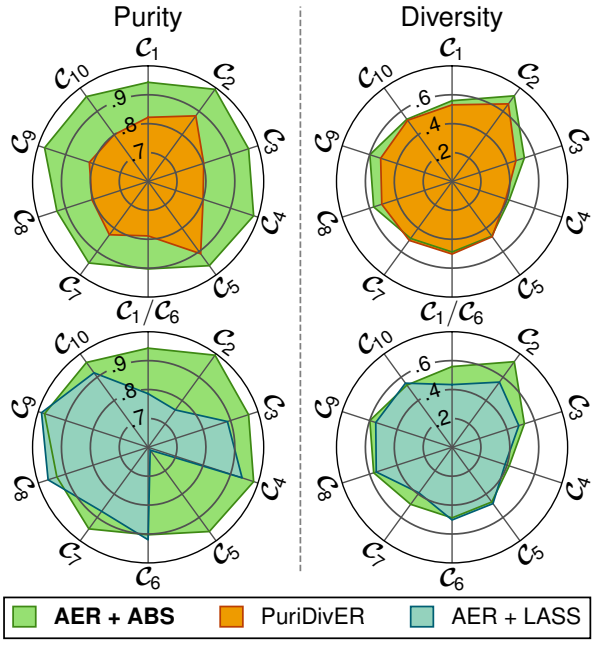


Figure 2.4: Final buffer composition at the end of training. Radar plots show purity (left) and diversity (right) across CIFAR-10 classes ($C_1 - C_{10}$) for three different methods.

entropy (*i.e.* ER in Table 2.3). The results indicate that the contribution of ACE is significant, aligning with both [27] and our initial expectations.

Purity of the buffer

Considering Seq. CIFAR-10 (40% noise), Figure 2.4 depicts the *purity* and the *diversity* of the buffer produced by ABS, PuriDivER.ME, and LASS. For each class, purity is defined as the percentage of examples labeled correctly for that class within the memory buffer. Instead, we model diversity as the intra-class variation within each class, thereby computing the average std. deviation of the features produced by the *Joint* ideal model. Finally, we scale all metrics according to their occurrence rate to account for potential imbalances in the number of examples from different classes. As shown in Figure 2.4, ABS clearly outperforms both LASS and PuriDivER.ME in terms of purity and diversity. Unexpectedly,

LASS yields a particularly unbalanced buffer, with only the most recent classes showing a good balance. In contrast, PuriDivER.ME achieves better balance but falls short in terms of purity.

Applicability to other methods

To evaluate whether AER/ABS can enhance other rehearsal methods, we apply them on **DER**++ [25] and conduct tests on Seq. CIFAR-100. We also report the results with and without the consolidation phase (Section 2.5.1). The gains shown in Figure 2.3 support the validity of our AER/ABS on enhancing other CL baselines.

2.6.2 ADDITIONAL RESULTS

In this Section, we provide: *i)* details about the experimental settings, the adapted baselines, noise injection process and hyperparameters, *ii)* the evaluation of Final Forgetting (FF), *iii)* an analysis of the computational costs, *iv)* an evaluation of the speed at which the model learns the noisy data, *v)* a sensitivity analysis conducted on the hyperparameter α, which controls the purity within the sample insertion strategy.

On the effectiveness of buffer consolidation

By combining AER with ABS we obtain a balance between purity – for samples of the current task – while preserving the complexity of those from the past. To achieve this, the backbone network had to be trained on a stream of noisy data. While we find that the effect of noise from the current task is mitigated by AER (Section 2.6.2), we can further reduce its influence with the help of the memory buffer.

In principle, with an ideal sample selection strategy we could simply train on samples from $\mathcal{M}$ to adjust the predictions of the network at the end of the task in a fully-supervised fashion (**buffer fit.**). While we empirically find in Section 2.5 that such a strategy delivers remarkable results, we can refine it to handle more complex noise scenarios.

In particular, we use a modified version of MixMatch [17] to obtain a more robust model, using the most *uncertain* samples as a source for unlabeled data. Similarly to [11], we fit a two-component Gaussian Mixture Model (GMM) $g(\mathcal{L})$ on the loss $\mathcal{L}$ of each $(\mathbf{x}, \tilde{y}) \in \mathcal{M}$. Then, we compute the perceived uncertainty of each sample $u(\mathbf{x})$ as the posterior $g(l|\mathcal{L})$, where l indicates the Gaussian component with the smaller mean. Samples are then separated into *pure* $\mathcal{P}$ and *uncertain* $\mathcal{U}$ with a simple threshold on $g(l|\mathcal{L})$.

From this, samples in $\mathcal{P}$ have label $\tilde{y} \approx y$, thus we can use them to compute a supervised loss term. Instead, for $\mathbf{x} \in \mathcal{U}$ we compute $\hat{y}$ using the model's response on different augmentations T of $\mathbf{x}$:

$$\hat{y} = u(\mathbf{x})\tilde{y} + \frac{1 - u(\mathbf{x})}{\eta} \sum_{i=1}^{\eta} f_\theta(T(\mathbf{x})), \tag{2.10}$$

Finally, we obtain the refined set $\mathcal{Q} = \{(\mathbf{x}, \hat{y}) : (\mathbf{x}, \tilde{y}) \in \mathcal{U}\}$ and follow up with the MixMatch procedure to compute the supervised and self-supervised loss terms $\mathcal{L}_s$ and $\mathcal{L}_u$ respectively. The overall loss term is computed as $\mathcal{L}_s + \lambda_u \mathcal{L}_u$, where λ_u is a regularization hyperparameter.

On the influence of the hyperparameter α

We want to carry out a sensitivity analysis targeting the value of α. Recall that alpha controls the proportion of samples to be discarded from the insertion phase within the buffer. We here report the results yielded by several α values under three different noise settings (asymm. 40%, symm. 40%, symm. 60%). The experiments, reported in Table 2.11, are conducted on Split CIFAR-100, with performance measured in terms of Final Average Accuracy. It can be concluded that $\alpha >= 50\%$ is a good choice, with gains that stabilize around $60\% - 90\%$. In our experiments, we remark that we avoided tuning α and set the same value for every dataset/noise ratio/noise type.

Table 2.11: FAA [↑] on CIFAR100 with varying noise to assess the influence of α

CIFAR-100	Parameter α					
	0%	**25%**	**50%**	**60%**	**75%**	**90%**
Asym 40%	24.76	26.69	28.76	29.70	29.26	29.90
Sym 40%	27.01	31.99	36.92	38.52	38.64	39.40
Sym 60%	15.30	17.35	20.74	24.61	26.34	30.67

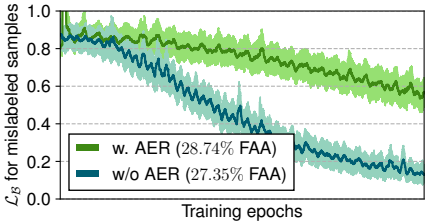


Figure 2.5: Effect of AER on the speed at which the model learns the noisy data

On the effectiveness of AER as a regularizer for CNL

Here, we further provide evidence of the impact of AER on the overall performance of the model. In particular, in Figure 2.5 we depict the final accuracy (FAA) and the loss of the noisy samples from the current task of ER-ACE with and without AER during the second task of Split CIFAR-10.

Surprisingly, we find that AER vastly reduces the rate of convergence of noisy samples, which just by itself improves over the baseline in terms of FAA. Indeed, in rehearsal CNL providing a purified and diverse set of examples to counter forgetting is only part of the challenge: as the model is subjected to a continuous stream of noisy data from the current task, an important effect is to reduce the speed with which noisy samples from the present are learned.

2.7 Conclusion

This chapter presents an innovative framework for Continual Learning in the presence of Noisy Labels, a common issue in real-world AI applications. We fo-

cus on the multi-epoch class-incremental scenario, arguing the shortcomings of current methods leveraging the small-loss criterion. We hence appeal to a long-standing enemy of continual learning – *forgetting* – and propose Alternate Experience Replay to maintain a clear separation between mislabeled and clean samples. Additionally, we introduce Asymmetric Balanced Sampling to enhance sample diversity and purity within the buffer. We demonstrate the merits of our approach through extensive experiments, showcasing its potential in noisy incremental scenarios.

3

EARL: Embracing Amnesic Replay for Learning with Noisy Labels

This chapter introduces EARL (*Embracing Amnesic Replay for Learning with Noisy Labels*), a direct evolution of the Alternate Experience Replay (AER) framework described previously. We expand on previous research in [118] 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 [11, 107], 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 (Figure 3.1 – *left*). Nevertheless, we warn against the limitations of such an experimental setup, which cannot fit tasks that demand multiple passes to achieve satisfactory performance [25] (Figure 3.1 – *right*), or those characterized by immense amounts of data (*e.g.,* training large language models). For this reason, we herein investigates the problem of LNL from the perspective of offline Continual Learning.

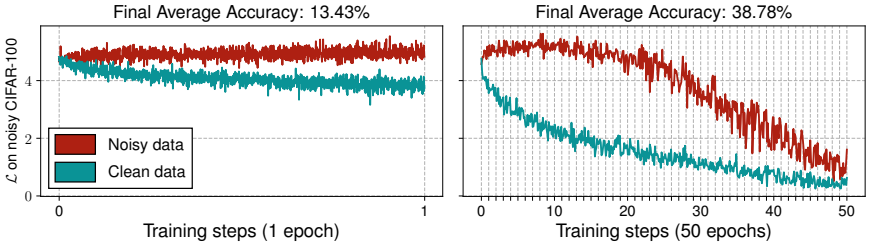


Figure 3.1: Accuracy and loss trend for clean (*blue*) and noisy (*red*) samples on CIFAR-100 with 40% symmetric noise. The *left* shows that training for 1 epoch leads to underfitting with 13.4% accuracy. Training for 50 epochs (*right*) improves convergence, accuracy and reduces the loss gap.

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 LNL. Notably, EARL can be applied to all the rehearsal-based techniques evaluated here and to 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 processes, e.g., the collection of data by crawling the web. To mimicking automatic labelling pipelines, we conduct an experiment where the data are automatically annotated through the prediction of the zero-shot CLIP [134]. Finally, unlike the current state-of-the-art, our analysis also covers the continual fine-tuning of pre-trained models, a prominent trend in Artificial Intelligence (AI).

In summary, building upon the previous work presented in Chapter 2, 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.

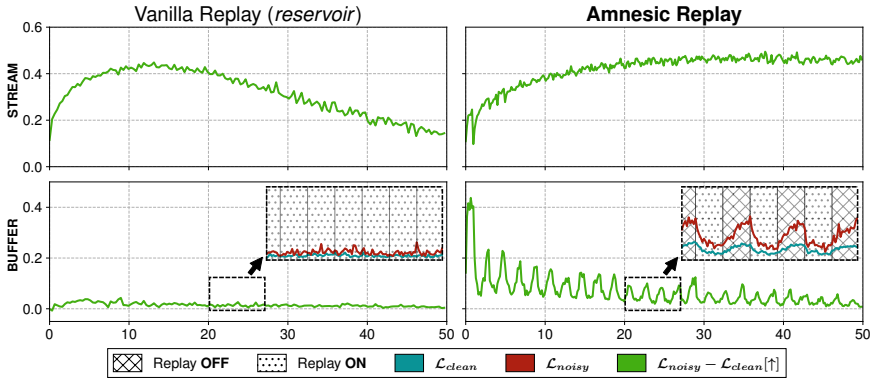


Figure 3.2: Trend of the loss difference between clean and noisy data among training epochs, for Vanilla Replay (*reservoir*) and Amnesic Replay *(ours)*, with forgetting induced every other epoch.

We believe that the contributions above enhance both this approach and our previous work [118], providing a more in-depth analysis, refined methodology, and broader experiments that may further progress the CL community.

3.1 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 [107], as they constitute a key foundation for justifying the rationale behind our work.

In Section 5 of their work [107], 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 $\mathcal{S}_{\mathcal{A}}$ (with noisy labels) for E epochs until 100% accuracy results in

weights $\mathbf{w}_A(e)$ that are close to the max-margin separator $\hat{\mathbf{w}}_A$ of $\mathcal{S}_A$:

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

where $\rho_A(E)$ is a small residual. This reflects the implicit max-margin dynamics of gradient descent [153], 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 $w_B(0) = w_A(E)$ and further trained on a clean set $\mathcal{S}_B$ for $E' = f(E)$ epochs. Since $\mathcal{S}_B$ 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 $\mathcal{S}_A$ 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 [107] (Intermediate-Time Forgetting) provides a high-probability guarantee that the following holds:

- Noisy samples from $\mathcal{S}_A$ are forgotten (their predictions flip to the correct label)
- Clean and rare examples from $\mathcal{S}_A$ are retained (their predictions remain correct).

The exact probability and the time E' depends on various factors, including class separability, model overparameterization, and the data's signal-to-noise ratio. Nonetheless, to quantify the aforementioned forgetting epoch E' and formalize what "rapid forgetting" means across different architectures, we indeed leverage the **second-split forgetting** metric. Following [107]'s protocol, we split the training set into two partitions $(\mathcal{S}_A, \mathcal{S}_B)$: the model is first trained on $\mathcal{S}_A$, then forgetting statistics are computed during fine-tuning on the second split $\mathcal{S}_B$. An example is said

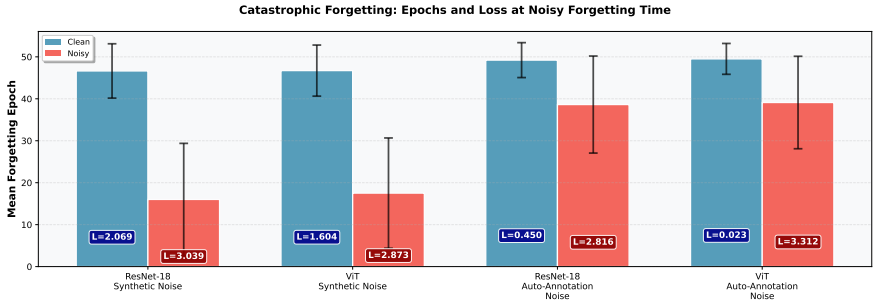


Figure 3.3: Forgetting Time Comparison across different backbones and noise types.

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 $\mathcal{S}_A$ is no longer classified correctly (forgotten) as the network is fine-tuned on a randomly held-out partition of the dataset $\mathcal{S}_B$. 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 Figure 3.3 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 ($\sim$16th epoch), while a more complex source of noise makes the forgetting time E' higher ($\sim$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 faster occurrence of natural forgetting for noisy samples justifies our approach of actively inducing forgetting events.

3.2 Experiments

For the reasons discussed above, we follow the established offline ClassIL scenario [137, 25, 14, 20], where multiple training epochs are allowed per task. We also use a task-aware approach to allow comparison with other works.

3.2.1 SETTING

Datasets. To comply with the current CNL literature, we start our evaluation on convolutional models with no pre-training (Section 3.2.3). We test on the **CIFAR-100** dataset [83] with *synthetic uniform noise*, where each label is randomly flipped to another class. We also consider **CIFAR-100N** [176]], 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 [134] in a zero-shot setup, which we name **CIFAR-100C** and simulates *automatic labeling pipelines*. In addition, to cover most sources of noise, we include **ANIMAL-10N** [150] and **FOOD-101N** [88], 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** [38, 158] and **EuroSAT-RGB** [61, 62], as these two datasets hold low domain similarity [123] w.r.t. the pre-train (ImageNet [43]).

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** [167], a widely used collection of language understanding tasks. Specifically, we consider six sentence- and sentence-pair classification tasks—**MNLI** [178], **SICK** [109], **RTE** [16], **SciTail** [79], **QNLI** [135], and **SNLI** [23]—which are encountered sequentially in this order. Together, they form a single dataset of 6

Table 3.1: Results in terms of FAA [↑] and (FF) [↓] on benchmarks using a pre-trained ViT.

	Benchmark	**EuroSAT**	**ISIC**
	noise	*40% symm*	*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 [149]	60.78 (16.61)	41.97 (4.09)
	L2P [173]	48.37 (02.78)	32.87 (03.49)
	SLCA [197]	35.87 (77.72)	31.85 (70.89)
$\mathcal{M} = 500$	CODA-Prompt [149]	62.97 (18.72)	43.37 (23.77)
	w. **EARL**	87.95 (06.88)	52.99 (22.34)
	L2P [173]	77.72 (12.07)	49.03 (09.26)
	w. **EARL**	80.34 (08.67)	51.88 (12.09)
	SLCA [197]	56.26 (11.49)	41.01 (30.49)
	w. **EARL**	92.28 (03.68)	54.18 (17.05)
	ER-ACE [27]	76.39 (18.10)	52.16 (20.67)
	w. **EARL**	93.62 (02.48)	56.00 (21.47)

tasks.

Backbones. We employ a Vision Transformer (ViT) [45] for ISIC, EuroSAT-RGB, and Food-101N, and a ResNet18 [60] for CIFAR-100 and ANIMAL-10N.

Metrics. All results are presented in terms of 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.

3.2.2 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** [173] and **CODA-Prompt** [149], as they represent the most widely adopted methods for rehearsal-free learning. Moreover, we also consider **SLCA** [197], as it stands out for achiev-

ing higher performance w.r.t. prompting in most scenarios. For rehearsal-based methods, we employ **ER-ACE** [27] and **DER**++ [25] 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** [15], **SPR** [80], and **CNLL** [78]. 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. We adapted the **CLTR** [96] 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^{2}L** [29]. We include an additional baseline that applies the regularization of DivideMix [89] 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 [15, 176].

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).

3.2.3 RESULTS

Not pre-trained backbones. We analyze the benefits brought by EARL on popular rehearsal baselines by computing the FAA these exhibit on different datasets, before and after applying EARL to them (Tables 3.2 and 3.3). 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

Table 3.2: 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 [89]	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 [15]	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 [96]	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 [29]	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++ [25]	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 [27]	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)

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 3.1 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 3.1 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.

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

Table 3.3: 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 [89]	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 [15]	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 [96]	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 [29]	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++ [25]	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 [27]	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)

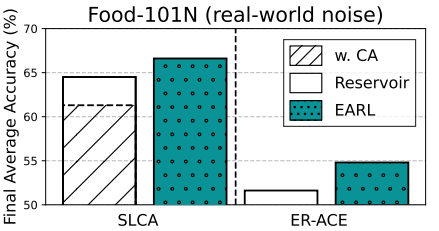


Figure 3.4: Final Average Accuracy (FAA) [↑] of EARL when applied to SLCA and ER-ACE to learn on the Food-101N dataset.

trained on the same datasets without any noise.

In Figure 3.4 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 3.1, 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 Model (VLM) with high zero-shot performance [180]. However, these models are not infallible, and incorrect pseudo-labels introduce chal-

Table 3.4: 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	

lenging 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 3.2.3 and present results in Tables 3.2 and 3.3 (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 [167], a widely used collection of natural language understanding tasks including question answering, sentiment analysis, and textual entailment. Specifically, we consider six sentence- and sentence-pair classification tasks MNLI [178], SICK [109], RTE [16], SciTail [79], QNLI [135], and SNLI [23]—which are encountered sequentially in this order, forming a single dataset of 6 tasks. Multitask (also refferred to as Joint Training (JT)) and finetuning (FT) 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 3.4, 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.

Table 3.5: 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

3.3 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**?

3.3.1 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** [15]: 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 3.6: 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** [177, 137]: 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** [14]: 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 3.5 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 3.6 (right part) we compare three strategies on the human-annotated CIFAR-100N: *Herding*, which generates a representative set of samples from the stream

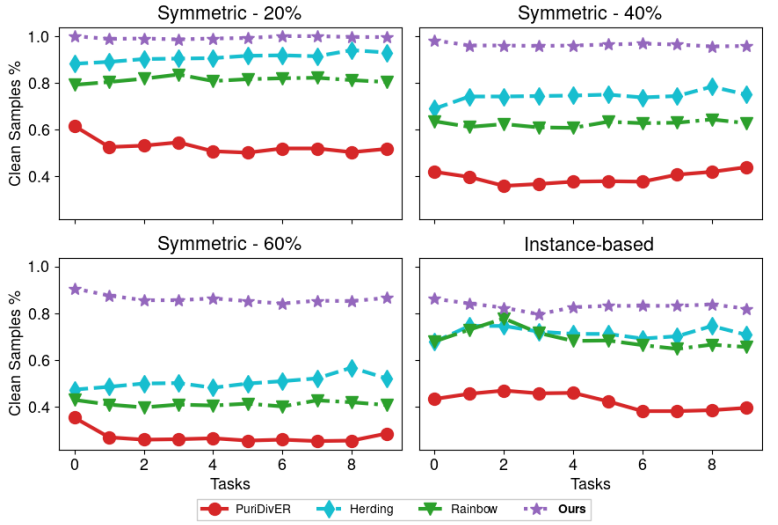


Figure 3.5: Purity levels achieved through different sampling strategies. Trend of buffer clean percentages across incremental training tasks on CIFAR100

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 3.6 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.

3.3.2 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. Figure 3.5 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.

3.3.3 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 [89, 15]. Specifically, we evaluate our α-threshold insertion method (same as in Section 2.3) against a Gaussian Mixture Model (GMM) approach that partitions samples into clean and noisy subsets. We evaluate both methods using two key metrics: FAA and buffer Purity, measured as the average percentage of clean samples retained in the buffer at the end of each task. Table 3.7 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.

Table 3.7: 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 3.8: 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**

3.3.4 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) [176, 94]. Therefore, for the majority of our experiments, we maintain a fixed noise rate of 40% on synthetically noised datasets.

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 3.8. 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.

3.4 Conclusion

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.

4

A Second-Order Perspective on Model Compositionality and Incremental Learning

Over the last two decades, AI research has largely relied on monolithic models trained on single, large-scale datasets. Although this paradigm has delivered strong empirical performance, it poorly matches the requirements of real-world applications, where adaptability, customization, and computational efficiency are critical. As a result, modular alternatives such as model averaging [115] and Mixture of Experts [186] have recently regained attention. These approaches aim to address the high maintenance cost of monolithic models and the limited resources available to many practitioners, aligning with broader efforts to democratize AI through flexible and adaptive architectures [93, 131]. A key ingredient enabling such flexibility is model compositionality, which allows the construction of task-specific models via inexpensive editing operations [101, 22]. Beyond efficient transfer with limited data [204], fine-tuning has revealed that independently trained models can often be meaningfully combined. In particular, simple linear combinations of weights have been

shown to yield robust representations without additional inference or memory costs [179, 199]. While this principle underlies model soups [179], where models differ only in optimization choices, task arithmetic extends it to models trained on distinct tasks or datasets [70, 101, 124], with recent applications to language modeling [44]. In this work, we address two related questions. First, we investigate the conditions under which models trained on different tasks can be successfully composed. Prior analyses are either empirical [70] or restricted to linearized networks and tangent fine-tuning [101, 124]. We instead consider standard non-linear networks and general fine-tuning strategies, including parameter-efficient methods such as LoRA [66]. Using a second-order Taylor approximation of the loss around the pre-training weights, we derive a principled relationship between individual model performance and that of their composition, highlighting the importance of retaining out-of-distribution accuracy. Second, we exploit this insight to study incremental learning of composable models, where each task is assigned a dedicated module. While modularity has been proposed as a natural solution for incremental training [22, 101], we argue that compositionality itself requires continual learning capabilities. Preserving performance outside the task-specific training distribution can be framed as a continual learning objective [81], in which each module must maintain the general knowledge acquired during pre-training. Based on this formulation, we propose two algorithms for incremental fine-tuning of composable models. Both rely on the same second-order approximation but differ in whether they optimize individual modules or their composition. Evaluated in the class-incremental setting [161], our methods produce accurate and editable multi-task models, supporting both task specialization and selective unlearning.

4.1 Framework

We consider $f(\cdot;\theta) : \mathcal{X} \to \mathcal{Y}$ as a twice-differentiable deep network with weights $\theta \in \Theta \subseteq \mathbb{R}^m$. It takes inputs $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ and yields a conditional distribution $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ over the targets $\boldsymbol{y} \in \mathcal{Y} \subseteq \mathbb{R}^c$. In this paper, we focus on **incremental training**, which progresses sequentially through a series of T classification tasks

$\mathcal{T} = \{1, 2, \dots, T\}$. Each task $t \sim \mathcal{T}$ is characterized by a dataset $\mathcal{D}_t$ with n_t training samples $\boldsymbol{x}, \boldsymbol{y} \sim p_t(\boldsymbol{x}, \boldsymbol{y})$ drawn from a distribution varying across tasks. We assume that tasks share the same loss function $\ell(\theta|\boldsymbol{x}, \boldsymbol{y})$, *i.e.* the negative log-likelihood $-\log p_\theta(\boldsymbol{y}|\boldsymbol{x})$. In this setting, we maintain a pool of composable networks $\mathcal{P} = \{f(\cdot; \theta_t) \mid \theta_t \triangleq \theta_0 + \tau_t\}_{t \in \mathcal{T}}$, where each model is fine-tuned from a common set of pre-training weights θ_0. The *task vector* [70] τ_t indicates the displacement in weight space w.r.t. θ_0 after training on task t. We obtain the weights of the composed model $f_\mathcal{P}$ by averaging the weights within the pool:

$$f_\mathcal{P} \triangleq f(\cdot; \theta_\mathcal{P}) \quad \text{s.t.} \quad \theta_\mathcal{P} = \theta_0 + \textstyle\sum_{t=1}^{T} w_t \tau_t, \quad \sum_{t=1}^{T} w_t = 1 \tag{4.1}$$

where w_t balances the contribution of the t-th learner. While some works [13, 67] optimize these coefficients, we devise uniform weights $w_t = 1/T$ in our algorithms.

Scope. How can we learn multiple disjoint tasks through a pool $\mathcal{P}$ of models, so that the composed model performs well on their union? To answer this question, we introduce the concept of **empirical risk**, *i.e.* the average loss $\hat{\ell}(\theta|\mathcal{D})$, computed over the union $\mathcal{D} = \bigcup_{t=1}^{T} \mathcal{D}_t$ of all training tasks:

$$\hat{\ell}(\theta|\mathcal{D}) = \tfrac{1}{\sum_{t=1}^{T} n_t} \textstyle\sum_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}} \ell(\theta|\boldsymbol{x}, \boldsymbol{y}) \approx \mathbb{E}_{\substack{t \sim \mathcal{T} \\ \boldsymbol{x}, \boldsymbol{y} \sim p_t(\boldsymbol{x}, \boldsymbol{y})}} [\ell(\theta|\boldsymbol{x}, \boldsymbol{y})] \tag{4.2}$$

To simplify notation, we will henceforth omit the explicit dependence of the loss on the data, denoting the individual loss $\ell(\theta|\boldsymbol{x}, \boldsymbol{y})$ simply as $\ell(\theta)$, and the empirical risk $\hat{\ell}(\theta|\mathcal{D})$ as $\hat{\ell}(\theta)$. On this basis, the rest of this section will delve into the following research questions.

Question i) Given the empirical risk of each individual model, what can we say about the composed model $f(\cdot;\theta_{\mathcal{P}})$? *Question ii)* How can we train individual learners $f(\cdot;\theta_t)$ on distinct tasks (***individual training***) to still achieve a reliable composition $f(\cdot;\theta_{\mathcal{P}})$? *Question iii)* Instead of optimizing each model on its individual loss, could we optimize each model based on the loss of the whole composed model $f(\cdot;\theta_{\mathcal{P}})$ (***ensemble training***)?

INDIVIDUAL LEARNERS *VS.* THE COMPOSED MODEL: A PRE-TRAINING PERSPECTIVE

We now relate the composed model $f_{\mathcal{P}}$ to the individual components $f(\cdot;\theta_t)$ of the pool. To do so, we introduce the **second-order** Taylor approximation $\ell_{\text{cur}}(\theta)$ of the loss around the pre-trained weights θ_0:

$$\ell(\theta) = \ell_{\text{cur}}(\theta) + \mathcal{O}(\|\theta-\theta_0\|^3) \text{ where} \tag{4.3}$$

$$\ell_{\text{cur}}(\theta) = \ell(\theta_0) + (\theta-\theta_0)^{\mathrm{T}}\nabla\ell(\theta_0) + \frac{1}{2}(\theta-\theta_0)^{\mathrm{T}}\mathbf{H}_{\ell}(\theta_0)(\theta-\theta_0). \tag{4.4}$$

$\nabla\ell(\theta_0) \triangleq \nabla_{\theta}\ell(\theta_0|\boldsymbol{x},\boldsymbol{y})$ and $\mathbf{H}_{\ell}(\theta_0) \triangleq \nabla^2_{\theta}\ell(\theta_0|\boldsymbol{x},\boldsymbol{y})$ are the gradient and the Hessian around θ_0. Similarly, we can define the second-order approximation $\hat{\ell}_{\text{cur}}(\theta) \approx \hat{\ell}(\theta)$ of the empirical risk, which corresponds to averaging the approximated loss across examples from all tasks.

Assumption. We now assume that $\theta = \theta_0$ is a point of **local minimum** of the empirical risk $\hat{\ell}(\theta)$*across all tasks (Eq. 4.2). Under this hypothesis, the Hessian of the empirical risk $\mathbf{H}_{\hat{\ell}}(\theta_0) \succeq 0$ is positive semidefinite. In light of this and the quadratic nature of $\hat{\ell}_{\text{cur}}(\theta)$, we can state that the second-order approximation of the empirical risk $\hat{\ell}_{\text{cur}}(\theta)$ is **convex**. Therefore, we apply the **Jensen's inequality**

*As shown in [125], techniques like linear probing, latent replay, and incremental linear discriminant analysis can be employed to enforce the optimality of the base model θ_0, along with Instruction Tuning for language modeling [188]. See Section 4.2 for additional implementation details.

to derive a relation between the composed model and the individual components:

$$\hat{\ell}_{\text{cur}}(\theta_{\mathcal{P}} = \theta_0 + \sum_{t=1}^{T} w_t \tau_t) \leq \sum_{t=1}^{T} w_t \, \hat{\ell}_{\text{cur}}(\theta_t = \theta_0 + \tau_t). \tag{4.5}$$

The relation states that, under the second-order approximation, the empirical risk $\hat{\ell}_{\text{cur}}(\theta_{\mathcal{P}})$ of the composed model is upper-bounded by the convex combination of its individuals. In other words, if each individual model is trained to the optimum with near-zero loss value, there are some guarantees that - under the stated local assumptions - the loss function attained by the composed model. Notably, this relation could help reduce the computational footprint during inference, as it enables the reduction of forward passes, from multiple (one for each individual) to a singular pass (performed on the composed model).

At a first glance, the result of Eq. 4.5 appears similar to the statement of Eq. 2 in [101]:

$$\hat{\ell}(\theta_{\mathcal{P}} = \theta_0 + \sum_{t=1}^{T} w_t \tau_t) \leq \sum_{t=1}^{T} w_t \hat{\ell}(\theta_t = \theta_0 + \tau_t) \text{ given that} \tag{4.6}$$

$$f_t(\cdot;\theta_t) \triangleq f_{\text{lin}}(\cdot;\theta_t) = f(\cdot;\theta_0) + (\theta_t - \theta_0)^{\mathrm{T}} \nabla f(\cdot;\theta_0) \text{ (\textit{tangentness})} \tag{4.7}$$

However, some notable distinctions remain. Their inequality applies to the exact risk $\hat{\ell}$ but is valid only for linearized models (*i.e.* fine-tuned in the tangent space of pre-training weights). In contrast, our result pertains to the *second-order* approximation $\hat{\ell}_{\text{cur}}$ of the risk and applies to *any* fine-tuning strategy (*e.g.* LoRA, adapters, etc.). Intuitively, our inequality provides more flexibility to the training of individual learners, as long as: *i)* the learners remain in the pre-training basin, such that $\mathcal{O}(\|\theta - \theta_0\|^3) \to 0$ and ℓ_{cur} can be considered a good proxy of ℓ; *ii)* θ_0 is a local minimum of ℓ.

ENABLING INDIVIDUAL TRAINING IN INCREMENTAL SCENARIOS

As mentioned above, a possible application of Eq. 4.5 is to devote each learner to a distinct task and optimize them in isolation (*i.e.* **individual training**). Indeed, the upper bound in Eq. 4.5 describes a sort of worst-case scenario for the risk

of the composed model: at worst, it collapses to that given by the upper bound. Nonetheless, if every individual model were *accurate* on all tasks, the right-side term of Eq. 4.5 would yield an appealing upper-bound to the risk of the composed model.

Limits of Eq. 4.5. Under the constraints of individual training, each individual learner $f(\cdot;\theta_t)$ can be optimized only on the current distribution $p_t(\boldsymbol{x},\boldsymbol{y})$. Therefore, when considering examples from other data distributions $p_{t'\neq t}(\boldsymbol{x},\boldsymbol{y})$, the loss $\ell_{\text{cur}}(\theta_t|\boldsymbol{x},\boldsymbol{y})$ of the t-th individual learner is likely to be much higher for examples outside its training distribution $p_t(\boldsymbol{x},\boldsymbol{y})$. As a consequence, the upper bound delivered by the right-side of Eq. 4.5 is likely to increase one task after the other, and we cannot rely on it to recover a reliable composed model $f(\cdot;\theta_{\mathcal{P}})$.

In this respect, our proposal is to tighten the upper bound of Eq. 4.5 through explicit regularization, which we devise during the optimization of each individual learner. In practice, each model is provided with a learning objective that extends beyond minimizing the loss on the assigned data distribution $p_t(\boldsymbol{x},\boldsymbol{y})$. Specifically, to ensure decent performance on external distributions $p_{t'\neq t}(\boldsymbol{x},\boldsymbol{y})$, we anchor the model to the pre-training knowledge for out-of-distribution examples. We do so by encouraging the predictions of $f(\cdot;\theta_t)$ to be close to those generated by the pre-trained model $f(\cdot;\theta_0)$, given examples $\boldsymbol{x},\boldsymbol{y} \sim p_{t'\neq t}(\boldsymbol{x},\boldsymbol{y})$. Denoting the pre-training posterior distribution as $p_{\theta_0}(\boldsymbol{y}|\boldsymbol{x})$, we have:

$$\underset{\theta_t}{\text{minimize}} \quad \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p_t(\boldsymbol{x},\boldsymbol{y})}\left[\ell_{\text{cur}}(\theta_t|\boldsymbol{x},\boldsymbol{y})\right] + \mathcal{D}_{\text{KL}}(p_{\theta_0}(\boldsymbol{y}|\boldsymbol{x})||p_{\theta_t}(\boldsymbol{y}|\boldsymbol{x})). \tag{4.8}$$

It is noted that the computation of the KL term $\mathcal{D}_{\text{KL}}(\cdot)$ requires sampling from the external distributions $p_{t'\neq t}(\boldsymbol{x},\boldsymbol{y})$. Theoretically, this clashes with the constraints of individual training. Thankfully, if $\tau_t \to 0$, the KL term can be approximated [31] with the *distance* between θ_t and pre-training weights θ_0:

$$\mathcal{D}_{\text{KL}}(p_{\theta_0}(\boldsymbol{y}|\boldsymbol{x}) \,||\, p_{\theta_t}(\boldsymbol{y}|\boldsymbol{x})) \approx \tfrac{1}{2}(\theta_t-\theta_0)^{\text{T}}\text{F}_{\theta_0}(\theta_t-\theta_0) = \tfrac{1}{2}\|\tau_t\|^2_{\text{F}_{\theta_0}}. \tag{4.9}$$

The distance is computed in the Riemannian manifold [87] induced by the Fisher Information Matrix (FIM) [69]; namely, a $|\theta| \times |\theta|$ positive semi-definite matrix

given by:

$$\mathrm{F}_{\theta_0} = \mathbb{E}_{\substack{t\sim\mathcal{T} \\ \boldsymbol{x}\sim p_t(\boldsymbol{x},\boldsymbol{y})}} \left[\mathbb{E}_{\boldsymbol{y}\sim p_{\theta_0}(\boldsymbol{y}|\boldsymbol{x})} \left[\nabla \log p_{\theta_0}(\boldsymbol{y}|\boldsymbol{x}) \, \nabla \log p_{\theta_0}(\boldsymbol{y}|\boldsymbol{x})^\top \right] \right]. \tag{4.10}$$

Under the local minimum hypothesis on pre-training weights, the FIM at θ_0 equals the expected[†] Hessian of the negative log-likelihood [111]: $\mathrm{F}_{\theta_0} = \mathbb{E}_{\boldsymbol{x}} [\mathbf{H}_\ell(\theta_0)]$. Due to this connection, the FIM yields insights on the sensitivity of each parameter to changes in the data distribution.

Application to incremental learning. Given that optimizing the regularization above does not necessitate data from external tasks, it can be readily adapted to incremental learning. Following [31, 146], we introduce a few additional approximations. Firstly, we limit to estimate the diagonal $\hat{\mathrm{F}}_{\theta_0}$ of the FIM, thus avoiding the prohibitive footprint required to treat a $|\theta| \times |\theta|$ matrix. Basically, the diagonal $\hat{\mathrm{F}}_{\theta_0}$ consists of a Monte Carlo estimate of the (squared) gradients of the log-likelihood. Secondly, we note that the expectation $\mathbb{E}_{\boldsymbol{x}} [\mathbf{H}_\ell(\theta_0)]$ in the FIM cannot be directly estimated, as examples from all tasks are not available simultaneously but rather sequentially. We hence compute the expectation incrementally [31, 146] one task at a time. As each new task becomes available, we calculate a *local* Fisher matrix on the data of the new task and then accumulate it into a *global* running Fisher estimate. As outlined by Algorithm 2, the accumulation can be thought as a simple summation, net of re-normalizing operations reflecting the number of samples of each task (see Section 4.6).

Given Eq. 4.9 and the diagonal FIM, the augmented optimization problem for the t-th learner becomes:

$$\underset{\theta_t}{\text{minimize}} \quad \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}\sim p_t(\boldsymbol{x},\boldsymbol{y})} [\ell_{\text{cur}}(\theta_t|\boldsymbol{x},\boldsymbol{y})] + \frac{\alpha}{2} \, \mathrm{EWC}_{\theta_0}(\theta_t) \tag{4.11}$$

$$\text{where} \quad \mathrm{EWC}_{\theta_0}(\theta_t) = \textstyle\sum_i^{|\theta|} \hat{\mathrm{F}}_{\theta_0}^{(i)} (\theta_t^{(i)} - \theta_0^{(i)})^2. \tag{4.12}$$

where $\alpha \geq 0$ is an hyper-parameter and $\mathrm{EWC}_{\theta_0}(\cdot)$ indicates the Riemannian distance from the pre-training weights θ_0. The acronym EWC_{θ_0} highlights the

[†]Precisely, we take the expectation w.r.t. data from other tasks $t' \neq t$ to reflect the regularization in Eq. 4.8.

strong analogy with Elastic Weight Consolidation [81], a well-established approach against *catastrophic forgetting* [114]. In a sense, our term prevents forgetting pre-training knowledge; however, while our anchor is fixed at θ_0, the anchor of EWC instead shifts and focuses on the weights learned during the preceding task.

JOINT TRAINING OF THE COMPOSED MODEL IN INCREMENTAL SCENARIOS

Individual training profitably aligns with *decentralized learning* [22], emphasizing minimal interactions between learners and privacy preservation. Nevertheless, it might be inefficient when these constraints are not of interest and the goal is simply to create an accurate model. In fact, individual training prevents a learner from leveraging the knowledge of other ensemble members, eliminating the potential for beneficial mutual transfer. For these reasons, we adopt the dual perspective, in which each model is directly optimized using the loss of the composed model $f(\cdot;\theta_{\mathcal{P}})$ (**ensemble training**). Since the inequality in Eq. 4.5 does not provide much help for the explicit optimization of $\ell_{\text{cur}}(\theta_{\mathcal{P}})$, we quantify the exact gap between the two sides of the Jensen's inequality:

Theorem 1. *Let us assume a pool $\mathcal{P}$ with $T \geq 2$ models, with the t-th model parameterized by $\theta_t = \theta_0 + \tau_t$. If we compose them through coefficients $w_1, \ldots, w_T$ s.t. $w_t \in [0,1]$ and $\sum_{t=1}^{T} w_t = 1$, the 2nd order approximation $\ell_{\text{cur}}(\theta_{\mathcal{P}})$ evaluated on composed weights $\theta_{\mathcal{P}} = \theta_0 + \sum_{t=1}^{T} w_t \tau_t$ is:*

$$\ell_{\text{cur}}(\theta_{\mathcal{P}}) + \Omega(\theta_1, \ldots, \theta_T) = \sum_{t=1}^{T} w_t \ell_{\text{cur}}(\theta_t) \tag{4.13}$$

$$\textbf{where} \quad \Omega(\theta_1, \ldots, \theta_T) = \frac{1}{2} \sum_{t=1}^{T} \sum_{t'<t} w_t w_{t'} (\tau_t - \tau_{t'})^{\mathrm{T}} \mathbf{H}_{\ell}(\theta_0)(\tau_t - \tau_{t'}). \tag{4.14}$$

Proof in Section 4.4. The equality introduces a term $\Omega(\cdot)$ that is non-negative (due to $\mathbf{H}_{\ell}(\theta_0) \succeq 0$) and depends on weights $\theta_1, \ldots, \theta_T$. Therefore, $\Omega(\cdot)$ is proportional to the cumulative *distance* between every pair of learners, within the Riemannian manifold induced by the Hessian $\mathbf{H}_{\ell}(\theta_0)$ [87]. Notably, Eq. 4.13 permits to draw the following insights: *i)* optimizing both the loss of the com-

posed model $\ell_{\text{cur}}(\theta_{\mathcal{P}})$ and the term $\Omega(\cdot)$ is equivalent to individual training; *ii)* during *inference*, if $\Omega(\cdot)$ tends towards zero, performing a prediction with the composed model is akin to conducting multiple forward passes, each corresponding to an individual learner. Based on that interpretation, we now transition to a setting that considers the explicit auxiliary minimization of $\Omega(\cdot)$, as follows:

$$\underset{\theta_1,\ldots,\theta_T}{\text{minimize}} \quad \mathbb{E}_{\substack{t\sim\mathcal{T} \\ \boldsymbol{x},\boldsymbol{y}\sim p_t(\boldsymbol{x},\boldsymbol{y})}} \left[\ell_{\text{cur}}(\theta_{\mathcal{P}}|\boldsymbol{x},\boldsymbol{y}) + \beta\, \Omega(\theta_1,\ldots,\theta_T)\right] \tag{4.15}$$

where $\beta \geq 0$ is an hyper-parameter. It is noted that minimizing $\Omega(\cdot)$ encourages alignment among task vectors, especially for those weights that are sensitive/important in the pre-training loss landscape.

Similarly to [72], the objective of Eq. 4.15 can be interpreted as a smooth transition between individual and ensemble training. Indeed, given that $\Omega(\cdot) = \sum_t w_t \ell_{\text{cur}}(\theta_t) - \ell_{\text{cur}}(\theta_{\mathcal{P}})$, Eq. 4.15 can be restated:

$$\ell_{\text{cur}}(\theta_{\mathcal{P}}) + \beta\, \Omega(\theta_1,\ldots,\theta_T) \overset{\text{Eq. 4.13}}{=} (1-\beta)\, \ell_{\text{cur}}(\theta_{\mathcal{P}}) + \beta \textstyle\sum_{t=1}^{T} w_t \ell_{\text{cur}}(\theta_t) \tag{4.16}$$

This suggests that by minimizing both the loss $\ell_{\text{cur}}(\theta_{\mathcal{P}})$ of the joint composed model and $\Omega(\cdot)$, we also implicitly optimize the individual models. Notably, this result not only paves the way for a well-performing ensemble but also for components that are reliable when considered individually.

Incremental ensembles. We now refine the problem in Eq. 4.15 to account for incremental settings. We firstly bring the expectation inside $\Omega(\cdot)$ (Eq. 4.14) and replace the Hessian $\mathbf{H}_{\ell}(\theta_0)$ with its expectation, taken across data points $\boldsymbol{x},\boldsymbol{y} \sim p_t(\boldsymbol{x},\boldsymbol{y})$ from all tasks up to the current one. Afterwards, we capitalize on a property mentioned in Section 4.1, which states that, for a point of maximum likelihood like $\theta = \theta_0$, the expected Hessian of the negative log-likelihood coincides with the Fisher F_{θ_0}. As a result, we can approximate $\mathbb{E}_{\boldsymbol{x}}\left[\mathbf{H}_{\ell}(\theta_0)\right]$ with the tractable diagonal Fisher $\hat{\text{F}}_{\theta_0}$. Based on $\mathbf{H}_{\ell}(\theta_0) \approx \hat{\text{F}}_{\theta_0}$ and further steps (see

Section 4.4) , we can rearrange $\Omega(\cdot)$ as:

$$\Omega(\cdot) \approx \Omega_{\hat{\mathrm{F}}}(\theta_1, \ldots, \theta_T) = \tfrac{1}{2}\sum_{t=1}^{T} w_t(1-w_t) \underbrace{\mathrm{EWC}_{\theta_0}(\theta_t)}_{\text{see Eq. 4.12}} - \sum_{t=1}^{T}\sum_{t'<t} w_t w_{t'} \tau_t{}^{\mathrm{T}} \hat{\mathrm{F}}_{\theta_0} \tau_{t'}. \tag{4.17}$$

Intuitively, $\Omega_{\hat{\mathrm{F}}}(\cdot)$ aims to preserve pre-training knowledge embodied by θ_0 through the first term; through the second one, instead, it encourages pairwise alignment between task vectors. Also, we highlight that the summations depend on the number of models T. Therefore, the *more* components we manage within the ensemble, the *more* crucial it becomes to minimize $\Omega_{\hat{\mathrm{F}}}(\cdot)$ within Eq. 4.15.

Finally, we plug the approximated barrier term $\Omega_{\hat{\mathrm{F}}}(\cdot)$ into the optimization problem outlined in Eq. 4.15. As learning proceeds in subsequent tasks, we can optimize the composed model only one task at a time. To prevent biasing the previously learned components of the pool toward current data, at each round t we optimize only the weights $\theta_t \triangleq \theta_0 + \tau_t$ of the corresponding t-th learner, while freezing the others components of the pool. This yields the following form for the t-th learning task:

$$\underset{\theta_t}{\text{minimize}} \quad \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p_t(\boldsymbol{x},\boldsymbol{y})} \left[\ell_{\mathrm{cur}}(\theta_{\mathcal{P}}|\boldsymbol{x},\boldsymbol{y})\right] + \beta\, \Omega_{\hat{\mathrm{F}}}(\theta_1, \ldots, \theta_t). \tag{4.18}$$

Finally, if we optimize only one θ_t at a time and $w_t = \frac{1}{t}$, then several terms in Eq. 4.17 can be ignored, such as $\mathrm{EWC}_{\theta_0}(\theta_{t'})$ for $t' < t$. Hence, minimizing Eq. 4.18 is equivalent to minimize:

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p_t(\boldsymbol{x},\boldsymbol{y})} \left[\ell_{\mathrm{cur}}(\theta_{\mathcal{P}}|\boldsymbol{x},\boldsymbol{y})\right] + \tfrac{\beta}{t}\left[(1-\tfrac{1}{t})\tfrac{1}{2}\,\mathrm{EWC}_{\theta_0}(\theta_t) - \tfrac{1}{t}\sum_{t'<t} \tau_t{}^{\mathrm{T}} \hat{\mathrm{F}}_{\theta_0} \tau_{t'}\right]. \tag{4.19}$$

As shown in Section 4.4 for both full and LoRA fine-tuning, the gradients of the regularization term in Eq. 4.19 can be computed in **closed form**. Importantly, this derivation enables a lightweight optimization, as the analytical gradients maintain **constant complexity** w.r.t. the number of tasks.

4.2 Algorithm(s)

We present **Incremental Task Arithmetic** (**ITA**) and **Incremental Ensemble Learning** (**IEL**), two **distinct** algorithms for *individual* and *ensemble* incremental learning respectively. As shown in Algorithm 2, they divide each task into the *pre-consolidation* and *fine-tuning* stages, with differences occurring in the latter. We direct the reader to Section 4.6 for comprehensive implementation guidelines.

Task pre-consolidation. Due to the pivotal role of the local optimality of θ_0 (Section 4.1), we now follow up on this aspect and consider *closed vocabulary* models. Unlike models like CLIP [134], closed vocabulary models require the addition of a tailored classification head to handle novel classes. In line with [124], we denote the process of fine-tuning only the added classification head as **linear probing** (**LP**): specifically, LP keeps the rest of the layers fixed to the pre-training θ_0. In our work, we basically exploit linear probing to enforce the optimality of pre-training weights θ_0. Namely, during the pre-consolidation phase, we conduct a few preliminary training epochs on the t-th incoming task, with the sole purpose of fine-tuning the new classification head. From that point onward, the fine-tuned head $\boldsymbol{h}_0^t$ is regarded as a part of pre-training weights θ_0. Finally, the pre-consolidation stage concludes with the update of the diagonal Fisher matrix $\hat{\mathrm{F}}_{\theta_0}$.

Fine-tuning. While the pre-consolidation step is identical for both ITA and IEL, they differ during the fine-tuning phase. The shared goal is to learn a task vector τ_t s.t. $\theta_t \equiv \theta_0 + \tau_t$ for the current task. However, ITA treats τ_t as the weights of an individual model, whereas IEL interprets it as a new learnable component of the composition. In other words, ITA computes predictions through the t-th learner $f(\boldsymbol{x}; \theta_t)$, while IEL leverages the composed function $f(\boldsymbol{x}; \theta_{\mathcal{P}})$. Moreover, their regularizing objectives differ: ITA builds upon Eq. 4.11 (*i.e.* the additional EWC-like term computed w.r.t. θ_0), while IEL exploit Eqs. 4.18 and 4.19 to train the composed model. Notably, both approaches can be applied to any fine-tuning strategy in the form of $\theta_0 + \Delta\theta$. We conduct experiments on Full Fine-Tuning (*i.e.*

Algorithm 2 Incremental Task Arithmetic (ITA) ***vs.*** Incremental Ensemble Learning (IEL)

Input: T disjoint classification tasks $D_t = \{(\boldsymbol{x}, \boldsymbol{y})\}_{n_t}$, a pre-trained DNN $f(\cdot; \theta_0)$, learning rate $\texttt{lr}$, hyper-parameters α and β, initialized pool $\mathcal{P} = \emptyset$ and the diagonal Fisher $\hat{\mathrm{F}}_{\theta_0} = \mathbf{0}$.

for each task $t \in \{1, 2, \ldots, T\}$ **do**

- $\boldsymbol{b}_0^{(t)} \leftarrow$ Linear Probing on D_t with the pre-trained $f(\cdot; \theta_0)$ — Task pre-consolidation
- $\theta_0 \leftarrow \theta_0 \cup \boldsymbol{b}_0^{(t)}$ — ▷ add the pre-training classification head
- $\hat{\mathrm{F}}_{\theta_0} \leftarrow \hat{\mathrm{F}}_{\theta_0} + \hat{\mathrm{F}}_{\theta_0}^{(t)}$ — ▷ update the global Fisher with the local Fisher $\hat{\mathrm{F}}_{\theta_0}^{(t)}$ estimated on D_t
- $\tau_t \leftarrow \mathcal{N}(\mu_{\text{init}}, \sigma_{\text{init}})$ — Fine-tuning
- $\mathcal{P} \leftarrow \mathcal{P} \cup \{f(\cdot; \theta_t = \theta_0 + \tau_t)\}$ — ▷ extend $\mathcal{P}$ with the weights of the t-th learner
- **for** each example $(\boldsymbol{x}, \boldsymbol{y})$ **in** D_t **do**
  - $p_\theta(\boldsymbol{y}|\boldsymbol{x}) \leftarrow f(\boldsymbol{x}; \theta_t)$ — ▷ predict with the t-th learner $\theta_t \leftarrow \theta_0 + \tau_t$
  - $p_\theta(\boldsymbol{y}|\boldsymbol{x}) \leftarrow f(\boldsymbol{x}; \theta_{\mathcal{P}})$ — ▷ predict with the composed model $\theta_{\mathcal{P}} \leftarrow \theta_0 + \frac{1}{t}\sum_{t'=1}^{t} \tau_{t'}$
  - $\ell \leftarrow -\log p_\theta(\boldsymbol{y}|\boldsymbol{x})$
  - $\tau_t \leftarrow \tau_t - \texttt{lr} \cdot \nabla_{\tau_t}[\ell + \frac{\alpha}{2}\,\mathrm{EWC}_{\theta_0}(\theta_t)]$ — ▷ arithmetic-oriented regularization (Eq. 4.11)
  - $\tau_t \leftarrow \tau_t - \texttt{lr} \cdot \nabla_{\tau_t}[\ell + \beta\Omega_{\hat{\mathrm{F}}}(\mathcal{P})]$ — ▷ ensemble-oriented regularization (Eqs. 4.18 and 4.19)

$\tau_t \in \mathbb{R}^{F_{OUT} \times F_{IN}}$), Low-Rank Adaptation (LoRA) [66] (*i.e.* $\tau_t = B_t A_t$), and on `(IA)`^{3} [99]. About the latter, let h represent the hidden dimension and $l \in \mathbb{R}^h$ denote the `(IA)`^{3} task-specific vector: it can be shown that `(IA)`^{3} is equivalent to $\theta_t \equiv \theta_0 + \tau_t$ with $\tau_t = \theta_0 \odot ((l - 1_h) \otimes 1_h)$, where 1_h is the all-ones vector. For each fine-tuning strategy, we initialize their parameters so that the resulting task vector τ_t starts as a null vector at the beginning.

Computational analysis. As outlined in Section 4.5, by treating the composed model as a cumulative average of individual models, both training/inference stages of ITA/IEL maintain constant complexity $\mathcal{O}(1)$ with respect to the number of tasks T. Indeed, a single forward pass is sufficient to compute the output of the composed model (**constant time**). Moreover, we do not need to store a separate set of weights for each task (**constant memory**), provided we are not interested in more complex forms of composition than the simplest uniform average (as required for specialization and unlearning).

4.3 Experiments

Datasets. Following affine works [174, 22, 101], we evaluate on these **class-incremental** benchmarks: Split ImageNet-R [63] (10 tasks $\times$ 20 classes each), Split CIFAR-100 [84] (10 tasks $\times$ 10 classes), Split CUB-200 [166] (10 tasks $\times$ 20 classes), Split Caltech-256 [55] (10 tasks, as in [101]), and Split MIT-67 [133] (10 tasks, as in [101]). We conduct further tests in the **aerial** and **medical** domains using Split RESISC45 [35] (9 tasks × 5 classes) and Split CropDiseases [68] (7 tasks × 5 classes). They provide a challenging benchmark for our proposals due to their **low domain similarity** with the ImageNet pre-training [123].

Benchmarking. We compare against recognized incremental methods as EWC [81], LwF-MC [137], DER++ [25], L2P [174], and CODA [149]. We also asses SEED [143], InfLoRA [98], APT [22], and TMC [101], four approaches featuring compositionality and ensemble learning. Their description and the main differences from our approaches are provided in Section 4.7. All methods, including ours, utilize the **same backbone** – a ViT-B/16 [45] with supervised pre-

Table 4.1: Comparison with SOTA (Final Accuracy [↑]). Best results in **bold**, second-best underlined. EWC, LwF-MC, DER++ (buffer size of 1, 000 examples), SEED, and TMC rely on full fine-tuning; L2P, CODA, and APT utilize prompt-based learning. Finally, InfLoRA adopts LoRA fine-tuning.

Model	IN-R	C-100	CUB	Caltech	MIT	RESISC	CropDis.
Joint	86.08	91.74	87.12	93.21	87.19	96.79	99.71
Finetune	22.31	21.57	29.35	44.03	18.85	12.90	20.54
EWC	58.64	73.49	40.33	73.23	64.44	58.80	70.33
LWF-MC	50.93	72.16	21.88	79.73	67.04	68.09	78.28
DER++	60.53	83.02	76.10	86.11	68.88	67.23	**98.77**
L2P	67.17	87.32	78.95	91.22	83.17	63.47	75.18
CODA	74.12	86.48	78.54	90.57	77.73	69.50	74.65
SEED	55.87	83.39	85.35	90.04	86.34	74.81	92.77
APT	65.32	86.19	69.51	87.71	75.83	49.99	59.37
InfLoRA	76.97	87.17	79.14	90.53	79.14	79.92	89.05
TMC	60.01	78.42	71.72	82.30	68.66	60.66	66.56
ITA-FFT	76.43	89.38	84.80	92.32	85.35	80.50	91.81
ITA-LoRA	77.79	89.96	85.55	92.65	**86.60**	82.00	95.85
ITA-$(IA)^3$	77.04	**90.66**	**85.67**	92.67	84.74	**83.73**	95.41
IEL-FFT	**80.09**	89.38	84.89	92.23	82.79	81.42	95.83
IEL-LoRA	79.93	89.53	84.95	92.19	84.49	82.53	95.88
IEL-$(IA)^3$	77.86	89.72	84.57	**92.70**	85.54	81.50	95.68

training on ImageNet21K [140] – and the same batch size (128). We compute the accuracy on all classes at the end of the final task (Final Accuracy, FA). Following [25], the **hyperparameters** are chosen through a grid search on a validation set (*i.e.* 10% of the training set). This procedure was repeated for each method, ensuring careful tuning of the search space.

Comparison with SOTA. As reported in Table 4.1, ITA and IEL outperform existing approaches on all datasets except MIT-67 and CropDisease. At times, they match SEED; however, its reliance on Mixture of Experts makes its inference computationally demanding. Our results far exceed those of TMC, showcasing the potential of the non-linear regime over linearization. Considering the good results on RESISC and CropDis., ITA and IEL do not seem affected by large domain shifts, indicating that our formulation remains effective even when the

Table 4.2: For ITA, analysis of the impact of the proposed regularization loss (FA [↑]).

Model	IN-R	C-100	CUB	Caltech	MIT	RESISC	CropDis.
ITA-FFT *(reg)*	**76.43**	**89.38**	**84.80**	**92.32**	**85.35**	**80.50**	**91.81**
without Eq. 4.12 *reg.*	8.61	17.59	10.47	12.76	12.01	17.17	20.64
Eq. 4.12 *only on* CLS	76.00	87.60	83.54	91.04	81.45	75.26	77.00
ITA-LoRA *(reg)*	**77.79**	89.96	**85.55**	**92.65**	**86.60**	**82.00**	95.85
without Eq. 4.12 *reg.*	50.17	66.58	60.58	74.87	52.74	37.59	55.86
Eq. 4.12 *only on* CLS	77.33	**90.03**	**85.55**	92.59	84.86	80.64	**96.22**
ITA-`(IA)`^{3} *(reg)*	**77.04**	**90.66**	**85.67**	**92.67**	84.74	83.73	95.41
without Eq. 4.12 *reg.*	71.82	88.43	77.61	90.66	69.14	69.01	63.72
Eq. 4.12 *only on* CLS	76.72	90.48	85.56	92.56	**85.25**	**84.37**	**95.45**
IEL-FFT *(reg)*	**80.09**	**89.38**	84.89	**92.23**	**82.79**	**81.42**	95.83
without Eq. 4.19 *reg.*	40.85	52.56	14.02	53.76	47.63	39.20	31.24
Eq. 4.19 *reg. only on* CLS	77.99	85.82	**85.30**	91.43	77.58	76.87	**96.18**
IEL-LoRA *(reg)*	**79.93**	**89.53**	**84.95**	**92.19**	**84.49**	**82.53**	**95.88**
without Eq. 4.19 *reg.*	51.15	66.01	60.39	70.71	55.38	42.72	45.25
Eq. 4.19 *reg. only on* CLS	76.14	86.11	84.43	91.77	82.50	70.05	95.54
IEL-`(IA)`^{3} *(reg)*	**77.86**	**89.72**	84.57	92.70	**85.54**	81.50	95.68
without Eq. 4.19 *reg.*	73.72	84.00	74.72	89.58	69.82	62.52	66.29
Eq. 4.19 *reg. only on* CLS	77.23	89.38	**84.70**	**92.76**	85.43	**81.60**	**95.72**

pre-training optimality is challenged. Finally, given the comparable results of ITA and IEL, we recommend ITA as the preferred starting point for future research, in light of the greater flexibility offered by the individual training paradigm.

On the impact of regularization. Table 4.2 reports the results of both ITA and IEL when the regularization term in Eq. 4.12 is removed. To evaluate the effect on distinct layers, we additionally assess the model with only the last classification layer regularized (see *Eq.* 4.12 *only on* CLS in Table 4.2). As observed: *i)* applying Eq. 4.12 is beneficial for all examined fine-tuning strategies; *ii)* although regularizing all layers is the most consistent approach, applying Eq. 4.12 only on the classification head already yields good accuracy. This suggests that compositionality in closed-set models is largely dependent on the learning dynamics of the last layer. Finally, full fine-tuning (FFT) struggles the most when no regularization or partial regularization is applied, in contrast to PEFT modules like LoRA and `(IA)`^{3} that still manage to achieve decent results. We ascribe this evidence to the tendency of PEFT modules to forget less of the pre-trained knowledge [18],

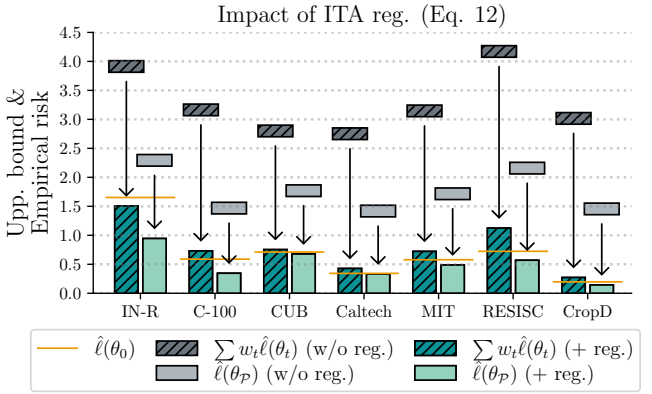


Figure 4.1: Effect of ITA. Best viewed in color.

a feature we identify as beneficial for model compositionality.

To gain insights into our regularization, we revisit the foundations of ITA: specifically the inequality in Eq. 4.5, which states that the risk of the composed model $\hat{\ell}_{\text{cur}}(\theta_{\mathcal{P}})$ is upper bounded by the weighted risk of the individual models $\sum w_t \hat{\ell}_{\text{cur}}(\theta_t)$. In practice, there is no guarantee that this upper bound is tight: as these individual models are trained on disjoint tasks, their risk is likely high for examples outside their training distribution. We hence expect the upper bound to be loose: this is shown in Figure 4.1, where we measure the effect of our regularization on the empirical risk of the composed model ▭ and on the upper bound ▨. From this lens, ITA significantly tightens the upper bound, with a remarkable reduction of the risk ▭ associated with the composed model $\hat{\ell}_{\text{cur}}(\theta_{\mathcal{P}})$. Notably, the latter consistently surpasses the pre-trained model $\hat{\ell}_{\text{cur}}(\theta_0)$ — (obtained with linear probing). This indicates that there is a good margin to find a point in parameter space that improves pre-training. Vice versa, the upper bound $\sum w_t \hat{\ell}_{\text{cur}}(\theta_t)$ does not exceed pre-training, even with regularization applied, raising the theoretical possibility of *regret* between the composed model and pre-training. This work provides empirical evidence to address it but a theoretical analysis is needed, which we leave for future works.

On the compositional skills of ITA and IEL. We herein investigate *specialization* and *unlearning* capabilities of our approaches. Specifically, given the task

vectors trained on the sequence of tasks, we freeze and use them to edit the composed model. As these adaptations involve simple additions and subtractions of task vectors, we can perform *specialization* and *unlearning* in a zero-shot fashion requiring no examples. In particular, through the former test on specialization, we examine whether composing only a subset of the task vectors can boost the composed model on the respective selected tasks. In doing so, we focus on three tasks of the incremental sequence *i.e.* the first, the central, and the last one. The performance is then measured considering the average FA across the tasks we wish to specialize in (FA_{TGT}) and the average FA across the others (FA_{CTRL}). In the second test, instead, our goal is to eliminate knowledge of a specific task without degrading performance on other tasks (*unlearning*). Specifically, to unlearn a given task, we subtract its associated task vector from the composition, which still includes the weights of the tasks that need to be preserved. Afterwards: *i)* we measure the accuracy for the unlearned task (FA_{TGT}) and the others (FA_{CTRL}); *ii)* the procedure is then repeated for all tasks, with the performance averaged across them. The results are in Table 4.3, where we compare with TMC by [101], *i.e.* model linearization and tangent fine-tuning. Regarding **specialization**, both ITA and TMC improve the accuracy on the target tasks. IEL, instead, leads to a severe drop during specialization. This yields an interesting finding: while Table 4.1 highlights its performance as satisfying, we observe that the ensemble struggles when any of its members are removed. In an era where ensemble models are experiencing renewed attention [98, 110], this evidence should be sobering. On the other hand, we underline the specialization capabilities of ITA, with the best absolute performance on the target tasks. We ascribe it to the devised regularization objective and the natural advantages of the non-linear regime. When focusing on **unlearning**, ITA shows mixed results. Notably, both ITA and TMC consistently reduce the performance of the task to be forgotten. While this also affects the tasks that should be preserved, ITA exhibits a greater distinction between the target and control tasks, indicating a more effective disentanglement of knowledge across tasks.

Table 4.3: Analysis of compositional capabilities. In parentheses, we report the gain (or loss) in accuracy on the target task.

Dataset	Model	*zero-shot specialization*		*zero-shot unlearning*	
		FA_{TGT} [↑]	FA_{CTRL}	FA_{TGT} [↓]	FA_{CTRL}
IN-R	ITA-LoRA	80.83 (+11.40)	50.52	22.77 (−55.02)	52.72 (−25.07)
	IEL-LoRA	73.46 (−06.68)	38.46	18.55 (−61.38)	41.97 (−37.96)
	TMC	69.93 (+08.36)	34.08	45.77 (−14.24)	54.37 (−05.64)
C-100	ITA-LoRA	92.80 (+01.63)	60.06	28.67 (−61.29)	71.96 (−17.99)
	IEL-LoRA	77.77 (−13.22)	37.90	19.48 (−70.05)	56.52 (−33.01)
	TMC	87.53 (+06.49)	45.75	55.63 (−22.79)	71.83 (−06.59)
CUB	ITA-LoRA	90.46 (+05.21)	57.87	68.19 (−17.36)	74.63 (−10.92)
	IEL-LoRA	74.03 (−10.57)	47.95	22.44 (−62.51)	30.99 (−53.96)
	TMC	71.06 (+09.92)	44.93	67.91 (−03.81)	53.22 (−18.50)
Caltech	ITA-LoRA	89.84 (−01.21)	65.47	80.02 (−12.63)	82.40 (−10.25)
	IEL-LoRA	79.70 (−12.99)	62.45	32.00 (−60.19)	42.03 (−50.16)
	TMC	89.23 (+10.63)	49.46	64.52 (−17.78)	73.33 (−08.97)
MIT	ITA-LoRA	89.66 (+08.17)	57.30	36.99 (−49.61)	74.75 (−11.85)
	IEL-LoRA	56.14 (−19.38)	37.54	06.39 (−78.10)	32.57 (−52.02)
	TMC	88.03 (+10.11)	30.56	29.77 (−38.89)	62.03 (−06.63)
RESISC	ITA-LoRA	89.47 (+06.70)	49.75	33.38 (−48.62)	64.06 (−17.94)
	IEL-LoRA	90.13 (+04.92)	48.06	32.77 (−49.76)	65.19 (−17.34)
	TMC	75.77 (+27.63)	17.00	06.64 (−54.02)	52.31 (−08.35)
CropDis.	ITA-LoRA	97.63 (+00.11)	53.05	65.87 (−29.98)	79.24 (−16.61)
	IEL-LoRA	90.12 (−04.68)	48.31	34.01 (−61.87)	54.43 (−41.45)
	TMC	73.60 (+09.23)	15.95	06.24 (−60.32)	53.21 (−13.35)

4.4 Proofs

PROOF OF THEOREM 1

Proof. For the sake of notation, we will use the shortcuts $\ell_0 \triangleq \ell(\theta_0)$, $\mathbf{J}_\ell \triangleq \nabla\ell(\theta_0)$, and $\mathbf{H}_\ell \triangleq \mathbf{H}_\ell(\theta_0)$. Given a setting with only two learners A and B, the second-order approximation of the loss function of the composed model $\mathcal{P}$ is:

$$\ell_{\text{cur}}(\theta_{\mathcal{P}}) = \ell_0 + (\theta_{\mathcal{P}} - \theta_0)^{\text{T}}\mathbf{J}_\ell + \frac{1}{2}(\theta_{\mathcal{P}} - \theta_0)^{\text{T}}\mathbf{H}_\ell(\theta - \theta_0)$$

$$
\begin{aligned}
&= \underbrace{(w_a + w_b)}_{=1}\ell_0 + (\underbrace{\theta_{\mathcal{P}} - \theta_0}_{w_a\tau_a + w_b\tau_b})^{\mathrm{T}}\mathbf{J}_\ell + \frac{1}{2}(\theta_{\mathcal{P}} - \theta_0)^{\mathrm{T}}\mathbf{H}_\ell(\theta - \theta_0) \\
&= w_a\ell_0 + w_b\ell_0 + (w_a\tau_a + w_b\tau_b)^{\mathrm{T}}\mathbf{J}_\ell + \frac{1}{2}(w_a\tau_a + w_b\tau_b)^{\mathrm{T}}\mathbf{H}_\ell(w_a\tau_a + w_b\tau_b) \\
&= w_a\ell_0 + w_b\ell_0 + w_a\tau_a^{\mathrm{T}}\mathbf{J}_\ell + w_b\tau_b^{\mathrm{T}}\mathbf{J}_\ell + \frac{1}{2}(w_a\tau_a^{\mathrm{T}} + w_b\tau_b^{\mathrm{T}})\mathbf{H}_\ell(w_a\tau_a + w_b\tau_b) \\
&= w_a[\ell_0 + \tau_a^{\mathrm{T}}\mathbf{J}_\ell] + w_b[\ell_0 + \tau_b^{\mathrm{T}}\mathbf{J}_\ell] + \frac{1}{2}(w_a\tau_a^{\mathrm{T}}\mathbf{H}_\ell + w_b\tau_b^{\mathrm{T}}\mathbf{H}_\ell)(w_a\tau_a + w_b\tau_b) \\
&= w_a[\ell_0 + \tau_a^{\mathrm{T}}\mathbf{J}_\ell] + w_b[\ell_0 + \tau_b^{\mathrm{T}}\mathbf{J}_\ell] + \frac{1}{2}(w_a^2\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_a + w_b^2\tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_b + \\
&\quad + w_aw_b\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_b + w_aw_b\tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_a)
\end{aligned}
$$

Given that $\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_b$ is a scalar and $(ABC)^{\mathrm{T}} = (C^{\mathrm{T}}B^{\mathrm{T}}A^{\mathrm{T}})$

$$
\rightarrow \tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_b = (\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_b)^{\mathrm{T}} = (\tau_b^{\mathrm{T}}\underbrace{\mathbf{H}_\ell}_{\text{symm.}}{}^{\mathrm{T}}\tau_a) = \tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_a
$$

$$
\begin{aligned}
\ell_{\mathrm{cur}}(\theta_{\mathcal{P}}) &= w_a[\ell_0 + \tau_a^{\mathrm{T}}\mathbf{J}_\ell] + \\
&\quad + w_b[\ell_0 + \tau_b^{\mathrm{T}}\mathbf{J}_\ell] + \frac{1}{2}(w_a^2\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_a + w_b^2\tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_b + 2w_aw_b\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_b) \\
&= w_a[\ell_0 + \tau_a^{\mathrm{T}}\mathbf{J}_\ell] + \\
&\quad + w_b[\ell_0 + \tau_b^{\mathrm{T}}\mathbf{J}_\ell] + \frac{1}{2}(\underbrace{w_a^2\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_a}_{\substack{w_a^2 = w_a(1-w_b) \\ = w_a - w_aw_b}} + w_b^2\tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_b + 2w_aw_b\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_b) \\
&= w_a[\ell_0 + \tau_a^{\mathrm{T}}\mathbf{J}_\ell + \frac{1}{2}\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_a] + w_b[\ell_0 + \tau_b^{\mathrm{T}}\mathbf{J}_\ell] + \frac{1}{2}(w_a\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_a + \\
&\quad + w_b\tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_b - w_aw_b\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_a - w_aw_b\tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_b + 2w_aw_b\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_b) \\
&= w_a[\ell_0 + \tau_a^{\mathrm{T}}\mathbf{J}_\ell + \frac{1}{2}\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_a] + w_b[\ell_0 + \tau_b^{\mathrm{T}}\mathbf{J}_\ell + \\
&\quad + \frac{1}{2}\tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_b] - \frac{1}{2}w_aw_b(\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_a + \tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_b - 2\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_b) \\
&= w_a\ell_{\mathrm{cur}}(\theta_A) + w_b\ell_{\mathrm{cur}}(\theta_B) - \frac{1}{2}w_aw_b(\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_a + \tau_b^{\mathrm{T}}\mathbf{H}_\ell\tau_b - 2\tau_a^{\mathrm{T}}\mathbf{H}_\ell\tau_b) \\
&= w_a\ell_{\mathrm{cur}}(\theta_A) + w_b\ell_{\mathrm{cur}}(\theta_B) - \frac{1}{2}w_aw_b(\tau_a - \tau_b)^{\mathrm{T}}\mathbf{H}_\ell(\tau_a - \tau_b).
\end{aligned}
$$

In the multiple learning setting, we have that $\theta_{\mathcal{P}} = \theta_0 + \tau_{\mathcal{P}} = \theta_0 + \sum_{t=1}^{T} w_t \tau_t$ with $\sum_{t=1}^{T} w_t = 1$. Therefore:

$$\begin{aligned}
\ell_{\text{cur}}(\theta_{\mathcal{P}}) &= \ell_0 + (\theta_{\mathcal{P}} - \theta_0)^{\mathrm{T}} \mathbf{J}_\ell + \frac{1}{2}(\theta_{\mathcal{P}} - \theta_0)^{\mathrm{T}} \mathbf{H}_\ell (\theta - \theta_0) \\
&= \underbrace{\sum_{t=1}^{T} w_t}_{=1} \ell_0 + \tau_{\mathcal{P}}{}^{\mathrm{T}} \mathbf{J}_\ell + \frac{1}{2} \tau_{\mathcal{P}}{}^{\mathrm{T}} \mathbf{H}_\ell \tau_{\mathcal{P}} \\
&= \sum_{t=1}^{T} w_t \ell_0 + (\sum_{t=1}^{T} w_t \tau_t)^{\mathrm{T}} \mathbf{J}_\ell + \frac{1}{2} \tau_{\mathcal{P}}{}^{\mathrm{T}} \mathbf{H}_\ell \tau_{\mathcal{P}} \\
&= \sum_{t=1}^{T} w_t \left[\ell_0 + \tau_t{}^{\mathrm{T}} \mathbf{J}_\ell \right] + \frac{1}{2} \tau_{\mathcal{P}}{}^{\mathrm{T}} \mathbf{H}_\ell \tau_{\mathcal{P}}.
\end{aligned}$$

Let us now focus on the quadratic term:

$$\begin{aligned}
\tau_{\mathcal{P}}{}^{\mathrm{T}} \mathbf{H}_\ell \tau_{\mathcal{P}} &= (\sum_{t=1}^{T} w_t \tau_t)^{\mathrm{T}} \mathbf{H}_\ell (\sum_{t=1}^{T} w_t \tau_t) = (\sum_{t=1}^{T} w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell)(\sum_{t=1}^{T} w_t \tau_t) \\
&= (\sum_{t=1}^{T} w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell)(w_t \tau_t + \sum_{t' \neq t}^{T} w_{t'} \tau_{t'}) = \\
&= \sum_{t=1}^{T} {w_t}^2 \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell \sum_{t' \neq t}^{T} w_{t'} \tau_{t'} \\
&= \sum_{t=1}^{T} {w_t}^2 \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + \sum_{t=1}^{T} w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell \sum_{t' \neq t}^{T} w_{t'} \tau_{t'} \\
&= \sum_{t=1}^{T} {w_t}^2 \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + \sum_{t=1}^{T} \sum_{t' \neq t}^{T} w_t w_{t'} \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'} \\
&\text{Since } \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'} = \tau_{t'}^{\mathrm{T}} \mathbf{H}_\ell \tau_t \text{ (symmetry)} \\
&= \sum_{t=1}^{T} {w_t}^2 \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + 2 \sum_{t,t'<t}^{T} w_t w_{t'} \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'}
\end{aligned}$$

Given that ${w_t}^2 = w_t \cdot w_t = w_t(1 - \sum_{t' \neq t}^{T} w_{t'}) = w_t - w_t \sum_{t' \neq t}^{T} w_{t'}$:

$$
\begin{aligned}
&= \sum_{t=1}^{T} \left(w_t - w_t \sum_{t' \neq t}^{T} w_{t'} \right) \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + 2 \sum_{t=1, t'<t}^{T} w_t w_{t'} \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'} \\
&= \sum_{t=1}^{T} w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t - \sum_{t=1}^{T} \left(w_t \sum_{t' \neq t}^{T} w_{t'} \right) \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + 2 \sum_{t=1, t'<t}^{T} w_t w_{t'} \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'} \\
&= \sum_{t=1}^{T} w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t - \sum_{t=1}^{T} \sum_{t' \neq t}^{T} w_t w_{t'} \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + 2 \sum_{t=1, t'<t}^{T} w_t w_{t'} \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'} \\
&= \sum_{t=1}^{T} w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t - \sum_{t=1, t'<t}^{T} w_t w_{t'} (\tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + \tau_{t'}^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'}) + 2 \sum_{t=1, t'<t}^{T} w_t w_{t'} \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'} \\
&= \sum_{t=1}^{T} w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t - \sum_{t=1, t'<t}^{T} w_t w_{t'} (\tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + \tau_{t'}^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'}) - 2 w_t w_{t'} \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'} \\
&= \sum_{t=1}^{T} w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t - \sum_{t=1, t'<t}^{T} w_t w_{t'} (\tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t + \tau_{t'}^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'} - 2 \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_{t'}) \\
&= \sum_{t=1}^{T} w_t \tau_t^{\mathrm{T}} \mathbf{H}_\ell \tau_t - \sum_{t=1, t'<t}^{T} w_t w_{t'} (\tau_t - \tau_{t'})^{\mathrm{T}} \mathbf{H}_\ell (\tau_t - \tau_{t'})
\end{aligned}
$$

Therefore:

$$\begin{aligned}
\ell_{\text{cur}}(\theta_{\mathcal{P}}) &= \ell_0 + (\theta_{\mathcal{P}} - \theta_0)^{\mathrm{T}}\mathbf{J}_\ell + \frac{1}{2}(\theta_{\mathcal{P}} - \theta_0)^{\mathrm{T}}\mathbf{H}_\ell(\theta - \theta_0) \\
&= \sum_{t=1}^{T} w_t \left[\ell_0 + {\tau_t}^{\mathrm{T}}\mathbf{J}_\ell\right] + \frac{1}{2}{\tau_{\mathcal{P}}}^{\mathrm{T}}\mathbf{H}_\ell\tau_{\mathcal{P}}. \\
&= \sum_{t=1}^{T} w_t \left[\ell_0 + {\tau_t}^{\mathrm{T}}\mathbf{J}_\ell\right] + \frac{1}{2}\sum_{t=1}^{T} w_t\tau_t^{\mathrm{T}}\mathbf{H}_\ell\tau_t + \\
&- \frac{1}{2}\sum_{t=1,t'<t}^{T} w_t w_{t'}(\tau_t - \tau_{t'})^{\mathrm{T}}\mathbf{H}_\ell(\tau_t - \tau_{t'}) \\
&= \sum_{t=1}^{T} w_t \left[\ell_0 + {\tau_t}^{\mathrm{T}}\mathbf{J}_\ell + \frac{1}{2}w_t\tau_t^{\mathrm{T}}\mathbf{H}_\ell\tau_t\right] + \\
&- \frac{1}{2}\sum_{t=1,t'<t}^{T} w_t w_{t'}(\tau_t - \tau_{t'})^{\mathrm{T}}\mathbf{H}_\ell(\tau_t - \tau_{t'}) \\
&= \sum_{t=1}^{T} w_t\ell_{\text{cur}}(\theta_t) - \frac{1}{2}\sum_{t=1,t'<t}^{T} w_t w_{t'}(\tau_t - \tau_{t'})^{\mathrm{T}}\mathbf{H}_\ell(\tau_t - \tau_{t'}),
\end{aligned}$$

which ends the proof of Theorem 1. □

PROOF OF EQ. 4.17

Proof. In the dual-learner setting we have:

$$\Omega(A, B) = \frac{1}{2}w_A w_B(\tau_A - \tau_B)^{\mathrm{T}}\mathbf{H}_\ell(\theta_0)(\tau_A - \tau_B).$$

If we replace the Hessian matrix with the diagonal Fisher matrix $\hat{\mathrm{F}}_{\theta_0}$, we get:

$$\begin{aligned}
\Omega(A, B) &= \frac{1}{2}w_A w_B(\tau_A - \tau_B)^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}(\tau_A - \tau_B) = \\
&= \frac{1}{2}w_A w_B\left[{\tau_A}^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_A - 2{\tau_A}^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_B + {\tau_B}^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_B\right].
\end{aligned}$$

If we focus on the first term inside the parenthesis, we obtain:

$$\begin{aligned}\tau_A{}^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_A &= \tau_A{}^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}^{1/2}\hat{\mathrm{F}}_{\theta_0}^{1/2}\tau_A = (\hat{\mathrm{F}}_{\theta_0}^{1/2}\tau_A)^{\mathrm{T}}(\hat{\mathrm{F}}_{\theta_0}^{1/2}\tau_A) = \|\hat{\mathrm{F}}_{\theta_0}^{1/2}\tau_A\|_2^2 \\ &= \sum_{i=1}^{|\theta_A|}\hat{\mathrm{F}}_{\theta_0}^{(i)}(\tau_A^{(i)})^2 = \sum_{i=1}^{|\theta_A|}\hat{\mathrm{F}}_{\theta_0}^{(i)}(\theta_A^{(i)} - \theta_0^{(i)})^2 = \mathrm{EWC}_{\theta_0}(\theta_A).\end{aligned}$$

Therefore, we can rewrite $\Omega(A, B)$ as:

$$\Omega(A, B) = \frac{1}{2}w_A w_B\left[\mathrm{EWC}_{\theta_0}(\theta_A) + \mathrm{EWC}_{\theta_0}(\theta_B) - 2\tau_A{}^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_B\right].$$

We now generalize to $T \geq 2$. Starting from Eq. 4.14 and replacing the Hessian with the diagonal Fisher approximation $\hat{\mathrm{F}}_{\theta_0}$, we obtain:

$$\begin{aligned}&\frac{1}{2}\sum_{t=1,t'<t}^{T} w_t w_{t'}(\tau_t - \tau_{t'})^{\mathrm{T}}\mathbf{H}_\ell(\tau_t - \tau_{t'}) \approx \\ &\qquad\approx \frac{1}{2}\sum_{t=1,t'<t}^{T} w_t w_{t'}\left[\mathrm{EWC}_{\theta_0}(\theta_t) + \mathrm{EWC}_{\theta_0}(\theta_{t'}) - 2\tau_t^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'}\right] \\ &\qquad= \frac{1}{2}\sum_{t=1,t'<t}^{T} w_t w_{t'}\left[\mathrm{EWC}_{\theta_0}(\theta_t) + \mathrm{EWC}_{\theta_0}(\theta_{t'})\right] - \sum_{t=1,t'<t}^{T} w_t w_{t'}\tau_t^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'} \\ &\qquad= \frac{1}{2}\sum_{t=1,t'\neq t}^{T} w_t w_{t'}\,\mathrm{EWC}_{\theta_0}(\theta_t) - \sum_{t=1,t'<t}^{T} w_t w_{t'}\tau_t^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'} \\ &\qquad= \frac{1}{2}\sum_{t=1}^{T} w_t\,\mathrm{EWC}_{\theta_0}(\theta_t)\underbrace{\sum_{t'\neq t}^{T} w_{t'}}_{1-w_t} - \sum_{t=1,t'<t}^{T} w_t w_{t'}\tau_t^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'} \\ &\qquad= \frac{1}{2}\sum_{t=1}^{T} w_t(1 - w_t)\,\mathrm{EWC}_{\theta_0}(\theta_t) - \sum_{t=1,t'<t}^{T} w_t w_{t'}\tau_t^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'},\end{aligned}$$

which aligns with the result of Eq. 4.17. □

CLOSED FORM GRADIENTS FOR EQ. 4.19

We start by deriving the gradients of $\Omega_{\hat{\mathrm{F}}}(\cdot)$ (Eq. 4.14) w.r.t. the generic task vector τ_k. We initially consider the case of full-fine tuning, and then generalize the results to LoRA and `(IA)`^{3}.

To simplify the calculations, we first rearrange $\Omega_{\hat{\mathrm{F}}}(\cdot)$ as:

$$\begin{aligned}\Omega_{\hat{\mathrm{F}}}(\theta_1,\dots,\theta_T) &= \frac{1}{2}\sum_{t=1}^{T}\sum_{t'<t} w_t w_{t'}(\tau_t-\tau_{t'})^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}(\tau_t-\tau_{t'}) \\ &\approx \frac{1}{2}\sum_{t=1}^{T} w_t(1-w_t)\,\mathrm{EWC}_{\theta_0}(\theta_t) - \sum_{t=1,t'<t}^{T} w_t w_{t'}\tau_t^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'} \quad \text{(see Eq. 4.17)},\end{aligned}$$

Therefore, we can write:

$$\begin{aligned}\frac{\partial\Omega_{\hat{\mathrm{F}}}}{\partial\tau_k} &= \frac{\partial\left[\frac{1}{2}\sum_{t=1}^{T} w_t(1-w_t)\,\mathrm{EWC}_{\theta_0}(\theta_t)\right]}{\partial\tau_k} - \frac{\partial\left[\sum_{t=1,t'<t}^{T} w_t w_{t'}\tau_t^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'}\right]}{\partial\tau_k} \\ &= \frac{\partial\left[\frac{1}{2}w_k(1-w_k)\,\mathrm{EWC}_{\theta_0}(\theta_k)\right]}{\partial\tau_k} - \\ &\qquad - \frac{\partial\left[\sum_{t'<k} w_k w_{t'}\tau_k^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'} + \sum_{t\neq k,t'<t} w_t w_{t'}\tau_t^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'}\right]}{\partial\tau_k} \qquad (4.20) \\ &= w_k(1-w_k)\frac{\partial\left[\frac{1}{2}\,\mathrm{EWC}_{\theta_0}(\theta_k)\right]}{\partial\tau_k} - \sum_{t'\neq k}^{T} w_k w_{t'}\frac{\partial\tau_k^{\mathrm{T}}\hat{\mathrm{F}}_{\theta_0}\tau_{t'}}{\partial\tau_k} \\ &= w_k(1-w_k)\,\hat{\mathrm{F}}_{\theta_0}\odot\tau_k - w_k\sum_{t'\neq k}^{T} w_{t'}\hat{\mathrm{F}}_{\theta_0}\odot\tau'_t \\ &= w_k\left[(1-w_k)\,\hat{\mathrm{F}}_{\theta_0}\odot\tau_k - \sum_{t'\neq k}^{T} w_{t'}\hat{\mathrm{F}}_{\theta_0}\odot\tau'_t\right] \\ &= w_k\left[\hat{\mathrm{F}}_{\theta_0}\odot\left[(1-w_k)\tau_k - \sum_{t'\neq k}^{T} w_{t'}\tau'_t\right]\right]. \qquad (4.21)\end{aligned}$$

We now suppose that training is performed incrementally on a sequence of $1,2,\dots,k,\dots,T$ tasks. During the k-th task, IEL optimizes only τ_k, namely the

task vector instantiated at the beginning of the k-th task. Indeed, as discussed in Section 4.1 (Eq. 4.19), the task vectors $\tau_1, \tau_2, \ldots, \tau_{k-1}$ introduced in preceding tasks are kept frozen. Moreover, at each round, we devise uniform weights, such that $w_t = \frac{1}{k}$, with $t < k$. On top of that, we can rewrite the gradients in Eq. 4.20 as:

$$\frac{\partial \Omega_{\hat{\mathrm{F}}}}{\partial \tau_k} = \frac{1}{k}\left[\hat{\mathrm{F}}_{\theta_0} \odot \left[(1-\frac{1}{k})\tau_k - \frac{1}{k}\sum_{t<k}\tau_t\right]\right] \tag{4.22}$$

Notably, the term $\frac{1}{k}\sum_{t<k}\tau_t$ is a running average of the preceding task vectors. As also discussed in Section 4.5, this reduces the memory/time computational cost for computing the gradients of $\Omega_{\hat{\mathrm{F}}}$ to $\mathcal{O}(1)$.

As a final step, we can exploit the result in Eq. 4.22 to compute the gradients of LoRA ($\tau_k = B_k A_k$) and `(IA)³`. For LoRA, we have that:

$$\frac{\partial \Omega}{\partial B_k} = \frac{\partial \Omega}{\partial \tau_k}\frac{\partial \tau_k}{\partial B_k} = \frac{1}{k}\left[\hat{\mathrm{F}}_{\theta_0} \odot \left[(1-\frac{1}{k})\tau_k - \frac{1}{k}\sum_{t<k}\tau_t\right]\right] A^{\mathrm{T}}$$

$$\frac{\partial \Omega}{\partial A_k} = \frac{\partial \Omega}{\partial \tau_k}\frac{\partial \tau_k}{\partial A_k} = \frac{1}{k}B^{\mathrm{T}}\left[\hat{\mathrm{F}}_{\theta_0} \odot \left[(1-\frac{1}{k})\tau_k - \frac{1}{k}\sum_{t<k}\tau_t\right]\right]$$

In the case of `(IA)³`, where $\tau_k = \theta_0 \odot ((l_k - 1_d) \otimes 1_d)$:

$$\frac{\partial \Omega}{\partial l_k} = \frac{\partial \Omega}{\partial \tau_k}\frac{\partial \tau_k}{\partial l_k} = \left[\frac{1}{k}\left[\hat{\mathrm{F}}_{\theta_0} \odot \left[(1-\frac{1}{k})\tau_k - \frac{1}{k}\sum_{t<k}\tau_t\right]\right] \odot \theta_0^{\mathrm{T}}\right] 1_d$$

where 1_d is a d-dimensional column vector of ones with shape $d \times 1$.

4.5 Computational analysis

Analysis of ITA. Regarding ITA (*i.e.* individual training), the learning phase has constant $\mathcal{O}(1)$ time and memory complexity. As each model is optimized in isolation, the training cost is similar to that of EWC [81] and stems from the compu-

tation and storage of the Fisher Information Matrix, along with the calculations of the penalty term. During the evaluation phase, the time complexity of ITA remains $\mathcal{O}(1)$, as it involves a single forward pass on the composed model $f(\cdot;\theta_{\mathcal{P}})$. Memory complexity, on the other hand, is $\mathcal{O}(T)$ if maintaining separate models with distinct expertise is desired (*e.g.* for later re-composition). Otherwise, this can be avoided by considering the composed model as a cumulative average of individual models (see later), resulting in a memory cost of $\mathcal{O}(1)$.

Analysis of IEL. Referring to IEL (*i.e.* ensemble training), it might initially appear expensive due to theJT of the ensemble. However, the complexity of IEL remains constant to $\mathcal{O}(1)$ in terms of both memory and time. This reduction is intuitive when we view the ensemble as a cumulative average of individual weights. To demonstrate this, we begin by considering the following cascade of models to be learned:

$$
\begin{aligned}
&\text{Task}_{\#1} \rightarrow f(\cdot;\theta_{\mathcal{P}} = \theta_0 + \tau_1)\\
&\text{Task}_{\#2} \rightarrow f(\cdot;\theta_{\mathcal{P}} = \theta_0 + \frac{1}{2}\tau_1 + \frac{1}{2}\tau_2)\\
&\quad \dots\\
&\text{Task}_{\#t} \rightarrow f(\cdot;\theta_{\mathcal{P}} = \theta_0 + \frac{1}{t}\tau_1 + \frac{1}{t}\tau_2 + \dots + \frac{1}{t}\tau_{t-1} + \frac{1}{t}\tau_t)
\end{aligned}
$$

At each task, only the latter component of the composed model is learnable, while the preceding components are frozen:

$$
\text{Task}_{\#t} \rightarrow f(\cdot;\theta_{\mathcal{P}} = \theta_0 + \underbrace{\frac{1}{t}\tau_1 + \frac{1}{t}\tau_2 + \dots \frac{1}{t}\tau_{t-1}}_{\text{frozen components}} + \frac{1}{t}\underbrace{\tau_t}_{\text{learnable}}).
$$

Therefore, as the preceding $\#t-1$ components are kept frozen, we can incorporate them into the initialization weights $\theta_0 \rightarrow \theta_0^{(t)}$ and optimize:

$$
\text{Task}_{\#t} \rightarrow f(\cdot;\theta_{\mathcal{P}} = \theta_0^{(t)} + \frac{1}{t}\tau_t) \quad \text{where} \quad \theta_0^{(t)} = \theta_0 + \frac{1}{t}\sum_{t'=1}^{t-1}\tau_{t'}.
$$

Under this perspective, the learning of the t-th task is comparable to standard fine-tuning with a re-scaling factor $= 1/t$. Therefore, provided that $\theta_0^{(t)}$ is computed once at the beginning of the t-th task, the learning of the t-th task has constant $\mathcal{O}(1)$ time complexity.

It is noted that optimizing Eq. 4.19 via gradient descent can be similarly simplified, resulting in a constant $\mathcal{O}(1)$ time complexity. In fact, the gradients derived in Section 4.4 involve averaging the weights learned during previous tasks, specifically $\frac{1}{t}\sum_{t'=1}^{t-1}\tau_{t'}$. Since the weights from previous tasks are frozen during the current task, we can compute this average once and cache it.

To reduce memory complexity to $\mathcal{O}(1)$, we must avoid storing $\tau_1, \ldots, \tau_{t-1}$ separately. This can be accomplished by assuming an initial null displacement $\tau_{\mathbf{0}} = \mathbf{0}$ and redefining $\theta_0^{(t)}$ as:

$$\theta_0^{(t)} = \theta_0 + \frac{1}{t}\sum_{t'=1}^{t-1}\tau_{t'} = \theta_0 + \tau_{\mathbf{0}} + \frac{1}{t}\sum_{t'=1}^{t-1}\tau_{t'} = \theta_0 + \frac{1}{t}\sum_{t'=0}^{t-1}\tau_{t'} = \\ = \theta_0 + \tau_{\text{AVG}}^{(t)}$$

The term $\tau_{\text{AVG}}^{(t)} = \frac{1}{t}\sum_{t'=0}^{t-1}\tau_{t'}$ is basically the cumulative average of the displacements up to the current task (***excluded***). The cumulative average is straightforward to compute and, as is well-known, eliminates the need to store all previous values appearing in the sum, with resulting memory complexity $\mathcal{O}(1)$.

4.6 Implementation details of ITA and IEL

Task pre-consolidation – Linear Probing. At the beginning of each task, during the pre-consolidation phase, we train a new classification head to account for the classes introduced by the current task. While the rest of the backbone remains frozen, a new linear classification layer is then trained with standard Stochastic Gradient Descent (SGD) for a varying number of epochs, depending on the dataset (typically either 3 or 8 epochs, as detailed in Section 4.9).

With standard linear probing, the newly trained classification head can be con-

sidered reliable only for the classes of the current task. However, it may be miscalibrated for classes from previous tasks. This misalignment could undermine the role of pre-training in regularization. To mitigate this and enforce the hypothesis of pre-training optimality for the global empirical risk **across all tasks**, we adopt a classifier alignment approach inspired by [196]. For each new class, we fit a class-specific Mixture of multivariate Gaussians (MoG) model on the feature representations obtained from the frozen pre-trained model. To capture intra-class variations, we use $K = 5$ Gaussian components per MoG. In subsequent tasks, we generate $N = 256$ synthetic feature vectors using the respective MoGs. These generated feature vectors, encompassing both past and present classes, are then used to fine-tune the new classification head. This approach allows us to enforce the hypothesis of pre-training optimality without relying on input data from other tasks.

Task pre-consolidation – Update of the FIM. Afterward, the diagonal Fisher Information Matrix (FIM) has to be updated to incorporate new information from the current task. Similar to [146], we consider the estimated FIM as an online cumulative average of the squared gradients of the negative log-likelihood. Unlike [146], we do not introduce the hyper-parameter γ to down-weight the importance of the previous estimate. Moreover, following [85], we compute the **true** FIM as defined in Eq. 4.10: *i.e.* taking the expected gradient on the prediction vector $\hat{y}$. This approach differs from the majority of existing methods [146, 81, 31], which apply a further approximation by relying on the empirical FIM: *i.e.* only considering the gradient of the ground truth label. Consequently, our estimate is more accurate but requires multiple backward passes, but the computational impact of this operation can be significantly reduced through batch-wise parallelization as in [50].

Fine-tuning – Initialization. All methods utilize the same pre-training (supervised) on ImageNet21K (‘`vit_base_patch16_224.augreg_in21k`‘ from the ‘`timm`‘ library). Following the original works, we initialize the learnable parameters such that task vectors start from the pre-train initialization:

4.7 Discussion on competing methods

To deliver the most comprehensive and significant comparison with the state of the art in incremental learning, we chose a combination of well-established standard approaches (such as EWC, DER++, L2P, and CODA) and recent proposals emphasizing compositionality skills (*e.g*. TMC and APT) and ensemble learning (*i.e*. SEED). It is important to note that these methods have been heavily influenced by the research trends prevalent at the time they were originally proposed, and therefore, they tend to employ different strategies for fine-tuning the model. To sum up:

- EWC, LwF, DER++, SEED and TMC leverage on full-fine tuning.
- L2P, CODA, APT resort instead to prompting.

Therefore, achieving a direct apple-to-apple comparison is challenging, as the approaches present in the literature are themselves prone to this issue.

We herein summarize the main aspects of the more important recent methods we compared with.

Elastic Weight Consolidation (EWC) As discussed in the main paper, while our approach regularizes the distance in parameter space with respect to θ_0, the regularization in EWC [81] instead focuses on the weights learned during the preceding task. However, beyond the different regularizing strategies, there is another significant difference between ITA and EWC. While ITA fine-tunes each task starting from the same original pre-trained model θ_0, EWC begins with the weights of the previous task.

It is noted that an EWC-like term that protects the last task weights could work as well in our framework based on task vectors. We chose to anchor the model to the pre-training weights to allow for more flexible decentralized learning, wherein multiple task vectors can be trained on different tasks *in parallel*, with minimal interactions. In contrast, an EWC-like term, which regularizes each successor based on its predecessor, would necessitate training the multiple task vectors *in sequence*.

Learning to Prompt (L2P) [174] and **CODA-Prompt** [149] are two continual learning techniques based on prompting. They fine-tune a pre-trained model through a few learnable parameters stored in a prompt pool, which can be either shared or tied to different tasks. At inference time, the prompts are retrieved from the pool through a query-key search. In terms of memory complexity, our ITA is in line with L2P and CODA when coupled with a parameter-efficient fine-tuning technique such as IA3. Moreover, ITA does not require the additional forward pass on the frozen pre-trained model required to retrieve the prompts.

À-la-carte Prompt Tuning (APT) [22] is a prompt-based strategy similar to L2P and CODA. The prompts are trained in isolation (similarly to ITA) and concatenated at inference time to create the composed predictor (no prior query-key search is required). To avoid destructive interference, a tailored masking mechanism is employed in self-attention layers. Differently, our approaches fuse parameters through linear combinations.

Tangent Model Composition (TMC) [101] is a recent approach addressing continual learning through task arithmetic in the tangent space. TMC builds upon task vectors and is similar to ITA (full fine-tuning). They differ in two aspects: *i)* TMC applies a first-order approximation of the forward pass to support compositionality, making it two to three times slower than a non-linear forward pass; *ii)* TMC does not include auxiliary regularization during training.

Selection of Experts for Ensemble Diversification (SEED) [143] is a recent approach that trains an ensemble of models incrementally. For each incoming task, an expert model is chosen from the pool and trained. The major difference with respect to our IEL concerns the inference stage: while IEL performs an ensemble prediction in $\mathcal{O}(1)$ time (thanks to weight averaging), SEED makes inference on all models at test time and averages their predictions.

Model merging. While we address compositionality during training, other approaches focus on post-training techniques, as simple averaging leads to interference [187] when parameters are redundant or have conflicting signs. TIES [187] discards uninformative parameters and addresses conflicts via majority voting. Zipit! [154] merges redundant parameters that produce similar features, while

RegMean [75] exploits a closed-form solution for linear layers. Notably, [113] weighs the contribution of each parameter through the Fisher matrix, computed by each individual model at its optimum. This differs from our approach that evaluates the FIM at the pre-training optimum.

- Full Fine-Tuning: we apply zero-initialization.
- LoRA: we use Gaussian initialization for matrix A and zero initialization for matrix B.
- `(IA)³`: we initialize the vectors l with ones.

Fine-tuning – Loss function. Importantly, while our derivations regard the second-order approximation ℓ_{cur}, the full loss ℓ is instead employed in our algorithms. Indeed, as is common in frameworks similar to ours [31, 119], the Taylor approximation is employed to build a surrogate of the exact loss that is both accurate and mathematically tractable. However, when it comes to practice, this proxy is often relaxed, and the full target function is used instead for simplicity.

Following existing works [149, 174], we employ the **local cross-entropy loss** as the learning objective. Given an example from the current task, while the standard cross-entropy loss considers logits related to all classes, including those learned in previous tasks, the local cross-entropy focuses only on logits corresponding to the classes introduced in the current task. This approach prevents the logits of past classes from being overly penalized during the current task [27, 20]. To ensure a fair comparison in our experiments, we apply this modification to other competing methods (*e.g.* EWC [81]).

Fine-tuning – Optimization. In each experiment, we use the AdamW optimizer [103] with a learning rate of 3×10^{-4} for LoRA and `(IA)³` fine-tuning, and 1×10^{-4} for full fine-tuning. Importantly, in both ITA and IEL, we employ a decoupled strategy [103] to incorporate the gradients of the regularization term. Specifically, we apply the gradients of the regularizing objectives directly to the parameters before the gradient update step, ensuring that the regularization term does not interfere with momentum in the optimization dynamics. By

adopting this approach, we observe an empirical improvement in final accuracy, attributed to a more effective minimization of the Riemannian distance relative to the pre-training initialization (see Eq. 4.12). Finally, we apply this decoupled gradient update exclusively to LoRA and `(IA)`^{3}. For full fine-tuning, we refrain from using it as we observed numerical instabilities (*i.e.* exploding loss).

Furthermore, we decouple the regularization strength applied to the final classification layer from that applied to the rest of the learnable parameters. This introduces two additional hyper-parameters: α_{CLS} for ITA and β_{CLS} for IEL. Intuitively, by decoupling the regularization weights, we can increase the regularization strength of intermediate layers without causing numerical instabilities, which often stem from the final classification layer.

4.8 Datasets

We conduct a comprehensive evaluation using a variety of benchmarks. Following the current literature on pre-trained CL models [174, 172, 149], we include conventional image datasets such as Split CIFAR-100 and Split ImageNet-R. We also include Split CUB-200, Split Caltech-256 and Split MIT-67, recently used in the context of composable incremental methods [22, 101]. Finally, we assess the adaptability of these pre-trained methods in settings with decreasing domain similarity to the ImageNet pre-training utilized by our backbone model [123, 40]. Specifically, these settings include the satellite and medical domains, represented by Split RESISC45 and Split CropDiseases, respectively. In the following, we outline the due details:

- **Standard domains**: *Split CIFAR-100* [84] and *Split ImageNet-R* [63], with respectively 100 and 200 classes split into 10 tasks. We train each task of Split ImageNet-R for 30 epochs and each task of Split CIFAR-100 for 20 epochs. In particular, IN-R is a variant of the ImageNet dataset that includes artistic renditions such as sketches, cartoons, and paintings. It is used to evaluate the robustness and generalization capabilities of models trained on the original ImageNet when tested on out-of-distribution

data. Following [101], we also employ Split Caltech-256 [55] and Split MIT-67 [133], dividing both into 10 tasks (5 epoch each).

- **Specialized domain**: We adopt *Split CUB-200* [166] to evaluate compositional capabilities in a more fine-grained classification scenario, namely recognizing 200 species of birds. The classes are split across 10 tasks, each lasting for 50 epochs.
- **Aerial domain**: we use *Split RESISC45* [35], which comprises 30000 RGB satellite images for land use and land cover classification. The dataset contains 45 classes (*e.g.* airport, cloud, island, and so on) divided into 9 tasks, with each task lasting 30 epochs.
- **Medical domain**: we finally explore the medical setting (*i.e.* plant diseases) and conduct experiments on *Split CropDiseases* [68]. It regards infected leaves with 7 tasks of 5 classes each (5 epochs).

We base our code on Mammoth [25, 24], a widely adopted framework in the class-incremental learning literature.

4.9 Hyperparameters

The hyperparameters employed for each experimenmt are reported in Tables 4.4 to 4.10.

Table 4.4: Hyperparameters for each method tested on **ImageNet-R**.

Method	Hyperparameters
CODA-Prompt	$\text{lr} = 1.0 \times 10^{-3}$
DER++	$\alpha = 3.0 \times 10^{-1}$; $\beta = 8.0 \times 10^{-1}$; $\text{lr} = 1.0 \times 10^{-3}$
EWC	$\varepsilon = 1.0 \times 10^{2}$; $\gamma = 1.0$; $\text{lr} = 1.0 \times 10^{-2}$
L2P	$\text{lr} = 2.5 \times 10^{-3}$
LWF-MC	$\text{wd} = 0.0$; $\text{lr} = 1.0 \times 10^{-2}$
SEED	$\text{lr} = 3.0 \times 10^{-4}$
SGD	$\text{lr} = 3.0 \times 10^{-2}$
TMC	$\text{lr} = 1.0 \times 10^{-4}$
InfLoRA	$\text{lr} = 5.0 \times 10^{-4}$; $r = 10$; $\varepsilon = 0.98$
Shared (ITA, IEL): $\#\,\text{epochs}_{\text{pre-tuning}} = 3$; $\text{lr}_{\text{pre-tuning}} = 1.0 \times 10^{-2}$.	
ITA-FFT	$\alpha = 5.0 \times 10^{1}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 1.0 \times 10^{-4}$
ITA-LoRA	$\alpha = 2.0 \times 10^{-2}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-3}$; $\text{lr} = 3.0 \times 10^{-4}$
ITA-`(IA)`^{3}	$\alpha = 7.0 \times 10^{-1}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-3}$
IEL-FFT	$\beta = 5.0 \times 10^{1}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 3.0 \times 10^{-3}$; $\text{lr} = 1.0 \times 10^{-4}$
IEL-LoRA	$\beta = 2.0 \times 10^{1}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-3}$; $\text{lr} = 3.0 \times 10^{-4}$
IEL-`(IA)`^{3}	$\beta = 2.0 \times 10^{1}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-3}$

Table 4.5: Hyperparameters for each method tested on **CIFAR-100**.

Method	Hyperparameters
CODA-Prompt	$\text{lr} = 1.0 \times 10^{-3}$
DER++	$\alpha = 3.0 \times 10^{-1}$; $\beta = 8.0 \times 10^{-1}$; $\text{lr} = 1.0 \times 10^{-4}$
EWC	$\varepsilon = 1.0 \times 10^{2}$; $\gamma = 1.0$; $\text{lr} = 1.0 \times 10^{-3}$
L2P	$\text{lr} = 2.5 \times 10^{-3}$
LWF-MC	$\text{wd} = 0.0$; $\text{lr} = 1.0 \times 10^{-2}$
SEED	$\text{lr} = 3.0 \times 10^{-4}$
SGD	$\text{lr} = 1.0 \times 10^{-2}$
TMC	$\text{lr} = 1.0 \times 10^{-4}$
InfLoRA	$\text{lr} = 5.0 \times 10^{-4}$; $r = 10$; $\varepsilon = 0.95$
Shared (ITA, IEL): $\#\,\text{epochs}_{\text{pre-tuning}} = 3$; $\text{lr}_{\text{pre-tuning}} = 1.0 \times 10^{-2}$.	
ITA-FFT	$\alpha = 2.0 \times 10^{3}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 3.0 \times 10^{-3}$; $\text{lr} = 1.0 \times 10^{-4}$
ITA-LoRA	$\alpha = 2.0 \times 10^{-2}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-3}$; $\text{lr} = 3.0 \times 10^{-4}$
ITA-`(IA)`^{3}	$\alpha = 2.0 \times 10^{-2}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 3.0 \times 10^{-3}$; $\text{lr} = 3.0 \times 10^{-3}$
IEL-FFT	$\beta = 2.0 \times 10^{3}$; $\beta_{\text{CLS}} = 2.5 \times 10^{-2}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 1.0 \times 10^{-4}$
IEL-LoRA	$\beta = 2.0 \times 10^{1}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-3}$; $\text{lr} = 3.0 \times 10^{-4}$
IEL-`(IA)`^{3}	$\beta = 2.0 \times 10^{1}$; $\beta_{\text{CLS}} = 2.5 \times 10^{-2}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-3}$; $\text{lr} = 3.0 \times 10^{-4}$

Table 4.6: Hyperparameters for each method tested on **CUB-200**.

Method	Hyperparameters
CODA-Prompt	$\text{lr} = 1.0 \times 10^{-3}$
DER++	$\alpha = 3.0 \times 10^{-1}$; $\beta = 8.0 \times 10^{-1}$; $\text{lr} = 1.0 \times 10^{-3}$
EWC	$\varepsilon = 1.0 \times 10^{1}$; $\gamma = 9.0 \times 10^{-1}$; $\text{lr} = 1.0 \times 10^{-2}$
L2P	$\text{lr} = 2.5 \times 10^{-3}$
LWF-MC	$\text{wd} = 1.0 \times 10^{-4}$; $\text{lr} = 1.0 \times 10^{-2}$
SEED	$\text{lr} = 3.0 \times 10^{-4}$
SGD	$\text{lr} = 3.0 \times 10^{-2}$
TMC	$\text{lr} = 1.0 \times 10^{-4}$
InfLoRA	$\text{lr} = 5.0 \times 10^{-4}$; $r = 10$; $\varepsilon = 0.98$
Shared (ITA, IEL): $\text{lr}_{\text{pre-tuning}} = 1.0 \times 10^{-2}$; $\#\,\text{epochs}_{\text{pre-tuning}} = 8$.	
ITA-FFT	$\alpha = 5.0 \times 10^{2}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 1.0 \times 10^{-4}$
ITA-LoRA	$\alpha = 5.0$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-4}$
ITA-`(IA)³`	$\alpha = 7.0 \times 10^{-1}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-3}$
IEL-FFT	$\beta = 5.0 \times 10^{3}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 1.0 \times 10^{-5}$
IEL-LoRA	$\beta = 2.0 \times 10^{1}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-4}$
IEL-`(IA)³`	$\beta = 5.0 \times 10^{2}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-4}$

Table 4.7: Hyperparameters for each method tested on **Caltech-256**.

Method	Hyperparameters
CODA-Prompt	$\text{lr} = 1.0 \times 10^{-3}$
DER++	$\alpha = 3.0 \times 10^{-1}$; $\beta = 8.0 \times 10^{-1}$; $\text{lr} = 1.0 \times 10^{-3}$
EWC	$\varepsilon = 1.0 \times 10^{2}$; $\gamma = 1.0$; $\text{lr} = 1.0 \times 10^{-2}$
L2P	$\text{lr} = 2.5 \times 10^{-3}$
LWF-MC	$\text{wd} = 0.0$; $\text{lr} = 1.0 \times 10^{-2}$
SEED	$\text{lr} = 3.0 \times 10^{-4}$
SGD	$\text{lr} = 1.0 \times 10^{-2}$
TMC	$\text{lr} = 1.0 \times 10^{-4}$
InfLoRA	$\text{lr} = 5.0 \times 10^{-4}$; $r = 10$; $\varepsilon = 0.99$
Shared (ITA, IEL): $\text{lr}_{\text{pre-tuning}} = 1.0 \times 10^{-2}$.	
ITA-FFT	$\#\,\text{epochs}_{\text{pre-tuning}} = 3$; $\alpha = 2.0 \times 10^{3}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 1.0 \times 10^{-4}$
ITA-LoRA	$\#\,\text{epochs}_{\text{pre-tuning}} = 3$; $\alpha = 2.0 \times 10^{-2}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 3.0 \times 10^{-3}$; $\text{lr} = 3.0 \times 10^{-4}$
ITA-`(IA)³`	$\#\,\text{epochs}_{\text{pre-tuning}} = 8$; $\alpha = 7.0 \times 10^{-1}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-3}$
IEL-FFT	$\#\,\text{epochs}_{\text{pre-tuning}} = 3$; $\beta = 5.0 \times 10^{1}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 1.0 \times 10^{-4}$
IEL-LoRA	$\#\,\text{epochs}_{\text{pre-tuning}} = 3$; $\beta = 2.0 \times 10^{1}$; $\beta_{\text{CLS}} = 5.0$; $\beta_{\text{CLS-prior}} = 3.0 \times 10^{-3}$; $\text{lr} = 3.0 \times 10^{-4}$
IEL-`(IA)³`	$\#\,\text{epochs}_{\text{pre-tuning}} = 3$; $\beta = 2.0 \times 10^{1}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 3.0 \times 10^{-3}$; $\text{lr} = 3.0 \times 10^{-4}$

Table 4.8: Hyperparameters for each method tested on **MIT-67**.

Method	Hyperparameters
CODA-Prompt	$\text{lr} = 1.0 \times 10^{-3}$
DER++	$\alpha = 3.0 \times 10^{-1}; \beta = 8.0 \times 10^{-1}; \text{lr} = 1.0 \times 10^{-3}$
EWC	$\varepsilon = 1.0 \times 10^{2}; \gamma = 1.0; \text{lr} = 1.0 \times 10^{-3}$
L2P	$\text{lr} = 2.5 \times 10^{-3}$
LWF-MC	$\text{wd} = 0.0; \text{lr} = 1.0 \times 10^{-2}$
SEED	$\text{lr} = 3.0 \times 10^{-4}$
SGD	$\text{lr} = 1.0 \times 10^{-2}$
TMC	$\text{lr} = 1.0 \times 10^{-4}$
InfLoRA	$\text{lr} = 1.0 \times 10^{-3}; r = 10; \varepsilon = 0.95$
Shared (ITA, IEL): $\text{lr}_{\text{pre-tuning}} = 1.0 \times 10^{-2}$.	
ITA-FFT	$\#\,\text{epochs}_{\text{pre-tuning}} = 8; \alpha = 5.0 \times 10^{3}; \alpha_{\text{CLS}} = 1.0 \times 10^{-1}; \alpha_{\text{CLS-prior}} = 1.0 \times 10^{-1}; \text{lr} = 1.0 \times 10^{-4}$
ITA-LoRA	$\#\,\text{epochs}_{\text{pre-tuning}} = 8; \alpha = 5.0; \alpha_{\text{CLS}} = 1.0 \times 10^{-1}; \alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}; \text{lr} = 3.0 \times 10^{-4}$
ITA-`(IA)`^{3}	$\#\,\text{epochs}_{\text{pre-tuning}} = 8; \alpha = 5.0; \alpha_{\text{CLS}} = 1.0 \times 10^{-1}; \alpha_{\text{CLS-prior}} = 3.0 \times 10^{-3}; \text{lr} = 3.0 \times 10^{-4}$
IEL-FFT	$\#\,\text{epochs}_{\text{pre-tuning}} = 8; \beta = 5.0 \times 10^{3}; \beta_{\text{CLS}} = 1.0 \times 10^{-1}; \beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}; \text{lr} = 1.0 \times 10^{-4}$
IEL-LoRA	$\#\,\text{epochs}_{\text{pre-tuning}} = 3; \beta = 2.0 \times 10^{1}; \beta_{\text{CLS}} = 1.0 \times 10^{-1}; \beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}; \text{lr} = 3.0 \times 10^{-4}$
IEL-`(IA)`^{3}	$\#\,\text{epochs}_{\text{pre-tuning}} = 3; \beta = 2.0 \times 10^{1}; \beta_{\text{CLS}} = 1.0 \times 10^{-1}; \beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}; \text{lr} = 3.0 \times 10^{-4}$

Table 4.9: Hyperparameters for each method tested on **RESISC**.

Method	Hyperparameters
CODA-Prompt	$\text{lr} = 1.0 \times 10^{-3}$
DER++	$\alpha = 3.0 \times 10^{-1}; \beta = 8.0 \times 10^{-1}; \text{lr} = 1.0 \times 10^{-3}$
EWC	$\varepsilon = 1.0 \times 10^{2}; \gamma = 9.0 \times 10^{-1}; \text{lr} = 1.0 \times 10^{-2}$
L2P	$\text{lr} = 2.5 \times 10^{-3}$
LWF-MC	$\text{wd} = 0.0; \text{lr} = 1.0 \times 10^{-2}$
SEED	$\text{lr} = 3.0 \times 10^{-4}$
SGD	$\text{lr} = 1.0 \times 10^{-2}$
TMC	$\text{lr} = 3.0 \times 10^{-4}$
InfLoRA	$\text{lr} = 5.0 \times 10^{-4}; r = 10; \varepsilon = 0.98$
Shared (ITA, IEL): $\#\,\text{epochs}_{\text{pre-tuning}} = 8; \text{lr}_{\text{pre-tuning}} = 1.0 \times 10^{-2}$.	
ITA-FFT	$\alpha = 1.0 \times 10^{4}; \alpha_{\text{CLS}} = 1.0 \times 10^{-1}; \alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}; \text{lr} = 1.0 \times 10^{-4}$
ITA-LoRA	$\alpha = 2.0 \times 10^{-2}; \alpha_{\text{CLS}} = 1.0 \times 10^{-1}; \alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}; \text{lr} = 3.0 \times 10^{-4}$
ITA-`(IA)`^{3}	$\alpha = 5.0; \alpha_{\text{CLS}} = 1.0 \times 10^{-1}; \alpha_{\text{CLS-prior}} = 3.0 \times 10^{-3}; \text{lr} = 3.0 \times 10^{-3}$
IEL-FFT	$\beta = 5.0 \times 10^{3}; \beta_{\text{CLS}} = 1.0 \times 10^{-1}; \beta_{\text{CLS-prior}} = 3.0 \times 10^{-3}; \text{lr} = 1.0 \times 10^{-4}$
IEL-LoRA	$\beta = 2.0 \times 10^{2}; \beta_{\text{CLS}} = 1.0 \times 10^{-2}; \beta_{\text{CLS-prior}} = 1.0 \times 10^{-3}; \text{lr} = 3.0 \times 10^{-4}$
IEL-`(IA)`^{3}	$\beta = 2.0; \beta_{\text{CLS}} = 1.0 \times 10^{-2}; \beta_{\text{CLS-prior}} = 1.0 \times 10^{-3}; \text{lr} = 3.0 \times 10^{-4}$

Table 4.10: Hyperparameters for each method tested on **CropDisease**.

Method	Hyperparameters
CODA-Prompt	$\text{lr} = 1.0 \times 10^{-3}$
DER++	$\alpha = 3.0 \times 10^{-1}$; $\beta = 8.0 \times 10^{-1}$; $\text{lr} = 1.0 \times 10^{-3}$
EWC	$\varepsilon = 1.0$; $\gamma = 1.0$; $\text{lr} = 1.0 \times 10^{-2}$
L2P	$\text{lr} = 2.5 \times 10^{-3}$
LWF-MC	$\text{wd} = 1.0 \times 10^{-4}$; $\text{lr} = 1.0 \times 10^{-2}$
SEED	$\text{lr} = 3.0 \times 10^{-4}$
SGD	$\text{lr} = 1.0 \times 10^{-2}$
TMC	$\text{lr} = 1.0 \times 10^{-4}$
InfLoRA	$\text{lr} = 5.0 \times 10^{-4}$; $r = 10$; $\varepsilon = 0.98$
Shared (ITA, IEL): $\#\,\text{epochs}_{\text{pre-tuning}} = 8$; $\text{lr}_{\text{pre-tuning}} = 1.0 \times 10^{-2}$.	
ITA-FFT	$\alpha = 5.0 \times 10^{3}$; $\alpha_{\text{CLS}} = 2.5 \times 10^{-2}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 1.0 \times 10^{-4}$
ITA-LoRA	$\alpha = 5.0$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-4}$
ITA-`(IA)`^{3}	$\alpha = 2.0 \times 10^{-2}$; $\alpha_{\text{CLS}} = 1.0 \times 10^{-1}$; $\alpha_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-4}$
IEL-FFT	$\beta = 2.0 \times 10^{2}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 1.0 \times 10^{-5}$
IEL-LoRA	$\beta = 5.0 \times 10^{1}$; $\beta_{\text{CLS}} = 1.0 \times 10^{-1}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-4}$
IEL-`(IA)`^{3}	$\beta = 2.0$; $\beta_{\text{CLS}} = 2.5 \times 10^{-2}$; $\beta_{\text{CLS-prior}} = 1.0 \times 10^{-2}$; $\text{lr} = 3.0 \times 10^{-4}$

Discussion of limitations and future directions

Second-order Taylor approximations are valid tools to make the theory more tractable. For instance, a similar approach was used by [119] to support the importance of wider local minima against forgetting. Nonetheless, these approximations often rely on certain concessions, the first regarding their validity during training. Since our approximation is performed at the pre-training weights, it may become inaccurate as parameters drift. While importance-based terms like ours may partially mitigate the drift, other techniques could be used (*e.g.* a L2-norm penalty on task vectors or low learning rate). Even without explicit countermeasures, existing studies suggest that deep networks often fall into a regime where the loss tends to be almost convex in a neighborhood around their local minima [54, 104, 193]. Also, under some conditions, their training dynamics can enter the lazy regime [36, 71], where these models rapidly achieve near-zero loss with minimal changes to their weights.

Moreover, two other concessions warrant a discussion: *approx. i)* we implicitly model the weight distribution with a Gaussian [31] (Eq. 4.9); *approx. ii)* the Fisher matrix is approximated by its diagonal (Eq. 4.11). Regarding *approx. i)*, while non-Gaussian posteriors are often cumbersome to manage, [47] challenge the common belief that mean-field approximations are overly restrictive for deep networks. Moreover, although *approx. ii)* may seem crude, the diagonal approximation is efficient, with a low memory footprint. Still, our approach could profit from more accurate estimations of the Hessian, like the Kronecker factored approximation [112], which considers the interactions between weights of the same layer.

Future work. Through theoretical and empirical analyses, we support the importance of remaining within the pre-training basin to achieve composable deep networks. However, there is still much to explore along this path. We mainly focus on closed-set classification models but it would be noteworthy to extend our analysis to self-supervised pre-training and open-vocabulary models like CLIP. Indeed, recent works [203] have shown their tendency to forget zero-shot pre-training capabilities while fine-tuning. In this respect, our second-order regularization could significantly aid compositionality in CLIP-based models. Finally, we believe that staying within the pre-train basin is only one aspect to consider; future research should emphasize the **exploration** of the pre-train basin, to achieve composable modules with a higher degree of specialization on their respective tasks.

5

Conclusions

This thesis investigated the challenges of designing Continual Learning methods capable of operating effectively under noisy supervision and through compositional mechanisms. The work presented herein focused on two distinct research directions: addressing the vulnerability of rehearsal-based continual learning to label noise, and enabling modular, composable adaptation of pre-trained models through second-order analysis.

The work began in **Chapter 2** with an in-depth analysis of how label noise propagates and accumulates in rehearsal-based continual learning systems. This research led to the development of Alternate Experience Replay (AER), a novel training strategy that deliberately exploits forgetting dynamics to maintain separability between clean and corrupted samples in the memory buffer. By alternating between buffer learning and buffer forgetting phases, coupled with Asymmetric Balanced Sampling (ABS), the proposed approach demonstrated that forgetting—traditionally viewed as an obstacle in continual learning—can be transformed into a mechanism for noise detection and buffer purification.

In **Chapter 3**, the thesis expanded this investigation to more realistic noise scenarios through EARL (Embracing Amnesic Replay for Learning with Noisy

Labels). This work provided theoretical grounding for the forgetting dynamics under noise, validated the approach across multiple noise sources including human annotations, web-scraped labels, and machine-generated annotations, and extended the evaluation beyond vision to natural language understanding tasks. Crucially, this research demonstrated that the amnesic replay mechanism remains effective across modern pre-trained architectures and prompt-based continual learning baselines, establishing robustness to noisy supervision as a fundamental requirement rather than an auxiliary concern.

The thesis then shifted to a complementary perspective on continual learning, with **Chapter 4** exploring model compositionality through the lens of task arithmetic and second-order optimization. This research addressed a fundamental question: under what conditions can models fine-tuned on different tasks be meaningfully combined? By deriving a second-order Taylor approximation of the loss around pre-trained weights, the work established a principled relationship between individual model performance and their composition, highlighting that successful compositionality requires each module to preserve out-of-distribution accuracy—a continual learning property. This insight led to two dual algorithms: Incremental Task Arithmetic (ITA), which optimizes individual task modules with Fisher-based regularization, and Incremental Ensemble Learning (IEL), which directly optimizes the composed model while encouraging alignment between task vectors. Both approaches demonstrated that compositional capabilities and continual learning objectives are intrinsically connected, offering new perspectives on modular adaptation, task specialization, and selective unlearning.

Overall, this thesis advances continual learning by tackling two underexplored issues: robustness to noisy supervision in rehearsal-based methods and the theoretical conditions enabling compositional model adaptation. The proposed amnesic replay mechanisms demonstrate that controlled forgetting can maintain buffer quality across diverse noise regimes, while the second-order analysis of task arithmetic clarifies when and why model composition is theoretically sound.

At the same time, these contributions reveal intrinsic limitations of current approaches. Noise-robust replay relies on loss-based separation, which can deterio-

rate under strong instance-dependent corruptions, and it remains inherently tied to rehearsal, raising scalability and privacy concerns due to data storage. Similarly, the compositional framework assumes models remain close to their pre-training state, an assumption that weakens as tasks accumulate or as domain shifts grow. Despite these constraints, the work shows that explicitly leveraging learning dynamics—through selective forgetting and geometry-aware regularization—provides a principled path toward continual learning systems that remain reliable under realistic, imperfect conditions.

Ph.D. Activities

This final section presents a list of the main activities carried out by the candidate during the Ph.D. program in Information and Communication Technologies.

EXCHANGE PERIODS

14 April - 11 October 2025: Visiting PhD Student at the Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI) of the University of Granada (Spain), supervised by Professor Natalia Díaz-Rodríguez.

TEACHING ACTIVITIES

2021 - 2024: Teaching Assistant for "Machine Learning and Deep Learning" course held by Prof. Simone Calderara;

2024: Lecturer for the "Machine Learning and Deep Learning" executive program held by the BI-REX Consortium

2024: Lecturer for the "AI and ML for Smart Factory" intensive master

CONFERENCE ATTENDANCES

May 29th - 31st, 2023 3rd CINI National Conference on Artificial Intelligence (Ital-IA), 2023, *Pisa, Italy*;

July 21st - 27th, 2024 41st International Conference on Machine Learning (ICML), 2024, *Vienna, Austria*;

November 25th - 28th, 2024 35th British Machine vision Conference (BMVC), 2024, *Glasgow, UK*.

SEMINARS AND WORKSHOPS

November 2022: Attendance at "Graph Signal Processing for Machine Learning: Challenges and Use-cases" seminar, speaker: Prof. Laura Toni;

November 2022: Attendance at "Digital Humanities and Artificial Intelligence for humans in today society" seminar, speaker: Prof. Rita Cucchiara;

December 2022: Attendance at "3D Computer Vision for Animals" seminar, speaker: Prof. Silvia Zuffi;

June 2023: Attendance at "Academic English Workshop II" course, speaker: Prof. Silvia Cavalieri;

May 2025: Attendance at "Automatic Machine Learning for Multi-Target Task"seminar, speaker: Prof. Ricardo Cerri;

June 2025: Attendance at "Advanced Training Strategies for Incremental and Decentralized Learning" 10 hours course, speaker: Prof. Angelo Porrello;

SCHOOLS

September 4th - 8th, 2023 International Summer School on Machine Vision (VISMAC), 2023, *Padova, Italy*.

September 18th - 22nd, 2023 ELLIS Summer School on Large-Scale AI for Research and Industry, 2023, *Modena, Italy*.

TECHNICAL PROGRAM COMMITTEES

Conferences

18th European Conference on Computer Vision ECCV 2024 (ECCV), 2024, *Milan, Italy*;

13th International Conference on Learning Representations (ICLR), 2025, *Singapore, Asia*;

November 25th - 28th, 2024 35th British Machine vision Conference (BMVC), 2024, *Glasgow, UK*.

List of Publications

[1] Idrissi, B. Y., Millunzi, M., Sorrenti, A., Baraldi, L., and Dementieva, D. (2025). Temperature matters: Enhancing watermark robustness against paraphrasing attacks. *arXiv preprint arXiv:2506.22623*.

[2] Millunzi, M., Bonicelli, L., Porrello, A., Credi, J., Kolm, P. N., and Calderara, S. (2024). May the forgetting be with you: Alternate replay for learning with noisy labels. In *35th British Machine Vision Conference (BMVC 2024)*, Glasgow, UK. BMVA Press.

[3] Millunzi, M., Bonicelli, L., Porrello, A., Credi, J., Kolm, P. N., and Calderara, S. (2026). Earl: Embracing amnesic replay for learning with noisy labels. *Pattern Recognition*, 179:113514.

[4] Millunzi, M., Bonicelli, L., Zurli, A., Salman, A., Credi, J., and Calderara, S. (2023). Novel continual learning techniques on noisy label datasets. In *Ital-IA 2023 Thematic Workshops, co-located with the 3rd CINI National Lab AIIS Conference on Artificial Intelligence*, volume 3486 of *CEUR Workshop Proceedings*, pages 517–521, Pisa, Italy. CEUR-WS.org.

[5] Porrello, A., Bonicelli, L., Buzzega, P., Millunzi, M., Calderara, S., and Cucchiara, R. (2025). A second-order perspective on model compositionality and incremental learning. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, Singapore. OpenReview.net.

Bibliography

[6] Abati, D., Tomczak, J., Blankevoort, T., Calderara, S., Cucchiara, R., and Bejnordi, B. E. (2020). Conditional channel gated networks for task-aware continual learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[7] Ahn, H., Kwak, J., Lim, S., Bang, H., Kim, H., and Moon, T. (2021). SS-IL: Separated Softmax for Incremental Learning. In *IEEE International Conference on Computer Vision*.

[8] Aljundi, R., Kelchtermans, K., and Tuytelaars, T. (2019a). Task-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11254–11263.

[9] Aljundi, R., Lin, M., Goujaud, B., and Bengio, Y. (2019b). Gradient based sample selection for online continual learning. *Advances in Neural Information Processing Systems*.

[10] Amid, E., Warmuth, M. K., Anil, R., and Koren, T. (2019). Robust bi-tempered logistic loss based on bregman divergences. In *Proc. NeurIPS*, pages 14987–14996.

[11] Arazo, E., Ortego, D., Albert, P., O'Connor, N., and McGuinness, K. (2019). Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*.

[12] Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. (2017). A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR.

[13] Asadi, N., Beitollahi, M., Khalil, Y., Li, Y., Zhang, G., and Chen, X. (2024). Does combining parameter-efficient modules improve few-shot transfer accuracy? *arXiv preprint arXiv:2402.15414*.

[14] Bang, J., Kim, H., Yoo, Y., Ha, J.-W., and Choi, J. (2021). Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[15] Bang, J., Koh, H., Park, S., Song, H., Ha, J.-W., and Choi, J. (2022). Online continual learning on a contaminated data stream with blurry task boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[16] Bentivogli, L., Dagan, I., Dang, H. T., Giampiccolo, D., and Magnini, B. (2009). The fifth pascal recognizing textual entailment challenge. In *Proceedings of TAC*.

[17] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*.

[18] Biderman, D., Portes, J., Ortiz, J. J. G., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., Chiley, V., Frankle, J., Blakeney, C., and Cunningham, J. P. (2024). LoRA learns less and forgets less. *Transactions on Machine Learning Research*.

[19] Bonicelli, L., Boschini, M., Porrello, A., Spampinato, C., and Calderara, S. (2022). On the Effectiveness of Lipschitz-Driven Rehearsal in Continual Learning. In *Advances in Neural Information Processing Systems*.

[20] Boschini, M., Bonicelli, L., Buzzega, P., Porrello, A., and Calderara, S. (2022a). Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[21] Boschini, M., Bonicelli, L., Porrello, A., Bellitto, G., Pennisi, M., Palazzo, S., Spampinato, C., and Calderara, S. (2022b). Transfer without forgetting. In *Proceedings of the European Conference on Computer Vision*.

[22] Bowman, B., Achille, A., Zancato, L., Trager, M., Perera, P., Paolini, G., and Soatto, S. (2023). A-la-carte prompt tuning (apt): Combining distinct data via composable prompting. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[23] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, pages 632–642.

[24] Buzzega, P., Boschini, M., Bonicelli, L., and Porrello, A. (2020a). Mammoth - an extendible (general) continual learning framework for pytorch.

[25] Buzzega, P., Boschini, M., Porrello, A., Abati, D., and Calderara, S. (2020b). Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems*.

[26] Buzzega, P., Boschini, M., Porrello, A., and Calderara, S. (2020c). Rethinking Experience Replay: a Bag of Tricks for Continual Learning. In *International Conference on Pattern Recognition*.

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

[28] Cermelli, F., Mancini, M., Bulo, S. R., Ricci, E., and Caputo, B. (2020). Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9233–9242.

[29] Cha, H., Lee, J., and Shin, J. (2021). Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 9516–9525.

[30] Chang, H.-S., Learned-Miller, E., and McCallum, A. (2017). Active Bias: Training more accurate neural networks by emphasizing high variance samples. In *Proc. NeurIPS*, pages 1002–1012.

[31] 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*.

[32] Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. (2019). On tiny episodic memories in continual learning. In *International Conference on Machine Learning Workshop*.

[33] Chen, H., Wu, Z., Han, X., Jia, M., and Jiang, Y.-G. (2024). Promptfusion: Decoupling stability and plasticity for continual learning. In *European Conference on Computer Vision*, pages 196–212. Springer.

[34] Chen, P., Ye, J., Chen, G., Zhao, J., and Heng, P.-A. (2021). Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. In *Proc. AAAI*.

[35] Cheng, G., Han, J., and Lu, X. (2017). Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10).

[36] Chizat, L., Oyallon, E., and Bach, F. (2019). On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*.

[37] Codella, N. et al. (2018). Skin lesion analysis toward melanoma detection. *IEEE Journal of Biomedical and Health Informatics*.

[38] Codella, N., Rotemberg, V., Tschandl, P., Celebi, M. E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., et al. (2019). Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*.

[39] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123.

[40] Cui, Y., Song, Y., Sun, C., Howard, A., and Belongie, S. (2018). Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[41] 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.

[42] Del Chiaro, R., Twardowski, B., Bagdanov, A., and Van de Weijer, J. (2020). Ratt: Recurrent attention to transient tasks for continual image captioning. *Advances in Neural Information Processing Systems*, 33:16736–16748.

[43] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[44] Don-Yehiya, S., Venezian, E., Raffel, C., Slonim, N., and Choshen, L. (2023). Cold fusion: Collaborative descent for distributed multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[45] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[46] Farquhar, S. and Gal, Y. (2018). Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*.

[47] Farquhar, S., Smith, L., and Gal, Y. (2020). Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations. *Advances in Neural Information Processing Systems*.

[48] Frascaroli, E., Panariello, A., Buzzega, P., Bonicelli, L., Porrello, A., and Calderara, S. (2024). Clip with generative latent replay: a strong baseline for incremental learning. *arXiv preprint arXiv:2407.15793*.

[49] Frénay, B. and Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

[50] George, T. (2021). NNGeometry: Easy and Fast Fisher Information Matrices and Neural Tangent Kernels in PyTorch.

[51] Ghosh, A., Kumar, H., and Sastry, P. S. (2017). Robust loss functions under label noise for deep neural networks. *AAAI Conference on Artificial Intelligence*.

[52] Goldberger, J. and Ben-Reuven, E. (2017). Training deep neural-networks using a noise adaptation layer. *International Conference on Learning Representations*.

[53] Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

[54] Goodfellow, I. J., Vinyals, O., and Saxe, A. M. (2015). Qualitatively characterizing neural network optimization problems. In *International Conference on Learning Representations*.

[55] Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. *CalTech Report*.

[56] Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., and Sugiyama, M. (2018a). Masking: A new perspective of noisy supervision. In *Proc. NeurIPS*, pages 5836–5846.

[57] Han, B., Yao, Q., Liu, T., Niu, G., Tsang, I. W., Kwok, J. T., and Sugiyama, M. (2020). A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*.

[58] Han, B., Yao, Q., Yu, X., et al. (2018b). Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*.

[59] Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. (2018c). Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*.

[60] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[61] Helber, P., Bischke, B., Dengel, A., and Borth, D. (2018). 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*, pages 204–207. IEEE.

[62] Helber, P., Bischke, B., Dengel, A., and Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

[63] Hendrycks, D., Basart, S., et al. (2021). The many faces of robustness: A critical analysis of out-of-distribution generalization. In *International Conference on Computer Vision*.

[64] Hendrycks, D., Lee, K., and Mazeika, M. (2019). Using pre-training can improve model robustness and uncertainty. In *Proc. ICML*.

[65] Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. (2019). Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[66] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

[67] Huang, C., Liu, Q., Lin, B. Y., Pang, T., Du, C., and Lin, M. (2024). Lorahub: Efficient cross-task generalization via dynamic loRA composition. In *First Conference on Language Modeling*.

[68] Hughes, D., Salathé, M., et al. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*.

[69] Huszár, F. (2018). Note on the quadratic penalties in elastic weight consolidation. *Proceedings of the National Academy of Sciences*, 115(11).

[70] Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. (2023). Editing models with task arithmetic. In *International Conference on Learning Representations*.

[71] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*.

[72] Jeffares, A., Liu, T., Crabbé, J., and van der Schaar, M. (2024). Joint training of deep ensembles fails due to learner collusion. *Advances in Neural Information Processing Systems*.

[73] Jenni, S. and Favaro, P. (2018). Deep bilevel learning. In *Proc. ECCV*, pages 618–633.

[74] Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*.

[75] Jin, X., Ren, X., Preotiuc-Pietro, D., and Cheng, P. (2023). Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations*.

[76] Jolicoeur-Martineau, A., Gervais, E., Fatras, K., Zhang, Y., and Lacoste-Julien, S. (2024). Population parameter averaging (PAPA). *Transactions on Machine Learning Research*.

[77] Joseph, K., Khan, S., Khan, F. S., and Balasubramanian, V. N. (2021). Towards open world object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5830–5840.

[78] Karim, N., Khalid, U., Esmaeili, A., and Rahnavard, N. (2022). Cnll: A semi-supervised approach for continual noisy label learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Continual Learning in Vision Workshop.

[79] Khot, T., Sabharwal, A., and Clark, P. (2018). Scitail: A textual entailment dataset from science question answering. In *Proceedings of AAAI*, pages 5189–5197.

[80] Kim, C. D., Jeong, J., Moon, S., and Kim, G. (2021). Continual learning on noisy data streams via self-purified replay. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

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

[82] Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*.

[83] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

[84] Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.

[85] Kunstner, F., Hennig, P., and Balles, L. (2019). Limitations of the empirical fisher approximation for natural gradient descent. *Advances in Neural Information Processing Systems*.

[86] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

[87] Lee, J. M. (2006). *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media.

[88] Lee, K.-H., He, X., Zhang, L., and Yang, L. (2018). Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[89] Li, J., Socher, R., and Hoi, S. C. (2020). Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*.

[90] Li, J., Socher, R., and Hoi, S. C. H. (2019). Dividemix: Learning with noisy labels as semi-supervised learning. *International Conference on Learning Representations*.

[91] Li, T., Huang, Z., Tao, Q., Wu, Y., and Huang, X. (2022). Trainable weight averaging: Efficient training by optimizing historical solutions. In *International Conference on Learning Representations*.

[92] Li, T., Huang, Z., Tao, Q., Wu, Y., and Huang, X. (2023a). Trainable weight averaging: A general approach for subspace training. *arXiv preprint arXiv:2205.13104*.

[93] Li, W., Peng, Y., Zhang, M., Ding, L., Hu, H., and Shen, L. (2023b). Deep model fusion: A survey. *arXiv preprint arXiv:2309.15698*.

[94] Li, W., Wang, L., Li, W., Agustsson, E., Berent, J., Gupta, A., Sukthankar, R., and Gool, L. V. (2017). Webvision challenge: Visual learning and understanding with web data. *arXiv preprint arXiv:1705.05640*.

[95] Li, Y., Guo, Z., and Wang, L. (2025). Cltr: Continual learning time-varying regularization for robust classification of noisy label images. *Pattern Recognition*, 171:112137.

[96] Li, Y., Guo, Z., and Wang, L. (2026). Cltr: Continual learning time-varying regularization for robust classification of noisy label images. *Pattern Recognition*, 171:112137.

[97] Li, Z. and Hoiem, D. (2017). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[98] Liang, Y.-S. and Li, W.-J. (2024). Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[99] Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. A. (2022). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*.

[100] Liu, J., Ji, Z., Yu, Y., Cao, J., Pang, Y., Han, J., and Li, X. (2024). Parameter-efficient fine-tuning for continual learning: A neural tangent kernel perspective. *arXiv preprint arXiv:2407.17120*.

[101] Liu, T. Y. and Soatto, S. (2023). Tangent model composition for ensembling and continual fine-tuning. In *IEEE International Conference on Computer Vision*.

[102] Lopez-Paz, D. and Ranzato, M. (2017). Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*.

[103] Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

[104] Lucas, J., Bae, J., Zhang, M. R., Fort, S., Zemel, R., and Grosse, R. (2021). Analyzing monotonic linear interpolation in neural network loss landscapes. In *International Conference on Machine Learning*.

[105] Lyu, Y. and Tsang, I. W. (2020). Curriculum loss: Robust learning and generalization against label corruption. In *Proc. ICLR*.

[106] Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S.-T., Wijewickrema, S., and Bailey, J. (2018). Dimensionality-driven learning with noisy labels. In *Proc. ICML*.

[107] Maini, P., Garg, S., Lipton, Z., and Kolter, J. Z. (2022). Characterizing datapoints via second-split forgetting. *Advances in Neural Information Processing Systems*.

[108] 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*.

[109] Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014). A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC*, pages 216–223.

[110] Marouf, I. E., Roy, S., Tartaglione, E., and Lathuilière, S. (2024). Weighted ensemble models are strong continual learners. In *Proceedings of the European Conference on Computer Vision*.

[111] Martens, J. (2020). New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21.

[112] Martens, J. and Grosse, R. (2015). Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*.

[113] Matena, M. S. and Raffel, C. A. (2022). Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*.

[114] McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Elsevier.

[115] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*.

[116] Menabue, M., Frascaroli, E., Boschini, M., Sangineto, E., Bonicelli, L., Porrello, A., and Calderara, S. (2024). Semantic residual prompts for continual learning. In *Proceedings of the European Conference on Computer Vision*.

[117] Menon, A. K., Rawat, A. S., Reddi, S. J., and Kumar, S. (2020). Can gradient clipping mitigate label noise? In *Proc. ICLR*.

[118] Millunzi, M., Bonicelli, L., Porrello, A., Credi, J., Kolm, P. N., and Calderara, S. (2024). May the forgetting be with you: Alternate replay for learning with noisy labels. In *British Machine Vision Conference*.

[119] Mirzadeh, S. I., Farajtabar, M., Pascanu, R., and Ghasemzadeh, H. (2020). Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*.

[120] Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. *Advances in Neural Information Processing Systems*.

[121] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop*.

[122] Nguyen, D. T., Mummadi, C. K., Ngo, T. P. N., Nguyen, T. H. P., Beggel, L., and Brox, T. (2020). SELF: Learning to filter noisy labels with self-ensembling. In *Proc. ICLR*.

[123] Oh, J., Kim, S., Ho, N., Kim, J.-H., Song, H., and Yun, S.-Y. (2022). Understanding cross-domain few-shot learning based on domain similarity and few-shot difficulty. *Advances in Neural Information Processing Systems*.

[124] Ortiz-Jimenez, G., Favero, A., and Frossard, P. (2024). Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*.

[125] Ostapenko, O., Lesort, T., Rodriguez, P., Arefin, M. R., Douillard, A., Rish, I., and Charlin, L. (2022). Continual learning with foundation models: An empirical study of latent replay. In *Conference on lifelong learning agents*.

[126] Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. (2017). Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[127] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. (2019). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415.

[128] Perera, P., Trager, M., Zancato, L., Achille, A., and Soatto, S. (2023). Prompt algebra for task composition. *arXiv preprint arXiv:2306.00310*.

[129] Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., and Hinton, G. (2017). Regularizing neural networks by penalizing confident output distributions. In *Proc. ICLR W*.

[130] Perez-Rua, J.-M., Zhu, X., Hospedales, T. M., and Xiang, T. (2020). Incremental few-shot object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13846–13855.

[131] Pfeiffer, J., Ruder, S., Vulić, I., and Ponti, E. (2023). Modular deep learning. *Transactions on Machine Learning Research*.

[132] Prabhu, A., Torr, P. H., and Dokania, P. K. (2020). Gdumb: A simple approach that questions our progress in continual learning. In *European conference on computer vision*, pages 524–540. Springer.

[133] Quattoni, A. and Torralba, A. (2009). Recognizing indoor scenes. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE.

[134] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

[135] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*, pages 2383–2392.

[136] Ratcliff, R. (1990). Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological Review*.

[137] 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*.

[138] Reed, S., Lee, H., Anguelov, D., et al. (2015). Training deep neural networks on noisy labels with bootstrapping. *International Conference on Learning Representations*.

[139] Ren, M., Zeng, W., Yang, B., and Urtasun, R. (2018). Learning to reweight examples for robust deep learning. *International Conference on Machine Learning*.

[140] Ridnik, T., Ben-Baruch, E., Noy, A., and Zelnik-Manor, L. (2021). Imagenet-21k pretraining for the masses. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

[141] Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*.

[142] Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

[143] Rypeść, G., Cygert, S., Khan, V., Trzcinski, T., Zieliński, B. M., and Twardowski, B. (2024). Divide and not forget: Ensemble of selectively trained experts in continual learning. In *International Conference on Learning Representations*.

[144] Sadrtdinov, I., Pozdeev, D., Vetrov, D. P., and Lobacheva, E. (2024). To stay or not to stay in the pre-train basin: Insights on ensembling in transfer learning. *Advances in Neural Information Processing Systems*.

[145] Schmidt, F. D., Vulić, I., and Glavaš, G. (2023). Free lunch: Robust cross-lingual transfer via model checkpoint averaging. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[146] Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. (2018). Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*.

[147] Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. (2016). Ntu rgb&d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[148] Shu, J., Xiong, Q., Wang, Q., et al. (2019). Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in Neural Information Processing Systems*.

[149] Smith, J. S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., and Kira, Z. (2023). Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[150] Song, H., Kim, M., and Lee, J.-G. (2019). Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*.

[151] Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. (2020). Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*.

[152] Song, Y.-F., Zhang, Z., Shan, C., and Wang, L. (2021). Constructing Stronger and Faster Baselines for Skeleton-based Action Recognition. *arXiv preprint arXiv:2106.15125*.

[153] Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. (2018). The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57.

[154] Stoica, G., Bolya, D., Bjorner, J., Hearn, T., and Hoffman, J. (2024). Zipit! merging models from different tasks without training. In *International Conference on Learning Representations*.

[155] Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., and Fergus, R. (2015). Training convolutional networks with noisy labels. In *International Conference on Learning Representations*.

[156] Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., and Silberman, N. (2019). Learning from noisy labels by regularized estimation of annotator confusion. In *Proc. CVPR*, pages 11244–11253.

[157] Toneva, M., Sordoni, A., des Combes, R. T., Trischler, A., Bengio, Y., and Gordon, G. J. (2019). An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*.

[158] Tschandl, P., Rosendahl, C., and Kittler, H. (2018). The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9.

[159] Van de Ven, G. M. and Tolias, A. S. (2019). Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*.

[160] Van de Ven, G. M., Tuytelaars, T., and Tolias, A. S. (2022a). Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197.

[161] Van de Ven, G. M., Tuytelaars, T., and Tolias, A. S. (2022b). Three types of incremental learning. *Nature Machine Intelligence*, 4(12).

[162] Verwimp, E., De Lange, M., and Tuytelaars, T. (2021). Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *IEEE International Conference on Computer Vision*.

[163] Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. (2016). Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*.

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

[165] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011a). The caltech-ucsd birds-200-2011 dataset. *Technical Report CNS-TR-2011-001*.

[166] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011b). The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

[167] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019a). 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.

[168] Wang, R., Duan, X., Kang, G., Liu, J., Lin, S., Xu, S., Lü, J., and Zhang, B. (2023). Attriclip: A non-incremental learner for incremental knowledge learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3654–3663.

[169] Wang, R., Liu, T., and Tao, D. (2017a). Multiclass learning with partially corrupted labels. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2568–2580.

[170] Wang, X. et al. (2017b). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification. *IEEE Conference on Computer Vision and Pattern Recognition*.

[171] Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. (2019b). Symmetric cross entropy for robust learning with noisy labels. In *Proc. ICCV*, pages 322–330.

[172] Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.-Y., Ren, X., Su, G., Perot, V., Dy, J., et al. (2022a). Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pages 631–648. Springer.

[173] Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T. (2022b). Learning to prompt for continual learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 139–149.

[174] Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T. (2022c). Learning to prompt for continual learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[175] Wei, H., Feng, L., Chen, X., and An, B. (2020). Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[176] Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., and Liu, Y. (2022). Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*.

[177] Welling, M. (2009). Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128.

[178] Williams, A., Nangia, N., and Bowman, S. R. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

[179] Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. (2022). Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*.

[180] Wu, C.-E., Tian, Y., Yu, H., Wang, H., Morgado, P., Hu, Y. H., and Yang, L. (2023). Why is prompt tuning for vision-language models robust to noisy labels? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[181] Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. (2019). Large scale incremental learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[182] Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. (2019). Are anchor points really indispensable in label-noise learning? In *Proc. NeurIPS*.

[183] Xiang, J. and Shlizerman, E. (2023). Tkil: Tangent kernel optimization for class balanced incremental learning. In *IEEE International Conference on Computer Vision*.

[184] Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. (2015a). Learning from massive noisy labeled data for image classification. *IEEE Conference on Computer Vision and Pattern Recognition*.

[185] Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. (2015b). Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[186] Yadav, P., Raffel, C., Muqeeth, M., Caccia, L., Liu, H., Chen, T., Bansal, M., Choshen, L., and Sordoni, A. (2024a). A survey on model moerging: Recycling and routing among specialized experts for collaborative learning. *arXiv preprint arXiv:2408.07057*.

[187] Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. (2024b). Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*.

[188] Yadav, P., Vu, T., Lai, J., Chronopoulou, A., Faruqui, M., Bansal, M., and Munkhdalai, T. (2024c). What matters for model merging at scale? *arXiv preprint arXiv:2410.03617*.

[189] Yang, G., Fini, E., Xu, D., Rota, P., Ding, M., Nabi, M., Alameda-Pineda, X., and Ricci, E. (2022). Uncertainty-aware contrastive distillation for incremental semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2567–2581.

[190] Yao, J., Wang, J., Tsang, I. W., Zhang, Y., Sun, J., Zhang, C., and Zhang, R. (2018). Deep learning from noisy image labels with quality embedding. *IEEE Transactions on Image Processing*, 28(4):1909–1922.

[191] Yu, J., Zhuge, Y., Zhang, L., Hu, P., Wang, D., Lu, H., and He, Y. (2024). Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23219–23230.

[192] Yu, X., Han, B., Yao, Q., et al. (2019). Does data cleaning really work? *IEEE International Conference on Computer Vision*.

[193] Yunis, D., Patel, K. K., Savarese, P. H. P., Vardi, G., Frankle, J., Walter, M., Livescu, K., and Maire, M. (2022). On convexity and linear mode connectivity in neural networks. In *OPT: Optimization for Machine Learning (NeurIPS 2022 Workshop)*.

[194] Zhai, M., Chen, L., He, J., Nawhal, M., Tung, F., and Mori, G. (2020). Piggyback gan: Efficient lifelong learning for image conditioned generation. In *European Conference on Computer Vision*, pages 397–413. Springer.

[195] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations*.

[196] Zhang, G., Wang, L., Kang, G., Chen, L., and Wei, Y. (2023a). Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *IEEE International Conference on Computer Vision*.

[197] Zhang, G., Wang, L., Kang, G., Chen, L., and Wei, Y. (2023b). Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *IEEE International Conference on Computer Vision*, pages 19148–19158.

[198] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). Mixup: Beyond empirical risk minimization. In *Proc. ICLR*.

[199] Zhang, J. and Bottou, L. (2023). Learning useful representations for shifting tasks and distributions. In *International Conference on Machine Learning*.

[200] Zhang, J., Liu, J., He, J., et al. (2024). Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information Processing Systems*.

[201] Zhang, Z. and Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proc. NeurIPS*, pages 8778–8788.

[202] Zhang, Z., Zhang, H., Arik, S. O., Lee, H., and Pfister, T. (2020). Distilling effective supervision from severe label noise. In *Proc. CVPR*, pages 9294–9303.

[203] Zheng, Z., Ma, M., Wang, K., Qin, Z., Yue, X., and You, Y. (2023). Preventing zero-shot transfer degradation in continual learning of vision-language models. In *IEEE International Conference on Computer Vision*.

[204] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1).

Glossary

ABS Asymmetric Balanced Sampling

AER Alternated Experience Replay

AI Artificial Intelligence

ANN Artificial Neural Network

CL Continual Learning

Class-IL Class-Incremental Learning

CLN Continual Learning with Noisy Labels

DER++ Dark Experience Replay++

DL Deep Learning

DNN Deep Neural Network

Domain-IL Domain-Incremental Learning

FAA Final Average Accuracy

FF Final Average Forgetting

FT fine-tuning

GLUE General Language Understanding Evaluation

GMM Gaussian Mixture Model

JT Joint Training

LNL Learning with Noisy Labels

ML Machine Learning

NLU Natural Language Understanding

SSL Self-Supervised Learning

Task-IL Task-Incremental Learning

VLM Vision-Language Model